PSS®E 33.4

APPLICATION PROGRAM INTERFACE (API)

March, 2013

SIEMENS

Siemens Industry, Inc.
Siemens Power Technologies International
400 State Street, PO Box 1058
Schenectady, NY 12301-1058 USA
+1 518-395-5000
www.siemens.com/power-technologies

© Copyright 1990-2013 Siemens Industry, Inc., Siemens Power Technologies International

Information in this manual and any software described herein is confidential and subject to change without notice and does not represent a commitment on the part of Siemens Industry, Inc., Siemens Power Technologies International. The software described in this manual is furnished under a license agreement or nondisclosure agreement and may be used or copied only in accordance with the terms of the agreement. No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, for any purpose other than the purchaser's personal use, without the express written permission of Siemens Industry, Inc., Siemens Power Technologies International.

PSS®E high-performance transmission planning software is a registered trademark of Siemens Industry, Inc., Siemens Power Technologies International in the United States and other countries.

The Windows® 2000 operating system, the Windows XP® operating system, the Windows Vista® operating system, the Windows 7® operating system, the Visual C++® development system, Microsoft Office Excel® and Microsoft Visual Studio® are registered trademarks of Microsoft Corporation in the United States and other countries.

Intel® Visual Fortran Compiler for Windows is a trademark of Intel Corporation in the United States and other countries.

The Python™ programming language is a trademark of the Python Software Foundation.

Other names may be trademarks of their respective owners.

SIEMENS

Table of Contents

Preface i

Chapter 1 - Power Flow Operation 1-1

1.1 ACCC 1-2
1.2 ACCC_MULTIPLE_RUN_REPORT 1-4
1.3 ACCC_MULTIPLE_RUN_REPORT_2 1-8
1.4 ACCC_PARALLEL 1-12
1.5 ACCC_PARALLEL_2 1-15
1.6 ACCC_SINGLE_RUN_REPORT 1-18
1.7 ACCC_SINGLE_RUN_REPORT_2 1-21
1.8 ACCC_SINGLE_RUN_REPORT_3 1-24
1.9 ACCC_SINGLE_RUN_REPORT_4 1-27
1.10 ACCC_TRIP_COR 1-30
1.11 ACCC_TRIP_COR_2 1-34
1.12 ACCC_TRIP_COR_3 1-38
1.13 ACCC_WITH_COR 1-43
1.14 ACCC_WITH_COR_2 1-47
1.15 ACCC_WITH_COR_3 1-51
1.16 ACCC_WITH_DSP 1-56
1.17 ACCC_WITH_DSP_2 1-58
1.18 ACCC_WITH_DSP_3 1-61
1.19 ACCC_WITH_TRIP 1-64
1.20 ACCC_WITH_TRIP_2 1-67
1.21 ACCC_WITH_TRIP_PARALLEL 1-71
1.22 ACCOR 1-75
1.23 ACCOR_2 1-78
1.24 ACCOR_3 1-82
1.25 ALERT_OUTPUT 1-86
1.26 ALPH 1-87
1.27 ANSI 1-88
1.28 ANSI_2 1-91
1.29 APPEND_ACCC 1-94
1.30 AREA 1-96

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.31 AREA_2 1-97
1.32 AREA_ZONE 1-98
1.33 ARNM 1-99
1.34 ARNM_2 1-100
1.35 ASCC 1-101
1.36 ASCC_1A 1-103
1.37 ASCC_2 1-105
1.26 ASCC_3 1-108
1.38 ASCC_SCFILE 1-112
1.39 BASE_FREQUENCY 1-113
1.40 BGEN 1-114
1.41 BKDY 1-115
1.42 BRCH 1-116
1.43 BRCH_2 1-118
1.44 BSNM 1-120
1.45 BUS_INPUT 1-125
1.46 BUS_OUTPUT 1-126
1.47 BUS_SIZE_LEVEL 1-127
1.48 BUSN 1-128
1.49 CASE 1-129
1.50 CHECK_POWERFLOW_DATA 1-130
1.51 CLOSE_POWERFLOW 1-131
1.52 CHECK_SEQUENCE_DATA 1-132
1.53 CheckVoltageLimits 1-133
1.54 CLOSE_REPORT 1-134
1.55 CMPR 1-135
1.56 CNTB 1-137
1.57 CONG 1-140
1.58 CONL 1-141
1.59 CONNECTIVITY_CHECK 1-143
1.60 CONTROL_AREA_INTERCHANGE 1-144
1.61 DC_TAP_ADJUSTMENT 1-145
1.62 DCCC 1-146
1.63 DCCOR 1-148
1.64 DCLF 1-151
1.65 DFAX 1-153
1.66 DFAX_CONTINGENCY 1-155

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.67 DFTI 1-156
1.68 DIFF 1-158
1.69 DSCN 1-161
1.70 ECDI 1-162
1.71 ECHO 1-164
1.72 EEQV 1-165
1.73 EQRD 1-166
1.74 EXAM 1-168
1.75 EXTR 1-169
1.76 FACT 1-170
1.77 FDNS 1-171
1.78 FILE_OVERWRITE 1-173
1.79 FIND 1-174
1.80 FLAT 1-175
1.81 FLAT_2 1-177
1.82 FNSL 1-179
1.83 GCAP 1-181
1.84 GDIF 1-182
1.85 GENDSP 1-184
1.86 GENS 1-186
1.87 GEOL 1-187
1.88 GetContingencySavedCase 1-188
1.89 GIC 1-189
1.90 GNESTATUS 1-192
1.91 GNET 1-193
1.92 IECS 1-194
1.93 IECS_2 1-196
1.94 IECS_3 1-199
1.77 IECS_4 1-202
1.95 IECS_SCFILE 1-206
1.96 ImNet 1-207
1.97 ImOutage 1-208
1.98 IMPC 1-209
1.99 IMPLEMENT_TRANSFER 1-210
1.100 IMPLEMENT_TRANSFER_2 1-213
1.101 INLF 1-216
1.102 INLF_2 1-218

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.103 INTA 1-220
1.104 INTZ 1-221
1.105 JOIN 1-222
1.106 LAMP 1-224
1.107 LINE_SHUNT_REPORTING 1-225
1.108 LINES_PER_PAGE 1-226
1.109 LINES_PER_PAGE_ONE_DEVICE 1-228
1.110 LIST 1-229
1.111 ListContingencySavedCases 1-231
1.112 LLRF 1-232
1.113 LOAD_REDUCTION 1-234
1.114 LOUT 1-235
1.115 LTAP 1-236
1.116 MACCC 1-237
1.117 MACCC_2 1-242
1.118 MACCC_PARALLEL 1-247
1.119 MACCC_TRIP_COR 1-252
1.120 MACCC_TRIP_COR_2 1-258
1.121 MACCC_TRIP_COR_3 1-264
1.122 MACCC_WITH_COR 1-271
1.123 MACCC_WITH_COR_2 1-277
1.124 MACCC_WITH_COR_3 1-283
1.125 MACCC_WITH_TRIP 1-289
1.126 MACCC_WITH_TRIP_2 1-294
1.127 MACCC_WITH_TRIP_PARALLEL 1-300
1.128 MATRIX_GROWTH_FACTOR 1-306
1.129 MBID2DC 1-307
1.130 MBID3WND 1-308
1.131 MBIDATRN 1-309
1.132 MBIDBRN 1-310
1.133 MBIDFACTS 1-311
1.134 MBIDGNE 1-312
1.135 MbidIndMac 1-313
1.136 MoveIndMac 1-314
1.137 MoveIndMacs 1-315
1.138 MBIDLOAD 1-316
1.139 MBIDMAC 1-317

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.140 MBIDMDC 1-318
1.141 MBIDMSL 1-319
1.142 MBIDSHUNT 1-320
1.143 MBIDVSC 1-321
1.144 MCRE 1-322
1.145 MODR 1-323
1.146 MOVE3WND 1-325
1.147 MOVEBRN 1-326
1.148 MOVELOAD 1-327
1.149 MOVELOADS 1-328
1.150 MOVEMAC 1-329
1.151 MOVEPLNT 1-330
1.152 MOVESHUNT 1-331
1.153 MOVESHUNTS 1-332
1.154 MOVESWS 1-333
1.155 MSLV 1-334
1.156 MSUM 1-336
1.157 MTDC 1-337
1.158 MULTISECTION_REPORTING 1-338
1.159 MWMI 1-339
1.160 MWOMWMSF 1-341
1.161 MWOMWOSF 1-342
1.162 N11_ACCC 1-343
1.163 N11_ACCC_2 1-348
1.164 N11_ACCC_PSCOPF 1-352
1.165 N11_ACCC_PSCOPF_2 1-360
1.166 NETG 1-367
1.167 NEW_DIMENSION 1-368
1.168 NEWCAS 1-369
1.169 NEWCASE_2 1-370
1.170 NEWSEQ 1-371
1.171 NEWTON_TOLERANCE 1-372
1.172 NON_DIVERGENT 1-373
1.173 NON_TRANS_PERCENT_UNITS 1-374
1.174 NSOL 1-375
1.175 NUMBER_THREADS 1-377
1.176 OPENUCTEFILE 1-378

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.177 ORDR 1-379
1.178 OTDF 1-380
1.179 OUTPUT_Y_MATRIX 1-381
1.180 OUTS 1-382
1.181 OWNM 1-383
1.182 OWNM_2 1-385
1.183 OWNM_3 1-387
1.184 OWNR 1-389
1.185 PATH 1-390
1.186 PBUS_DELETE 1-391
1.187 PHASE_SHIFT_ADJUSTMENT 1-392
1.188 POLY 1-393
1.189 POUT 1-397
1.190 POWER_OUTPUT 1-398
1.191 PP_ACCC 1-399
1.192 PP_ACCC_MULTI_CASE 1-401
1.193 PRINT_OUTAGED_BRANCHES 1-404
1.194 PRINT_WINDING_BUSES 1-405
1.195 PROGRESS_OUTPUT 1-406
1.196 PROMPT_OUTPUT 1-407
1.197 PRTI 1-408
1.198 PSCOPF 1-409
1.199 PSCOPF_2 1-415
1.200 PSEB 1-421
1.201 PSSEHALT 1-422
1.202 PSSEHALT_2 1-423
1.203 PURG 1-424
1.204 PURG2DC 1-425
1.205 PURG3WND 1-426
1.206 PURGAREA 1-427
1.207 PURGATRN 1-428
1.208 PURGBRN 1-429
1.209 PURGFACTS 1-430
1.210 PURGGNE 1-431
1.211 PurgIndMac 1-432
1.212 PurgIndMacs 1-433
1.213 PURGLOAD 1-434

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.214 PURGLOADS 1-435
1.215 PURGMAC 1-436
1.216 PURGMDC 1-437
1.217 PURGMSL 1-438
1.218 PURGMUT 1-439
1.219 PURGOWNER 1-440
1.220 PURGPLNT 1-441
1.221 PURGSHUNT 1-442
1.222 PURGSHUNTS 1-443
1.223 PURGSWS 1-444
1.224 PURGVSC 1-445
1.225 PURGZONE 1-446
1.226 PURGE_MULTI_TERM_DC_BUS 1-447
1.227 PURGE_MULTI_TERM_DC_CONVR 1-448
1.228 PURGE_MULTI_TERM_DC_LINK 1-449
1.229 PV_ENGINE 1-450
1.230 PV_ENGINE_1A 1-455
1.231 PV_ENGINE_2 1-460
1.232 PV_ENGINE_3 1-465
1.233 PV_ENGINE_4 1-470
1.234 PV_ENGINE_5 1-475
1.235 PV_ENGINE_6 1-481
1.236 QV_ENGINE 1-488
1.237 QV_ENGINE_2 1-490
1.238 QV_ENGINE_3 1-492
1.239 QV_ENGINE_4 1-495
1.240 RANK 1-499
1.241 RANK_BRN_AND_MAC 1-501
1.242 RATE 1-503
1.243 RATE_2 1-504
1.244 RATING_SET 1-506
1.245 RAWD 1-507
1.246 RAWD_2 1-509
1.247 RDCH 1-511
1.248 RDCHRAWVERSION 1-512
1.249 RDEQ 1-513
1.250 READ 1-515

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.251 READRAWVERSION 1-516
1.252 READSUB 1-517
1.253 READSUBRAWVERSION 1-519
1.254 RECN 1-521
1.255 RECORDING_OPTIONS 1-522
1.256 REGB 1-523
1.257 RELIND 1-524
1.258 RELIND_2 1-527
1.259 REMM 1-530
1.260 REPORT_OUTPUT 1-531
1.261 RESQ 1-532
1.262 RESQVERSION 1-533
1.263 RETI 1-534
1.264 REV29_NAMES 1-535
1.265 RNFI 1-536
1.266 RWCM 1-538
1.267 RWMA 1-539
1.268 RWMM 1-541
1.269 RWSQ 1-542
1.270 RWSQ_2 1-544
1.271 SAVE 1-546
1.272 SAVEUCTEFILE 1-547
1.273 SCAL 1-548
1.274 SCAL_2 1-551
1.275 SCEQ 1-554
1.276 SCGR 1-555
1.277 SCMU 1-557
1.278 SCOP 1-563
1.279 SENSITIVITY_FLOW 1-565
1.280 SENSITIVITY_FLOWS 1-569
1.281 SENSITIVITY_VOLTAGE 1-573
1.282 SENSITIVITY_VOLTAGES 1-576
1.283 SENSITIVITY_INTERFACE 1-579
1.284 SCOP 1-583
1.285 SEQD 1-585
1.286 SEQD_2 1-586
1.287 SEQUENCE_NETWORK_SETUP 1-587

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.288 SET_INPUT_DEV 1-588
1.289 SHNT 1-589
1.290 SHORT_CIRCUIT_COORDINATES 1-590
1.291 SHORT_CIRCUIT_MODELING 1-591
1.292 SHORT_CIRCUIT_UNITS 1-592
1.293 SHORT_CIRCUIT_WARNING 1-593
1.294 SHOW 1-594
1.295 SIZE 1-595
1.296 SOLV 1-596
1.297 SPCB 1-598
1.298 SPIL 1-600
1.299 SPLT 1-604
1.300 SQEX 1-605
1.301 SRAIND 1-606
1.302 SQLI 1-608
1.303 STOP 1-609
1.304 STOP_2 1-610
1.305 SUBS 1-611
1.306 SWITCHED_SHUNT_ADJUSTMENT 1-612
1.307 TAP_ADJUSTMENT 1-613
1.308 TEXT 1-614
1.309 TFLG 1-615
1.310 TIES 1-616
1.311 TIEZ 1-617
1.312 TIME 1-618
1.313 TLST 1-619
1.314 TLTG 1-620
1.315 TPCH 1-624
1.316 TRANSACTION_DELETE 1-626
1.317 TRANSFORMER_PERCENT_UNITS 1-627
1.318 TRANSMISSION_LINE_UNITS 1-628
1.319 TREE 1-629
1.320 TYSL 1-630
1.321 USER 1-631
1.322 VAMM 1-632
1.323 VAMO 1-633
1.324 VCHK 1-634

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.325 VOLTAGE_INPUT 1-635
1.326 VOLTAGE_OUTPUT 1-636
1.327 VSMO 1-637
1.328 WRITERAWVERSION 1-638
1.329 WRITESEQVERSION 1-639
1.330 WRITESEQVERSION_2 1-640
1.331 WRITE_OPTIONS_FILE 1-641
1.332 XEQV 1-642
1.333 ZONE 1-643
1.334 ZONE_2 1-644
1.335 ZONE_AREA 1-645
1.336 ZONM 1-646
1.337 ZONM_2 1-647

Chapter 2 - Power Flow Data Changing 2-1

2.1 AREA_DATA 2-2
2.2 BRANCH_CHNG 2-3
2.3 BRANCH_DATA 2-5
2.4 BUS_CHNG_3 2-8
2.5 BUS_DATA 2-10
2.6 BUS_DATA_2 2-12
2.7 BUS_DATA_3 2-14
2.8 BUS_NUMBER 2-16
2.9 CASE_TITLE_DATA 2-17
2.10 FACTS_CHNG_2 2-18
2.11 FACTS_DATA 2-20
2.12 FACTS_DATA_2 2-23
2.13 GNE_CHNG 2-26
2.14 GNE_DATA 2-28
2.15 IMPEDANCE_CORRECTION_DATA 2-30
2.16 INDUCTION_MACHINE_CHNG 2-31
2.17 INDUCTION_MACHINE_DATA 2-33
2.18 LOAD_CHNG_4 2-36
2.19 LOAD_DATA 2-37
2.20 LOAD_DATA_3 2-39
2.21 LOAD_DATA_4 2-41
2.22 LONG_TITLE_DATA 2-43

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.23 MACHINE_CHNG_2 2-44
2.24 MACHINE_DATA 2-46
2.25 MACHINE_DATA_2 2-48
2.26 MULTI_SECTION_LINE_DATA 2-51
2.27 MULTI_SECTION_LINE_EDIT 2-53
2.28 MULTI_TERM_DC_BUS_DATA 2-54
2.29 MULTI_TERM_DC_CONVR_DATA 2-56
2.30 MULTI_TERM_DC_LINE_CHNG 2-59
2.31 MULTI_TERM_DC_LINE_DATA 2-60
2.32 MULTI_TERM_DC_LINK_DATA 2-62
2.33 OWNER_DATA 2-64
2.34 PBUS_ADD_MOD 2-65
2.35 PLANT_CHNG 2-66
2.36 PLANT_DATA 2-67
2.37 SEQ_3_WIND_GROUNDING_DATA 2-68
2.38 SEQ_3_WIND_WINDING_DATA 2-69
2.39 SEQ_BRANCH_DATA 2-70
2.40 SEQ_BRANCH_DATA_3 2-71
2.41 SEQ_BUS_DATA 2-73
2.42 SEQ_FIXED_SHUNT_DATA 2-74
2.43 SEQ_INDUCTION_MACH_DATA 2-75
2.44 SEQ_LOAD_DATA 2-77
2.45 SEQ_MACHINE_DATA 2-79
2.46 SEQ_MACHINE_DATA_3 2-80
2.47 SEQ_MUTUAL_DATA 2-82
2.48 SEQ_SWITCHED_SHUNT_DATA 2-84
2.49 SEQ_THREE_WINDING_DATA 2-86
2.50 SEQ_THREE_WINDING_DATA_3 2-88
2.51 SEQ_TWO_WINDING_DATA 2-90
2.52 SEQ_TWO_WINDING_DATA_3 2-92
2.53 SHUNT_CHNG 2-94
2.54 SHUNT_DATA 2-95
2.55 SOLUTION_PARAMETERS 2-96
2.56 SOLUTION_PARAMETERS_2 2-98
2.57 SOLUTION_PARAMETERS_3 2-100
2.58 SOLUTION_PARAMETERS_4 2-103
2.59 SWITCHED_SHUNT_CHNG_3 2-105

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.60 SWITCHED_SHUNT_DATA 2-107
2.61 SWITCHED_SHUNT_DATA_3 2-110
2.62 THREE_WINDING_DATA 2-114
2.63 THREE_WND_IMPED_CHNG_3 2-118
2.64 THREE_WND_IMPED_DATA_3 2-120
2.65 THREE_WND_IMPEDANCE_DATA 2-123
2.66 THREE_WND_WINDING_DATA 2-126
2.67 THREE_WND_WINDING_DATA_3 2-129
2.68 TRANSACTION_ADD_MOD 2-132
2.69 TRANSFER_CHNG 2-133
2.70 TRANSFER_DATA 2-134
2.71 TWO_TERM_DC_CONVR_DATA 2-135
2.72 TWO_TERMINAL_DC_LINE_CHNG 2-138
2.73 TWO_TERMINAL_DC_LINE_DATA 2-140
2.74 TWO_WINDING_CHNG_4 2-142
2.75 TWO_WINDING_DATA 2-145
2.76 TWO_WINDING_DATA_3 2-149
2.77 TWO_WINDING_DATA_4 2-153
2.78 VSC_DC_LINE_CHNG 2-157
2.79 VSC_DC_CONVERTER_DATA 2-158
2.80 VSC_DC_LINE_DATA 2-160
2.81 ZONE_DATA 2-161

Chapter 3 - Optimal Power Flow 3-1

3.1 Optimal Power Flow Program Operation APIs 3-2
3.1.1 ADD_DETAILS_TO_OPF_LOG 3-2
3.1.2 APPLY_ALTERNATE_STEP_SIZE 3-3
3.1.3 APPLY_AUTOMATIC_SCALING 3-4
3.1.4 BAD_ITER_COARSE_LIMIT 3-5
3.1.5 BAD_ITER_FINE_LIMIT 3-6
3.1.6 CLAMP_NONOPTIMIZED_GENS 3-7
3.1.7 CONSTRAIN_INTERFACE_FLOWS 3-8
3.1.8 FINAL_OPF_BARRIER_COEFF 3-9
3.1.9 IMPORT_ECDI 3-10
3.1.10 INITIAL_OPF_BARRIER_COEFF 3-11
3.1.11 INTERFACE_FLOW_COST_COEFF 3-12
3.1.12 LSTO 3-13
3.1.13 MINIMIZE_ADJ_BUS_SHUNTS 3-14
3.1.14 MINIMIZE_FUEL_COST 3-15
3.1.15 MINIMIZE_INTERFACE_FLOWS 3-16
3.1.16 MINIMIZE_LOAD_ADJUSTMENTS 3-17
3.1.17 MINIMIZE_P_LOSSES 3-18
3.1.18 MINIMIZE_P_SLACK 3-19

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.19 MINIMIZE_Q_LOSSES 3-20
3.1.20 MINIMIZE_Q_SLACK 3-21
3.1.21 MINIMIZE_REACTIVE_RESERVE 3-22
3.1.22 MINIMIZE_SERIES_COMP 3-23
3.1.23 NOPF 3-24
3.1.24 OPEN_BUS_VOLTAGE_LIMITS 3-25
3.1.25 OPF_BARRIER_STEP_LENGTH 3-26
3.1.26 OPF_CLAMP_DECAY_FACTOR 3-27
3.1.27 OPF_FINAL_CLAMP_TOL 3-28
3.1.28 OPF_FIX_ALL_GENERATORS 3-29
3.1.29 OPF_FIX_PHASE_SHIFTERS 3-30
3.1.30 OPF_FIX_SWITCHED_SHUNTS 3-31
3.1.31 OPF_FIX_TAP_RATIOS 3-32
3.1.32 OPF_FIXED_VOLTAGE_PENALTY 3-33
3.1.33 OPF_INITIAL_CLAMP_TOL 3-34
3.1.34 OPF_INTERIOR_SHIFT_FACTOR 3-35
3.1.35 OPF_LAMBDA_TOLERANCE 3-36
3.1.36 OPF_LF_CONTROL_PENALTY 3-37
3.1.37 OPF_MAX_TAP_RATIO_STEP 3-38
3.1.38 OPF_MIN_TAP_RATIO_STEP 3-39
3.1.39 OPF_REGULATE_AREA_INT 3-40
3.1.40 OPF_ROUND_SWITCHED_SHUNTS 3-41
3.1.41 OPF_ROUND_TAP_RATIOS 3-42
3.1.42 OPF_SCALE_QGEN_LIMITS 3-43
3.1.43 OPF_STEP_LENGTH_TOLERANCE 3-44
3.1.44 OPF_USE_GENERATOR_VSCHED 3-45
3.1.45 OPTO 3-46
3.1.46 P_LOSSES_COST_COEFF 3-52
3.1.47 PRODUCE_OPF_LOG_FILE 3-53
3.1.48 Q_LOSSES_COST_COEFF 3-54
3.1.49 REACTIVE_RESV_COST_COEFF 3-55
3.1.50 ROPF 3-56
3.1.51 RWOP 3-57
3.1.52 SET_OPF_REPORT_SUBSYSTEM 3-58
3.1.53 USE_DUAL_CRITERIA 3-59
3.1.54 USE_EMERGENCY_FLOW_LIMITS 3-60
3.1.55 USE_EMERGENCY_VOLT_LIMITS 3-61
3.1.56 WRITE_OPF_OPTIONS_FILE 3-62

3.2 Optimal Power Flow Data Changing APIs 3-63
3.2.1 NEWOPF 3-63
3.2.2 OPF_ADJBRX_INDV 3-64
3.2.3 OPF_ADJBRX_SUBSYS 3-66
3.2.4 OPF_ADJLOAD_TBL 3-69
3.2.5 OPF_ADJVAR_INDV 3-71
3.2.6 OPF_ADJVAR_SUBSYS 3-73
3.2.7 OPF_APDSP_TBL 3-76
3.2.8 OPF_BRFLW_3WT_INDV 3-78
3.2.9 OPF_BRFLW_BRN_INDV 3-80
3.2.10 OPF_BRFLW_SUBSYS 3-82
3.2.11 OPF_BUS_INDV 3-85
3.2.12 OPF_BUS_SUBSYS 3-87
3.2.13 OPF_CSTTBL_LIN 3-90

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.14 OPF_CSTTBL_POLY 3-91
3.2.15 OPF_CSTTBL_QUAD 3-92
3.2.16 OPF_GEN_RCAP_INDV 3-93
3.2.17 OPF_GEN_RCAP_SUBSYS 3-95
3.2.18 OPF_GENDSP_INDV 3-98
3.2.19 OPF_GENDSP_SUBSYS 3-99
3.2.20 OPF_GENRSV_INDV 3-100
3.2.21 OPF_GENRSV_SUBSYS 3-101
3.2.22 OPF_INTFLW_3WT 3-103
3.2.23 OPF_INTFLW_BRN 3-104
3.2.24 OPF_INTFLW_MAIN 3-105
3.2.25 OPF_INTFLW_RESET 3-107
3.2.26 OPF_LNCEQN_ADJLOAD 3-108
3.2.27 OPF_LNCEQN_ADJVAR 3-109
3.2.28 OPF_LNCEQN_BRFLOW 3-110
3.2.29 OPF_LNCEQN_INTFLOW 3-111
3.2.30 OPF_LNCEQN_MAIN 3-112
3.2.31 OPF_LNCEQN_PGEN 3-113
3.2.32 OPF_LNCEQN_QGEN 3-114
3.2.33 OPF_LNCEQN_RESET 3-115
3.2.34 OPF_LNCEQN_SWSHUNT 3-116
3.2.35 OPF_LNCEQN_TRAN 3-117
3.2.36 OPF_LNCEQN_VANG 3-118
3.2.37 OPF_LNCEQN_VMAG 3-119
3.2.38 OPF_LOAD_INDV 3-120
3.2.39 OPF_LOAD_SUBSYS 3-121
3.2.40 OPF_PERRSV_GEN 3-122
3.2.41 OPF_PERRSV_MAIN 3-123
3.2.42 OPF_PERRSV_RESET 3-124
3.2.43 PURGE_ALL_OPF_DATA 3-125
3.2.44 PURGE_OPF_ADJBRX_INDV 3-126
3.2.45 PURGE_OPF_ADJBRX_SUBSYS 3-127
3.2.46 PURGE_OPF_ADJLOAD_TBL 3-128
3.2.47 PURGE_OPF_ADJVAR_INDV 3-129
3.2.48 PURGE_OPF_ADJVAR_SUBSYS 3-130
3.2.49 PURGE_OPF_APDSP_TBL 3-131
3.2.50 PURGE_OPF_BRFLW_3WT 3-132
3.2.51 PURGE_OPF_BRFLW_BRN 3-133
3.2.52 PURGE_OPF_BRFLW_SUBSYS 3-134
3.2.53 PURGE_OPF_BUS_INDV 3-135
3.2.54 PURGE_OPF_BUS_SUBSYS 3-136
3.2.55 PURGE_OPF_CSTTBL_LIN 3-137
3.2.56 PURGE_OPF_CSTTBL_POLY 3-138
3.2.57 PURGE_OPF_CSTTBL_QUAD 3-139
3.2.58 PURGE_OPF_GEN_RCAP_INDV 3-140
3.2.59 PURGE_OPF_GEN_RCAP_SUBSYS 3-141
3.2.60 PURGE_OPF_GENDSP_INDV 3-142
3.2.61 PURGE_OPF_GENDSP_SUBSYS 3-143
3.2.62 PURGE_OPF_GENRSV_INDV 3-144
3.2.63 PURGE_OPF_GENRSV_SUBSYS 3-145
3.2.64 PURGE_OPF_INTFLW 3-146
3.2.65 PURGE_OPF_INTFLW_BRN 3-147

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.66 PURGE_OPF_INTFLW_3WT 3-148
3.2.67 PURGE_OPF_LNCEQN 3-149
3.2.68 PURGE_OPF_LNCEQN_ADJLOAD 3-150
3.2.69 PURGE_OPF_LNCEQN_ADJVAR 3-151
3.2.70 PURGE_OPF_LNCEQN_BRFLOW 3-152
3.2.71 PURGE_OPF_LNCEQN_INTFLOW 3-153
3.2.72 PURGE_OPF_LNCEQN_PGEN 3-154
3.2.73 PURGE_OPF_LNCEQN_QGEN 3-155
3.2.74 PURGE_OPF_LNCEQN_SWSHUNT 3-156
3.2.75 PURGE_OPF_LNCEQN_TRAN 3-157
3.2.76 PURGE_OPF_LNCEQN_VANG 3-158
3.2.77 PURGE_OPF_LNCEQN_VMAG 3-159
3.2.78 PURGE_OPF_LOAD_INDV 3-160
3.2.79 PURGE_OPF_LOAD_SUBSYS 3-161
3.2.80 PURGE_OPF_PERRSV 3-162
3.2.81 PURGE_OPF_PERRSV_GEN 3-163

Chapter 4 - Dynamics Simulation 4-1

4.1 Dynamic Simulation Function APIs 4-2
4.1.1 ADDMODELFOLDER 4-2
4.1.2 ADDMODELLIBRARY 4-3
4.1.3 ASTR 4-4
4.1.4 AULIST 4-6
4.1.5 CHANGE_CHANNEL_OUT_FILE 4-7
4.1.6 CHANGE_CON 4-8
4.1.7 CHANGE_ICON 4-9
4.1.8 CHANGE_CHRICN 4-10
4.1.9 CHANGE_GREF 4-11
4.1.10 CHANGE_STATE 4-12
4.1.11 CHANGE_SWSREF 4-13
4.1.12 CHANGE_VAR 4-14
4.1.13 CHANGE_VREF 4-15
4.1.14 DCLIST 4-16
4.1.15 DELMODELFOLDER 4-17
4.1.16 DLST 4-18
4.1.17 DOCU 4-19
4.1.18 DROPMODELLIBRARY 4-22
4.1.19 DROPMODELPROGRAM 4-23
4.1.20 DYDA 4-24
4.1.21 DYNAMICS_SOLUTION_PARAMS 4-27
4.1.22 DYNAMICS_SOLUTION_PARAM_2 4-29
4.1.23 DYNAMICSMODE 4-31
4.1.24 DYRE_ADD 4-32
4.1.25 DYRE_NEW 4-33
4.1.26 ERUN 4-34
4.1.27 ESTR_OPEN_CIRCUIT_TEST 4-35
4.1.28 ESTR_RESPONSE_RATIO_TEST 4-36
4.1.29 FCLIST 4-37
4.1.30 GRUN 4-38
4.1.31 GSTR 4-39
4.1.32 INCREMENT_GREF 4-40
4.1.33 INCREMENT_SWSREF 4-41

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.34 INCREMENT_VREF 4-42
4.1.35 LDLIST 4-43
4.1.36 MLST 4-44
4.1.37 MOVEMODELFOLDER 4-45
4.1.38 MRUN 4-46
4.1.39 MSTR 4-47
4.1.40 POWERFLOWMODE 4-48
4.1.41 PSAS 4-49
4.1.42 RLLIST 4-50
4.1.43 RSTR 4-51
4.1.44 RUN 4-52
4.1.45 RWDY 4-53
4.1.46 SET_GENANG 4-54
4.1.47 SET_GENANG_2 4-55
4.1.48 SET_GENPWR 4-57
4.1.49 SET_NETFRQ 4-58
4.1.50 SET_NEXT_CHANNEL 4-59
4.1.51 SET_NEXT_ICON 4-60
4.1.52 SET_NEXT_VAR 4-61
4.1.53 SET_OSSCAN 4-62
4.1.54 SET_RELANG 4-63
4.1.55 SET_RELSCN 4-64
4.1.56 SET_VLTSCN 4-65
4.1.57 SET_VOLT_VIOL_SUBSYS_FLAG 4-66
4.1.58 SET_VOLTAGE_DIP_CHECK 4-67
4.1.59 SET_VOLTAGE_REC_CHECK 4-68
4.1.60 SETMODELFOLDER 4-70
4.1.61 SETMODELFOLDERS 4-71
4.1.62 SIZE_DS 4-72
4.1.63 SNAP 4-73
4.1.64 SRRS 4-74
4.1.65 STRT 4-75
4.1.66 SWSLIST 4-76
4.1.67 TRIG_VOLT_VIOLATION_CHECK 4-77
4.1.68 WNLIST 4-78

4.2 Output Channel APIs 4-79
4.2.1 General Output Channel Related Functions 4-79
4.2.1.1 CHSB 4-79
4.2.1.2 CHANGE_CHANNEL 4-82
4.2.1.3 CRT_PLOT_CHANNEL 4-83
4.2.1.4 DELETE_ALL_PLOT_CHANNELS 4-84
4.2.1.5 DMPC 4-85
4.2.1.6 LIST_CHANNEL_MODELS 4-87
4.2.1.7 REMOVE_UNUSED_CHAN_MODELS 4-88
4.2.2 Assign Machine Quantities to Output Channels 4-89
4.2.2.1 MACHINE_APP_R_X_CHANNEL 4-89
4.2.2.2 MACHINE_ARRAY_CHANNEL 4-91
4.2.2.3 MACHINE_ITERM_CHANNEL 4-93
4.2.3 Assign Branch Quantities to Output Channels 4-95
4.2.3.1 BRANCH_APP_R_X_CHANNEL 4-95
4.2.3.2 BRANCH_MVA_CHANNEL 4-97
4.2.3.3 BRANCH_P_AND_Q_CHANNEL 4-99

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.2.3.4 BRANCH_P_CHANNEL 4-101
4.2.3.5 THREE_WND_APP_R_X_CHANNEL 4-103
4.2.3.6 THREE_WND_MVA_CHANNEL 4-105
4.2.3.7 THREE_WND_P_AND_Q_CHANNEL 4-107
4.2.3.8 THREE_WND_P_CHANNEL 4-109
4.2.4 Assign Other Quantities to Output Channels 4-111
4.2.4.1 BUS_FREQUENCY_CHANNEL 4-111
4.2.4.2 LOAD_ARRAY_CHANNEL 4-112
4.2.4.3 STATE_CHANNEL 4-113
4.2.4.4 VAR_CHANNEL 4-114
4.2.4.5 VOLTAGE_CHANNEL 4-115
4.2.4.6 VOLTAGE_AND_ANGLE_CHANNEL 4-116

4.3 Plant Related Model APIs 4-118
4.3.1 ADD_PLANT_MODEL 4-118
4.3.2 CHANGE_PLMOD_CHRICN 4-120
4.3.3 CHANGE_PLMOD_CON 4-121
4.3.4 CHANGE_PLMOD_DATA 4-122
4.3.5 CHANGE_PLMOD_ICON 4-124
4.3.6 CHANGE_PLMOD_VAR 4-125
4.3.7 GMB_ADD_PLANT_MODEL 4-126
4.3.8 PLMOD_CONSISTENCY 4-128
4.3.9 PLMOD_PACK 4-129
4.3.10 PLMOD_REMOVE 4-130
4.3.11 PLMOD_STATUS 4-131
4.3.12 PLMOD_UNCONNECTED 4-132
4.3.13 PLMOD_USER 4-133

4.4 Wind Related Model APIs 4-134
4.4.1 ADD_WIND_MODEL 4-134
4.4.2 CHANGE_WNMOD_CHRICN 4-136
4.4.3 CHANGE_WNMOD_CON 4-137
4.4.4 CHANGE_WNMOD_ICON 4-138
4.4.5 CHANGE_WNMOD_VAR 4-139
4.4.6 WNMOD_CONSISTENCY 4-140
4.4.7 WNMOD_PACK 4-141
4.4.8 WNMOD_REMOVE 4-142
4.4.9 WNMOD_STATUS 4-143
4.4.10 WNMOD_UNCONNECTED 4-144
4.4.11 WNMOD_USER 4-145

4.5 Load Related Model APIs 4-146
4.5.1 ADD_LOAD_MODEL 4-146
4.5.2 CHANGE_LDMOD_CHRICN 4-148
4.5.3 CHANGE_LDMOD_CON 4-149
4.5.4 CHANGE_LDMOD_DATA 4-150
4.5.5 CHANGE_LDMOD_ICON 4-152
4.5.6 CHANGE_LDMOD_VAR 4-153
4.5.7 LDMOD_PACK 4-154
4.5.8 LDMOD_REMOVE 4-155
4.5.9 LDMOD_STATUS 4-156
4.5.10 LDMOD_STATUS2 4-157
4.5.11 LDMOD_UNCONNECTED 4-158
4.5.12 LDMOD_USER 4-159

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.6 Line Relay Model APIs 4-160
4.6.1 ADD_RELAY_MODEL 4-160
4.6.2 CHANGE_RLMOD_CHRICN 4-162
4.6.3 CHANGE_RLMOD_CON 4-163
4.6.4 CHANGE_RLMOD_DATA 4-164
4.6.5 CHANGE_RLMOD_ICON 4-166
4.6.6 CHANGE_RLMOD_VAR 4-167
4.6.7 RLMOD_PACK 4-168
4.6.8 RLMOD_REMOVE 4-169
4.6.9 RLMOD_STATUS 4-170
4.6.10 RLMOD_UNCONNECTED 4-171
4.6.11 RLMOD_USER 4-172

4.7 DC Line Model APIs 4-173
4.7.1 ADD_MTDCLINE_MODEL 4-173
4.7.2 ADD_TTDCLINE_MODEL 4-175
4.7.3 ADD_VSDCLINE_MODEL 4-177
4.7.4 CHANGE_MTDCLMOD_CHRICN 4-178
4.7.5 CHANGE_MTDCLMOD_CON 4-179
4.7.6 CHANGE_MTDCLMOD_DATA 4-180
4.7.7 CHANGE_MTDCLMOD_ICON 4-182
4.7.8 CHANGE_MTDCLMOD_VAR 4-183
4.7.9 CHANGE_TTDCLMOD_CHRICN 4-184
4.7.10 CHANGE_TTDCLMOD_CON 4-185
4.7.11 CHANGE_TTDCLMOD_DATA 4-186
4.7.12 CHANGE_TTDCLMOD_ICON 4-188
4.7.13 CHANGE_TTDCLMOD_VAR 4-189
4.7.14 CHANGE_VSDCLMOD_CHRICN 4-190
4.7.15 CHANGE_VSDCLMOD_CON 4-191
4.7.16 CHANGE_VSDCLMOD_DATA 4-192
4.7.17 CHANGE_VSDCLMOD_ICON 4-194
4.7.18 CHANGE_VSDCLMOD_VAR 4-195
4.7.19 DCLMOD_PACK 4-196
4.7.20 DCLMOD_UNCONNECTED 4-197
4.7.21 DCLMOD_USER 4-198
4.7.22 MTDCLMOD_REMOVE 4-199
4.7.23 MTDCLMOD_STATUS 4-200
4.7.24 TTDCLMOD_REMOVE 4-201
4.7.25 TTDCLMOD_STATUS 4-202
4.7.26 VSDCLMOD_REMOVE 4-203
4.7.27 VSDCLMOD_STATUS 4-204

4.8 FACTS Device Model APIs 4-205
4.8.1 ADD_FACTS_DEVICE_MODEL 4-205
4.8.2 CHANGE_FCTMOD_CHRICN 4-207
4.8.3 CHANGE_FCTMOD_CON 4-208
4.8.4 CHANGE_FCTMOD_DATA 4-209
4.8.5 CHANGE_FCTMOD_ICON 4-211
4.8.6 CHANGE_FCTMOD_VAR 4-212
4.8.7 FCTMOD_PACK 4-213
4.8.8 FCTMOD_REMOVE 4-214
4.8.9 FCTMOD_STATUS 4-215
4.8.10 FCTMOD_UNCONNECTED 4-216

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.8.11 FCTMOD_USER 4-217
4.8.12 GMB_ADD_FACTS_DEV_MODEL 4-218

4.9 Switched Shunt Model APIs 4-220
4.9.1 ADD_SWSHUNT_MODEL 4-220
4.9.2 CHANGE_SWSMOD_CHRICN 4-221
4.9.3 CHANGE_SWSMOD_CON 4-222
4.9.4 CHANGE_SWSMOD_ICON 4-223
4.9.5 CHANGE_SWSMOD_VAR 4-224
4.9.6 GMB_ADD_SWSHUNT_MODEL 4-225
4.9.7 SWSMOD_PACK 4-226
4.9.8 SWSMOD_REMOVE 4-227
4.9.9 SWSMOD_STATUS 4-228
4.9.10 SWSMOD_UNCONNECTED 4-229
4.9.11 SWSMOD_USER 4-230

4.10 Auxiliary Signal Model APIs 4-231
4.10.1 ADD_FCTSAUXSIGNAL_MODEL 4-231
4.10.2 ADD_MTDCAUXSIGNAL_MODEL 4-233
4.10.3 ADD_TTDCAUXSIGNAL_MODEL 4-235
4.10.4 ADD_VSDCAUXSIGNAL_MODEL 4-237
4.10.5 AUXMOD_PACK 4-239
4.10.6 AUXMOD_UNCONNECTED 4-240
4.10.7 AUXMOD_USER 4-241
4.10.8 CHANGE_FCTSAUXMOD_CHRICN 4-242
4.10.9 CHANGE_FCTSAUXMOD_CON 4-244
4.10.10 CHANGE_FCTSAUXMOD_DATA 4-245
4.10.11 CHANGE_FCTSAUXMOD_ICON 4-247
4.10.12 CHANGE_FCTSAUXMOD_VAR 4-249
4.10.13 CHANGE_MTDCAUXMOD_CHRICN 4-250
4.10.14 CHANGE_MTDCAUXMOD_CON 4-251
4.10.15 CHANGE_MTDCAUXMOD_DATA 4-252
4.10.16 CHANGE_MTDCAUXMOD_ICON 4-254
4.10.17 CHANGE_MTDCAUXMOD_VAR 4-255
4.10.18 CHANGE_TTDCAUXMOD_CHRICN 4-256
4.10.19 CHANGE_TTDCAUXMOD_CON 4-257
4.10.20 CHANGE_TTDCAUXMOD_DATA 4-258
4.10.21 CHANGE_TTDCAUXMOD_ICON 4-260
4.10.22 CHANGE_TTDCAUXMOD_VAR 4-261
4.10.23 CHANGE_VSDCAUXMOD_CHRICN 4-262
4.10.24 CHANGE_VSDCAUXMOD_CON 4-263
4.10.25 CHANGE_VSDCAUXMOD_DATA 4-264
4.10.26 CHANGE_VSDCAUXMOD_ICON 4-266
4.10.27 CHANGE_VSDCAUXMOD_VAR 4-267
4.10.28 FCTSAUXMOD_REMOVE 4-268
4.10.29 FCTSAUXMOD_STATUS 4-269
4.10.30 MTDCAUXMOD_REMOVE 4-270
4.10.31 MTDCAUXMOD_STATUS 4-271
4.10.32 TTDCAUXMOD_REMOVE 4-272
4.10.33 TTDCAUXMOD_STATUS 4-273
4.10.34 VSDCAUXMOD_REMOVE 4-274
4.10.35 VSDCAUXMOD_STATUS 4-275

4.11 Induction Machine Model APIs 4-276

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.11.1 ADD_INDMAC_MODEL 4-276
4.11.2 CHANGE_IMMOD_CHRICN 4-278
4.11.3 CHANGE_IMMOD_CON 4-279
4.11.4 CHANGE_IMMOD_ICON 4-280
4.11.5 CHANGE_IMMOD_VAR 4-281
4.11.6 IMMOD_PACK 4-282
4.11.7 IMMOD_REMOVE 4-283
4.11.8 IMMOD_STATUS 4-284
4.11.9 IMMOD_UNCONNECTED 4-285
4.11.10 IMMOD_USER 4-286

4.12 CCT Model APIs 4-287
4.12.1 ADD_CCTBRND_MODEL 4-287
4.12.2 ADD_CCT2WTD_MODEL 4-289
4.12.3 ADD_CCT3WTD_MODEL 4-291
4.12.4 ADD_CCTMCNP_MODEL 4-293
4.12.5 ADD_CCTBUSO_MODEL 4-295
4.12.6 ADD_CCT2DCO_MODEL 4-297
4.12.7 ADD_CCTMCNO_MODEL 4-299
4.12.8 ADD_CCTMSCO_MODEL 4-301
4.12.9 CHANGE_CCTBRNDMOD_CHRICN 4-303
4.12.10 CHANGE_CCTBRNDMOD_CON 4-304
4.12.11 CHANGE_CCTBRNDMOD_ICON 4-305
4.12.12 CHANGE_CCTBRNDMOD_VAR 4-306
4.12.13 CHANGE_CCT2WTDMOD_CHRICN 4-307
4.12.14 CHANGE_CCT2WTDMOD_CON 4-308
4.12.15 CHANGE_CCT2WTDMOD_ICON 4-309
4.12.16 CHANGE_CCT2WTDMOD_VAR 4-310
4.12.17 CHANGE_CCT3WTDMOD_CHRICN 4-311
4.12.18 CHANGE_CCT3WTDMOD_CON 4-313
4.12.19 CHANGE_CCT3WTDMOD_ICON 4-315
4.12.20 CHANGE_CCT3WTDMOD_VAR 4-317
4.12.21 CHANGE_CCTMCNPMOD_CHRICN 4-319
4.12.22 CHANGE_CCTMCNPMOD_CON 4-320
4.12.23 CHANGE_CCTMCNPMOD_ICON 4-321
4.12.24 CHANGE_CCTMCNPMOD_VAR 4-322
4.12.25 CHANGE_CCTBUSOMOD_CHRICN 4-323
4.12.26 CHANGE_CCTBUSOMOD_CON 4-324
4.12.27 CHANGE_CCTBUSOMOD_ICON 4-325
4.12.28 CHANGE_CCTBUSOMOD_VAR 4-326
4.12.29 CHANGE_CCT2DCOMOD_CHRICN 4-327
4.12.30 CHANGE_CCT2DCOMOD_CON 4-328
4.12.31 CHANGE_CCT2DCOMOD_ICON 4-329
4.12.32 CHANGE_CCT2DCOMOD_VAR 4-330
4.12.33 CHANGE_CCTMCNOMOD_CON 4-331
4.12.34 CHANGE_CCTMCNOMOD_CHRICN 4-332
4.12.35 CHANGE_CCTMCNOMOD_ICON 4-333
4.12.36 CHANGE_CCTMCNOMOD_VAR 4-334
4.12.37 CHANGE_CCTMSCOMOD_CHRICN 4-335
4.12.38 CHANGE_CCTMSCOMOD_CON 4-336
4.12.39 CHANGE_CCTMSCOMOD_ICON 4-337
4.12.40 CHANGE_CCTMSCOMOD_VAR 4-338
4.12.41 CCT2WTDMOD_REMOVE 4-339

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.42 CCT3WTDMOD_REMOVE 4-340
4.12.43 CCTMCNPMOD_REMOVE 4-341
4.12.44 CCTBUSOMOD_REMOVE 4-342
4.12.45 CCT2DCOMOD_REMOVE 4-343
4.12.46 CCTMCNOMOD_REMOVE 4-344
4.12.47 CCTMSCOMOD_REMOVE 4-345
4.12.48 CCTBRNDMOD_STATUS 4-346
4.12.49 CCT2WTDMOD_STATUS 4-347
4.12.50 CCT3WTDMOD_STATUS 4-348
4.12.51 CCTMCNPMOD_STATUS 4-349
4.12.52 CCTBUSOMOD_STATUS 4-350
4.12.53 CCT2DCOMOD_STATUS 4-351
4.12.54 CCTMCNOMOD_STATUS 4-352
4.12.55 CCTMSCOMOD_STATUS 4-353
4.12.56 CCTMOD_USER 4-354
4.12.57 CCTMOD_PACK 4-355

4.13 Disturbance APIs 4-356
4.13.1 DIST_3WIND_CLOSE 4-356
4.13.2 DIST_3WIND_FAULT 4-357
4.13.3 DIST_3WIND_TRIP 4-359
4.13.4 DIST_BRANCH_CLOSE 4-360
4.13.5 DIST_BRANCH_FAULT 4-361
4.13.6 DIST_BRANCH_TRIP 4-363
4.13.7 DIST_BUS_FAULT 4-364
4.13.8 DIST_BUS_TRIP 4-365
4.13.9 DIST_CLEAR_FAULT 4-366
4.13.10 DIST_MACHINE_TRIP 4-367
4.13.11 DIST_SCMU_FAULT 4-368
4.13.12 DIST_SPCB_FAULT 4-370
4.13.13 DIST_SPCB_FAULT_2 4-372

Chapter 5 - Subsystem Definition 5-1

5.1 Area Subsystem Definition APIs 5-2
5.1.1 ASYS 5-2
5.1.2 ASYSDEF 5-3
5.1.3 ASYSINIT 5-4

5.2 Bus Subsystem Definition APIs 5-5
5.2.1 BSYS 5-5
5.2.2 BSYSADD 5-7
5.2.3 BSYSDEF 5-8
5.2.4 BSYSINIT 5-9
5.2.5 BSYSMEM 5-10
5.2.6 BSYSO 5-11
5.2.7 BSYSRCL 5-12

5.3 Owner Subsystem Definition APIs 5-13
5.3.1 OSYS 5-13
5.3.2 OSYSDEF 5-14
5.3.3 OSYSINIT 5-15

5.4 Zone Subsystem Definition APIs 5-16
5.4.1 ZSYS 5-16

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

5.4.2 ZSYSDEF 5-17
5.4.3 ZSYSINIT 5-18

Chapter 6 - Diagram View APIs. 6-1

6.1 CHANGEDIAGCONTOUR 6-2
6.2 CHECKDIAGFILE 6-3
6.3 CLOSEDIAGFILE 6-4
6.4 DELETEDIAGCONTOUR 6-5
6.5 ENABLEDIAGCONTOUR 6-6
6.6 EXPORTBUSLOCFILE 6-7
6.7 EXPORTIMAGEFILE 6-8
6.8 GEXMBUS 6-9
6.9 GROWBUS 6-10
6.10 GROWDIAGRAM 6-11
6.11 GROWBUSLEVELS 6-12
6.12 GRPG 6-13
6.13 IMPORTDRAWFILE 6-14
6.14 IMPORTIMAGEFILE 6-15
6.15 NEWDIAGFILE 6-16
6.16 OPENACCFILE 6-17
6.17 OPENBUSLOCFILE 6-18
6.18 OPENDIAGFILE 6-19
6.19 PRINTDIAGFILE 6-20
6.20 REFRESHDIAGCONTOUR 6-21
6.21 REFRESHDIAGFILE 6-22
6.22 RENUMDIAGFILE 6-23
6.23 SAVEDIAGFILE 6-24
6.24 SETDIAGAUTOFILE 6-25
6.25 SETDIAGFONT 6-26
6.26 SETDIAGPREC 6-27
6.27 SETDIAGPROP 6-28
6.28 SETDIAGPROP_2 6-31
6.29 SETDIAGPROP_3 6-33
6.30 SETDIAGRESACCC 6-35
6.31 SETDIAGRESACCC_2 6-36
6.32 SETDIAGRESASCC 6-37
6.33 SETDIAGRESDATA 6-39
6.34 SETDIAGRESDATA_2 6-42

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.35 SETDIAGRESGDIF 6-43
6.36 SETDIAGRESDYN 6-45
6.37 SETDIAGRESIEC 6-46
6.38 SETDIAGRESPFLOWCHECK 6-48
6.39 SETDIAGRESPFLOWCHECK_2 6-52
6.40 SETDIAGRESPFLOWOPTIONS 6-55
6.41 SETDIAGRESPFLOWOPTIONS_2 6-57
6.42 SETDIAGRESREL 6-59
6.43 SETDIAGRESREL_2 6-61
6.44 SETDIAGRESSCGR 6-63
6.45 SETDIAGRESVRCS 6-66
6.46 SETDIAGRESTYPEASCC 6-69
6.47 SETDIAGRESTYPEDATA 6-70
6.48 SETDIAGRESTYPEDYN 6-71
6.49 SETDIAGRESTYPEGDIF 6-72
6.50 SETDIAGRESTYPEIEC 6-73
6.51 SETDIAGRESTYPEPFLOW 6-74
6.52 SETDIAGRESTYPEREL 6-75
6.53 SETDIAGRESTYPESCGR 6-76
6.54 UPDATEBUSLOCDIAGFILE 6-77
6.55 NEXTDIAGFILEDATASET 6-78
6.56 PREVDIAGFILEDATASET 6-79
6.57 SWITCHDIAGFILEDATASET 6-80

Chapter 7 - Single Element Data Retrieval 7-1

7.1 ARDAT 7-2
7.2 AREDAT 7-4
7.3 AREINT 7-5
7.4 ARENAM 7-6
7.5 ARENUM 7-7
7.6 AREUSE 7-8
7.7 ARITOJ 7-9
7.8 BRNCUR 7-10
7.9 BRNDAT 7-11
7.10 BRNDT2 7-12
7.11 BRNFLO 7-13
7.12 BRNINT 7-14

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.13 BRNMSC 7-15
7.14 BRNMVA 7-17
7.15 BRNSTT 7-18
7.16 BSYSISDEF 7-19
7.17 BUSDAT 7-20
7.18 BUSDT1 7-21
7.19 BUSDT2 7-23
7.20 BUSEXS 7-25
7.21 BUSINT 7-26
7.22 BUSMSM 7-27
7.23 CHKTRE 7-28
7.24 CHNVAL 7-29
7.25 DC2AUXMIND 7-30
7.26 DC2AUXMNAM 7-31
7.27 DC2DAT 7-32
7.28 DC2DAT_2 7-34
7.29 DC2INT 7-36
7.30 DC2INT_2 7-37
7.31 DC2MIND 7-38
7.32 DC2MNAM 7-39
7.33 DCNAUXMIND 7-40
7.34 DCNAUXMNAM 7-41
7.35 DCNCIN 7-42
7.36 DCNCIN_2 7-43
7.37 DCNDAT 7-44
7.38 DCNDAT_2 7-46
7.39 DCNINT 7-48
7.40 DCNINT_2 7-49
7.41 DCNMIND 7-50
7.42 DCNMNAM 7-51
7.43 DSCVAL 7-52
7.44 DSIVAL 7-53
7.45 DSRVAL 7-54
7.46 FCDAUXMIND 7-55
7.47 FCDAUXMNAM 7-56
7.48 FCDDAT 7-57
7.49 FCDDAT_2 7-60

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.50 FCDINT 7-63
7.51 FCDINT_2 7-64
7.52 FCDMIND 7-65
7.53 FCDMNAM 7-66
7.54 FXSDT1 7-67
7.55 FXSDT2 7-68
7.56 FXSINT 7-70
7.57 GENCNV 7-71
7.58 GENDAT 7-72
7.59 GENDT1 7-73
7.60 GET_MSTATE 7-74
7.61 GNECHR 7-75
7.62 GNEDAT 7-76
7.63 GNEINT 7-77
7.64 INI2DC 7-78
7.65 INDDT1 7-79
7.66 INDDT2 7-81
7.67 INDINT 7-82
7.68 INIBRN 7-84
7.69 INIBRX 7-85
7.70 INIBUS 7-86
7.71 INIBUX 7-87
7.72 INIFAX 7-88
7.73 INIFXS 7-89
7.74 INIIND 7-90
7.75 INILOD 7-91
7.76 INIMAC 7-92
7.77 INIMDC 7-93
7.78 INIMSL 7-94
7.79 ININAM 7-95
7.80 ININAX 7-96
7.81 INITIE 7-97
7.82 INITIX 7-98
7.83 INIVSC 7-99
7.84 ISLAND 7-100
7.85 ITERAT 7-101
7.86 LMODIND 7-102

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.87 LMODNAM 7-104
7.88 LODCNV 7-105
7.89 LODDT1 7-106
7.90 LODDT2 7-108
7.91 LODIND 7-110
7.92 LODINT 7-111
7.93 MACDAT 7-112
7.94 MACDT2 7-114
7.95 MACIND 7-116
7.96 MACINT 7-117
7.97 MACSTT 7-118
7.98 MAXMSM 7-119
7.99 MDLFLDCNT 7-120
7.100 MDLFOLDER 7-121
7.101 MDLIND 7-122
7.102 MDLLIBCNT 7-124
7.103 MDLPGMCNT 7-125
7.104 MDLNAM 7-126
7.105 NATONO 7-127
7.106 NOTONA 7-128
7.107 NXT2DC 7-129
7.108 NXTBRN 7-130
7.109 NXTBRN3 7-131
7.110 NXTBUS 7-132
7.111 NXTFAX 7-133
7.112 NXTFXS 7-134
7.113 NXTIND 7-135
7.114 NXTLOD 7-136
7.115 NXTMAC 7-137
7.116 NXTMDC 7-138
7.117 NXTMSL 7-139
7.118 NXTTIE 7-140
7.119 NXTTIE3 7-141
7.120 NXTVSC 7-142
7.121 OKSTRT 7-143
7.122 OPENUN 7-144
7.123 ORDBUS 7-145

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.124 OWNDAT 7-146
7.125 OWNNAM 7-148
7.126 OWNNUM 7-149
7.127 OWNUSE 7-150
7.128 PRMDAT 7-151
7.129 PRMINT 7-152
7.130 RMODIND 7-153
7.131 RMODNAM 7-155
7.132 SC3WND 7-156
7.133 SCBRN2 7-157
7.134 SCBUS2 7-158
7.135 SCDONE 7-160
7.136 SCINIT 7-161
7.137 SCMAC2 7-162
7.138 SELCTR 7-163
7.139 SFILES 7-164
7.140 SLMODIND 7-165
7.141 SLMODNAM 7-167
7.142 SOLVED 7-168
7.143 SOLVED_CA 7-169
7.144 STAREA 7-170
7.145 STBSKV 7-171
7.146 STZONE 7-172
7.147 SWSBLK 7-173
7.148 SWSBLZ 7-174
7.149 SWSBLZ2 7-175
7.150 SWSDAT 7-176
7.151 SWSDT1 7-177
7.152 SWSINT 7-178
7.153 SWSMIND 7-179
7.154 SWSMNAM 7-180
7.155 SYSMSM 7-181
7.156 SYSMVA 7-182
7.157 SYSTOT 7-183
7.158 TITLDT 7-184
7.159 TOTBUS 7-185
7.160 TR3DAT 7-186

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.161 TR3DT2 7-187
7.162 TR3INT 7-189
7.163 TR3NAM 7-191
7.164 TRNDAT 7-192
7.165 TRXDAT 7-193
7.166 VSCAUXMIND 7-194
7.167 VSCAUXMNAM 7-195
7.168 VSCCDT 7-196
7.169 VSCCIN 7-197
7.170 VSCINT 7-198
7.171 VSCMIND 7-199
7.172 VSCMNAM 7-200
7.173 WINDMIND 7-201
7.174 WINDMNAM 7-203
7.175 WNDDAT 7-204
7.176 WNDDT2 7-206
7.177 WNDINT 7-207
7.178 XFRDAT 7-208
7.179 XFRINT 7-210
7.180 XFRNAM 7-212
7.181 ZNDAT 7-213
7.182 ZNITOJ 7-215
7.183 ZONNAM 7-216
7.184 ZONNUM 7-217
7.185 ZONUSE 7-218

Chapter 8 - Subsystem Data Retrieval 8-1

8.1 Introduction 8-1
8.2 Character Data 8-2
8.2.1 GET_CHAR_LENGTH 8-2
8.3 Bus Data 8-3
8.3.1 aBusCount 8-3
8.3.2 aBusInt 8-4
8.3.3 aBusReal 8-6
8.3.4 aBusCplx 8-8
8.3.5 aBusChar 8-10
8.3.6 aBusTypes 8-11
8.4 Plant Bus Data 8-12
8.4.1 aGenBusCount 8-12
8.4.2 aGenBusInt 8-13

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.4.3 aGenBusReal 8-15
8.4.4 aGenBusCplx 8-17
8.4.5 aGenBusChar 8-19
8.4.6 aGenBusTypes 8-21

8.5 Machine Data 8-22
8.5.1 aMachCount 8-22
8.5.2 aMachInt 8-23
8.5.3 aMachReal 8-25
8.5.4 aMachCplx 8-28
8.5.5 aMachChar 8-30
8.5.6 aMachTypes 8-32

8.6 Load Bus Data 8-33
8.6.1 aLodBusCount 8-33
8.6.2 aLodBusInt 8-35
8.6.3 aLodBusReal 8-37
8.6.4 aLodBusCplx 8-40
8.6.5 aLodBusChar 8-43
8.6.6 aLodBusTypes 8-45

8.7 Load Data 8-46
8.7.1 aLoadCount 8-46
8.7.2 aLoadInt 8-47
8.7.3 aLoadReal 8-49
8.7.4 aLoadCplx 8-51
8.7.5 aLoadChar 8-53
8.7.6 aLoadTypes 8-55

8.8 Fixed Shunt Bus Data 8-56
8.8.1 aFxShntBusCount 8-56
8.8.2 aFxShntBusInt 8-57
8.8.3 aFxShntBusReal 8-59
8.8.4 aFxShntBusCplx 8-61
8.8.5 aFxShntBusChar 8-63
8.8.6 aFxShntBusTypes 8-65

8.9 Fixed Shunt Data 8-66
8.9.1 aFxShuntCount 8-66
8.9.2 aFxShuntInt 8-67
8.9.3 aFxShuntReal 8-69
8.9.4 aFxShuntCplx 8-71
8.9.5 aFxShuntChar 8-73
8.9.6 aFxShuntTypes 8-75

8.10 Switched Shunt Data 8-76
8.10.1 aSwshCount 8-76
8.10.2 aSwshInt 8-77
8.10.3 aSwshReal 8-79
8.10.4 aSwshCplx 8-83
8.10.5 aSwshChar 8-85
8.10.6 aSwshTypes 8-87

8.11 Branch Flow Data 8-88
8.11.1 aFlowCount 8-89
8.11.2 aFlowInt 8-90

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.11.3 aFlowReal 8-92
8.11.4 aFlowCplx 8-95
8.11.5 aFlowChar 8-97
8.11.6 aFlowTypes 8-99

8.12 Branch Data 8-100
8.12.1 aBrnCount 8-101
8.12.2 aBrnInt 8-103
8.12.3 aBrnReal 8-105
8.12.4 aBrnCplx 8-109
8.12.5 aBrnChar 8-111
8.12.6 aBrnTypes 8-113

8.13 Two-Winding Transformer Data 8-114
8.13.1 aTrnCount 8-115
8.13.2 aTrnInt 8-116
8.13.3 aTrnReal 8-119
8.13.4 aTrnCplx 8-123
8.13.5 aTrnChar 8-126
8.13.6 aTrnTypes 8-128

8.14 Three-Winding Transformer Data 8-129
8.14.1 aTr3Count 8-130
8.14.2 aTr3Int 8-131
8.14.3 aTr3Real 8-133
8.14.4 aTr3Cplx 8-135
8.14.5 aTr3Char 8-138
8.14.6 aTr3Types 8-140

8.15 Three-Winding Transformer Winding Data 8-141
8.15.1 aWndCount 8-142
8.15.2 aWndInt 8-144
8.15.3 aWndReal 8-147
8.15.4 aWndCplx 8-152
8.15.5 aWndChar 8-154
8.15.6 aWndTypes 8-157

8.16 Two-Terminal dc Line Data 8-158
8.16.1 a2TrmDcCount 8-159
8.16.2 a2TrmDcInt 8-160
8.16.3 a2TrmDcReal 8-162
8.16.4 a2TrmDcCplx 8-166
8.16.5 a2TrmDcChar 8-168
8.16.6 a2TrmDcTypes 8-170

8.17 Two-Terminal dc Line Converter Data 8-171
8.17.1 a2TrmDcConvCount 8-172
8.17.2 a2TrmDcConvInt 8-173
8.17.3 a2TrmDcConvReal 8-175
8.17.4 a2TrmDcConvCplx 8-178
8.17.5 a2TrmDcConvChar 8-180
8.17.6 a2TrmDcConvTypes 8-182

8.18 Multi-Terminal dc Line Data 8-183
8.18.1 aMultiTrmDcCount 8-184
8.18.2 aMultiTrmDcInt 8-185

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.18.3 aMultiTrmDcReal 8-187
8.18.4 aMultiTrmDcCplx 8-189
8.18.5 aMultiTrmDcChar 8-191
8.18.6 aMultiTrmDcTypes 8-193

8.19 Multi-Terminal dc Line Converter Data 8-194
8.19.1 aMultiTrmDcConvCount 8-195
8.19.2 aMultiTrmDcConvInt 8-196
8.19.3 aMultiTrmDcConvReal 8-198
8.19.4 aMultiTrmDcConvCplx 8-201
8.19.5 aMultiTrmDcConvChar 8-203
8.19.6 aMultiTrmDcConvTypes 8-205

8.20 VSC dc Line Data 8-206
8.20.1 aVscDcCount 8-207
8.20.2 aVscDcInt 8-208
8.20.3 aVscDcReal 8-210
8.20.4 aVscDcCplx 8-212
8.20.5 aVscDcChar 8-214
8.20.6 aVscDcTypes 8-216

8.21 VSC dc Line Converter Data 8-217
8.21.1 aVscDcConvCount 8-218
8.21.2 aVscDcConvInt 8-219
8.21.3 aVscDcConvReal 8-221
8.21.4 aVscDcConvCplx 8-224
8.21.5 aVscDcConvChar 8-226
8.21.6 aVscDcConvTypes 8-228

8.22 FACTS Device Data 8-229
8.22.1 aFactsCount 8-230
8.22.2 aFactsInt 8-231
8.22.3 aFactsReal 8-233
8.22.4 aFactsCplx 8-236
8.22.5 aFactsChar 8-238
8.22.6 aFactsTypes 8-240

8.23 FACTS Device Bus Data 8-241
8.23.1 aFactsBusCount 8-242
8.23.2 aFactsBusInt 8-244
8.23.3 aFactsBusReal 8-246
8.23.4 aFactsBusCplx 8-249
8.23.5 aFactsBusChar 8-251
8.23.6 aFactsBusTypes 8-253

8.24 Area Data 8-254
8.24.1 aAreaCount 8-254
8.24.2 aAreaInt 8-255
8.24.3 aAreaReal 8-257
8.24.4 aAreaCplx 8-260
8.24.5 aAreaChar 8-262
8.24.6 aAreaTypes 8-264

8.25 Owner Data 8-265
8.25.1 aOwnerCount 8-265
8.25.2 aOwnerInt 8-266

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.25.3 aOwnerReal 8-268
8.25.4 aOwnerCplx 8-271
8.25.5 aOwnerChar 8-273
8.25.6 aOwnerTypes 8-275

8.26 Zone Data 8-276
8.26.1 aZoneCount 8-276
8.26.2 aZoneInt 8-277
8.26.3 aZoneReal 8-279
8.26.4 aZoneCplx 8-282
8.26.5 aZoneChar 8-284
8.26.6 aZoneTypes 8-286

8.27 Induction Machine Bus Data 8-287
8.27.1 aIndMacBusCount 8-287
8.27.2 aIndMacBusInt 8-289
8.27.3 aIndMacBusReal 8-291
8.27.4 aIndMacBusCplx 8-293
8.27.5 aIndMacBusChar 8-295
8.27.6 aIndMacBusTypes 8-297

8.28 Induction Machine Data 8-298
8.28.1 aIndMacCount 8-298
8.28.2 aIndMacInt 8-300
8.28.3 aIndMacReal 8-302
8.28.4 aIndMacCplx 8-305
8.28.5 aIndMacChar 8-307
8.28.6 aIndMacTypes 8-309

Chapter 9 - Results Data Retrieval Modules 9-1

9.1 PSSARRAYS module 9-2
9.1.1 ACCC_SUMMARY 9-3
9.1.2 ACCC_SOLUTION 9-5
9.1.3 ACCC_SUMMARY_REPORT 9-7
9.1.4 SENSITIVITY_FLOW_TO_MW 9-8
9.1.5 ACCC_SOLUTION_REPORT 9-11
9.1.6 ACCC_VIOLATIONS_REPORT 9-12
9.1.7 ASCC_CURRENTS 9-13
9.1.8 IECS_CURRENTS 9-17
9.1.9 OTDF_FACTORS 9-22
9.1.10 PV_SUMMARY 9-24
9.1.11 PV_SUMMARY_REPORT 9-26
9.1.12 PV_SOLUTION 9-27
9.1.13 PV_SOLUTION_REPORT 9-28
9.1.14 QV_SOLUTION 9-29
9.1.15 QV_SOLUTION_REPORT 9-30
9.1.16 QV_SUMMARY 9-31
9.1.17 QV_SUMMARY_REPORT 9-33

9.2 PSSEXCEL module 9-34
9.2.1 ACCC 9-35
9.2.2 iec_data_file 9-37
9.2.3 pv 9-38
9.2.4 qv 9-40

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

9.3 EXCELPY module 9-41
9.3.1 Methods 9-44
9.3.2 Workbook Functions 9-50
9.4 GICDATA module 9-51

Chapter 10 - Graphical Analysis Output 10-1

10.1 POLY_PRINT 10-2
10.2 PV_PRINT 10-3
10.3 QV_PRINT 10-5
10.4 SETFULLVIEWSCALE 10-7
10.5 SETFULLVIEWGRAPHAREA 10-9

Chapter 11 - PSSPLOT Module 11-1

11.1 INSERTPAGE 11-2
11.2 INSERTPLOT 11-3
11.3 DELETESELPLOT 11-4
11.4 EXPORTPLOT 11-5
11.5 PRINTSELPLOT 11-6
11.6 SHOWPLOTEDITOR 11-7
11.7 NEWPLOTBOOK 11-8
11.8 OPENCHANDATAFILE 11-9
11.9 DRAGDROPPLOTDATA 11-10
11.10 SETSELECTEDPAGE 11-11
11.11 SETSELECTEDPLOT 11-12

Chapter 12 - Non-Engineering Functions 12-1

12.1 ACTIVATE_LICENSE 12-2
12.2 ALERT 12-3
12.3 BEGINREPORT 12-4
12.4 FILEIN 12-5
12.5 IOFLUSH 12-6
12.6 GETBATDEFAULTS 12-7
12.7 GETDEFAULTCHAR 12-8
12.8 GETDEFAULTINT 12-9
12.9 GETDEFAULTREAL 12-10
12.10 PAGEREPORT 12-11
12.11 PROGRESS 12-12
12.12 PROMPT 12-13
12.13 PSSEINIT 12-14

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

12.14 PSSEVERSION 12-15
12.15 REFRESHGUI 12-16
12.16 REPORT 12-17
12.17 RUNIPLANFILE 12-18
12.18 RUNRSPNSFILE 12-19
12.19 STARTRECORDING 12-20
12.20 STOPRECORDING 12-21
12.21 USERIN 12-22

Chapter 13 - Scenario and Event Study APIs 13-1

13.1 OPENEVENTSTUDYFILE 13-2
13.2 SAVEEVENTSTUDYFILE 13-3
13.3 CLOSEEVENTSTUDYFILE 13-4
13.4 SETACTIVEEVENTSTUDY 13-5
13.5 RUNPOWERFLOWEVENTSTUDY 13-6
13.6 RUNDYNAMICEVENTSTUDY 13-7
13.7 NEWSCENARIOFILE 13-8
13.8 OPENSCENARIOFILE 13-9
13.9 SAVESCENARIOFILE 13-10
13.10 CLOSESCENARIOFILE 13-11
13.11 SCENARIONEW 13-12
13.12 SCENARIOOPEN 13-13
13.13 SCENARIOSAVE 13-14
13.14 SCENARIOCLOSE 13-15
13.15 SCENARIOISVALID 13-16
13.16 SCENARIOISDIRTY 13-17
13.17 SCENARIOZIPACTIVE 13-18
13.18 SCENARIOGETATTRS 13-19
13.19 SCENARIOSETATTRS 13-20
13.20 SCENARIOLENGTHCOMMENTS 13-21
13.21 SCENARIOGETCOMMENTS 13-22
13.22 SCENARIOSETCOMMENTS 13-23
13.23 SCENARIOGROUPOPEN 13-24
13.24 SCENARIOGROUPRENAME 13-25
13.25 SCENARIOGROUPLENGTHCOMMENTS 13-26
13.26 SCENARIOGROUPSETCOMMENTS 13-27
13.27 SCENARIOGROUPCOUNT 13-28

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.28 SCENARIOGROUPLIST 13-29
13.29 SCENARIOGROUPADD 13-30
13.30 SCENARIOGROUPREMOVE 13-31
13.31 SCENARIOFILEGETATTRS 13-32
13.32 SCENARIOFILELENGTHCOMMENTS 13-33
13.33 SCENARIOFILEGETCOMMENTS 13-34
13.34 SCENARIOFILESETCOMMENTS 13-35
13.35 SCENARIOFILECOUNT 13-36
13.36 SCENARIOFILELIST 13-37
13.37 SCENARIOFILEADD 13-38
13.38 SCENARIOFILEREMOVE 13-39
13.39 SCENARIOMODELGROUP 13-40
13.40 SCENARIOMOVEFILE 13-41
13.41 SCENARIOCOPYFILE 13-42
13.42 SCENARIOSETOPT 13-43
13.43 SCENARIOGETOPT 13-45
13.44 SCENARIOTRACKFILENUM 13-47
13.45 SCENARIOTRACKFILELIST 13-48
13.46 SCENARIOTRACKFILEADD 13-49
13.47 SCENARIOZIP 13-50
13.48 SCENARIOZIPOPEN 13-51
13.49 SCENARIOZIPSAVE 13-52
13.50 SCENARIOUNZIP 13-53
13.51 SCENARIOUNZIPPROBLEMS 13-54
13.52 SCENARIOUNZIPSOLUTIONS 13-55
13.53 SCENARIOFILESETATTRS 13-56
13.54 SCENARIOFILERENAME 13-57

Chapter 14 - Other Python Modules 14-1

14.1 Modules Supplied with a GUI 14-2
14.2 Modules that Define Functions 14-3

Index

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

SIEMENS

Preface

This document defines the PSS®E Application Program Interface (API) to the various engineering functions in PSS®E.

Structure of Manual

The manual is broken into chapters, each chapter representing a category of API routines. Some chapters have sub-sections. Each API routine is described in an individual section.

How API Routines are Described

The description for each API routine starts with a brief explanation of its function, followed by one or more language specific syntaxes for calling the routine, followed by a detailed description of the arguments used in those calling sequences. The nomenclature used in the argument descriptions is most consistent with the Fortran programming language.

The syntaxes for calling the API routines will include at least one of the PSS®E automation file languages, and will usually include a syntax for use from a Fortran program. The set of syntaxes will include all the PSS®E automation file languages where the API routine can be used. Some general comments about those syntaxes may be helpful, as follows:

PSS®E Automation File Languages

Batch commands: this is a free-format input syntax. Items can be separated by a comma or spaces, strings need only be quoted if they contain spaces or commas, omitted items can be indicated by consecutive commas, arrays are entered as a sequence of scalar values, and a semi-colon can be used to terminate the command and return the remaining items to default values (it must be a separate data item, i.e., it must be separated from the previous data item by a space or comma). Except for unquoted filenames, a forward slash will terminate a line. Commands may be continued over multiple lines. The sequence @! at the beginning of the line will cause that line to be ignored.

Python commands: Python is an open source language; its syntax is well documented. Some of are arguments are defined as arrays (in keeping with Fortran nomenclature, as mentioned above). Python does not have arrays, and in general you should interpret arrays to mean lists (although tuples will usually work on input). Arrays that have some elements as input and some as output are in the Fortran syntax represented in the Python syntax as two separate lists. Keywords can be used for arguments, and any argument can be omitted.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Fortran call statements: Fortran is a well-documented language (where it matters, we will use the Fortran90 standard). We will not use features that require explicit interfaces without stating so clearly in the description. Unless specifically stated, arguments cannot be omitted.

IPLAN commands: the IPLAN language is described in the IPLAN manual. For purposes of the CALL statement it resembles Fortran. Unless specifically stated, arguments cannot be omitted.

Return Codes / Error Codes

Most API routines return a integer return code, where zero indicates success. In Python, where multiple values are returned (a tuple), the return code will always be the first value. When non-zero error codes are returned, other returned values are set to None in Python, and are either uninitialized or unchanged (for input/output values) in other syntaxes.

The Python interface can be made to throw exceptions instead of returning error codes values (the default). Enter:

```
psspy.throwPsseExceptions = True
```

Default Values

When values are omitted, special “no input” values are supplied. These special values – for each variable type – are meant to indicate that “no input” was supplied. Many of the API routines permit this and will use documented default values when “no input” values are supplied.

The actual “no input” values can be retrieved from the API routines GETDEFAULTINT, GETDEFAULTREAL, and GETDEFAULTCHAR (capitalization depends on syntax). Users should never depend on the actual values returned, by these functions. They should be used only for supplying values to the PSS®E API routines.

When automation files are recorded, items that have the default value are omitted in batch commands (i.e. recorded as ",,") and recorded as _i, _f, _s, or _o (for integer, real/float, character/string, and filename, respectively). In the PSS®E embedded Python environment there will be local variables _i, _f, and _s set to the values returned from GETDEFAULTxxx routines mentioned above (_o will be treated as blanks). Outside of PSS®E it is the user’s responsibility to set those variables.

Recording

Two recording formats are available: batch commands and Python.

Not all API routines are recordable. Routines that only retrieve data do not record, for example. If an API routine is recordable, it will have a batch command format specified; routines without a batch command format specified are not recordable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Python Modules

Most of the PSS®E API is defined in the module psspy. Chapters 9, 11, 14, and 15 discuss routines defined in other modules. In keeping with normal Python documentation the namespace is not included as part of the syntax for the individual routines. This may seem confusing to newcomers to Python. For those whose first exposure to Python is inside of PSS®E, be aware that, normally, you will use the, say, case API routine by entering:

```
psspy.case(filename)
```

not

```
case(filename)
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

SIEMENS

Chapter 1

Power Flow Operation

This chapter describes the API routines corresponding to the operational functions of the PSS®E Power Flow and Fault Analysis. Also included are those routines that control basic program operations and the Open Access option.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.1 ACCC

This API routine is obsolete. It has been replaced by the API routine ACCC_WITH_DSP_3, and is implemented by a call to ACCC_WITH_DSP_3.

Use this API routine to run the AC contingency calculation function (activity ACCC).

Batch command syntax:

```
BAT_ACCC  TOL  OPTIONS(1)...OPTIONS(7)  DFXFILE  ACCFILE  THRFILE
```

Python syntax:

```
ierr = accc(tol, options, dfxfile, accfile, thrfile)
```

Fortran syntax:

```
CALL ACCCAPI(TOL, OPTIONS, DFXFILE, ACCFILE, THRFILE, IERR)
```

where:

Real TOL	Is the mismatch tolerance (input; Newton solution convergence tolerance, TOLN, by default).
Integer OPTIONS(7)	Is an array of seven elements specifying solution options (input). The values are as follows:
	OPTIONS(1) — tap adjustment flag (tap adjustment option setting by default).
	OPTIONS(1) = 0 — disable.
	OPTIONS(1) = 1 — enable stepping adjustment.
	OPTIONS(1) = 2 — enable direct adjustment.
	OPTIONS(2) — area interchange adjustment flag (area interchange adjustment option setting by default).
	OPTIONS(2) = 0 — disable.
	OPTIONS(2) = 1 — enable using tie line flows only in calculating area interchange.
	OPTIONS(2) = 2 — enable using tie line flows and loads in calculating area interchange.
	OPTIONS(3) — phase shift adjustment flag (phase shift adjustment option setting by default).
	OPTIONS(3) = 0 — disable.
	OPTIONS(3) = 1 — enable.
	OPTIONS(4) — dc tap adjustment flag (dc tap adjustment option setting by default).
	OPTIONS(4) = 0 — disable.
	OPTIONS(4) = 1 — enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTIONS(5) switched shunt adjustment flag (switched shunt adjustment option setting by default). OPTIONS(5) = 0 disable. OPTIONS(5) = 1 enable. OPTIONS(5) = 2 enable continuous mode, disable discrete mode. OPTIONS(6) solution method flag (0 by default). OPTIONS(6) = 0 FDNS. OPTIONS(6) = 1 FNSL. OPTIONS(6) = 2 optimized FDNS. OPTIONS(7) non-divergent solution flag (non-divergent solution option setting by default). OPTIONS(7) = 0 disable. OPTIONS(7) = 1 enable.
Character DFXFILE*260	Is the name of the Distribution Factor Data File (input; no default allowed).
Character ACCFILE*260	Is the name of the Contingency Solution Output file (input; no default allowed).
Character THRFILE*260	Is the name of the Load Throwover Data file; blank for none (input; blank by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 as for ACCC_WITH_DSP_3.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.2 ACCC_MULTIPLE_RUN_REPORT

The API is obsolete. It has been replaced by ACCC_MULTIPLE_RUN_REPORT_2 and is implemented by a call to ACCC_MULTIPLE_RUN_REPORT_2.

Use this API, the AC Contingency Multiple Run Report function, to report the results of up to twenty-two executions of the AC Contingency Calculation function.

Batch command syntax:

```
BAT_ACCC_MULTIPLE_RUN_REPORT  OPTIONS(1)...OPTIONS(14)
VALUES(1)...VALUES(7)  NFILES  ACFILES(1)...ACFILES(NFILES)
```

Python syntax:

```
ierr = accc_multiple_run_report(options, values, nfiles, acfiles)
```

Fortran syntax:

```
CALL ACCC_MULTIPLE_RUN_REPORT(OPTIONS, VALUES, NFILES, ACFILES, IERR)
```

where:

Integer OPTIONS(14) Is an array of fourteen elements defining reporting options (input). The values are as follows.

OPTIONS(1)	column headings code (1 by default).
OPTIONS(1) = 1	Contingency Solution Output File names.
OPTIONS(1) = 2	Saved Case File names as contained in the Contingency Solution Output Files.
OPTIONS(2)	base case rating set (rating set program option setting by default).
OPTIONS(2) = 1	rate A.
OPTIONS(2) = 2	rate B.
OPTIONS(2) = 3	rate C.
OPTIONS(3)	contingency case rating set (rating set program option setting by default).
OPTIONS(3) = 1	rate A.
OPTIONS(3) = 2	rate B.
OPTIONS(3) = 3	rate C.
OPTIONS(4)	print monitored elements summary report (0 by default).
OPTIONS(4) = 0	no.
OPTIONS(4) = 1	yes.
OPTIONS(5)	print missing monitored elements report (0 by default).
OPTIONS(5) = 0	no.
OPTIONS(5) = 1	yes.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(6)	print missing monitored voltage buses report (0 by default).
OPTIONS(6) = 0	no.
OPTIONS(6) = 1	yes.
OPTIONS(7)	print contingency legend (2 by default).
OPTIONS(7) = 0	no.
OPTIONS(7) = 1	complete master legend report only.
OPTIONS(7) = 2	reduced legend with each table.
OPTIONS(7) = 3	both master legend report and legend with each table.
OPTIONS(8)	print missing contingencies report (0 by default).
OPTIONS(8) = 0	no.
OPTIONS(8) = 1	yes.
OPTIONS(9)	print non-converged contingencies report (0 by default).
OPTIONS(9) = 0	no.
OPTIONS(9) = 1	yes.
OPTIONS(10)	print loading violation reports (0 by default).
OPTIONS(10) = 0	no.
OPTIONS(10) = 1	base case and worst case contingency violations only.
OPTIONS(10) = 2	base case and all contingency violations only.
OPTIONS(10) = 3	base case, worst case contingency, and all contingency violations.
OPTIONS(11)	print voltage violation reports (0 by default).
OPTIONS(11) = 0	no.
OPTIONS(11) = 1	base case and worst case contingency violations only.
OPTIONS(11) = 2	base case and all contingency violations only.
OPTIONS(11) = 3	base case, worst case contingency, and all contingency violations.
OPTIONS(12)	option for processing interfaces in loading violation reports (1 by default).
OPTIONS(12) = 0	exclude interfaces from loading violation checking and reporting.
OPTIONS(12) = 1	check and report interface loading violations.
OPTIONS(13)	in loading violation reports, exclude monitored branches and interfaces that show loading violations in the base case from being checked and reported in contingency case reports (0 by default).
OPTIONS(13) = 0	no.
OPTIONS(13) = 1	yes.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTIONS(14)	in voltage range violation reports, exclude monitored buses that show voltage range violations in the base case from the corresponding check in contingency case reports (0 by default).
	OPTIONS(14) = 0	no.
	OPTIONS(14) = 1	yes.
Real VALUES(7)	Is an array of seven elements specifying reporting parameters (input). The values are as follows.	
	VALUES(1)	bus mismatch converged tolerance (MW or Mvar) (0.5 by default).
	VALUES(2)	system mismatch converged tolerance (MVA) (5.0 by default).
	VALUES(3)	in loading violation reports, percent of flow rating for reporting (100.0 by default).
	VALUES(4)	in the worst case contingency loading violation reports, percent of flow rating for counting (≥ VALUES(3)) (100.0 by default).
	VALUES(5)	in loading violation reports, minimum contingency case flow change from base case value for reporting and counting (0.0 by default).
	VALUES(6)	in loading violation reports, minimum contingency case percent loading increase from base case value for reporting and counting (0.0 by default).
	VALUES(7)	minimum contingency case voltage change for reporting and counting in voltage range violations (0.0 by default).
Integer NFILES	Is the number of Contingency Solution Output Files to be processed (1 through 22) (input; no default allowed).	
Character ACFILES (NFILES)*260	Is an array of NFILES names of Contingency Solution Output Files (input; no defaults allowed).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid OPTIONS value.
IERR = 2	invalid VALUES value.
IERR = 3	invalid NFILES value.
IERR = 4	the percent for counting is less than the percent for reporting.
IERR = 5	file is not in the form of a current Contingency Solution Output File.
IERR = 6	error opening a Contingency Solution Output File.
IERR = 7	error reading a Contingency Solution Output File.
IERR = 8	error opening a temporary file.
IERR = 9	error opening a subfile in the temporary file.
IERR = 10	error reading or writing the temporary file.
IERR = 11	error closing a subfile in the temporary file.
IERR = 12	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.3 ACCC_MULTIPLE_RUN_REPORT_2

This API is the second release of AC Contingency Multiple Run Report function. Use this API, the AC Contingency Multiple Run Report function, to report the results of up to twenty-two executions of the AC Contingency Calculation function.

Batch command syntax:

```
BAT_ACCC_MULTIPLE_RUN_REPORT_2  OPTIONS(1)...OPTIONS(16)
VALUES(1)...VALUES(7)  NFILES  ACFILES(1)...ACFILES(NFILES)
```

Python syntax:

```
ierr = accc_multiple_run_report_2(options, values, nfiles, acfiles)
```

Fortran syntax:

```
CALL ACCC_MULTIPLE_RUN_REPORT_2(OPTIONS, VALUES, NFILES, ACFILES,
IERR)
```

where:

Integer OPTIONS(16)	Is an array of sixteen elements defining reporting options (input). The values are as follows.	
	OPTIONS(1)	column headings code (1 by default).
	OPTIONS(1) = 1	Contingency Solution Output File names.
	OPTIONS(1) = 2	Saved Case File names as contained in the Contingency Solution Output Files.
	OPTIONS(2)	base case rating set (rating set program option setting by default).
	OPTIONS(2) = 1	rate A.
	OPTIONS(2) = 2	rate B.
	OPTIONS(2) = 3	rate C.
	OPTIONS(3)	contingency case rating set (rating set program option setting by default).
	OPTIONS(3) = 1	rate A.
	OPTIONS(3) = 2	rate B.
	OPTIONS(3) = 3	rate C.
	OPTIONS(4)	base case voltage limit.
	OPTIONS(4) = 1	normal.
	OPTIONS(4) = 2	emergency.
	OPTIONS(5)	contingency case voltage limit.
	OPTIONS(5) = 1	normal.
	OPTIONS(5) = 2	emergency.
	OPTIONS(6)	print monitored elements summary report (0 by default).
	OPTIONS(6) = 0	no.
	OPTIONS(6) = 1	yes.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(7)	print missing monitored elements report (0 by default).
OPTIONS(7) = 0	no.
OPTIONS(7) = 1	yes.
OPTIONS(8)	print missing monitored voltage buses report (0 by default).
OPTIONS(8) = 0	no.
OPTIONS(8) = 1	yes.
OPTIONS(9)	print contingency legend (2 by default).
OPTIONS(9) = 0	no.
OPTIONS(9) = 1	complete master legend report only.
OPTIONS(9) = 2	reduced legend with each table.
OPTIONS(9) = 3	both master legend report and legend with each table.
OPTIONS(10)	print missing contingencies report (0 by default).
OPTIONS(10) = 0	no.
OPTIONS(10) = 1	yes.
OPTIONS(11)	print non-converged contingencies report (0 by default).
OPTIONS(11) = 0	no.
OPTIONS(11) = 1	yes.
OPTIONS(12)	print loading violation reports (0 by default).
OPTIONS(12) = 0	no.
OPTIONS(12) = 1	base case and worst case contingency violations only.
OPTIONS(12) = 2	base case and all contingency violations only.
OPTIONS(12) = 3	base case, worst case contingency, and all contingency violations.
OPTIONS(13)	print voltage violation reports (0 by default).
OPTIONS(13) = 0	no.
OPTIONS(13) = 1	base case and worst case contingency violations only.
OPTIONS(13) = 2	base case and all contingency violations only.
OPTIONS(13) = 3	base case, worst case contingency, and all contingency violations.
OPTIONS(14)	option for processing interfaces in loading violation reports (1 by default).
OPTIONS(14) = 0	exclude interfaces from loading violation checking and reporting.
OPTIONS(14) = 1	check and report interface loading violations.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTIONS(15)	in loading violation reports, exclude monitored branches and interfaces that show loading violations in the base case from being checked and reported in contingency case reports (0 by default).
	OPTIONS(15) = 0	no.
	OPTIONS(15) = 1	yes.
	OPTIONS(16)	in voltage range violation reports, exclude monitored buses that show voltage range violations in the base case from the corresponding check in contingency case reports (0 by default).
	OPTIONS(16) = 0	no.
	OPTIONS(16) = 1	yes.
Real VALUES(7)	Is an array of seven elements specifying reporting parameters (input). The values are as follows.	
	VALUES(1)	bus mismatch converged tolerance (MW or Mvar) (0.5 by default).
	VALUES(2)	system mismatch converged tolerance (MVA) (5.0 by default).
	VALUES(3)	in loading violation reports, percent of flow rating for reporting (100.0 by default).
	VALUES(4)	in the worst case contingency loading violation reports, percent of flow rating for counting (≥ VALUES(3)) (100.0 by default).
	VALUES(5)	in loading violation reports, minimum contingency case flow change from base case value for reporting and counting (0.0 by default).
	VALUES(6)	in loading violation reports, minimum contingency case percent loading increase from base case value for reporting and counting (0.0 by default).
	VALUES(7)	minimum contingency case voltage change for reporting and counting in voltage range violations (0.0 by default).
Integer NFILES	Is the number of Contingency Solution Output Files to be processed (1 through 22) (input; no default allowed).	
Character ACFILES (NFILES)*260	Is an array of NFILES names of Contingency Solution Output Files (input; no defaults allowed).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid OPTIONS value.
IERR = 2	invalid VALUES value.
IERR = 3	invalid NFILES value.
IERR = 4	the percent for counting is less than the percent for reporting.
IERR = 5	file is not in the form of a current Contingency Solution Output File.
IERR = 6	error opening a Contingency Solution Output File.
IERR = 7	error reading a Contingency Solution Output File.
IERR = 8	error opening a temporary file.
IERR = 9	error opening a subfile in the temporary file.
IERR = 10	error reading or writing the temporary file.
IERR = 11	error closing a subfile in the temporary file.
IERR = 12	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.4 ACCC_PARALLEL

This API routine is obsolete. It has been replaced by the API routine ACCC_PARALLEL_2, and is implemented by a call to ACCC_PARALLEL_2.

Use this API routine to run the initial release of the parallel implementation of the AC contingency calculation function (use ACCC_WITH_DSP_3 to run AC contingency calculation function serially). The API routine distributes contingency cases among multiple processors evenly, and tests the contingency cases assigned to each processor in parallel.

Batch command syntax:

```
BAT_ACCC_PARALLEL TOL OPTIONS(1)...OPTIONS(10) LABEL DFXFILE ACCFILE
THRFILE INLFILE
```

Python syntax:

```
ierr = accc_parallel(tol, options, label, dfxfile, accfile, thrfile,
inlfile)
```

Fortran syntax:

```
CALL ACCC_PARALLEL(TOL, OPTIONS, LABEL, DFXFILE, ACCFILE, THRFILE,
INLFILE, IERR)
```

where:

Real TOL	Is the mismatch tolerance (input; Newton solution convergence tolerance, TOLN, by default).
Integer OPTIONS(10)	Is an array of ten elements specifying solution options (input). The values are as follows:
	OPTIONS(1) — tap adjustment flag (tap adjustment option setting by default).
	OPTIONS(1) = 0 — disable.
	OPTIONS(1) = 1 — enable stepping adjustment.
	OPTIONS(1) = 2 — enable direct adjustment.
	OPTIONS(2) — area interchange adjustment flag (area interchange adjustment option setting by default).
	OPTIONS(2) = 0 — disable.
	OPTIONS(2) = 1 — enable using tie line flows only in calculating area interchange.
	OPTIONS(2) = 2 — enable using tie line flows and loads in calculating area interchange.
	OPTIONS(3) — phase shift adjustment flag (phase shift adjustment option setting by default).
	OPTIONS(3) = 0 — disable.
	OPTIONS(3) = 1 — enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer OPTIONS(10)	OPTIONS(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
	OPTIONS(4) = 0	disable.
	OPTIONS(4) = 1	enable.
	OPTIONS(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
	OPTIONS(5) = 0	disable.
	OPTIONS(5) = 1	enable.
	OPTIONS(5) = 2	enable continuous mode, disable discrete mode.
	OPTIONS(6)	solution method flag (0 by default).
	OPTIONS(6) = 0	FDNS.
	OPTIONS(6) = 1	FNSL.
	OPTIONS(6) = 2	optimized FDNS.
	OPTIONS(7)	non-divergent solution flag (non-divergent solution option setting by default).
	OPTIONS(7) = 0	disable.
	OPTIONS(7) = 1	enable.
	OPTIONS(8)	induction motor treatment flag (applied when an induction motor fails to solve due to low terminal bus voltage, 0 by default)
	OPTIONS(8) = 0	stall.
	OPTIONS(8) = 1	trip.
	OPTIONS(9)	induction machine failure flag(0 by default)
	OPTIONS(9) = 0	treat contingency as non-converged if any induction machines are placed in the “stalled” or “tripped” state.
	OPTIONS(9) = 1	treat contingency as solved if it converges, even if any induction machines are placed in the “stalled” or “tripped” state.
	OPTIONS(10)	dispatch mode (0 by default).
	OPTIONS(10) = 0	disable.
	OPTIONS(10) = 1	subsystem machines (reserve).
	OPTIONS(10) = 2	subsystem machines (pmax).
	OPTIONS(10) = 3	subsystem machines (inertia).
	OPTIONS(10) = 4	subsystem machines (governor droop).
Character LABEL*12	Is the name of the generation dispatch subsystem (blank by default, no default allowed if OPTIONS(10) is not 0).	
Character DFXFILE*260	Is the name of the Distribution Factor Data file (input; no default allowed).	
Character ACCFILE*260	Is the name of the Contingency Solution Output file (input; no default allowed).	
Character THRFILE*260	Is the name of the Load Throwover Data file (input; blank by default).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character INLFILE*260	Is the name of the Unit Inertia and Governor Data File (input; blank by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 as for ACCC_PARALLEL_2.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.5 ACCC_PARALLEL_2

Use this API routine to run the second release of the parallel implementation of the AC contingency calculation function (use ACCC_WITH_DSP_3 to run AC contingency calculation function serially). The API routine distributes contingency cases among multiple processors evenly, and tests the contingency cases assigned to each processor in parallel.

Batch command syntax:

```
BAT_ACCC_PARALLEL_2  TOL  OPTIONS(1)...OPTIONS(11)  LABEL  DFXFILE
ACCFILE  THRFILE  INLFILE  ZIPFILE
```

Python syntax:

```
ierr = accc_parallel_2(tol, options, label, dfxfile, accfile,
thrfile, inlfile, zipfile)
```

Fortran syntax:

```
CALL ACCC_PARALLEL_2(TOL, OPTIONS, LABEL, DFXFILE, ACCFILE, THRFILE,
INLFILE, ZIPFILE, IERR)
```

where:

Real TOL	Is the mismatch tolerance (input; Newton solution convergence tolerance, TOLN, by default).
Integer OPTIONS(11)	Is an array of eleven elements specifying solution options (input). The values are as follows:

OPTIONS(1)	tap adjustment flag (tap adjustment option setting by default).
OPTIONS(1) = 0	disable.
OPTIONS(1) = 1	enable stepping adjustment.
OPTIONS(1) = 2	enable direct adjustment.
OPTIONS(2)	area interchange adjustment flag (area interchange adjustment option setting by default).
OPTIONS(2) = 0	disable.
OPTIONS(2) = 1	enable using tie line flows only in calculating area interchange.
OPTIONS(2) = 2	enable using tie line flows and loads in calculating area interchange.
OPTIONS(3)	phase shift adjustment flag (phase shift adjustment option setting by default).
OPTIONS(3) = 0	disable.
OPTIONS(3) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer OPTIONS(11)	OPTIONS(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
	OPTIONS(4) = 0	disable.
	OPTIONS(4) = 1	enable.
	OPTIONS(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
	OPTIONS(5) = 0	disable.
	OPTIONS(5) = 1	enable.
	OPTIONS(5) = 2	enable continuous mode, disable discrete mode.
	OPTIONS(6)	solution method flag (0 by default).
	OPTIONS(6) = 0	FDNS.
	OPTIONS(6) = 1	FNSL.
	OPTIONS(6) = 2	optimized FDNS.
	OPTIONS(7)	non-divergent solution flag (non-divergent solution option setting by default).
	OPTIONS(7) = 0	disable.
	OPTIONS(7) = 1	enable.
	OPTIONS(8)	induction motor treatment flag (applied when an induction motor fails to solve due to low terminal bus voltage, 0 by default)
	OPTIONS(8) = 0	stall.
	OPTIONS(8) = 1	trip.
	OPTIONS(9)	induction machine failure flag(0 by default)
	OPTIONS(9) = 0	treat contingency as non-converged if any induction machines are placed in the “stalled” or “tripped” state.
	OPTIONS(9) = 1	treat contingency as solved if it converges, even if any induction machines are placed in the “stalled” or “tripped” state.
	OPTIONS(10)	dispatch mode (0 by default)
	OPTIONS(10) = 0	disable.
	OPTIONS(10) = 1	subsystem machines (reserve).
	OPTIONS(10) = 2	subsystem machines (pmax).
	OPTIONS(10) = 3	subsystem machines (inertia).
	OPTIONS(10) = 4	subsystem machines (governor droop).
	OPTIONS(11)	ZIP archive flag (0 by default)
	OPTIONS(11) = 0	do not write a ZIP archive file.
	OPTIONS(11) = 1	write a ZIP archive using the file specified as ZIPFILE.
Character LABEL*12	Is the name of the generation dispatch subsystem (blank by default, no default allowed if OPTIONS(10) is not 0).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character DFXFILE*260 Is the name of the Distribution Factor Data file (input; no default allowed).

Character ACCFILE*260 Is the name of the Contingency Solution Output file (input; no default allowed).

Character THRFILE*260 Is the name of the Load Throwover Data file (input; blank by default).

Character INLFILE*260 Is the name of the Unit Inertia and Governor Data File (input; blank by default).

Character ZIPFILE*260 Is the name of the ZIP Archive Output File (input; blank by default).

Integer IERR Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid TOL value.
IERR = 2	invalid OPTIONS value.
IERR = 3	generators are converted.
IERR = 4	buses in island(s) without a swing bus; use activity TREE.
IERR = 5	largest mismatch exceeds mismatch tolerance.
IERR = 6	generation dispatch subsystem is not defined.
IERR = 7	too many islands in base case.
IERR = 8	no Distribution Factor Data File specified.
IERR = 9	no Contingency Solution Output File specified.
IERR = 10	in-service induction machines are in the “stalled” or “tripped” state.
IERR = 11	buses with bus type code and series element status inconsistencies.
IERR = 12	no ZIP Archive Output File specified.
IERR = 21	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
IERR = 22	monitored elements exceed limit when adding multi-section line members.
IERR = 51	error opening Contingency Solution Output File.
IERR = 52	error opening Distribution Factor Data File.
IERR = 53	error opening Load Throwover Data File.
IERR = 54	error opening Unit Inertia and Governor Data File.
IERR = 55	error opening ZIP Archive Output File.
IERR = 56	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.6 ACCC_SINGLE_RUN_REPORT

This API routine is obsolete. It has been replaced by the API routine ACCC_SINGLE_RUN_REPORT_2, and is implemented by a call to ACCC_SINGLE_RUN_REPORT_2.

Use this API, the AC Contingency Report function, to report the results of the AC Contingency Calculation function.

Batch command syntax:

```
BAT_ACCC_SINGLE_RUN_REPORT  STATUS(1)...STATUS(8)
INTVAL(1)...INTVAL(5)  REALVAL(1)...REALVAL(7)  RFILE
```

Python syntax:

```
ierr = accc_single_run_report(status, intval, realval, rfile)
```

Fortran syntax:

```
CALL ACCC_SINGLE_RUN_REPORT(STATUS, INTVAL, REALVAL, RFILE, IERR)
```

where:

Integer STATUS(8)	Is an array of eight elements (input). The values are as follows.	
	STATUS(1)	report format (3 by default).
	STATUS(1) = 0	spreadsheet overload report.
	STATUS(1) = 1	spreadsheet loading table.
	STATUS(1) = 2	available capacity table.
	STATUS(1) = 3	non-spreadsheet overload report.
	STATUS(1) = 4	non-spreadsheet loading table.
	STATUS(1) = 5	non-converged networks report.
	STATUS(2)	base case rating set; used only when STATUS(1) is 0, 1, 3 or 4 (rating set program option setting by default).
	STATUS(2) = 1	rate A.
	STATUS(2) = 2	rate B.
	STATUS(2) = 3	rate C.
	STATUS(3)	contingency case rating set when STATUS(1) is 0, 1, 3 or 4; base case and contingency case rating set when STATUS(1) is 2 (rating set program option setting by default).
	STATUS(3) = 1	rate A.
	STATUS(3) = 2	rate B.
	STATUS(3) = 3	rate C.
	STATUS(4)	exclude interfaces from report; used only when STATUS(1) is 0, 1, 2, 3 or 4 (0 by default).
	STATUS(4) = 0	no.
	STATUS(4) = 1	yes.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	STATUS(5)	run voltage limit check; used only when STATUS(1) is 0, 1, 3 or 4 (0 by default).
	STATUS(5) = 0	no.
	STATUS(5) = 1	yes.
	STATUS(6)	in overload reports, exclude monitored branches and interfaces that show loading violations in the base case from being checked and reported in contingency cases; used only when STATUS(1) is 0 or 3 (0 by default).
	STATUS(6) = 0	no.
	STATUS(6) = 1	yes.
	STATUS(7)	in voltage range violation reports, exclude monitored buses that show voltage range violations in the base case from the corresponding check in contingency case reports; used only when STATUS(1) is 0 or 3 (0 by default).
	STATUS(7) = 0	no.
	STATUS(7) = 1	yes.
	STATUS(8)	exclude cases with no overloads from non-spreadsheet overload report; used only when STATUS(1) is 3 (0 by default).
	STATUS(8) = 0	no.
	STATUS(8) = 1	yes.
Integer INTVAL(5)	Is an array of five elements (input). The values are as follows.	
	INTVAL(1)	number of low voltage range violations filtering criterion (0 by default).
	INTVAL(2)	number of high voltage range violations filtering criterion (0 by default).
	INTVAL(3)	number of voltage deviation violations filtering criterion; not applied to base case (0 by default).
	INTVAL(4)	number of buses in the largest disconnected island filtering criterion; not applied to base case (0 by default).
	INTVAL(5)	maximum number of elements in the available capacity table (no limit by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real REALVAL(7)	Is an array of seven elements (input). The values are as follows.	
	REALVAL(1)	bus mismatch converged tolerance (MW or Mvar) (0.5 by default).
	REALVAL(2)	system mismatch converged tolerance (MVA) (5.0 by default).
	REALVAL(3)	percent of flow rating; used only when STATUS(1) is 0, 3 or 4 (100.0 by default).
	REALVAL(4)	in overload reports, minimum contingency case flow change from base case value; used only when STATUS(1) is 0 or 3 (0.0 by default).
	REALVAL(5)	in overload reports, minimum contingency case percent loading increase from base case value; used only when STATUS(1) is 0 or 3 (0.0 by default).
	REALVAL(6)	in voltage range violation reports, minimum contingency case voltage change from base case value; used only when STATUS(1) is 0, 1, 3 or 4 (0.0 by default).
	REALVAL(7)	cutoff threshold for available capacity table; used only when STATUS(1) is 2 (99999.0 by default).
Character RFILE*260	Is the Contingency Solution Output File (input; no default allowed).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid STATUS value.
	IERR = 2	invalid INTVAL value.
	IERR = 3	invalid REALVAL value.
	IERR = 4	error opening RFILE.
	IERR = 5	error reading RFILE.
	IERR = 6	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.7 ACCC_SINGLE_RUN_REPORT_2

This API routine is obsolete. It has been replaced by the API routine ACCC_SINGLE_RUN_REPORT_3, and is implemented by a call to ACCC_SINGLE_RUN_REPORT_3.

This API is the second release of the AC Contingency Report function. Use this API to report the results of the AC Contingency Calculation function.

Batch command syntax:

```
BAT_ACCC_SINGLE_RUN_REPORT_2  STATUS(1)...STATUS(9)
INTVAL(1)...INTVAL(5)  REALVAL(1)...REALVAL(7)  RFILE
```

Python syntax:

```
ierr = accc_single_run_report_2(status, intval, realval, rfile)
```

Fortran syntax:

```
CALL ACCC_SINGLE_RUN_REPORT_2(STATUS, INTVAL, REALVAL, RFILE, IERR)
```

where:

Integer STATUS(9)	Is an array of nine elements (input). The values are as follows.	
	STATUS(1)	report format (3 by default).
	STATUS(1) = 0	spreadsheet overload report.
	STATUS(1) = 1	spreadsheet loading table.
	STATUS(1) = 2	available capacity table.
	STATUS(1) = 3	non-spreadsheet overload report.
	STATUS(1) = 4	non-spreadsheet loading table.
	STATUS(1) = 5	non-converged networks report.
	STATUS(1) = 6	non-spreadsheet corrective actions report.
	STATUS(2)	base case rating set; used only when STATUS(1) is 0, 1, 3 or 4 (rating set program option setting by default).
	STATUS(2) = 1	rate A.
	STATUS(2) = 2	rate B.
	STATUS(2) = 3	rate C.
	STATUS(3)	contingency case rating set when STATUS(1) is 0, 1, 3 or 4; base case and contingency case rating set when STATUS(1) is 2 (rating set program option setting by default).
	STATUS(3) = 1	rate A.
	STATUS(3) = 2	rate B.
	STATUS(3) = 3	rate C.
	STATUS(4)	exclude interfaces from report; used only when STATUS(1) is 0, 1, 2, 3, 4 (0 by default).
	STATUS(4) = 0	no.
	STATUS(4) = 1	yes.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	STATUS(5)	run voltage limit check; used only when STATUS(1) is 0, 1, 3, 4 (0 by default).
	STATUS(5) = 0	no.
	STATUS(5) = 1	yes.
	STATUS(6)	in overload reports, exclude monitored branches and interfaces that show loading violations in the base case from being checked and reported in contingency cases; used only when STATUS(1) is 0 or 3 (0 by default).
	STATUS(6) = 0	no.
	STATUS(6) = 1	yes.
	STATUS(7)	in voltage range violation reports, exclude monitored buses that show voltage range violations in the base case from the corresponding check in contingency case reports; used only when STATUS(1) is 0 or 3 (0 by default).
	STATUS(7) = 0	no.
	STATUS(7) = 1	yes.
	STATUS(8)	exclude cases with no overloads from non-spreadsheet overload report or non-spreadsheet corrective actions report; used only when STATUS(1) is 3 (0 by default).
	STATUS(8) = 0	no.
	STATUS(8) = 1	yes.
	STATUS(9)	report post-tripping action solutions (0 by default).
	STATUS(9) = 0	no.
	STATUS(9) = 1	yes.
Integer INTVAL(5)	Is an array of five elements (input). The values are as follows.	
	INTVAL(1)	number of low voltage range violations filtering criterion (0 by default).
	INTVAL(2)	number of high voltage range violations filtering criterion (0 by default).
	INTVAL(3)	number of voltage deviation violations filtering criterion; not applied to base case (0 by default).
	INTVAL(4)	number of buses in the largest disconnected island filtering criterion; not applied to base case (0 by default).
	INTVAL(5)	maximum number of elements in the available capacity table (no limit by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real REALVAL(7)	Is an array of seven elements (input). The values are as follows.	
	REALVAL(1)	bus mismatch converged tolerance (MW or Mvar) (0.5 by default).
	REALVAL(2)	system mismatch converged tolerance (MVA) (5.0 by default).
	REALVAL(3)	percent of flow rating; used only when STATUS(1) is 0, 3 or 4 (100.0 by default).
	REALVAL(4)	in overload reports, minimum contingency case flow change from base case value; used only when STATUS(1) is 0 or 3 (0.0 by default).
	REALVAL(5)	in overload reports, minimum contingency case percent loading increase from base case value; used only when STATUS(1) is 0 or 3 (0.0 by default).
	REALVAL(6)	in voltage range violation reports, minimum contingency case voltage change from base case value; used only when STATUS(1) is 0, 1, 3 or 4 (0.0 by default).
	REALVAL(7)	cutoff threshold for available capacity table; used only when STATUS(1) is 2 (99999.0 by default).
Character RFILE*260	Is the Contingency Solution Output File (input; no default allowed).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid STATUS value.
	IERR = 2	invalid INTVAL value.
	IERR = 3	invalid REALVAL value.
	IERR = 4	error opening RFILE.
	IERR = 5	error reading RFILE.
	IERR = 6	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.8 ACCC_SINGLE_RUN_REPORT_3

The API is obsolete. It has been replaced by ACCC_SINGLE_RUN_REPORT_4 and is implemented by a call to ACCC_SINGLE_RUN_REPORT_4.
This API is the third release of the AC Contingency Report function. Use this API to report the results of the AC Contingency Calculation function.
Batch command syntax:

```
BAT_ACCC_SINGLE_RUN_REPORT_3  STATUS(1)...STATUS(10)
INTVAL(1)...INTVAL(5)  REALVAL(1)...REALVAL(7)  RFILE
```

Python syntax:

```
ierr = accc_single_run_report_3(status, intval, realval, rfile)
```

Fortran syntax:

```
CALL ACCC_SINGLE_RUN_REPORT_3(STATUS, INTVAL, REALVAL, RFILE, IERR)
```

where:

Integer STATUS(10) Is an array of ten elements (input). The values are as follows.

STATUS(1)	report format (3 by default).
STATUS(1) = 0	spreadsheet overload report.
STATUS(1) = 1	spreadsheet loading table.
STATUS(1) = 2	available capacity table.
STATUS(1) = 3	non-spreadsheet overload report.
STATUS(1) = 4	non-spreadsheet loading table.
STATUS(1) = 5	non-converged networks report.
STATUS(1) = 6	non-spreadsheet corrective actions report.
STATUS(2)	base case rating set; used only when STATUS(1) is 0, 1, 3 or 4 (rating set program option setting by default).
STATUS(2) = 1	rate A.
STATUS(2) = 2	rate B.
STATUS(2) = 3	rate C.
STATUS(3)	contingency case rating set when STATUS(1) is 0, 1, 3 or 4; base case and contingency case rating set when STATUS(1) is 2 (rating set program option setting by default).
STATUS(3) = 1	rate A.
STATUS(3) = 2	rate B.
STATUS(3) = 3	rate C.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

STATUS(4)	exclude interfaces from report; used only when STATUS(1) is 0, 1, 2, 3, 4 (0 by default).
STATUS(4) = 0	no.
STATUS(4) = 1	yes.
STATUS(5)	run voltage limit check; used only when STATUS(1) is 0, 1, 3, 4 (0 by default).
STATUS(5) = 0	no.
STATUS(5) = 1	yes.
STATUS(6)	in overload reports, exclude monitored branches and interfaces that show loading violations in the base case from being checked and reported in contingency cases; used only when STATUS(1) is 0 or 3 (0 by default).
STATUS(6) = 0	no.
STATUS(6) = 1	yes.
STATUS(7)	in voltage range violation reports, exclude monitored buses that show voltage range violations in the base case from the corresponding check in contingency case reports; used only when STATUS(1) is 0 or 3 (0 by default).
STATUS(7) = 0	no.
STATUS(7) = 1	yes.
STATUS(8)	exclude cases with no overloads from non-spreadsheet overload report or non-spreadsheet corrective actions report; used only when STATUS(1) is 3 (0 by default).
STATUS(8) = 0	no.
STATUS(8) = 1	yes.
STATUS(9)	report post-tripping action solutions (0 by default).
STATUS(9) = 0	no.
STATUS(9) = 1	yes.
STATUS(10)	report loss of loads (0 by default).
STATUS(10) = 0	no.
STATUS(10) = 1	yes.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer INTVAL(5)	Is an array of five elements (input). The values are as follows.	
	INTVAL(1)	number of low voltage range violations filtering criterion (0 by default).
	INTVAL(2)	number of high voltage range violations filtering criterion (0 by default).
	INTVAL(3)	number of voltage deviation violations filtering criterion; not applied to base case (0 by default).
	INTVAL(4)	number of buses in the largest disconnected island filtering criterion; not applied to base case (0 by default).
	INTVAL(5)	maximum number of elements in the available capacity table (no limit by default).
Real REALVAL(7)	Is an array of seven elements (input). The values are as follows.	
	REALVAL(1)	bus mismatch converged tolerance (MW or Mvar) (0.5 by default).
	REALVAL(2)	system mismatch converged tolerance (MVA) (5.0 by default).
	REALVAL(3)	percent of flow rating; used only when STATUS(1) is 0, 3 or 4 (100.0 by default).
	REALVAL(4)	in overload reports, minimum contingency case flow change from base case value; used only when STATUS(1) is 0 or 3 (0.0 by default).
	REALVAL(5)	in overload reports, minimum contingency case percent loading increase from base case value; used only when STATUS(1) is 0 or 3 (0.0 by default).
	REALVAL(6)	in voltage range violation reports, minimum contingency case voltage change from base case value; used only when STATUS(1) is 0, 1, 3 or 4 (0.0 by default).
	REALVAL(7)	cutoff threshold for available capacity table; used only when STATUS(1) is 2 (99999.0 by default).
Character RFILE*260	Is the Contingency Solution Output File (input; no default allowed).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid STATUS value.
	IERR = 2	invalid INTVAL value.
	IERR = 3	invalid REALVAL value.
	IERR = 4	error opening RFILE.
	IERR = 5	error reading RFILE.
	IERR = 6	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.9 ACCC_SINGLE_RUN_REPORT_4

This API is the fourth release of the AC Contingency Report function. Use this API to report the results of the AC Contingency Calculation function.

Batch command syntax:

```
BAT_ACCC_SINGLE_RUN_REPORT_4  STATUS(1)...STATUS(12)
INTVAL(1)...INTVAL(5)  REALVAL(1)...REALVAL(7)  RFILE
```

Python syntax:

```
ierr = accc_single_run_report_4(status, intval, realval, rfile)
```

Fortran syntax:

```
CALL ACCC_SINGLE_RUN_REPORT_4(STATUS, INTVAL, REALVAL, RFILE, IERR)
```

where:

Integer STATUS(12)	Is an array of twelve elements (input). The values are as follows.	
	STATUS(1)	report format (3 by default).
	STATUS(1) = 0	spreadsheet overload report.
	STATUS(1) = 1	spreadsheet loading table.
	STATUS(1) = 2	available capacity table.
	STATUS(1) = 3	non-spreadsheet overload report.
	STATUS(1) = 4	non-spreadsheet loading table.
	STATUS(1) = 5	non-converged networks report.
	STATUS(1) = 6	non-spreadsheet corrective actions report.
	STATUS(2)	base case rating set; used only when STATUS(1) is 0, 1, 3 or 4 (rating set program option setting by default).
	STATUS(2) = 1	rate A.
	STATUS(2) = 2	rate B.
	STATUS(2) = 3	rate C.
	STATUS(3)	contingency case rating set when STATUS(1) is 0, 1, 3 or 4; base case and contingency case rating set when STATUS(1) is 2 (rating set program option setting by default).
	STATUS(3) = 1	rate A.
	STATUS(3) = 2	rate B.
	STATUS(3) = 3	rate C.
	STATUS(4)	base case voltage limit when STATUS(1) is 0, 1, 3 or 4.
	STATUS(4) = 1	normal limit.
	STATUS(4) = 2	emergency limit.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

STATUS(5)	contingency case voltage limit when STATUS(1) is 0, 1, 3 or 4.
STATUS(5) = 1	normal limit.
STATUS(5) = 2	emergency limit.
STATUS(6)	exclude interfaces from report; used only when STATUS(1) is 0, 1, 2, 3, 4 (0 by default).
STATUS(6) = 0	no.
STATUS(6) = 1	yes.
STATUS(7)	run voltage limit check; used only when STATUS(1) is 0, 1, 3, 4 (0 by default).
STATUS(7) = 0	no.
STATUS(7) = 1	yes.
STATUS(8)	in overload reports, exclude monitored branches and interfaces that show loading violations in the base case from being checked and reported in contingency cases; used only when STATUS(1) is 0 or 3 (0 by default).
STATUS(8) = 0	no.
STATUS(8) = 1	yes.
STATUS(9)	in voltage range violation reports, exclude monitored buses that show voltage range violations in the base case from the corresponding check in contingency case reports; used only when STATUS(1) is 0 or 3 (0 by default).
STATUS(9) = 0	no.
STATUS(9) = 1	yes.
STATUS(10)	exclude cases with no overloads from non-spreadsheet overload report or non-spreadsheet corrective actions report; used only when STATUS(1) is 3 (0 by default).
STATUS(10) = 0	no.
STATUS(10) = 1	yes.
STATUS(11)	report post-tripping action solutions (0 by default).
STATUS(11) = 0	no.
STATUS(11) = 1	yes.
STATUS(12)	report loss of loads (0 by default).
STATUS(12) = 0	no.
STATUS(12) = 1	yes.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer INTVAL(5)	Is an array of five elements (input). The values are as follows.	
	INTVAL(1)	number of low voltage range violations filtering criterion (0 by default).
	INTVAL(2)	number of high voltage range violations filtering criterion (0 by default).
	INTVAL(3)	number of voltage deviation violations filtering criterion; not applied to base case (0 by default).
	INTVAL(4)	number of buses in the largest disconnected island filtering criterion; not applied to base case (0 by default).
	INTVAL(5)	maximum number of elements in the available capacity table (no limit by default).
Real REALVAL(7)	Is an array of seven elements (input). The values are as follows.	
	REALVAL(1)	bus mismatch converged tolerance (MW or Mvar) (0.5 by default).
	REALVAL(2)	system mismatch converged tolerance (MVA) (5.0 by default).
	REALVAL(3)	percent of flow rating; used only when STATUS(1) is 0, 3 or 4 (100.0 by default).
	REALVAL(4)	in overload reports, minimum contingency case flow change from base case value; used only when STATUS(1) is 0 or 3 (0.0 by default).
	REALVAL(5)	in overload reports, minimum contingency case percent loading increase from base case value; used only when STATUS(1) is 0 or 3 (0.0 by default).
	REALVAL(6)	in voltage range violation reports, minimum contingency case voltage change from base case value; used only when STATUS(1) is 0, 1, 3 or 4 (0.0 by default).
	REALVAL(7)	cutoff threshold for available capacity table; used only when STATUS(1) is 2 (99999.0 by default).
Character RFILE*260	Is the Contingency Solution Output File (input; no default allowed).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid STATUS value.
	IERR = 2	invalid INTVAL value.
	IERR = 3	invalid REALVAL value.
	IERR = 4	error opening RFILE.
	IERR = 5	error reading RFILE.
	IERR = 6	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.10 ACCC_TRIP_COR

Use this API to apply the AC contingency calculation with tripping simulations and corrective actions. When there are violations in the post-tripping simulation solutions, the corrections actions analysis is launched to eliminate the violations, with the objective of minimizing the controls adjustments, including load shedding, MW generation dispatch and phase shifter angle adjustments. Rating set program option setting and 100.0 percent of rating set are used to check branch and interface flow overload violations.

Batch command syntax:

```
BAT_ACCC_TRIP_COR  TOL  OPTACC(1)...OPTACC(8) OPTTRP(1)...OPTTRP(3)
OPTCOR(1)...OPTCOR(5)  VALUES(1)...VALUES(5)  LABELS(1)...LABELS(4)
DFXFILE  ACCFILE  THRFILE  INLFILE  TRPFILE
```

Python syntax:

```
ierr = accc_trip_cor(tol, optacc, opttrp, optcor, values, labels,
dfxfile, accfile, thrfile, inlfile, trpfile)
```

Fortran syntax:

```
CALL ACCC_TRIP_COR(TOL, OPTACC, OPTTRP, OPTCOR, VALUES, LABELS,
DFXFILE, ACCFILE, THRFILE, INLFILE, TRPFILE, IERR)
```

where:

Real TOL	Is the mismatch tolerance (input; Newton solution convergence tolerance, TOLN, by default).
Integer OPTACC(8)	Is an array of eight elements specifying solution options (input). The values are as follows:

OPTACC(1)	tap adjustment flag (tap adjustment option setting by default).
OPTACC(1) = 0	disable.
OPTACC(1) = 1	enable stepping adjustment.
OPTACC(1) = 2	enable direct adjustment.
OPTACC(2)	area interchange adjustment flag (area interchange adjustment option setting by default).
OPTACC(2) = 0	disable.
OPTACC(2) = 1	enable using tie line flows only in calculating area interchange.
OPTACC(2) = 2	enable using tie line flows and loads in calculating area interchange.
OPTACC(3)	phase shift adjustment flag (phase shift adjustment option setting by default).
OPTACC(3) = 0	disable.
OPTACC(3) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTACC(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
	OPTACC(4) = 0	disable.
	OPTACC(4) = 1	enable.
	OPTACC(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
	OPTACC(5) = 0	disable.
	OPTACC(5) = 1	enable.
	OPTACC(5) = 2	enable continuous mode, disable discrete mode.
	OPTACC(6)	solution method flag (0 by default).
	OPTACC(6) = 0	FDNS.
	OPTACC(6) = 1	FNSL.
	OPTACC(6) = 2	optimized FDNS.
	OPTACC(7)	non-divergent solution flag (non-divergent solution option setting by default).
	OPTACC(7) = 0	disable.
	OPTACC(7) = 1	enable.
	OPTACC(8)	dispatch mode (0 by default).
	OPTACC(8) = 0	disable.
	OPTACC(8) = 1	subsystem machines (reserve).
	OPTACC(8) = 2	subsystem machines (pmax).
	OPTACC(8) = 3	subsystem machines (inertia).
	OPTACC(8) = 4	subsystem machines (governor droop).
Integer OPTTRP(3)	Is an array of three elements specifying tripping simulation options (input). The values are as follows:	
	OPTTRP(1)	tripping simulation flag (0 by default).
	OPTTRP(1) = 0	disable.
	OPTTRP(1) = 1	enable tripping simulation.
	OPTTRP(2)	number of tripping events per power flow (1 by default).
	OPTTRP(3)	number of power flows per contingency (1 by default).
Integer OPTCOR(5)	Is an array of five elements specifying corrective actions options (input). The values are as follows:	
	OPTCOR(1)	run corrective actions (0 by default)
	OPTCOR(1) = 0	disable.
	OPTCOR(1) = 1	enable.
	OPTCOR(2)	number of power flow solutions (from 0 to 10, 1 by default)
	OPTCOR(3)	generator control flag (1 by default).
	OPTCOR(3) = 0	disable.
	OPTCOR(3) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTCOR(4)	load control flag (0 by default).
	OPTCOR(4) = 0	disable.
	OPTCOR(4) = 1	enable.
	OPTCOR(5)	phase shifter control flag (1 by default).
	OPTCOR(5) = 0	disable.
	OPTCOR(5) = 1	enable.
Real VALUES(5)	Is an array of five elements specifying corrective actions parameters (input). The values are as follows:	
	VALUES(1)	bus voltage violation tolerance (0.1 by default).
	VALUES(2)	branch flow overload tolerance (0.1 by default).
	VALUES(3)	generator control weighting factor (1.0 by default).
	VALUES(4)	load control weighting factor (1.0 by default).
	VALUES(5)	phase shifter control weighting factor (1.0 by default).
Character LABELS(4)*12	Is an array of four elements specifying subsystem labels (input). The values are as follows:	
	LABELS(1)	label of generation dispatch subsystem (blank by default, no default allowed if OPTACC(8) is not 0).
	LABELS(2)	label of generator control subsystem (blank by default, no default allowed if OPTCOR(3) is 1).
	LABELS(3)	label of load control subsystem (blank by default, no default allowed if OPTCOR(4) is 1).
	LABELS(4)	label of phase shifter control subsystem (blank by default, no default allowed if OPTCOR(5) is 1).
Character DFXFILE*260	Is the name of the Distribution Factor Data File (input; no default allowed).	
Character ACCFILE*260	Is the name of the Contingency Solution Output File (input; no default allowed).	
Character THRFILE*260	Is the name of the Load Throwover Data File (input; blank by default).	
Character INLFILE*260	Is the name of the Unit Inertia and Governor Data File (input; blank by default).	
Character TRPFILE*260	Is the name of the Tripping Element Data File (input, blank for none; no default allowed if OPTTRP(1) is 1).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid TOL value.
IERR = 2	invalid OPTACC value.
IERR = 3	generators are converted.
IERR = 4	buses in island(s) without a swing bus; use activity TREE.
IERR = 5	largest mismatch exceeds mismatch tolerance.
IERR = 6	generation dispatch subsystem is not defined.
IERR = 7	too many islands in base case.
IERR = 8	invalid OPTTRP value.
IERR = 9	invalid OPTCOR value.
IERR = 10	invalid VALUES value.
IERR = 21	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
IERR = 22	monitored elements exceed limit when adding multi-section line members.
IERR = 49	error opening Tripping Element Data File.
IERR = 50	error opening Unit Inertia and Governor Data File.
IERR = 51	error opening Contingency Solution Output File.
IERR = 52	error opening Distribution Factor Data File.
IERR = 53	error opening Load Throwover Data File.
IERR = 54	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.11 ACCC_TRIP_COR_2

The API is obsolete. It has been replaced by ACCC_TRIP_COR_3, and implemented by a call to ACCC_TRIP_COR_3.

This API is the second release of the function to run AC contingency analysis with tripping simulation and corrective actions. Following a post contingency solution, trip events are simulated if their tripping conditions are satisfied. Corrective actions are then specified automatically to eliminate the violations in each post tripping solution with the objective of minimizing the control adjustments including load shedding, MW generation dispatch, phase shifter angle adjustments, off-line generator controls, tap setting adjustments and switched shunt controls. Please note that flows are checked against rating set program option setting and 100.0 percent of the rating to determine flow overload violations.

Batch command syntax:

```
BAT_ACCC_TRIP_COR_2 TOL OPTACC(1)…OPTACC(8), OPTTRP(1)…OPTRP(3),
OPTCOR(1)…OPTCOR(8), VALUES(1)…VALUES(8), LABELS(1)…LABLES(7),
DFXFILE, ACCFILE, THRFILE, INLFILE, TRPFILE
```

Python syntax:

```
ierr = accc_trip_cor_2(tol, optacc, opttrp, optcor, values, labels,
dfxfile, accfile, thrfile, inlfile, trpfile)
```

Fortran syntax:

```
CALL ACCC_TRIP_COR_2(TOL, OPTACC, OPTTRP, OPTCOR, VALUES, LABELS,
DFXFILE, ACCFILE, THRFILE, INLFILE, TRPFILE, IERR)
```

where:

Real TOL	is the mismatch tolerance (input, Newton solution convergence tolerance, TOLN, by default).	
Integer OPTACC(8)	Is an array of eight elements specifying power flow solution (input).	
	OPTACC(1)	tap adjustment flag (tap adjustment option setting by default).
	OPTACC(1) = 0	disable.
	OPTACC(1) = 1	enable stepping adjustment.
	OPTACC(1) = 2	enable direct adjustment.
	OPTACC(2)	area interchange adjustment flag (area interchange adjustment option setting by default).
	OPTACC(2) = 0	disable.
	OPTACC(2) = 1	enable using tie line flows only in calculating area interchange.
	OPTACC(2) = 2	enable using tie line flows and loads in calculating area interchange.
	OPTACC(3)	phase shift adjustment flag (phase shift adjustment option setting by default).
	OPTACC(3) = 0	disable.
	OPTACC(3) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTACC(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
	OPTACC(4) = 0	disable.
	OPTACC(4) = 1	enable.
	OPTACC(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
	OPTACC(5) = 0	disable.
	OPTACC(5) = 1	enable.
	OPTACC(5) = 2	enable continuous, disable discrete.
	OPTACC(6)	solution method flag (0 by default).
	OPTACC(6) = 0	FDNS.
	OPTACC(6) = 1	FNSL.
	OPTACC(6) = 2	optimized FDNS.
	OPTACC(7)	non-divergent solution flag (non-divergent solution option setting by default).
	OPTACC(7) = 0	disable.
	OPTACC(7) = 1	enable.
	OPTACC(8) = dispatch mode (0 by default).	
	OPTACC(8) = 0	disable.
	OPTACC(8) = 1	subsystem machines (RESERVE).
	OPTACC(8) = 2	subsystem machines (PMAX).
	OPTACC(8) = 3	subsystem machines (INERTIA).
	OPTACC(8) = 4	subsystem machines (GOVERNOR DROOP).
Integer OPTTRP(3)	Is an array of three elements specifying tripping simulation options:	
	OPTTRP(1)	tripping simulation flag (0 by default).
	OPTTRP(1) = 0	disable.
	OPTTRP(1) = 1	enable tripping simulation.
	OPTTRP(2)	number of tripping events per power flow (1 by default).
	OPTTRP(3)	number of power flows per contingency (1 by default).
Integer OPTCOR(8)	Is an array of eight elements specifying corrective actions options.	
	OPTCOR(1)	run corrective action (0 by default).
	OPTCOR(1) = 0	disable.
	OPTCOR(1) = 1	enable.
	OPTCOR(2)	number of power flow solutions (1 by default).
	OPTCOR(3)	generator control flag (1 by default).
	OPTCOR(3) = 0	disable.
	OPTCOR(3) = 1	enable.
	OPTCOR(4)	load control flag (0 by default).
	OPTCOR(4) = 0	disable.
	OPTCOR(4) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTCOR(5)	phase shifter control flag (1 by default).
	OPTCOR(5) = 0	disable.
	OPTCOR(5) = 1	enable.
	OPTCOR(6)	off-line generator control flag (0 by default).
	OPTCOR(6) = 0	disable.
	OPTCOR(6) = 1	enable.
	OPTCOR(7)	tap setting adjustment flag (0 by default).
	OPTCOR(7) = 0	disable.
	OPTCOR(7) = 1	enable.
	OPTCOR(8)	switched shunt control flag (0 by default).
	OPTCOR(8) = 0	disable.
	OPTCOR(8) = 1	enable.
Real VALUES(8)	Is the array of eight elements to specifying corrective actions parameters (input).	
	VALUES(1)	bus voltage violation tolerance (0.1 by default).
	VALUES(2)	branch overload violation tolerance (0.1 by default).
	VALUES(3)	generator control weighting factor (1.0 by default).
	VALUES(4)	load control weighting factor (1.0 by default).
	VALUES(5)	phase shifter control weighting factor (1.0 by default).
	VALUES(6)	off-line generator control weighting factor (1.0 by default).
	VALUES(7)	tap setting adjustment weighting factor (1.0 by default).
	VALUES(8)	switched shunt control weighting factor (1.0 by default).
Character LABELS(7)*12	Is an array of seven elements specifying subsystem labels.	
	LABELS(1)	label of generation dispatch subsystem (blank by default, no default allowed if OPTACC(8) is not 0).
	LABELS(2)	label of generator control subsystem (blank by default, no default allowed if OPTCOR(3) is 1).
	LABELS(3)	label of load control subsystem (blank by default, no default allowed if OPTCOR(4) is 1).
	LABELS(4)	label of phase shifter control subsystem (blank by default, no default allowed if OPTCOR(5) is 1).
	LABELS(5)	off-line generator control subsystem (blank by default; no default allowed if OPTCOR(6) is 1).
	LABELS(6)	tap setting adjustment subsystem (blank by default; no default allowed if OPTCOR(7) is 1).
	LABELS(7)	switched shunt control subsystem (blank by default; no default allowed if OPTCOR(8) is 1).
Character*260 DFXFILE	Is the name of Distribution Factor Data File (input, no default allowed).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character*260 ACCFILE	Is the name of Contingency Solution Output File (input, no default allowed).
Character*260 THRFILE	Is the name of Load Throwover Data File (input, blank for none).
Character*260 INLFILE	Is the name of Unit Inertia Data File (input, blank for none).
Character*260 TRPFILE	Is the name of Tripping Element Data File (input, blank for none; no default allowed if OPTTRP(1) is 1).
Integer IERR	Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid TOL value.
IERR = 2	invalid OPTACC value.
IERR = 3	generators are converted.
IERR = 4	buses in island(s) without a swing bus; use activity TREE.
IERR = 5	largest mismatch exceeds mismatch tolerance.
IERR = 6	generation subsystem is not defined.
IERR = 7	too many islands in base case.
IERR = 8	invalid OPTTRP value.
IERR = 9	invalid OPTCOR value.
IERR = 10	invalid VALUES value.
IERR = 21	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
IERR = 22	monitored elements exceed limit when adding multisection line members.
IERR = 49	error opening Tripping Element Data File.
IERR = 50	error opening Unit Inertia Data File.
IERR = 51	error opening Contingency Solution Output File.
IERR = 52	error opening Distribution Factor Data File.
IERR = 53	error opening Load Throwover Data File.
IERR = 54	prerequisite requirements for function not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.12 ACCC_TRIP_COR_3

This API is the third release of the function to run AC contingency analysis with tripping simulation and corrective actions. Following a post contingency solution, trip events are simulated if their tripping conditions are satisfied. Corrective actions are then specified automatically to eliminate the violations in each post tripping solution with the objective of minimizing the control adjustments including load shedding, MW generation dispatch, phase shifter angle adjustments, off-line generator controls, tap setting adjustments and switched shunt controls. Please note that flows are checked against rating set program option setting and 100.0 percent of the rating to determine flow overload violations.

Batch command syntax:

```
BAT_ACCC_TRIP_COR_3 TOL OPTACC(1)…OPTACC(10), OPTTRP(1)…OPTRP(3),
OPTCOR(1)…OPTCOR(8), VALUES(1)…VALUES(8), LABELS(1)…LABLES(7),
DFXFILE, ACCFILE, THRFILE, INLFILE, TRPFILE
```

Python syntax:

```
ierr = accc_trip_cor_3(tol, optacc, opttrp, optcor, values, labels,
dfxfile, accfile, thrfile, inlfile, trpfile)
```

Fortran syntax:

```
CALL ACCC_TRIP_COR_3(TOL, OPTACC, OPTTRP, OPTCOR, VALUES, LABELS,
DFXFILE, ACCFILE, THRFILE, INLFILE, TRPFILE, IERR)
```

where:

Real TOL	is the mismatch tolerance (input, Newton solution convergence tolerance, TOLN, by default).	
Integer OPTACC(10)	Is an array of ten elements specifying power flow solution (input).	
	OPTACC(1)	tap adjustment flag (tap adjustment option setting by default).
	OPTACC(1) = 0	disable.
	OPTACC(1) = 1	enable stepping adjustment.
	OPTACC(1) = 2	enable direct adjustment.
	OPTACC(2)	area interchange adjustment flag (area interchange adjustment option setting by default).
	OPTACC(2) = 0	disable.
	OPTACC(2) = 1	enable using tie line flows only in calculating area interchange.
	OPTACC(2) = 2	enable using tie line flows and loads in calculating area interchange.
	OPTACC(3)	phase shift adjustment flag (phase shift adjustment option setting by default).
	OPTACC(3) = 0	disable.
	OPTACC(3) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTACC(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
OPTACC(4) = 0	disable.
OPTACC(4) = 1	enable.
OPTACC(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
OPTACC(5) = 0	disable.
OPTACC(5) = 1	enable.
OPTACC(5) = 2	enable continuous, disable discrete.
OPTACC(6)	solution method flag (0 by default).
OPTACC(6) = 0	FDNS.
OPTACC(6) = 1	FNSL.
OPTACC(6) = 2	optimized FDNS.
OPTACC(7)	non-divergent solution flag (non-divergent solution option setting by default).
OPTACC(7) = 0	disable.
OPTACC(7) = 1	enable.
OPTACC(8)	induction motor treatment flag (applied when an induction motor fails to solve due to low terminal bus voltage, 0 by default)
OPTACC(8) = 0	stall.
OPTACC(8) = 1	trip.
OPTACC(9)	induction machine failure flag(0 by default)
OPTACC(9) = 0	treat contingency as non-converged if any induction machines are placed in the “stalled” or “tripped” state.
OPTACC(9) = 1	treat contingency as solved if it converges, even if any induction machines are placed in the “stalled” or “tripped” state.
OPTACC(10) =	dispatch mode (0 by default).
OPTACC(10) = 0	disable.
OPTACC(10) = 1	subsystem machines (RESERVE).
OPTACC(10) = 2	subsystem machines (PMAX).
OPTACC(10) = 3	subsystem machines (INERTIA).
OPTACC(10) = 4	subsystem machines (GOVERNOR DROOP).

Integer OPTTRP(3) Is an array of three elements specifying tripping simulation options:

OPTTRP(1)	tripping simulation flag (0 by default).
OPTTRP(1) = 0	disable.
OPTTRP(1) = 1	enable tripping simulation.
OPTTRP(2)	number of tripping events per power flow (1 by default).
OPTTRP(3)	number of power flows per contingency (1 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer OPTCOR(8)	Is an array of eight elements specifying corrective actions options.	
	OPTCOR(1)	run corrective action (0 by default).
	OPTCOR(1) = 0	disable.
	OPTCOR(1) = 1	enable.
	OPTCOR(2)	number of power flow solutions (1 by default).
	OPTCOR(3)	generator control flag (1 by default).
	OPTCOR(3) = 0	disable.
	OPTCOR(3) = 1	enable.
	OPTCOR(4)	load control flag (0 by default).
	OPTCOR(4) = 0	disable.
	OPTCOR(4) = 1	enable.
	OPTCOR(5)	phase shifter control flag (1 by default).
	OPTCOR(5) = 0	disable.
	OPTCOR(5) = 1	enable.
	OPTCOR(6)	off-line generator control flag (0 by default).
	OPTCOR(6) = 0	disable.
	OPTCOR(6) = 1	enable.
	OPTCOR(7)	tap setting adjustment flag (0 by default).
	OPTCOR(7) = 0	disable.
	OPTCOR(7) = 1	enable.
	OPTCOR(8)	switched shunt control flag (0 by default).
	OPTCOR(8) = 0	disable.
	OPTCOR(8) = 1	enable.
Real VALUES(8)	Is the array of eight elements to specifying corrective actions parameters (input).	
	VALUES(1)	bus voltage violation tolerance (0.1 by default).
	VALUES(2)	branch overload violation tolerance (0.1 by default).
	VALUES(3)	generator control weighting factor (1.0 by default).
	VALUES(4)	load control weighting factor (1.0 by default).
	VALUES(5)	phase shifter control weighting factor (1.0 by default).
	VALUES(6)	off-line generator control weighting factor (1.0 by default).
	VALUES(7)	tap setting adjustment weighting factor (1.0 by default).
	VALUES(8)	switched shunt control weighting factor (1.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character LABELS(7)*12	Is an array of seven elements specifying subsystem labels.	
	LABELS(1)	label of generation dispatch subsystem (blank by default, no default allowed if OPTACC(8) is not 0).
	LABELS(2)	label of generator control subsystem (blank by default, no default allowed if OPTCOR(3) is 1).
	LABELS(3)	label of load control subsystem (blank by default, no default allowed if OPTCOR(4) is 1).
	LABELS(4)	label of phase shifter control subsystem (blank by default, no default allowed if OPTCOR(5) is 1).
	LABELS(5)	off-line generator control subsystem (blank by default; no default allowed if OPTCOR(6) is 1).
	LABELS(6)	tap setting adjustment subsystem (blank by default; no default allowed if OPTCOR(7) is 1).
	LABELS(7)	switched shunt control subsystem (blank by default; no default allowed if OPTCOR(8) is 1).
Character*260 DFXFILE	Is the name of Distribution Factor Data File (input, no default allowed).	
Character*260 ACCFILE	Is the name of Contingency Solution Output File (input, no default allowed).	
Character*260 THRFILE	Is the name of Load Throwover Data File (input, blank for none).	
Character*260 INLFILE	Is the name of Unit Inertia Data File (input, blank for none).	
Character*260 TRPFILE	Is the name of Tripping Element Data File (input, blank for none; no default allowed if OPTTRP(1) is 1).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid TOL value.
IERR = 2	invalid OPTACC value.
IERR = 3	generators are converted.
IERR = 4	buses in island(s) without a swing bus; use activity TREE.
IERR = 5	largest mismatch exceeds mismatch tolerance.
IERR = 6	generation subsystem is not defined.
IERR = 7	too many islands in base case.
IERR = 8	in-service induction machines are in the “stalled” or “tripped” state.
IERR = 9	buses with bus type code and series element status inconsistencies.
IERR = 10	invalid OPTTRP value.
IERR = 11	invalid OPTCOR value.
IERR = 12	invalid VALUES value.
IERR = 21	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
IERR = 22	monitored elements exceed limit when adding multisection line members.
IERR = 49	error opening Tripping Element Data File.
IERR = 50	error opening Unit Inertia Data File.
IERR = 51	error opening Contingency Solution Output File.
IERR = 52	error opening Distribution Factor Data File.
IERR = 53	error opening Load Throwover Data File.
IERR = 54	prerequisite requirements for function not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.13 ACCC_WITH_COR

Use this API to apply the AC contingency calculation with corrective actions. When there are violations in the post-contingency solutions, the corrections actions analysis is launched to eliminate the violations, with the objective of minimizing the controls adjustments, including load shedding, MW generation dispatch, and phase shifter angle adjustments. Rating set program option setting and 100.0 percent of the rating set are used to check branch and interface flow overload violations.

Batch command syntax:

```
BAT_ACCC_WITH_COR  TOL  OPTACC(1)...OPTACC(8)  OPTCOR(1)...OPTCOR(5)
VALUES(1)...VALUES(5)  LABELS(1)...LABELS(4)  DFXFILE  ACCFILE
THRFILE  INLFILE
```

Python syntax:

```
ierr = accc_with_cor(tol, optacc, optcor, values, labels, dfxfile,
accfile, thrfile, inlfile)
```

Fortran syntax:

```
CALL ACCC_WITH_COR(TOL, OPTACC, OPTCOR, VALUES, LABELS, DFXFILE,
ACCFILE, THRFILE, INLFILE, IERR)
```

where:

Real TOL	Is the mismatch tolerance (input; Newton solution convergence tolerance, TOLN, by default).
Integer OPTACC(8)	Is an array of eight elements specifying power flow solution options (input). The values are as follows:

OPTACC(1)	tap adjustment flag (tap adjustment option setting by default).
OPTACC(1) = 0	disable.
OPTACC(1) = 1	enable stepping adjustment.
OPTACC(1) = 2	enable direct adjustment.
OPTACC(2)	area interchange adjustment flag (area interchange adjustment option setting by default).
OPTACC(2) = 0	disable.
OPTACC(2) = 1	enable using tie line flows only in calculating area interchange.
OPTACC(2) = 2	enable using tie line flows and loads in calculating area interchange.
OPTACC(3)	phase shift adjustment flag (phase shift adjustment option setting by default).
OPTACC(3) = 0	disable.
OPTACC(3) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTACC(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
	OPTACC(4) = 0	disable.
	OPTACC(4) = 1	enable.
	OPTACC(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
	OPTACC(5) = 0	disable.
	OPTACC(5) = 1	enable.
	OPTACC(5) = 2	enable continuous mode, disable discrete mode.
	OPTACC(6)	solution method flag (0 by default).
	OPTACC(6) = 0	FDNS.
	OPTACC(6) = 1	FNSL.
	OPTACC(6) = 2	optimized FDNS.
	OPTACC(7)	non-divergent solution flag (non-divergent solution option setting by default).
	OPTACC(7) = 0	disable.
	OPTACC(7) = 1	enable.
	OPTACC(8)	dispatch mode (0 by default).
	OPTACC(8) = 0	disable.
	OPTACC(8) = 1	subsystem machines (reserve).
	OPTACC(8) = 2	subsystem machines (pmax).
	OPTACC(8) = 3	subsystem machines (inertia).
	OPTACC(8) = 4	subsystem machines (governor droop).
Integer OPTCOR(5)	Is an array of five elements specifying corrective actions options (input). The values are as follows:	
	OPTCOR(1)	run corrective actions (0 by default).
	OPTCOR(1) = 0	disable.
	OPTCOR(1) = 1	enable.
	OPTCOR(2)	number of power flow solutions (from 0 to 10, 1 by default)
	OPTCOR(3)	generator control flag (1 by default).
	OPTCOR(3) = 0	disable.
	OPTCOR(3) = 1	enable.
	OPTCOR(4)	load curtailment flag (0 by default).
	OPTCOR(4) = 0	disable.
	OPTCOR(4) = 1	enable.
	OPTCOR(5)	phase shifter control flag (1 by default).
	OPTCOR(5) = 0	disable.
	OPTCOR(5) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real VALUES(5)	Is an array of five elements specifying corrective actions parameters (input). The values are as follows:	
	VALUES(1)	bus voltage violation tolerance (0.1 by default).
	VALUES(2)	branch flow overload tolerance (0.1 by default).
	VALUES(3)	generator control weighting factor (1.0 by default).
	VALUES(4)	load control weighting factor (1.0 by default).
	VALUES(5)	phase shifter control weighting factor (1.0 by default).
Character LABELS(4)*12	Is an array of four elements specifying subsystem labels (input). The values are as follows:	
	LABELS(1)	label of generation dispatch subsystem (blank by default, no default allowed if OPTACC(8) is not 0).
	LABELS(2)	label of generator control subsystem (blank by default, no default allowed if OPTCOR(3) is 1).
	LABELS(3)	label of load control subsystem (blank by default, no default allowed if OPTCOR(4) is 1).
	LABELS(4)	label of phase shifter control subsystem (blank by default, no default allowed if OPTCOR(5) is 1).
Character DFXFILE*260	Is the name of the Distribution Factor Data File (input; no default allowed).	
Character ACCFILE*260	Is the name of the Contingency Solution Output File (input; no default allowed).	
Character THRFILE*260	Is the name of the Load Throwover Data File (input; blank by default).	
Character INLFILE*260	Is the name of the Unit Inertia and Governor Data File (input; blank by default).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid TOL value.
IERR = 2	invalid OPTACC value.
IERR = 3	generators are converted.
IERR = 4	buses in island(s) without a swing bus; use activity TREE.
IERR = 5	largest mismatch exceeds mismatch tolerance.
IERR = 6	generation dispatch subsystem is not defined.
IERR = 7	too many islands in base case.
IERR = 8	invalid OPTCOR value.
IERR = 9	invalid VALUES value.
IERR = 21	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
IERR = 22	monitored elements exceed limit when adding multi-section line members.
IERR = 50	error opening Unit Inertia and Governor Data File.
IERR = 51	error opening Contingency Solution Output File.
IERR = 52	error opening Distribution Factor Data File.
IERR = 53	error opening Load Throwover Data File.
IERR = 54	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.14 ACCC_WITH_COR_2

The API is obsolete. It has bee replaced by ACCC_WITH_COR_3, and implemented by a call to ACCC_WITH_COR_3.

This API is the second release of function to run AC contingency analysis with corrective actions. Corrective actions are specified automatically to eliminate the violations in each post contingency solution with the objective of minimizing the controls adjustments including load shedding, MW generation dispatch, phase shifter angle adjustments, off-line generator controls, tap setting adjustments and switched shunt controls. Please note that flows are checked against rating set program option setting and 100.0 percent of the rating to determine flow overload violations.

Batch command syntax:

```
BAT_ACCC_WITH_COR_2 TOL OPTACC(1)…OPTACC(8), OPTCOR(1)…OPTCOR(8),
VALUES(1)…VALUES(8), LABELS(1)…LABELS(7),  DFXFILE, ACCFILE, THRFILE,
INLFILE
```

Python syntax:

```
ierr = accc_with_cor_2(tol, optacc, optcor, values, labels,  dfxfile,
accfile, thrfile, inlfile)
```

Fortran syntax:

```
CALL ACCC_WITH_COR_2(TOL, OPTACC, OPTCOR, VALUES, LABELS,  DFXFILE,
ACCFILE, THRFILE, INLFILE, IERR)
```

where:

Real TOL	Is the mismatch tolerance (input, Newton solution convergence tolerance, TOLN, by default).	
Integer OPTACC(8)	Is an array of eight elements specifying power flow solution (input).	
	OPTACC(1)	tap adjustment flag (tap adjustment option setting by default).
	OPTACC(1) = 0	disable.
	OPTACC(1) = 1	enable stepping adjustment.
	OPTACC(1) = 2	enable direct adjustment.
	OPTACC(2)	area interchange adjustment flag (area interchange adjustment option setting by default).
	OPTACC(2) = 0	disable.
	OPTACC(2) = 1	enable using tie line flows only in calculating area interchange.
	OPTACC(2) = 2	enable using tie line flows and loads in calculating area interchange.
	OPTACC(3)	phase shift adjustment flag (phase shift adjustment option setting by default).
	OPTACC(3) = 0	disable.
	OPTACC(3) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTACC(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
	OPTACC(4) = 0	disable.
	OPTACC(4) = 1	enable.
	OPTACC(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
	OPTACC(5) = 0	disable.
	OPTACC(5) = 1	enable.
	OPTACC(5) = 2	enable continuous, disable discrete.
	OPTACC(6)	solution method flag (0 by default).
	OPTACC(6) = 0	FDNS.
	OPTACC(6) = 1	FNSL.
	OPTACC(6) = 2	optimized FDNS.
	OPTACC(7)	non-divergent solution flag (non-divergent solution option setting by default).
	OPTACC(7) = 0	disable.
	OPTACC(7) = 1	enable.
	OPTACC(8) = dispatch mode (0 by default).	
	OPTACC(8) = 0	disable.
	OPTACC(8) = 1	subsystem machines (RESERVE).
	OPTACC(8) = 2	subsystem machines (PMAX).
	OPTACC(8) = 3	subsystem machines (INERTIA).
	OPTACC(8) = 4	subsystem machines (GOVERNOR DROOP).
Integer OPTCOR(8)	Is an array of eight elements specifying corrective actions options.	
	OPTCOR(1)	run corrective actions (0 by default).
	OPTCOR(1) = 0	disable.
	OPTCOR(1) = 1	enable.
	OPTCOR(2)	number of power flows (1 by default).
	OPTCOR(3)	generator control flag (1 by default).
	OPTCOR(3) = 0	disable.
	OPTCOR(3) = 1	enable.
	OPTCOR(4)	load control flag (0 by default).
	OPTCOR(4) = 0	disable.
	OPTCOR(4) = 1	enable.
	OPTCOR(5)	phase shifter control flag (1 by default).
	OPTCOR(5) = 0	disable.
	OPTCOR(5) = 1	enable.
	OPTCOR(6)	off-line generator control flag (0 by default).
	OPTCOR(6) = 0	disable.
	OPTCOR(6) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTCOR(7)	tap setting adjustment flag (0 by default).
	OPTCOR(7) = 0	disable.
	OPTCOR(7) = 1	enable.
	OPTCOR(8)	switched shunt control flag (0 by default).
	OPTCOR(8) = 0	disable.
	OPTCOR(8) = 1	enable.
Real VALUES(8)	Is an array of eight elements specifying corrective actions parameters (input).	
	VALUES(1)	bus voltage violation tolerance (0.1 by default).
	VALUES(2)	branch overload violation tolerance (0.1 by default).
	VALUES(3)	generator control weighting factor (1.0 by default).
	VALUES(4)	load control weighting factor (1.0 by default).
	VALUES(5)	phase shifter control weighting factor (1.0 by default).
	VALUES(6)	off-line generator control weighting factor (1.0 by default).
	VALUES(7)	tap setting adjustment weighting factor (1.0 by default).
	VALUES(8)	switched shunt control weighting factor (1.0 by default).
Character LABELS(7)*12	Is an array of seven elements specifying subsystem labels.	
	LABELS(1)	label of generation dispatch subsystem (blank by default, no default allowed if OPTACC(8) is not 0).
	LABELS(2)	label of generator control subsystem (blank by default, no default allowed if OPTCOR(3) is 1).
	LABELS(3)	label of load control subsystem (blank by default, no default allowed if OPTCOR(4) is 1).
	LABELS(4)	label of phase shifter control subsystem (blank by default, no default allowed if OPTCOR(5) is 1).
	LABELS(5)	off-line generator control subsystem (blank by default; no default allowed if OPTCOR(6) is 1).
	LABELS(6)	tap setting adjustment subsystem (blank by default; no default allowed if OPTCOR(7) is 1).
	LABELS(7)	switched shunt control subsystem (blank by default; no default allowed if OPTCOR(8) is 1).
Character*260 DFXFILE	Is the name of Distribution Factor Data File (input, no default allowed).	
Character*260 ACCFILE	Is the name of Contingency Solution Output File (input, no default allowed).	
Character*260 THRFILE	Is the name of Load Throwover Data File (input, blank for none).	
Character*260 INLFILE	Is the name of Unit Inertia Data File (input, blank for none).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid TOL value.
IERR = 2	invalid OPTACC value.
IERR = 3	generators are converted.
IERR = 4	buses in island(s) without a swing bus; use activity TREE.
IERR = 5	largest mismatch exceeds mismatch tolerance.
IERR = 6	generation subsystem is not defined.
IERR = 7	too many islands in base case.
IERR = 8	invalid OPTCOR value.
IERR = 9	invalid VALUES value.
IERR = 21	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
IERR = 22	monitored elements exceed limit when adding multisection line members.
IERR = 50	error opening Unit Inertia Data File.
IERR = 51	error opening Contingency Solution Output File.
IERR = 52	error opening Distribution Factor Data File.
IERR = 53	error opening Load Throwover Data File.
IERR = 54	prerequisite requirements for function not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.15 ACCC_WITH_COR_3

This API is the third release of function to run AC contingency analysis with corrective actions. Corrective actions are specified automatically to eliminate the violations in each post contingency solution with the objective of minimizing the controls adjustments including load shedding, MW generation dispatch, phase shifter angle adjustments, off-line generator controls, tap setting adjustments and switched shunt controls. Please note that flows are checked against rating set program option setting and 100.0 percent of the rating to determine flow overload violations.

Batch command syntax:

```
BAT_ACCC_WITH_COR_3 TOL OPTACC(1)…OPTACC(10), OPTCOR(1)…OPTCOR(8),
VALUES(1)…VALUES(8), LABELS(1)…LABELS(7),  DFXFILE, ACCFILE, THRFILE,
INLFILE
```

Python syntax:

```
ierr = accc_with_cor_3(tol, optacc, optcor, values, labels,  dfxfile,
accfile, thrfile, inlfile)
```

Fortran syntax:

```
CALL ACCC_WITH_COR_3(TOL, OPTACC, OPTCOR, VALUES, LABELS,  DFXFILE,
ACCFILE, THRFILE, INLFILE, IERR)
```

where:

Real TOL	Is the mismatch tolerance (input, Newton solution convergence tolerance, TOLN, by default).	
Integer OPTACC(10)	Is an array of ten elements specifying power flow solution (input).	
	OPTACC(1)	tap adjustment flag (tap adjustment option setting by default
	OPTACC(1) = 0	disable.
	OPTACC(1) = 1	enable stepping adjustment.
	OPTACC(1) = 2	enable direct adjustment.
	OPTACC(2)	area interchange adjustment flag (area interchange adjustment option setting by default).
	OPTACC(2) = 0	disable.
	OPTACC(2) = 1	enable using tie line flows only in calculating area interchange.
	OPTACC(2) = 2	enable using tie line flows and loads in calculating area interchange.
	OPTACC(3)	phase shift adjustment flag (phase shift adjustment option setting by default).
	OPTACC(3) = 0	disable.
	OPTACC(3) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTACC(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
	OPTACC(4) = 0	disable.
	OPTACC(4) = 1	enable.
	OPTACC(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
	OPTACC(5) = 0	disable.
	OPTACC(5) = 1	enable.
	OPTACC(5) = 2	enable continuous, disable discrete.
	OPTACC(6)	solution method flag (0 by default).
	OPTACC(6) = 0	FDNS.
	OPTACC(6) = 1	FNSL.
	OPTACC(6) = 2	optimized FDNS.
	OPTACC(7)	non-divergent solution flag (non-divergent solution option setting by default).
	OPTACC(7) = 0	disable.
	OPTACC(7) = 1	enable.
	OPTIONS(8)	induction motor treatment flag (applied when an induction motor fails to solve due to low terminal bus voltage, 0 by default)
	OPTIONS(8) = 0	stall.
	OPTIONS(8) = 1	trip.
	OPTIONS(9)	induction machine failure flag(0 by default)
	OPTIONS(9) = 0	treat contingency as non-converged if any induction machines are placed in the “stalled” or “tripped” state.
	OPTIONS(9) = 1	treat contingency as solved if it converges, even if any induction machines are placed in the “stalled” or “tripped” state.
	OPTACC(10) = dispatch mode (0 by default).	
	OPTACC(10) = 0	disable.
	OPTACC(10) = 1	subsystem machines (RESERVE).
	OPTACC(10) = 2	subsystem machines (PMAX).
	OPTACC(10) = 3	subsystem machines (INERTIA).
	OPTACC(10) = 4	subsystem machines (GOVERNOR DROOP).
Integer OPTCOR(8)	Is an array of eight elements specifying corrective actions options.	
	OPTCOR(1)	run corrective actions (0 by default).
	OPTCOR(1) = 0	disable.
	OPTCOR(1) = 1	enable.
	OPTCOR(2)	number of power flows (1 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTCOR(3)	generator control flag (1 by default).
OPTCOR(3) = 0	disable.
OPTCOR(3) = 1	enable.
OPTCOR(4)	load control flag (0 by default).
OPTCOR(4) = 0	disable.
OPTCOR(4) = 1	enable.
OPTCOR(5)	phase shifter control flag (1 by default).
OPTCOR(5) = 0	disable.
OPTCOR(5) = 1	enable.
OPTCOR(6)	off-line generator control flag (0 by default).
OPTCOR(6) = 0	disable.
OPTCOR(6) = 1	enable.
OPTCOR(7)	tap setting adjustment flag (0 by default).
OPTCOR(7) = 0	disable.
OPTCOR(7) = 1	enable.
OPTCOR(8)	switched shunt control flag (0 by default).
OPTCOR(8) = 0	disable.
OPTCOR(8) = 1	enable.

Real VALUES(8) Is an array of eight elements specifying corrective actions parameters (input).

VALUES(1)	bus voltage violation tolerance (0.1 by default).
VALUES(2)	branch overload violation tolerance (0.1 by default).
VALUES(3)	generator control weighting factor (1.0 by default).
VALUES(4)	load control weighting factor (1.0 by default).
VALUES(5)	phase shifter control weighting factor (1.0 by default).
VALUES(6)	off-line generator control weighting factor (1.0 by default).
VALUES(7)	tap setting adjustment weighting factor (1.0 by default).
VALUES(8)	switched shunt control weighting factor (1.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character LABELS(7)*12	Is an array of seven elements specifying subsystem labels.	
	LABELS(1)	label of generation dispatch subsystem (blank by default, no default allowed if OPTACC(8) is not 0).
	LABELS(2)	label of generator control subsystem (blank by default, no default allowed if OPTCOR(3) is 1).
	LABELS(3)	label of load control subsystem (blank by default, no default allowed if OPTCOR(4) is 1).
	LABELS(4)	label of phase shifter control subsystem (blank by default, no default allowed if OPTCOR(5) is 1).
	LABELS(5)	off-line generator control subsystem (blank by default; no default allowed if OPTCOR(6) is 1).
	LABELS(6)	tap setting adjustment subsystem (blank by default; no default allowed if OPTCOR(7) is 1).
	LABELS(7)	switched shunt control subsystem (blank by default; no default allowed if OPTCOR(8) is 1).
Character*260 DFXFILE	Is the name of Distribution Factor Data File (input, no default allowed).	
Character*260 ACCFILE	Is the name of Contingency Solution Output File (input, no default allowed).	
Character*260 THRFILE	Is the name of Load Throwover Data File (input, blank for none).	
Character*260 INLFILE	Is the name of Unit Inertia Data File (input, blank for none).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid TOL value.
IERR = 2	invalid OPTACC value.
IERR = 3	generators are converted.
IERR = 4	buses in island(s) without a swing bus; use activity TREE.
IERR = 5	largest mismatch exceeds mismatch tolerance.
IERR = 6	generation subsystem is not defined.
IERR = 7	too many islands in base case.
IERR = 8	in-service induction machines are in the “stalled” or “tripped” state.
IERR = 9	buses with bus type code and series element status inconsistencies.
IERR = 10	invalid OPTCOR value.
IERR = 11	invalid VALUES value.
IERR = 21	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
IERR = 22	monitored elements exceed limit when adding multisection line members.
IERR = 50	error opening Unit Inertia Data File.
IERR = 51	error opening Contingency Solution Output File.
IERR = 52	error opening Distribution Factor Data File.
IERR = 53	error opening Load Throwover Data File.
IERR = 54	prerequisite requirements for function not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.16 ACCC_WITH_DSP

This API routine is obsolete. It has been replaced by the API routine ACCC_WITH_DSP_3, and is implemented by a call to ACCC_WITH_DSP_3.

Use this API routine to run the AC contingency calculation function. A generation dispatch function to handle imbalances in power resources and demand due to contingencies may be enabled.

Batch command syntax:

```
BAT_ACCC_WITH_DSP  TOL  OPTACC(1)...OPTACC(8)  LABEL  DFXFILE
ACCFILE  THRFILE  INLFILE
```

Python syntax:

```
ierr = accc_with_dsp(tol, optacc, label, dfxfile, accfile, thrfile,
inlfile)
```

Fortran syntax:

```
CALL ACCC_WITH_DSP(TOL, OPTACC, LABEL, DFXFILE, ACCFILE, THRFILE,
INLFILE, IERR)
```

where:

Real TOL	Is the mismatch tolerance (input; Newton solution convergence tolerance, TOLN, by default).	
Integer OPTACC(8)	Is an array of eight elements specifying solution options (input). The values are as follows:	
	OPTACC(1)	tap adjustment flag (tap adjustment option setting by default).
	OPTACC(1) = 0	disable.
	OPTACC(1) = 1	enable stepping adjustment.
	OPTACC(1) = 2	enable direct adjustment.
	OPTACC(2)	area interchange adjustment flag (area interchange adjustment option setting by default).
	OPTACC(2) = 0	disable.
	OPTACC(2) = 1	enable using tie line flows only in calculating area interchange.
	OPTACC(2) = 2	enable using tie line flows and loads in calculating area interchange.
	OPTACC(3)	phase shift adjustment flag (phase shift adjustment option setting by default).
	OPTACC(3) = 0	disable.
	OPTACC(3) = 1	enable.
	OPTACC(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
	OPTACC(4) = 0	disable.
	OPTACC(4) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTACC(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
OPTACC(5) = 0	disable.
OPTACC(5) = 1	enable.
OPTACC(5) = 2	enable continuous mode, disable discrete mode.
OPTACC(6)	solution method flag (0 by default).
OPTACC(6) = 0	FDNS.
OPTACC(6) = 1	FNSL.
OPTACC(6) = 2	optimized FDNS.
OPTACC(7)	non-divergent solution flag (non-divergent solution option setting by default).
OPTACC(7) = 0	disable.
OPTACC(7) = 1	enable.
OPTACC(8)	dispatch mode (0 by default).
OPTACC(8) = 0	disable.
OPTACC(8) = 1	subsystem machines (reserve).
OPTACC(8) = 2	subsystem machines (pmax).
OPTACC(8) = 3	subsystem machines (inertia).
OPTACC(8) = 4	subsystem machines (governor droop).

Character LABEL*12	Is the name of the generation dispatch subsystem (blank by default, no default allowed if OPTACC(8) is not 0).
Character DFXFILE*260	Is the name of the Distribution Factor Data file (input; no default allowed).
Character ACCFILE*260	Is the name of the Contingency Solution Output file (input; no default allowed).
Character THRFILE*260	Is the name of the Load Throwover Data file (input; blank by default).
Character INLFILE*260	Is the name of the Unit Inertia and Governor Data File (input; blank by default).
Integer IERR	Is the error code (output).

IERR = 0	no error occurred.
IERR > 0	as for ACCC_WITH_DSP_3.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.17 ACCC_WITH_DSP_2

This API routine is obsolete. It has been replaced by the API routine ACCC_WITH_DSP_3, and is implemented by a call to ACCC_WITH_DSP_3.

Use this API routine to run the second release of the AC contingency calculation function. A generation dispatch function to handle imbalances in power resources and demand due to contingencies may be enabled.

Batch command syntax:

```
BAT_ACCC_WITH_DSP_2  TOL  OPTIONS(1)...OPTIONS(10)  LABEL  DFXFILE
ACCFILE  THRFILE  INLFILE
```

Python syntax:

```
ierr = accc_with_dsp_2(tol, options, label, dfxfile, accfile, thrfile,
inlfile)
```

Fortran syntax:

```
CALL ACCC_WITH_DSP_2(TOL, OPTIONS, LABEL, DFXFILE, ACCFILE, THRFILE,
INLFILE, IERR)
```

where:

Real TOL	Is the mismatch tolerance (input; Newton solution convergence tolerance, TOLN, by default).
Integer OPTIONS(10)	Is an array of ten elements specifying solution options (input). The values are as follows:

OPTIONS(1)	tap adjustment flag (tap adjustment option setting by default).
OPTIONS(1) = 0	disable.
OPTIONS(1) = 1	enable stepping adjustment.
OPTIONS(1) = 2	enable direct adjustment.
OPTIONS(2)	area interchange adjustment flag (area interchange adjustment option setting by default).
OPTIONS(2) = 0	disable.
OPTIONS(2) = 1	enable using tie line flows only in calculating area interchange.
OPTIONS(2) = 2	enable using tie line flows and loads in calculating area interchange.
OPTIONS(3)	phase shift adjustment flag (phase shift adjustment option setting by default).
OPTIONS(3) = 0	disable.
OPTIONS(3) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer OPTIONS(10)	OPTIONS(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
	OPTIONS(4) = 0	disable.
	OPTIONS(4) = 1	enable.
	OPTIONS(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
	OPTIONS(5) = 0	disable.
	OPTIONS(5) = 1	enable.
	OPTIONS(5) = 2	enable continuous mode, disable discrete mode.
	OPTIONS(6)	solution method flag (0 by default).
	OPTIONS(6) = 0	FDNS.
	OPTIONS(6) = 1	FNSL.
	OPTIONS(6) = 2	optimized FDNS.
	OPTIONS(7)	non-divergent solution flag (non-divergent solution option setting by default).
	OPTIONS(7) = 0	disable.
	OPTIONS(7) = 1	enable.
	OPTIONS(8)	induction motor treatment flag (applied when an induction motor fails to solve due to low terminal bus voltage, 0 by default)
	OPTIONS(8) = 0	stall.
	OPTIONS(8) = 1	trip.
	OPTIONS(9)	induction machine failure flag(0 by default)
	OPTIONS(9) = 0	treat contingency as non-converged if any induction machines are placed in the “stalled” or “tripped” state.
	OPTIONS(9) = 1	treat contingency as solved if it converges, even if any induction machines are placed in the “stalled” or “tripped” state.
	OPTIONS(10)	dispatch mode (0 by default).
	OPTIONS(10) = 0	disable.
	OPTIONS(10) = 1	subsystem machines (reserve).
	OPTIONS(10) = 2	subsystem machines (pmax).
	OPTIONS(10) = 3	subsystem machines (inertia).
	OPTIONS(10) = 4	subsystem machines (governor droop).
Character LABEL*12	Is the name of the generation dispatch subsystem (blank by default, no default allowed if OPTIONS(10) is not 0).	
Character DFXFILE*260	Is the name of the Distribution Factor Data file (input; no default allowed).	
Character ACCFILE*260	Is the name of the Contingency Solution Output file (input; no default allowed).	
Character THRFILE*260	Is the name of the Load Throwover Data file (input; blank by default).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character INLFILE*260	Is the name of the Unit Inertia and Governor Data File (input; blank by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 as for ACCC_WITH_DSP_3.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.18 ACCC_WITH_DSP_3

Use this API routine to run the third release the AC contingency calculation function. A generation dispatch function to handle imbalances in power resources and demand due to contingencies may be enabled.

Batch command syntax:

```
BAT_ACCC_WITH_DSP_3  TOL  OPTIONS(1)...OPTIONS(11)  LABEL  DFXFILE
ACCFILE  THRFILE  INLFILE  ZIPFILE
```

Python syntax:

```
ierr = accc_with_dsp_3(tol, options, label, dfxfile, accfile, thrfile,
inlfile, zipfile)
```

Fortran syntax:

```
CALL ACCC_WITH_DSP_3(TOL, OPTIONS, LABEL, DFXFILE, ACCFILE, THRFILE,
INLFILE, ZIPFILE, IERR)
```

where:

Real TOL	Is the mismatch tolerance (input; Newton solution convergence tolerance, TOLN, by default).
Integer OPTIONS(11)	Is an array of eleven elements specifying solution options (input). The values are as follows:

OPTIONS(1)	tap adjustment flag (tap adjustment option setting by default).
OPTIONS(1) = 0	disable.
OPTIONS(1) = 1	enable stepping adjustment.
OPTIONS(1) = 2	enable direct adjustment.
OPTIONS(2)	area interchange adjustment flag (area interchange adjustment option setting by default).
OPTIONS(2) = 0	disable.
OPTIONS(2) = 1	enable using tie line flows only in calculating area interchange.
OPTIONS(2) = 2	enable using tie line flows and loads in calculating area interchange.
OPTIONS(3)	phase shift adjustment flag (phase shift adjustment option setting by default).
OPTIONS(3) = 0	disable.
OPTIONS(3) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer OPTIONS(11)	OPTIONS(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
	OPTIONS(4) = 0	disable.
	OPTIONS(4) = 1	enable.
	OPTIONS(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
	OPTIONS(5) = 0	disable.
	OPTIONS(5) = 1	enable.
	OPTIONS(5) = 2	enable continuous mode, disable discrete mode.
	OPTIONS(6)	solution method flag (0 by default).
	OPTIONS(6) = 0	FDNS.
	OPTIONS(6) = 1	FNSL.
	OPTIONS(6) = 2	optimized FDNS.
	OPTIONS(7)	non-divergent solution flag (non-divergent solution option setting by default).
	OPTIONS(7) = 0	disable.
	OPTIONS(7) = 1	enable.
	OPTIONS(8)	induction motor treatment flag (applied when an induction motor fails to solve due to low terminal bus voltage, 0 by default).
	OPTIONS(8) = 0	stall.
	OPTIONS(8) = 1	trip.
	OPTIONS(9)	induction machine failure flag (0 by default)
	OPTIONS(9) = 0	treat contingency as non-converged if any induction machines are placed in the “stalled” or “tripped” state.
	OPTIONS(9) = 1	treat contingency as solved if it converges, even if any induction machines are placed in the “stalled” or “tripped” state.
	OPTIONS(10)	dispatch mode (0 by default)
	OPTIONS(10) = 0	disable.
	OPTIONS(10) = 1	subsystem machines (reserve).
	OPTIONS(10) = 2	subsystem machines (pmax).
	OPTIONS(10) = 3	subsystem machines (inertia).
	OPTIONS(10) = 4	subsystem machines (governor droop).
	OPTIONS(11)	ZIP archive flag (0 by default)
	OPTIONS(11) = 0	do not write a ZIP archive file.
	OPTIONS(11) = 1	write a ZIP archive using the file specified as ZIPFILE.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character LABEL*12	Is the name of the generation dispatch subsystem (blank by default, no default allowed if OPTIONS(10) is not 0).
Character DFXFILE*260	Is the name of the Distribution Factor Data file (input; no default allowed).
Character ACCFILE*260	Is the name of the Contingency Solution Output file (input; no default allowed).
Character THRFILE*260	Is the name of the Load Throwover Data file (input; blank by default).
Character INLFILE*260	Is the name of the Unit Inertia and Governor Data File (input; blank by default).
Character ZIPFILE*260	Is the name of the ZIP Archive Output File (input; blank by default).
Integer IERR	Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid TOL value.
IERR = 2	invalid OPTIONS value.
IERR = 3	generators are converted.
IERR = 4	buses in island(s) without a swing bus; use activity TREE.
IERR = 5	largest mismatch exceeds mismatch tolerance.
IERR = 6	generation dispatch subsystem is not defined.
IERR = 7	too many islands in base case.
IERR = 8	no Distribution Factor Data File specified.
IERR = 9	no AC Contingency Solution Output File specified.
IERR = 10	in-service induction machines are in the “stalled” or “tripped” state.
IERR = 11	buses with bus type code and series element status inconsistencies.
IERR = 12	no ZIP Archive Output File specified.
IERR = 21	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
IERR = 22	monitored elements exceed limit when adding multi-section line members.
IERR = 51	error opening Contingency Solution Output File.
IERR = 52	error opening Distribution Factor Data File.
IERR = 53	error opening Load Throwover Data File.
IERR = 54	error opening Unit Inertia and Governor Data File.
IERR = 55	error opening ZIP Archive Output File.
IERR = 56	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.19 ACCC_WITH_TRIP

The API is obsolete. It has been replaced by ACCC_WITH_TRIP_2, and implemented by a call to ACCC_WITH_TRIP_2.

Use this API to apply the AC contingency calculation function with a post-contingency tripping function. This may be used, for example, to simulate switchings that occur when specified monitored quantities are beyond threshold values.

Batch command syntax:

```
BAT_ACCC_WITH_TRIP  TOL  OPTACC(1)...OPTACC(8)  OPTTRP(1)...OPTTRP(3)
LABEL   DFXFILE   ACCFILE   THRFILE  INLFILE  TRPFILE
```

Python syntax:

```
ierr = accc_with_trip(tol, optacc, opttrp, label, dfxfile, accfile,
thrfile, inlfile, trpfile)
```

Fortran syntax:

```
CALL ACCC_WITH_TRIP(TOL, OPTACC, OPTTRP, LABEL, DFXFILE, ACCFILE,
THRFILE, INLFILE, TRPFILE, IERR)
```

where:

Real TOL	Is the mismatch tolerance (input, Newton solution convergence tolerance, TOLN, by default).	
Integer OPTACC(8)	Is an array of eight elements specifying power flow solution options (input). The values are as follows:	
	OPTACC(1)	tap adjustment flag (tap adjustment option setting by default).
	OPTACC(1) = 0	disable.
	OPTACC(1) = 1	enable stepping adjustment.
	OPTACC(1) = 2	enable direct adjustment.
	OPTACC(2)	area interchange adjustment flag (area interchange adjustment option setting by default).
	OPTACC(2) = 0	disable.
	OPTACC(2) = 1	enable using tie line flows only in calculating area interchange.
	OPTACC(2) = 2	enable using tie line flows and loads in calculating area interchange.
	OPTACC(3)	phase shift adjustment flag (phase shift adjustment option setting by default).
	OPTACC(3) = 0	disable.
	OPTACC(3) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTACC(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
	OPTACC(4) = 0	disable.
	OPTACC(4) = 1	enable.
	OPTACC(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
	OPTACC(5) = 0	disable.
	OPTACC(5) = 1	enable.
	OPTACC(5) = 2	enable continuous mode, disable discrete mode.
	OPTACC(6)	solution method flag (0 by default).
	OPTACC(6) = 0	FDNS.
	OPTACC(6) = 1	FNSL.
	OPTACC(6) = 2	optimized FDNS.
	OPTACC(7)	non-divergent solution flag (non-divergent solution option setting by default).
	OPTACC(7) = 0	disable.
	OPTACC(7) = 1	enable.
	OPTACC(8)	dispatch mode (0 by default).
	OPTACC(8) = 0	disable.
	OPTACC(8) = 1	subsystem machines (reserve).
	OPTACC(8) = 2	subsystem machines (pmax).
	OPTACC(8) = 3	subsystem machines (inertia).
	OPTACC(8) = 4	subsystem machines (governor droop).
Integer OPTTRP(3)	Is an array of three elements specifying tripping simulation options (input). The values are as follows:	
	OPTTRP(1)	tripping simulation flag (0 by default).
	OPTTRP(1) = 0	disable.
	OPTTRP(1) = 1	enable tripping simulation.
	OPTTRP(2)	number of tripping events per power flow (1 by default).
	OPTTRP(3)	number of power flows per contingency (1 by default).
Character LABEL*12	Is the name of the generation dispatch subsystem (blank by default; no default allowed if OPTACC(8) is not 0).	
Character DFXFILE*260	Is the name of the Distribution Factor Data File (input; no default allowed).	
Character ACCFILE*260	Is the name of the Contingency Solution Output File (input; no default allowed).	
Character THRFILE*260	Is the name of the Load Throwover Data File (input; blank by default).	
Character INLFILE*260	Is the name of the Unit Inertia and Governor Data File (input; blank by default).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character TRPFILE*260	Is the name of the Tripping Element Data File (input, blank for none; no default allowed if OPTTRP(1) is 1).
Integer IERR	Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid TOL value.
IERR = 2	invalid OPTACC value.
IERR = 3	generators are converted.
IERR = 4	buses in island(s) without a swing bus; use activity TREE.
IERR = 5	largest mismatch exceeds mismatch tolerance.
IERR = 6	generation subsystem is not defined.
IERR = 7	too many islands in base case.
IERR = 8	Invalid OPTTRP value.
IERR = 21	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
IERR = 22	monitored elements exceed limit when adding multi-section line members.
IERR = 49	error opening Tripping Element Data File.
IERR = 50	error opening Unit Inertia and Governor Data File.
IERR = 51	error opening Contingency Solution Output File.
IERR = 52	error opening Distribution Factor Data File.
IERR = 53	error opening Load Throwover Data File.
IERR = 54	prerequisite requirements for function not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.20 ACCC_WITH_TRIP_2

Use this API to run the second version of AC contingency calculation function with a post-contingency tripping function. This may be used, for example, to simulate switchings that occur when specified monitored quantities are beyond threshold values.

Batch command syntax:

```
BAT_ACCC_WITH_TRIP_2  TOL  OPTACC(1)...OPTACC(10)
OPTTRP(1)...OPTTRP(3)  LABEL   DFXFILE   ACCFILE   THRFILE  INLFILE
TRPFILE
```

Python syntax:

```
ierr = accc_with_trip_2(tol, optacc, opttrp, label, dfxfile, accfile,
thrfile, inlfile, trpfile)
```

Fortran syntax:

```
CALL ACCC_WITH_TRIP_2(TOL, OPTACC, OPTTRP, LABEL, DFXFILE, ACCFILE,
THRFILE, INLFILE, TRPFILE, IERR)
```

where:

Real TOL	Is the mismatch tolerance (input, Newton solution convergence tolerance, TOLN, by default).
Integer OPTACC(10)	Is an array of ten elements specifying power flow solution options (input). The values are as follows:
	OPTACC(1) — tap adjustment flag (tap adjustment option setting by default).
	OPTACC(1) = 0 — disable.
	OPTACC(1) = 1 — enable stepping adjustment.
	OPTACC(1) = 2 — enable direct adjustment.
	OPTACC(2) — area interchange adjustment flag (area interchange adjustment option setting by default).
	OPTACC(2) = 0 — disable.
	OPTACC(2) = 1 — enable using tie line flows only in calculating area interchange.
	OPTACC(2) = 2 — enable using tie line flows and loads in calculating area interchange.
	OPTACC(3) — phase shift adjustment flag (phase shift adjustment option setting by default).
	OPTACC(3) = 0 — disable.
	OPTACC(3) = 1 — enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTACC(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
OPTACC(4) = 0	disable.
OPTACC(4) = 1	enable.
OPTACC(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
OPTACC(5) = 0	disable.
OPTACC(5) = 1	enable.
OPTACC(5) = 2	enable continuous mode, disable discrete mode.
OPTACC(6)	solution method flag (0 by default).
OPTACC(6) = 0	FDNS.
OPTACC(6) =	1FNSL.
OPTACC(6) = 2	optimized FDNS.
OPTACC(7)	non-divergent solution flag (non-divergent solution option setting by default).
OPTACC(7) = 0	disable.
OPTACC(7) = 1	enable.
OPTACC(8)	induction motor treatment flag (applied when an induction motor fails to solve due to low terminal bus voltage, 0 by default).
OPTACC(8) = 0	stall.
OPTACC(8) = 1	trip.
OPTACC(9)	induction machine failure flag(0 by default).
OPTACC(9) = 0	treat contingency as non-converged if any induction machines are placed in the “stalled” or “tripped” state.
OPTACC(9) = 1	treat contingency as solved if it converges, even if any induction machines are placed in the “stalled” or “tripped” state.
OPTACC(10)	dispatch mode (0 by default).
OPTACC(10) = 0	disable.
OPTACC(10) = 1	subsystem machines (reserve).
OPTACC(10) = 2	subsystem machines (pmax).
OPTACC(10) = 3	subsystem machines (inertia).
OPTACC(10) = 4	subsystem machines (governor droop).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer OPTTRP(3)	Is an array of three elements specifying tripping simulation options (input). The values are as follows: OPTTRP(1) tripping simulation flag (0 by default). OPTTRP(1) = 0 disable. OPTTRP(1) = 1 enable tripping simulation. OPTTRP(2) number of tripping events per power flow (1 by default). OPTTRP(3) number of power flows per contingency (1 by default).
Character LABEL*12	Is the name of the generation dispatch subsystem (blank by default; no default allowed if OPTACC(8) is not 0).
Character DFXFILE*260	Is the name of the Distribution Factor Data File (input; no default allowed).
Character ACCFILE*260	Is the name of the Contingency Solution Output File (input; no default allowed).
Character THRFILE*260	Is the name of the Load Throwover Data File (input; blank by default).
Character INLFILE*260	Is the name of the Unit Inertia and Governor Data File (input; blank by default).
Character TRPFILE*260	Is the name of the Tripping Element Data File (input, blank for none; no default allowed if OPTTRP(1) is 1).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid TOL value. IERR = 2 invalid OPTACC value. IERR = 3 generators are converted. IERR = 4 buses in island(s) without a swing bus; use activity TREE. IERR = 5 largest mismatch exceeds mismatch tolerance. IERR = 6 generation subsystem is not defined. IERR = 7 too many islands in base case. IERR = 8 in-service induction machines are in the “stalled” or “tripped” state. IERR = 9 buses with bus type code and series element status inconsistencies. IERR = 10 Invalid OPTTRP value. IERR = 21 file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX. IERR = 22 monitored elements exceed limit when adding multi-sec-tion line members.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IERR = 49	error opening Tripping Element Data File.
IERR = 50	error opening Unit Inertia and Governor Data File.
IERR = 51	error opening Contingency Solution Output File.
IERR = 52	error opening Distribution Factor Data File.
IERR = 53	error opening Load Throwover Data File.
IERR = 54	prerequisite requirements for function not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.21 ACCC_WITH_TRIP_PARALLEL

Use this API to run the parallel version of AC contingency calculation function with a post-contingency tripping function (use ACCC_WITH_TRIP_2 to run the function sequentially). The API distributes contingency cases among multiple processors evenly, and tests the contingency cases and following tripping actions assigned to each processor in parallel.

Batch command syntax:

```
BAT_ACCC_WITH_TRIP_PARALLEL TOL OPTACC(1)...OPTACC(8)
OPTTRP(1)...OPTTRP(3)

LABEL DFXFILE ACCFILE THRFILE INLFILE TRPFILE
```

Python syntax:

```
ierr = accc_with_trip_parallel(tol, optacc, opttrp, label, dfxfile,
accfile, thrfile, inlfile, trpfile)
```

Fortran syntax:

```
CALL ACCC_WITH_TRIP_PARALLEL(TOL, OPTACC, OPTTRP, LABEL, DFXFILE,
ACCFILE, THRFILE, INLFILE, TRPFILE, IERR)
```

where:

Real TOL	Is the mismatch tolerance (input, Newton solution convergence tolerance, TOLN, by default).
Integer OPTACC(10)	Is an array of ten elements specifying power flow solution options (input). The values are as follows:

OPTACC(1)	tap adjustment flag (tap adjustment option setting by default).
OPTACC(1) = 0	disable.
OPTACC(1) = 1	enable stepping adjustment.
OPTACC(1) = 2	enable direct adjustment.
OPTACC(2)	area interchange adjustment flag (area interchange adjustment option setting by default).
OPTACC(2) = 0	disable.
OPTACC(2) = 1	enable using tie line flows only in calculating area interchange.
OPTACC(2) = 2	enable using tie line flows and loads in calculating area interchange.
OPTACC(3)	phase shift adjustment flag (phase shift adjustment option setting by default).
OPTACC(3) = 0	disable.
OPTACC(3) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTACC(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
OPTACC(4) = 0	disable.
OPTACC(4) = 1	enable.
OPTACC(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
OPTACC(5) = 0	disable.
OPTACC(5) = 1	enable.
OPTACC(5) = 2	enable continuous mode, disable discrete mode.
OPTACC(6)	solution method flag (0 by default).
OPTACC(6) = 0	FDNS.
OPTACC(6) =	1FNSL.
OPTACC(6) = 2	optimized FDNS.
OPTACC(7)	non-divergent solution flag (non-divergent solution option setting by default).
OPTACC(7) = 0	disable.
OPTACC(7) = 1	enable.
OPTACC(8)	induction motor treatment flag (applied when an induction motor fails to solve due to low terminal bus voltage, 0 by default).
OPTACC(8) = 0	stall.
OPTACC(8) = 1	trip.
OPTACC(9)	induction machine failure flag(0 by default).
OPTACC(9) = 0	treat contingency as non-converged if any induction machines are placed in the “stalled” or “tripped” state.
OPTACC(9) = 1	treat contingency as solved if it converges, even if any induction machines are placed in the “stalled” or “tripped” state.
OPTACC(10)	dispatch mode (0 by default).
OPTACC(10) = 0	disable.
OPTACC(10) = 1	subsystem machines (reserve).
OPTACC(10) = 2	subsystem machines (pmax).
OPTACC(10) = 3	subsystem machines (inertia).
OPTACC(10) = 4	subsystem machines (governor droop).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer OPTTRP(3)	Is an array of three elements specifying tripping simulation options (input). The values are as follows: OPTTRP(1) tripping simulation flag (0 by default). OPTTRP(1) = 0 disable. OPTTRP(1) = 1 enable tripping simulation. OPTTRP(2) number of tripping events per power flow (1 by default). OPTTRP(3) number of power flows per contingency (1 by default).
Character LABEL*12	Is the name of the generation dispatch subsystem (blank by default; no default allowed if OPTACC(8) is not 0).
Character DFXFILE*260	Is the name of the Distribution Factor Data File (input; no default allowed).
Character ACCFILE*260	Is the name of the Contingency Solution Output File (input; no default allowed).
Character THRFILE*260	Is the name of the Load Throwover Data File (input; blank by default).
Character INLFILE*260	Is the name of the Unit Inertia and Governor Data File (input; blank by default).
Character TRPFILE*260	Is the name of the Tripping Element Data File (input, blank for none; no default allowed if OPTTRP(1) is 1).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid TOL value. IERR = 2 invalid OPTACC value. IERR = 3 generators are converted. IERR = 4 buses in island(s) without a swing bus; use activity TREE. IERR = 5 largest mismatch exceeds mismatch tolerance. IERR = 6 generation subsystem is not defined. IERR = 7 too many islands in base case. IERR = 8 in-service induction machines are in the “stalled” or “tripped” state. IERR = 9 buses with bus type code and series element status inconsistencies. IERR = 10 Invalid OPTTRP value. IERR = 21 file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX. IERR = 22 monitored elements exceed limit when adding multi-sec-tion line members.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IERR = 49	error opening Tripping Element Data File.
IERR = 50	error opening Unit Inertia and Governor Data File.
IERR = 51	error opening Contingency Solution Output File.
IERR = 52	error opening Distribution Factor Data File.
IERR = 53	error opening Load Throwover Data File.
IERR = 54	prerequisite requirements for function not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.22 ACCOR

Use this API to apply corrective actions to the base case using successive linear programming methods. The corrective actions are to eliminate flow overloads and voltage violations with the objective of minimizing the control adjustments, including load shedding, MW generations dispatch, and phase shifter angle adjustments.

Batch command syntax:

```
BAT_ACCOR  OPTIONS(1)...OPTIONS(10)  VALUES(1)...VALUES(7)
LABELS(1)...LABELS(3)  DFXFILE
```

Python syntax:

```
ierr = accor(options, values, labels, dfxfile)
```

Fortran syntax:

```
CALL ACCORAPI(OPTIONS, VALUES, LABELS, DFXFILE, IERR)
```

where:

Integer OPTIONS(10)	Is an array of ten elements specifying solution options (input). The values are as follows:	
	OPTIONS(1)	tap adjustment flag (tap adjustment option setting by default).
	OPTIONS(1) = 0	disable.
	OPTIONS(1) = 1	enable stepping adjustment.
	OPTIONS(1) = 2	enable direct adjustment.
	OPTIONS(2)	area interchange adjustment flag (area interchange adjustment option setting by default).
	OPTIONS(2) = 0	disable.
	OPTIONS(2) = 1	enable using tie line flows only in calculating area interchange.
	OPTIONS(2) = 2	enable using tie line flows and loads in calculating area interchange.
	OPTIONS(3)	phase shift adjustment flag (phase shift adjustment option setting by default).
	OPTIONS(3) = 0	disable.
	OPTIONS(3) = 1	enable.
	OPTIONS(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
	OPTIONS(4) = 0	disable.
	OPTIONS(4) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTIONS(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
	OPTIONS(5) = 0	disable.
	OPTIONS(5) = 1	enable.
	OPTIONS(5) = 2	enable continuous mode, disable discrete mode.
	OPTIONS(6)	rating set (rating set program option setting by default).
	OPTIONS(6) = 1	rating A
	OPTIONS(6) = 2	rating B
	OPTIONS(6) = 3	rating C
	OPTIONS(7)	number of power flow solutions (from 1 to 10; 1 by default).
	OPTIONS(8)	generator control flag (1 by default).
	OPTIONS(8) = 0	disable.
	OPTIONS(8) = 1	enable.
	OPTIONS(9)	load control flag (0 by default).
	OPTIONS(9) = 0	disable.
	OPTIONS(9) = 1	enable.
	OPTIONS(10)	phase shifter control flag (1 by default).
	OPTIONS(10) = 0	disable.
	OPTIONS(10) = 1	enable.
Real VALUES(7)	Is an array of seven elements specifying AC corrective actions parameters (input). The values are as follows:	
	VALUES(1)	MW mismatch tolerance (Newton solution convergence tolerance, TOLN, by default)..
	VALUES(2)	percent of rating set (100.0 by default).
	VALUES(3)	bus voltage violation tolerance (0.1 by default).
	VALUES(4)	branch flow overload tolerance (0.1 by default).
	VALUES(5)	generator control weighting factor (1.0 by default).
	VALUES(6)	load control weighting factor (1.0 by default).
	VALUES(7)	phase shifter control weighting factor (1.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character LABELS(3)*12	Is an array of three elements specifying subsystem labels (input). The values are as follows:	
	LABELS(1)	label of generator control subsystem (blank by default; no default allowed if OPTIONS(8) is 1).
	LABELS(2)	label of load control subsystem (blank by default; no default allowed if OPTIONS(9) is 1).
	LABELS(3)	label of phase shifter control subsystem (blank by default; no default allowed if OPTIONS(10) is 1).
Character DFXFILE*260	Is the name of the Distribution Factor Data File (input; no default allowed).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid OPTIONS value.
	IERR = 2	invalid VALUES value.
	IERR = 3	generators are converted.
	IERR = 4	buses in island(s) without a swing bus; use activity TREE.
	IERR = 5	No controls.
	IERR = 6	control subsystem not defined.
	IERR = 7	Distribution Factor Data File not specified.
	IERR = 8	largest mismatch exceeds tolerance.
	IERR = 21	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
	IERR = 22	monitored elements exceed limit when adding multi-section line members.
	IERR = 51	error opening Distribution Factor Data File.
	IERR = 52	prerequisite requirements for function not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.23 ACCOR_2

This API routine is obsolete. It has been replaced by the API routine ACCOR_3, and is implemented by a call to ACCOR_3.

This API is the second release of corrective action function. Use this API to augment control strategies with off-line generator control, tap setting adjustments and switched shunt controls.

Batch command syntax:

```
BAT_ACCOR_2 OPTIONS(1)...OPTIONS(13)  VALUES(1)...VALUES(10)
LABELS(1)...LABELS(6)  DFXFILE
```

Python syntax:

```
ierr = accor_2(options, values, labels, dfxfile)
```

Fortran syntax:

```
CALL ACCOR_2(OPTIONS, VALUES, LABELS, DFXFILE, IERR)
```

where:

Integer OPTIONS(13)	Is an array of thirteen elements specifying solution options (input). The values are as follows:	
	OPTIONS(1)	tap adjustment flag (tap adjustment option setting by default).
	OPTIONS(1) = 0	disable.
	OPTIONS(1) = 1	enable stepping adjustment.
	OPTIONS(1) = 2	enable direct adjustment.
	OPTIONS(2)	area interchange adjustment flag (area interchange adjustment option setting by default).
	OPTIONS(2) = 0	disable.
	OPTIONS(2) = 1	enable using tie line flows only in calculating area interchange.
	OPTIONS(2) = 2	enable using tie line flows and loads in calculating area interchange.
	OPTIONS(3)	phase shift adjustment flag (phase shift adjustment option setting by default).
	OPTIONS(3) = 0	disable.
	OPTIONS(3) = 1	enable.
	OPTIONS(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
	OPTIONS(4) = 0	disable.
	OPTIONS(4) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
OPTIONS(5) = 0	disable.
OPTIONS(5) = 1	enable.
OPTIONS(5) = 2	enable continuous mode, disable discrete mode.
OPTIONS(6)	rating set (rating set program option setting by default).
OPTIONS(6) = 1	rating A
OPTIONS(6) = 2	rating B
OPTIONS(6) = 3	rating C
OPTIONS(7)	number of power flow solutions (from 1 to 10; 1 by default)
OPTIONS(8)	generator control flag (1 by default).
OPTIONS(8) = 0	disable.
OPTIONS(8) = 1	enable.
OPTIONS(9)	load control flag (0 by default).
OPTIONS(9) = 0	disable.
OPTIONS(9) = 1	enable.
OPTIONS(10)	phase shifter control flag (1 by default).
OPTIONS(10) = 0	disable.
OPTIONS(10) = 1	enable.
OPTIONS(11)	off-line generator control flag (0 by default).
OPTIONS(11) = 0	disable.
OPTIONS(11) = 1	enable.
OPTIONS(12)	tap setting adjustment flag (0 by default).
OPTIONS(12) = 0	disable.
OPTIONS(12) = 1	enable.
OPTIONS(13)	switched shunt control flag (0 by default).
OPTIONS(13) = 0	disable.
OPTIONS(13) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real VALUES(10)	Is an array of ten elements specifying AC corrective actions parameters (input). The values are as follows:	
	VALUES(1)	MW mismatch tolerance (Newton solution convergence tolerance, TOLN, by default).
	VALUES(2)	percent of rating set (100.0 by default).
	VALUES(3)	bus voltage violation tolerance (0.1 by default).
	VALUES(4)	branch flow overload tolerance (0.1 by default).
	VALUES(5)	generator control weighting factor (1.0 by default).
	VALUES(6)	load control weighting factor (1.0 by default).
	VALUES(7)	phase shifter control weighting factor (1.0 by default).
	VALUES(8)	off-line generator control weighting factor (1.0 by default).
	VALUES(9)	tap setting adjustment weighting factor (1.0 by default).
	VALUES(10)	switched shunt control weighting factor (1.0 by default).
Character LABELS(6)*12	Is an array of three elements specifying subsystem labels (input). The values are as follows:	
	LABELS(1)	label of generator control subsystem (blank by default; no default allowed if OPTIONS(8) is 1).
	LABELS(2)	label of load control subsystem (blank by default; no default allowed if OPTIONS(9) is 1).
	LABELS(3)	label of phase shifter control subsystem (blank by default; no default allowed if OPTIONS(10) is 1).
	LABELS(4)	off-line generator control subsystem (blank by default; no default allowed if OPTIONS(11) is 1).
	LABELS(5)	tap setting adjustment subsystem (blank by default; no default allowed if OPTIONS(12) is 1).
	LABELS(6)	switched shunt control subsystem (blank by default; no default allowed if OPTIONS(13) is 1).
Character DFXFILE*260	Is the name of the Distribution Factor Data File (input; no default allowed).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid OPTIONS value.
	IERR = 2	invalid VALUES value.
	IERR = 3	generators are converted.
	IERR = 4	buses in island(s) without a swing bus, use activity TREE.
	IERR = 5	No controls.
	IERR = 6	control subsystem not defined.
	IERR = 7	Distribution Factor Data File not specified.
	IERR = 8	largest mismatch exceeds tolerance.
	IERR = 21	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file, run DFAX.
	IERR = 22	monitored elements exceed limit when adding multi-section line members.
	IERR = 51	error opening Distribution Factor Data File.
	IERR = 52	prerequisite requirements for function not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.24 ACCOR_3

This API is the third release of corrective action function. Use this API to specify voltage limits in determining voltage violations.

Batch command syntax:

```
BAT_ACCOR_3 OPTIONS(1)...OPTIONS(14)  VALUES(1)...VALUES(10)
LABELS(1)...LABELS(6)  DFXFILE
```

Python syntax:

```
ierr = accor_3(options, values, labels, dfxfile)
```

Fortran syntax:

```
CALL ACCOR_3(OPTIONS, VALUES, LABELS, DFXFILE, IERR)
```

where:

Integer OPTIONS(14)	Is an array of fourteen elements specifying solution options (input). The values are as follows:	
	OPTIONS(1)	tap adjustment flag (tap adjustment option setting by default).
	OPTIONS(1) = 0	disable.
	OPTIONS(1) = 1	enable stepping adjustment.
	OPTIONS(1) = 2	enable direct adjustment.
	OPTIONS(2)	area interchange adjustment flag (area interchange adjustment option setting by default).
	OPTIONS(2) = 0	disable.
	OPTIONS(2) = 1	enable using tie line flows only in calculating area interchange.
	OPTIONS(2) = 2	enable using tie line flows and loads in calculating area interchange.
	OPTIONS(3)	phase shift adjustment flag (phase shift adjustment option setting by default).
	OPTIONS(3) = 0	disable.
	OPTIONS(3) = 1	enable.
	OPTIONS(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
	OPTIONS(4) = 0	disable.
	OPTIONS(4) = 1	enable.
	OPTIONS(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
	OPTIONS(5) = 0	disable.
	OPTIONS(5) = 1	enable.
	OPTIONS(5) = 2	enable continuous mode, disable discrete mode.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(6)	rating set (rating set program option setting by default).
OPTIONS(6) = 1	rating A
OPTIONS(6) = 2	rating B
OPTIONS(6) = 3	rating C
OPTIONS(7)	voltage limit
OPTIONS(7)=1	normal
OPTIONS(7)=2	emergency
OPTIONS(8)	number of power flow solutions (from 1 to 10; 1 by default)
OPTIONS(9)	generator control flag (1 by default).
OPTIONS(9) = 0	disable.
OPTIONS(9) = 1	enable.
OPTIONS(10)	load control flag (0 by default).
OPTIONS(10) = 0	disable.
OPTIONS(10) = 1	enable.
OPTIONS(11)	phase shifter control flag (1 by default).
OPTIONS(11) = 0	disable.
OPTIONS(11) = 1	enable.
OPTIONS(12)	off-line generator control flag (0 by default).
OPTIONS(12) = 0	disable.
OPTIONS(12) = 1	enable.
OPTIONS(13)	tap setting adjustment flag (0 by default).
OPTIONS(13) = 0	disable.
OPTIONS(13) = 1	enable.
OPTIONS(14)	switched shunt control flag (0 by default).
OPTIONS(14) = 0	disable.
OPTIONS(14) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real VALUES(10)	Is an array of ten elements specifying AC corrective actions parameters (input). The values are as follows:	
	VALUES(1)	MW mismatch tolerance (Newton solution convergence tolerance, TOLN, by default).
	VALUES(2)	percent of rating set (100.0 by default).
	VALUES(3)	bus voltage violation tolerance (0.1 by default).
	VALUES(4)	branch flow overload tolerance (0.1 by default).
	VALUES(5)	generator control weighting factor (1.0 by default).
	VALUES(6)	load control weighting factor (1.0 by default).
	VALUES(7)	phase shifter control weighting factor (1.0 by default).
	VALUES(8)	off-line generator control weighting factor (1.0 by default).
	VALUES(9)	tap setting adjustment weighting factor (1.0 by default).
	VALUES(10)	switched shunt control weighting factor (1.0 by default).
Character LABELS(6)*12	Is an array of six elements specifying subsystem labels (input). The values are as follows:	
	LABELS(1)	label of generator control subsystem (blank by default; no default allowed if OPTIONS(9) is 1).
	LABELS(2)	label of load control subsystem (blank by default; no default allowed if OPTIONS(10) is 1).
	LABELS(3)	label of phase shifter control subsystem (blank by default; no default allowed if OPTIONS(11) is 1).
	LABELS(4)	off-line generator control subsystem (blank by default; no default allowed if OPTIONS(12) is 1).
	LABELS(5)	tap setting adjustment subsystem (blank by default; no default allowed if OPTIONS(13) is 1).
	LABELS(6)	switched shunt control subsystem (blank by default; no default allowed if OPTIONS(14) is 1).
Character DFXFILE*260	Is the name of the Distribution Factor Data File (input; no default allowed).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid OPTIONS value.
IERR = 2	invalid VALUES value.
IERR = 3	generators are converted.
IERR = 4	buses in island(s) without a swing bus, use activity TREE.
IERR = 5	No controls.
IERR = 6	control subsystem not defined.
IERR = 7	Distribution Factor Data File not specified.
IERR = 8	largest mismatch exceeds tolerance.
IERR = 21	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file, run DFAX.
IERR = 22	monitored elements exceed limit when adding multi-section line members.
IERR = 51	error opening Distribution Factor Data File.
IERR = 52	prerequisite requirements for function not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.25 ALERT_OUTPUT

Use this API to specify the alert device.

Batch command syntax:

```
BAT_ALERT_OUTPUT  ISLCT  FILARG  OPTIONS(1)  OPTIONS(2)
```

Python syntax:

```
ierr = alert_output(islct, filarg, options)
```

Fortran syntax:

```
CALL OPEN_ALERT_DEVICE(ISLCT, FILARG, OPTIONS, IERR)
```

where:

Integer ISLCT	Is the virtual device selector (input).	
	ISLCT = 1	standard destination. In the PSS®E GUI standard output for alerts is the message box.
	ISLCT = 2	direct alert to a file.
	ISLCT = 3	direct alert to a printer.
	ISLCT = 4	direct alert to progress device.
	ISLCT = 5	direct alert to report device.
	ISLCT = 6	no output.
Character FILARG*260	Is the printer name or filename; ignored if ISLCT is not 2 or 3 (input).	
Integer OPTIONS(2)	Is an array of two elements specifying open and printing options; ignored if ISLCT is not 2 or 3 (input). The values are as follows:	
	OPTIONS(1)	file and printer open options:
	OPTIONS(1) = 0	open with carriage control format and, for files, for overwrite of existing files
	OPTIONS(1) = 1	open with list format.
	OPTIONS(1) = 2	open file for append (file output only).
	OPTIONS(1) = 2	set printer orientation to portrait mode (printer output only).
	OPTIONS(1) = 4	open printer in immediate mode (printer output only).
	The OPTIONS(1) values may be summed to select multiple options.	
	OPTIONS(2)	number of copies to print (printer output only).
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	prerequisite requirements for API are not met.
	IERR = 2	failed to open device.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.26 ALPH

Use this API to print an alphabetically sorted table of all buses in a specified subsystem of the working case (activity ALPH).

Batch command syntax:

```
BAT_ALPH  SID  ALL
```

Python syntax:

```
ierr = alph(sid, all)
```

Fortran syntax:

```
CALL ALPHNW(SID, ALL, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.27 ANSI

This API routine is obsolete. It has been replaced by the API routine ANSI_2, and is implemented by a call to the ANSI_2 API routine.

Use this API to calculate fault currents according to the ANSI standard (activity ANSI). To designate fault specification data as part of the API's calling sequence, INARG must be positive; to designate fault specification data in an ANSI Fault Specification Data input file, INARG must be zero and the filename specified as FILARG.

Batch command syntax:

```
BAT_ANSI  OPTIONS(1)...OPTIONS(3)  VALUES(1)...VALUES(4)  INARG
BUSARY(1)...BUSARY(INARG)  VLTARY(1)...VLTARY(INARG)
CPTARY(1)...CPTARY(INARG)  FILARG
```

Python syntax:

```
ierr = ansi(options, values, inarg, busary, vltary, cptary, filarg)
```

Fortran syntax:

```
CALL ANSIAPI(OPTIONS, VALUES, INARG, BUSARY, VLTARY, CPTARY, FILARG,
IERR)
```

where:

Integer OPTIONS(3)	Is an array of three elements specifying solution and reporting options (input). The values are as follows:	
	OPTIONS(1)	transformer impedance correction option (1 by default):
	OPTIONS(1) = 0	do not apply to zero sequence.
	OPTIONS(1) = 1	apply to zero sequence.
	OPTIONS(2)	set of ANSI decrement factors to use (1 by default):
	OPTIONS(2) = 0	account for dc decrement only (use ANSI table 3).
	OPTIONS(2) = 1	account for ac and dc decrements (use ANSI tables 1 and 2).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTIONS(3)	output option (0 by default):
	OPTIONS(3) = 0	summary output using ANSI X only, report format (blank delimited output fields).
	OPTIONS(3) = 1	summary output using ANSI X only, spreadsheet input format (comma delimited output fields).
	OPTIONS(3) = 2	detailed output using ANSI X only.
	OPTIONS(3) = 3	detailed output using ANSI R and X.
	OPTIONS(3) = 4	summary output using ANSI R and X, report format (blank delimited output fields).
	OPTIONS(3) = 5	summary output using ANSI R and X, spreadsheet input format (comma delimited output fields).
Real VALUES(4)	Is an array of four elements specifying divisors for calculating resistances of network elements where resistance is zero in the working case (input).	
	VALUES(1)	used for branches in the positive sequence network (40.0 by default).
	VALUES(2)	used for machines in the positive sequence network (80.0 by default).
	VALUES(3)	used for branches in the zero sequence network (40.0 by default).
	VALUES(4)	used for machines in the zero sequence network (80.0 by default).
Integer INARG	Is the fault specification option (input; 0 by default):	
	INARG = 0	fault specifications specified in input file.
	INARG = ≥1	fault specifications specified in calling sequence.
Integer BUSARY(INARG)	Is a list of faulted buses; ignored if INARG is 0 (input; no defaults allowed if INARG > 0).	
Real VLTARY(INARG)	Is a list of maximum operating voltages in pu; ignored if INARG is 0 (input; 1.0 by default).	
Real CPTARY(INARG)	Is a list of contact parting times in seconds; ignored if INARG is 0 (input; 1 cycle by default).	
Character FILARG*260	Is the filename of ANSI Fault Specification Data File; ignored if INARG > 0 (input; no default allowed if INARG is 0).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid OPTIONS value.
IERR = 2	invalid VALUES value.
IERR = 3	invalid INARG value.
IERR = 4	fault analysis is invalid due to the presence of in-service induction machines that are in the "stalled" or "tripped" state
IERR = 5	bus type code and series element status inconsistencies.
IERR = 6	error opening file FILARG.
IERR = 7	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.28 ANSI_2

This API routine executes the second release of the ANSI fault calculation function. Use this API routine to calculate fault currents according to the ANSI standard (activity ANSI). To designate fault specification data as part of the API's calling sequence, INARG must be positive; to designate fault specification data in an ANSI Fault Specification Data input file, INARG must be zero and the filename specified as FILARG.

Batch command syntax:

```
BAT_ANSI_2  OPTIONS(1)...OPTIONS(5)  VALUES(1)...VALUES(4)  INARG
BUSARY(1)...BUSARY(INARG)  VLTARY(1)...VLTARY(INARG)
CPTARY(1)...CPTARY(INARG)  FILARG
```

Python syntax:

```
ierr = ansi_2(options, values, inarg, busary, vltary, cptary, filarg)
```

Fortran syntax:

```
CALL ANSI_2(OPTIONS,VALUES,INARG,BUSARY,VLTARY,CPTARY,FILARG,IERR)
```

where:

Integer OPTIONS(5)	Is an array of four elements specifying solution and reporting options (input). The values are as follows:	
	OPTIONS(1)	fault analysis generator reactance option (0 by default).
	OPTIONS(1) = 0	use subtransient reactance.
	OPTIONS(1) = 1	use transient reactance.
	OPTIONS(1) = 2	use synchronous reactance.
	OPTIONS(2)	transformer impedance correction option (1 by default):
	OPTIONS(2) = 0	do not apply to zero sequence.
	OPTIONS(2) = 1	apply to zero sequence.
	OPTIONS(3)	set of ANSI decrement factors to use (1 by default):
	OPTIONS(3) = 0	account for dc decrement only (use ANSI table 3).
	OPTIONS(3) = 1	account for ac and dc decrements (use ANSI tables 1 and 2).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTIONS(4)	output option (0 by default):
	OPTIONS(4) = 0	summary output using ANSI X only, report format (blank delimited output fields).
	OPTIONS(4) = 1	summary output using ANSI X only, spreadsheet input format (comma delimited output fields).
	OPTIONS(4) = 2	detailed output using ANSI X only.
	OPTIONS(4) = 3	detailed output using ANSI R and X.
	OPTIONS(4) = 4	summary output using ANSI R and X, report format (blank delimited output fields).
	OPTIONS(4) = 5	summary output using ANSI R and X, spreadsheet input format (comma delimited output fields).
	OPTIONS(5)	breaker rating option (0 by default):
	OPTIONS(5) = 0	IEEE total current basis.
	OPTIONS(5) = 1	IEEE symmetrical current basis
Real VALUES(4)	Is an array of four elements specifying divisors for calculating resistances of network elements where resistance is zero in the working case (input).	
	VALUES(1)	used for branches in the positive sequence network (40.0 by default).
	VALUES(2)	used for machines in the positive sequence network (80.0 by default).
	VALUES(3)	used for branches in the zero sequence network (40.0 by default).
	VALUES(4)	used for machines in the zero sequence network (80.0 by default).
Integer INARG	Is the fault specification option (input; 0 by default):	
	INARG = 0	fault specifications specified in input file.
	INARG = ≥1	fault specifications specified in calling sequence.
Integer BUSARY(INARG)	Is a list of faulted buses; ignored if INARG is 0 (input; no defaults allowed if INARG > 0).	
Real VLTARY(INARG)	Is a list of maximum operating voltages in pu; ignored if INARG is 0 (input; 1.0 by default).	
Real CPTARY(INARG)	Is a list of contact parting times in seconds; ignored if INARG is 0 (input; 1 cycle by default).	
Character FILARG*260	Is the filename of ANSI Fault Specification Data File; ignored if INARG > 0 (input; no default allowed if INARG is 0).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid OPTIONS value.
IERR = 2	invalid VALUES value.
IERR = 3	invalid INARG value.
IERR = 4	fault analysis is invalid due to the presence of in-service induction machines that are in the "stalled" or "tripped" state
IERR = 5	bus type code and series element status inconsistencies.
IERR = 6	error opening file FILARG.
IERR = 7	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.29 APPEND_ACCC

Use this API to replicate system conditions of a contingency case solution, as contained in a designated Saved Case File, in the form of a Contingency Solution Output File. Results may be appended to an existing Contingency Solution Output File, or an new Contingency Solution Output File may be created.

Batch command syntax:

```
BAT_APPEND_ACCC  OPTION  DFXFILE  BASFILE  CNTFILE  ACCFILE  LABEL
NUMDESC  DESCRPT(1)...DESCRPT(NUMDESC)
```

Python syntax:

```
ierr = append_accc(option, dfxfile, basfile, cntfile, accfile, label,
numdesc, descrpt)
```

Fortran syntax:

```
CALL APPEND_ACCC(OPTION, DFXFILE, BASFILE, CNTFILE, ACCFILE, LABEL,
NUMDESC, DESCRPT, IERR)
```

where:

Integer OPTION	Is the append output flag (input; 1 by default): OPTION = 0 — create a new Contingency Solution Output File in the file specified in ACCFILE. OPTION = 1 — append results to the existing Contingency Solution Output File specified in ACCFILE
Character DFXFILE*260	Is the name of the Distribution Factor Data File (input; no default allowed).
Character BASFILE*260	Is the name of the Saved Case File containing the pre-contingency power flow working case. This is the case that was present when the Distribution Factor Data File specified in DFXFILE was constructed; it is also the case from which the contingency case was formed. Its solution is in the Saved Case File specified in CNTFILE (input; no default allowed).
Character CNTFILE*260	Is the name of the Saved Case File containing the contingency case solution for which results are to be placed in the Contingency Solution Output File specified in ACCFILE. This case must be formed by imposing status and/or boundary condition changes on the pre-contingency case specified in BASFILE, and then solving using one or more of the power flow solution activities (input; no default allowed).
Character ACCFILE*260	Is the name of the Contingency Solution Output File (input; no default allowed).
Character LABEL*12	Is the 12 character contingency label (input; blank by default).
Integer NUMDESC	Is the number of lines of text used to describe the contingency case; $1 \leq$ NUMDESC ≤ 16 (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character DESCRPT(NUMDESC)*120	Are NUMDESC lines of text describing the contingency case (input; blank by default).
Integer IERR	Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid OPTION value.
IERR = 2	invalid NUMDESC value.
IERR = 3	generators are converted.
IERR = 4	buses in island(s) without a swing bus; use activity TREE.
IERR = 5	no base case Saved Case File specified.
IERR = 6	no contingency case Saved Case File specified.
IERR = 7	Contingency Solution Output File not found.
IERR = 8	DFAX File and Contingency Solution Output File are not compatible.
IERR = 9	no Distribution Factor Data File specified.
IERR = 10	no Contingency Solution Output File specified.
IERR = 21	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
IERR = 22	monitored elements exceed limit when adding multi-section line members.
IERR = 51	error opening Contingency Solution Output File.
IERR = 52	error opening Distribution Factor Data File.
IERR = 53	error opening pre-contingency Saved Case File.
IERR = 54	error opening contingency case Saved Case File.
IERR = 55	error reading Contingency Solution Output File.
IERR = 56	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.30 AREA

This API is obsolete. It has been replaced by the API AREA_2, and is implemented by a call to AREA_2. Use this API to tabulate area totals by area, as well as the desired area net interchange (activity AREA).

Batch command syntax:

```
BAT_AREA  SID  ALL
```

Python syntax:

```
ierr = area(sid, all)
```

Fortran syntax:

```
CALL AREANW(SID, ALL, IERR)
```

where:

Integer SID	Is a valid area subsystem identifier. Valid subsystem identifiers range from 0 to 11. Area subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all areas* or specified subsystem flag (input; 1 by default). ALL = 1 process all areas. ALL = 0 process only areas in area subsystem SID.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.31 AREA_2

Use this API to tabulate area totals by area, as well as the desired area net interchange (activity AREA).

Batch command syntax:

```
BAT_AREA_2  SID  ALL  OPT
```

Python syntax:

```
ierr = area_2(sid, all, opt)
```

Fortran syntax:

```
CALL AREA_2(SID, ALL, OPT, IERR)
```

where:

Integer SID	Is a valid area subsystem identifier. Valid subsystem identifiers range from 0 to 11. Area subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all areas* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all areas.
	ALL = 0	process only areas in area subsystem SID.
Integer OPT	Is the load and induction machine reporting option (input; 2 by default).	
	OPT = 1	tabulate loads, induction generators and induction motors connected to area buses
	OPT = 2	tabulate loads, induction generators and induction motors assigned to the area
	OPT = 3	tabulate three load and induction machine subtotals for each area: • area load and induction machines connected to area buses • area load and induction machines connected to non-area buses • non-area load and induction machines connected to area buses
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	Invalid OPT value.
	IERR = 4	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.32 AREA_ZONE

Use this API to tabulate area totals by area, along with subtotals by zone. The desired net interchange of each area reported is also tabulated.

Batch command syntax:

```
BAT_AREA_ZONE  SID  ALL
```

Python syntax:

```
ierr = area_zone(sid, all)
```

Fortran syntax:

```
CALL AREA_ZONE(SID, ALL, IERR)
```

where:

Integer SID	Is a valid area subsystem identifier. Valid subsystem identifiers range from 0 to 11. Area subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all areas* or specified subsystem flag (input; 1 by default). ALL = 1 process all areas. ALL = 0 process only areas in area subsystem SID.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.33 ARNM

This API routine is obsolete. It has been replaced by the API routine ARNM_2, and is implemented by a call to the ARNM_2 API routine.

Use this API to reassign the buses and loads in a specified subsystem of the working case from their original area to a designated area (activity ARNM).

Batch command syntax:

```
BAT_ARNM  SID  ALL  STATUS(1)  STATUS(2)  IANEW
```

Python syntax:

```
ierr = arnm(sid, all, status, ianew)
```

Fortran syntax:

```
CALL ARNMNW(SID, ALL, STATUS, IANEW, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 — process all buses. ALL = 0 — process only buses in subsystem SID.
Integer STATUS(2)	Is an array of two elements (input). The values are as follows: STATUS(1) = 1 — change area assignments of buses to IANEW; otherwise = 0 (0 by default). STATUS(2) = 1 — change area assignments of loads to IANEW; otherwise = 0 (0 by default).
Integer IANEW	Is the new area to be assigned to specified buses and/or loads. IANEW must be between 1 and 9999 (input; 1 by default).
Integer IERR	Is the error code (output). IERR = 0 — no error occurred. IERR = 1 — invalid SID value or subsystem SID is not defined. IERR = 2 — invalid ALL value. IERR = 3 — invalid STATUS value. IERR = 4 — invalid new area number. IERR = 5 — area table is full; cannot add area to the working case. IERR = 6 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.34 ARNM_2

This API routine is the second release of the area renumbering function. Use this API routine to reassign the buses, loads and/or induction machines in a specified subsystem of the working case from their original area to a designated area (activity ARNM).

Batch command syntax:

```
BAT_ARNM_2  SID  ALL  STATUS(1)  STATUS(2)  STATUS(3)  IANEW
```

Python syntax:

```
ierr = arnm_2(sid, all, status, ianew)
```

Fortran syntax:

```
CALL ARNM_2(SID, ALL, STATUS, IANEW, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 — process all buses. ALL = 0 — process only buses in subsystem SID.
Integer STATUS(3)	Is an array of three elements (input). The values are as follows: STATUS(1) = 1 — change area assignments of buses to IANEW; 0 to leave area assignments of buses unchanged (0 by default). STATUS(2) = 1 — change area assignments of loads to IANEW; 0 to leave area assignments of loads unchanged (0 by default). STATUS(3) = 1 — change area assignments of induction machines to IANEW; 0 to leave area assignments of induction machines unchanged (0 by default).
Integer IANEW	Is the new area to be assigned to specified buses, load, and/or induction machines. IANEW must be between 1 and 9999 (input; 1 by default).
Integer IERR	Is the error code (output). IERR = 0 — no error occurred. IERR = 1 — invalid SID value or subsystem SID is not defined. IERR = 2 — invalid ALL value. IERR = 3 — invalid STATUS value. IERR = 4 — invalid new area number. IERR = 5 — area table is full; cannot add area to the working case. IERR = 6 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.35 ASCC

This API routine is obsolete. It has been replaced by the API routine ASCC_1A, and is implemented by a call to the ASCC_1A API routine.

There is a new release of this API, ASCC_3. This API is kept to be able to produce ASCC text output in the old format. Use this API to apply a series of faults at various locations in the working case (activity ASCC).

Batch command syntax:

```
BAT_ASCC  SID  ALL  STATUS(1)...STATUS(10)  RFILE  FFILE
```

Python syntax:

```
ierr = ascc(sid, all, status, rfile, ffile)
```

Fortran syntax:

```
CALL ASCCNW(SID, ALL, STATUS, RFILE, FFILE, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all buses.
	ALL = 0	process only buses in subsystem SID.
Integer STATUS(10)	Is an array of 10 elements (input). The values are as follows.	
	STATUS(1)	line-to-ground faults (0 by default).
	STATUS(1) = 0	three phase faults only.
	STATUS(1) = 1	three phase and line-to-ground faults.
	STATUS(2)	line-out faults (0 by default).
	STATUS(2) = 0	omit.
	STATUS(2) = 1	include.
	STATUS(3)	line-end faults (0 by default).
	STATUS(3) = 0	omit.
	STATUS(3) = 1	include.
	STATUS(4)	flat conditions option (0 by default).
	STATUS(4) = 0	pre-fault conditions taken from working case.
	STATUS(4) = 1	impose flat conditions as pre-fault condition.
	STATUS(5)	double-precision fault calculation option (ignored, but must be specified as 0 or 1; 0 by default; double precision always used).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

STATUS(6)	output option (1 by default).
STATUS(6) = 1	full output at home bus and N levels away.
STATUS(6) = 2	full output at home bus, summary N levels away.
STATUS(6) = 3	fault current summary table.
STATUS(7)	number of levels back for contributions output if STATUS(6) = 1 or 2. STATUS(7) ≥ 0. (0 by default)
STATUS(8)	branch quantity output code for line to ground faults (0 by default).
STATUS(8) = 0	A phase.
STATUS(8) = 1	3 ∗ IO.
STATUS(8) = 2	both.
STATUS(9)	dc line and FACTS device option (0 by default).
STATUS(9) = 0	blocked.
STATUS(9) = 1	represent as load.
STATUS(10)	transformer impedance correction option (1 by default).
STATUS(10) = 0	do not apply to zero sequence.
STATUS(10) = 1	apply to zero sequence.

Character RFILE*260 Is the Relay Output Data file; blank for none (input; blank by default).

Character FFILE*260 Is the Fault Control Data File; blank for none (input; blank by default).

Integer IERR Is the error code (output).

IERR = 0	no error occurred
IERR = 1	invalid SID value or subsystem SID is not defined
IERR = 2	invalid ALL value
IERR = 3	invalid STATUS value
IERR = 4	line end fault is invalid; bus table is full
IERR = 5	fault analysis is invalid due to the presence of in-service induction machines that are in the "stalled" or "tripped" state
IERR = 6	bus type code and series element status inconsistencies.
IERR = 7	error opening RFILE
IERR = 8	error opening FFILE
IERR = 9	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.36 ASCC_1A

This API routine has been replaced by the API routine ASCC_2. This API routine is retained so that fault output in the form of the original ASCC may be obtained.

Use this API routine to apply a series of faults at various locations in the working case (activity ASCC).

Batch command syntax:

```
BAT_ASCC_1A  SID  ALL  STATUS(1)...STATUS(10)  RFILE  FFILE
```

Python syntax:

```
ierr = ascc_1a(sid, all, status, rfile, ffile)
```

Fortran syntax:

```
CALL ASCC_1A(SID, ALL, STATUS, RFILE, FFILE, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all buses.
	ALL = 0	process only buses in subsystem SID.
Integer STATUS(10)	Is an array of 10 elements (input). The values are as follows.	
	STATUS(1)	line-to-ground faults (0 by default).
	STATUS(1) = 0	three phase faults only.
	STATUS(1) = 1	three phase and line-to-ground faults.
	STATUS(2)	line-out faults (0 by default).
	STATUS(2) = 0	omit.
	STATUS(2) = 1	include.
	STATUS(3)	line-end faults (0 by default).
	STATUS(3) = 0	omit.
	STATUS(3) = 1	include.
	STATUS(4)	flat conditions option (0 by default).
	STATUS(4) = 0	pre-fault conditions taken from working case.
	STATUS(4) = 1	impose flat conditions as pre-fault condition.
	STATUS(5)	fault analysis generator reactance option; ignored if generators are already "converted" (0 by default).
	STATUS(5) = 0	use subtransient reactance.
	STATUS(5) = 1	use transient reactance.
	STATUS(5) = 2	use synchronous reactance.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

STATUS(6)	output option (1 by default).
STATUS(6) = 1	full output at home bus and N levels away.
STATUS(6) = 2	full output at home bus, summary N levels away.
STATUS(6) = 3	fault current summary table.
STATUS(7)	number of levels back for contributions output if STATUS(6) = 1 or 2. STATUS(7) ≥ 0. (0 by default)
STATUS(8)	branch quantity output code for line to ground faults (0 by default).
STATUS(8) = 0	A phase.
STATUS(8) = 1	3 ∗ IO.
STATUS(8) = 2	both.
STATUS(9)	dc line and FACTS device option (0 by default).
STATUS(9) = 0	blocked.
STATUS(9) = 1	represent as load.
STATUS(10)	transformer impedance correction option (1 by default).
STATUS(10) = 0	do not apply to zero sequence.
STATUS(10) = 1	apply to zero sequence.

Character RFILE*260 Is the Relay Output Data file; blank for none (input; blank by default).

Character FFILE*260 Is the Fault Control Data File; blank for none (input; blank by default).

Integer IERR Is the error code (output).

IERR = 0	no error occurred
IERR = 1	invalid SID value or subsystem SID is not defined
IERR = 2	invalid ALL value
IERR = 3	invalid STATUS value
IERR = 4	line end fault is invalid; bus table is full
IERR = 5	fault analysis is invalid due to the presence of in-service induction machines that are in the "stalled" or "tripped" state
IERR = 6	bus type code and series element status inconsistencies.
IERR = 7	error opening RFILE
IERR = 8	error opening FFILE
IERR = 9	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.37 ASCC_2

This API routine is obsolete. It has been replaced by the API routine ASCC_3, and is imple-mented by a call to the ASCC_3 API routine.

This API executes the second release of the activity ASCC. This API is used to apply a series of faults at various locations in the working case (activity ASCC_2).

Batch command syntax:

```
BAT_ASCC_2 SID ALL STATUS(1)...STATUS(13) RELFIL FCDFIL SCFILE
```

Python syntax:

```
ierr = ascc_2(sid, all, status, relfil, fcdfil, scfile)
```

Fortran syntax:

```
CALL ASCC_2(SID, ALL, STATUS, RELFIL, FCDFIL, SCFILE, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the all buses or selected subsystem flag (input; 1 by default).	
	ALL = 1	process all buses
	ALL = 0	process only buses in subsystem SID
Integer STATUS(13)	Is an array of 13 elements (input). The values are as follows.	
	STATUS(1)	line-to-ground faults (0 by default)
	STATUS(1) = 0	omit
	STATUS(1) = 1	include
	STATUS(2)	line-out faults (0 by default)
	STATUS(2) = 0	omit
	STATUS(2) = 1	include
	STATUS(3)	line-end faults (0 by default)
	STATUS(3) = 0	omit
	STATUS(3) = 1	include
	STATUS(4)	flat conditions option (0 by default)
	STATUS(4) = 0	pre-fault conditions taken from working case
	STATUS(4) = 1	impose flat conditions as pre-fault condition
	STATUS(5)	double-precision fault calculation option (ignored, but must be specified as 0 or 1; 0 by default; double precision always used)
	STATUS(6)	output option (1 by default)
	STATUS(6) = 1	total fault currents
	STATUS(6) = 2	fault contributions to N levels away
	STATUS(6) = 3	total fault currents and fault contributions to N levels away

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

STATUS(7)	number of levels back for contributions output (0 by default),
STATUS(7) ≥ 0	used if STATUS(6) = 2 or 3
STATUS(8)	branch quantity output code (ignored, but must be specified as 0 or 1; 0 by default)
STATUS(9)	dc line and FACTS device option (0 by default)
STATUS(9) = 0	blocked
STATUS(9) = 1	represent as load
STATUS(10)	zero sequence transformer impedance correction option (0 by default)
STATUS(10) = 0	ignore
STATUS(10) = 1	apply
STATUS(11)	three phase faults, (0 by default)
STATUS(11) = 0	omit
STATUS(11) = 1	include
STATUS(12)	line-line-to-ground faults, (0 by default)
STATUS(12) = 0	omit
STATUS(12) = 1	include
STATUS(13)	line-to-line faults, (0 by default)
STATUS(13) = 0	omit
STATUS(13) = 1	include

Character RELFIL*260	Is the Relay Output Data (.rel) file; blank for none (input; blank by default).
Character FCDFIL*260	Is the Fault Control Input (.fcd) file; blank for none (input; blank by default).
Character SCFILE*260	Is the ASCC_2 Results (.sc) file; blank for none (output; blank by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output)	
	IERR = 0	no error occurred
	IERR = 1	invalid SID value or subsystem SID is not defined
	IERR = 2	invalid ALL value
	IERR = 3	invalid STATUS value
	IERR = 4	line end fault invalid; bus table is full
	IERR = 5	error opening RELFIL
	IERR = 6	error opening FCDFIL
	IERR = 7	prerequisite requirements for function not met
	IERR = 8	no faults specified
	IERR = 9	error opening SCFILE
	IERR = 10	SCFILE error
	IERR = 11	bus SC results not available
	IERR = 12	branch SC results not available
	IERR = 13	three winding transformer SC results not available
	IERR = 14	incorrect SCFILE
	IERR = 15	SCFILE not allowed

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.26 ASCC_3

This API executes the third release of the activity ASCC. This API is used to apply a series of faults at various locations in the working case (activity ASCC_3).
Batch command syntax:

```
BAT_ASCC_3 SID ALL STATUS(1)...STATUS(17) VALUES(1) RELFIL FCDFIL
SCFILE
```

Python syntax:

```
ierr = ascc_3(sid, all, status, values, relfil, fcdfil, scfile)
```

Fortran syntax:

```
CALL ASCC_3(SID, ALL, STATUS, VALUES, RELFIL, FCDFIL, SCFILE, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the all buses or selected subsystem flag (input; 1 by default).	
	ALL = 1	process all buses
	ALL = 0	process only buses in subsystem SID
Integer STATUS(17)	Is an array of 17 elements (input). The values are as follows.	
	STATUS(1)	line-to-ground faults (0 by default)
	STATUS(1) = 0	omit
	STATUS(1) = 1	include
	STATUS(2)	line-out faults (0 by default)
	STATUS(2) = 0	omit
	STATUS(2) = 1	include
	STATUS(3)	line-end faults (0 by default)
	STATUS(3) = 0	omit
	STATUS(3) = 1	include
	STATUS(4)	bus voltage option (0 by default)
	STATUS(4) = 0	use bus voltages from power flow solution
	STATUS(4) = 1	set all bus voltages at specified value and at 0 deg
	STATUS(4) = 2	set faulted bus voltage at specified value and at 0 deg

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

STATUS(5)	fault analysis generator reactance option; ignored if generators are already "converted" (0 by default).
STATUS(5) = 0	use subtransient reactance.
STATUS(5) = 1	use transient reactance.
STATUS(5) = 2	use synchronous reactance
STATUS(6)	output option (1 by default)
STATUS(6) = 0	fault current summary table
STATUS(6) = 1	total fault currents
STATUS(6) = 2	fault contributions to N levels away
STATUS(6) = 3	total fault currents and fault contributions to N levels away
STATUS(7)	number of levels back for contributions output (0 by default),
STATUS(7) ≥ 0	used if STATUS(6) = 2 or 3
STATUS(8)	transformer tap ratios and phase shift angles option (0 by default)
STATUS(8) = 0	leave tap ratios and phase shift angles unchanged
STATUS(8) = 1	set tap ratios to 1.0 pu and phase shift angles to 0
STATUS(8) = 2	set tap ratios to 1.0 pu and phase shift angles unchanged
STATUS(8) = 3	set tap ratios unchanged and phase shift angles to 0
STATUS(9)	dc line and FACTS device option (0 by default)
STATUS(9) = 0	blocked
STATUS(9) = 1	represent as load
STATUS(10)	zero sequence transformer impedance correction option (1 by default)
STATUS(10) = 0	ignore
STATUS(10) = 1	apply
STATUS(11)	three phase faults (0 by default)
STATUS(11) = 0	omit
STATUS(11) = 1	include
STATUS(12)	line-line-to-ground faults (0 by default)
STATUS(12) = 0	omit
STATUS(12) = 1	include
STATUS(13)	line-to-line faults (0 by default)
STATUS(13) = 0	omit
STATUS(13) = 1	include

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	STATUS(14)	line charging option (0 by default)
	STATUS(14) = 0	leave unchanged
	STATUS(14) = 1	set to 0.0 in the positive and negative sequences
	STATUS(14) = 2	set to 0.0 in all sequences
	STATUS(15)	line, fixed and switched shunts, and magnetizing admittance option (0 by default)
	STATUS(15) = 0	leave unchanged
	STATUS(15) = 1	set to 0.0 in the positive and negative sequences
	STATUS(15) = 2	set to 0.0 in all sequences
	STATUS(16)	load option (0 by default)
	STATUS(16) = 0	leave unchanged
	STATUS(16) = 1	set to 0.0 in the positive and negative sequences
	STATUS(16) = 2	set to 0.0 in all sequences
	STATUS(17)	synchronous and asynchronous machines power output option (0 by default)
	STATUS(17) = 0	use real and reactive power outputs from power flow solution
	STATUS(17) = 1	set real and reactive power outputs to 0.0
Real VALUES	Is an array of 1 elements (input). The values are as follows.	
	VALUES (1)	Specified pu bus voltage value (Used when STATUS (4) =1 or 2)
Character RELFIL*260	Is the Relay Output Data (.rel) file; blank for none (input; blank by default).	
Character FCDFIL*260	Is the Fault Control Input (.fcd) file; blank for none (input; blank by default).	
Character SCFILE*260	Is the ASCC_3 Results (.sc) file; blank for none (output; blank by default).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output)	
	IERR = 0	no error occurred
	IERR = 1	invalid SID value or subsystem SID is not defined
	IERR = 2	invalid ALL value
	IERR = 3	invalid STATUS value
	IERR = 4	line end fault invalid; bus table is full
	IERR = 5	error opening RELFIL
	IERR = 6	error opening FCDFIL
	IERR = 7	prerequisite requirements for function not met
	IERR = 8	no faults specified
	IERR = 9	error opening SCFILE
	IERR = 10	SCFILE error
	IERR = 11	bus SC results not available
	IERR = 12	branch SC results not available
	IERR = 13	three winding transformer SC results not available
	IERR = 14	incorrect SCFILE
	IERR = 15	SCFILE not allowed

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.38 ASCC_SCFILE

This API reads ASCC short circuit results from SCFILE and produces same ASCC output report that was generated when SCFILE was created. SCFILE is read into PSS®E memory to enable displaying ASCC results onto slider diagrams. For this API to work properly, it is required that working case in PSS®E memory should be the same Saved case that was used when creating the SCFILE.

Batch command syntax:

```
BAT_ASCC_SCFILE SCFILE
```

Python syntax:

```
ierr = ascc_scfile(scfile)
```

Fortran syntax:

```
CALL ASCC_SCFILE(SCFILE,IERR)
```

where:

Character SCFILE*260	Is the ASCC_2 Results (.sc) file (input; no default allowed)
Integer IERR	Is the error code
	IERR = 0, no error occurred
	IERR = 7, prerequisite requirements for function not met
	IERR = 9, error opening SCFILE
	IERR = 10, SCFILE error
	IERR = 11, bus SC results not available.
	IERR = 12, branch SC results not available.
	IERR = 13, three winding transformer SC results not available.
	IERR = 14, incorrect SCFILE.
	IERR = 15, SCFILE not allowed.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.39 BASE_FREQUENCY

Use this API to specify the value of the base frequency option setting (activity OPTN).

Batch command syntax:

```
BAT_BASE_FREQUENCY  RVAL
```

Python syntax:

```
ierr = base_frequency(rval)
```

or:

```
ierr,rval = base_frequency()
```

Fortran syntax:

```
CALL BASE_FREQUENCY_API(RVAL, IOCODE, IERR)
```

where:

Real RVAL	Is the base frequency (usually 50 or 60) (input if Batch command or for Python if specified as an argument, or for FORTRAN if IOCODE is 0; else output 60.0 by default).
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to the value specified in RVAL. IOCODE = 1 return the current value of the option setting in RVAL.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid RVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.40 BGEN

Use this API to convert the mismatch at boundary buses to equivalent load and/or generation (activity BGEN). It is intended to be used following the removal of a subsystem from a solved working case, with the flows to the deleted subsystem replaced by load and/or generation at those retained buses that were directly connected to the removed subsystem.

Batch command syntax:

```
BAT_BGEN  SID  ALL  STATUS
```

Python syntax:

```
ierr = bgen(sid, all, status)
```

Fortran syntax:

```
CALL BGENNW(SID, ALL, STATUS, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all buses.
	ALL = 0	process only buses in subsystem SID.
Integer STATUS	Is the type of boundary condition modification to be used (input; 0 by default).	
	STATUS = 0	generator for inflow; load for outflow.
	STATUS = 1	all equivalent generators.
	STATUS = 2	all equivalent loads.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	invalid STATUS value.
	IERR = 4	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.41 BKDY

Use this API to calculate and report circuit breaker interrupting duty for 3-phase faults at all buses in a specified subsystem of the working case (activity BKDY).

Batch command syntax:

```
BAT_BKDY  SID  ALL  APIOPT  LVLBAK  FLTTIM  BFILE  FFILE
```

Python syntax:

```
ierr = bkdy(sid, all, apiopt, lvlbak, flttim, bfile, ffile)
```

Fortran syntax:

```
CALL BKDYNW(SID, ALL, APIOPT, LVLBAK, FLTTIM, BFILE, FFILE, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all buses.
	ALL = 0	process only buses in subsystem SID.
Integer APIOPT	Is the mode of operation in the API (input; 1 by default).	
	APIOPT = 1	fault applications specified in subsystem.
	APIOPT = 2	fault applications specified in FFILE.
Integer LVLBAK	Is the number of levels back for contributions output (input; 0 by default):	
	LVLBAK < 0	fault currents only.
	LVLBAK = 0	fault currents and local contributions.
	LVLBAK > 0	fault currents and local contributions, plus contributions LVLBAK levels back.
Real FLTTIM	Is the default fault duty time. FLTTIM >= 0.0 (input; 0.0 by default).	
Character BFILE*260	Is the Breaker Duty Data file (input; no default allowed).	
Character FFILE*260	Is the Fault Specification Data file (input; no default allowed if APIOPT is 2).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred
	IERR = 1	invalid SID value or subsystem SID is not defined
	IERR = 2	invalid ALL value
	IERR = 3	invalid APIOPT value
	IERR = 4	invalid FLTTIM value
	IERR = 5	BFILE is blank
	IERR = 6	FFILE is blank
	IERR = 7	generators are not converted
	IERR = 8	fixed bus shunt table is full
	IERR = 9	error opening BFILE or FFILE
	IERR = 10	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.42 BRCH

This API routine is obsolete. It has been replaced by the API routine BRCH_2, and is implemented by a call to the BRCH_2 API routine.

Use this API to tabulate those branches where impedances or other characteristics are such that they may be detrimental to the rate of convergence of one or more of the power flow solution activities (activity BRCH).

Batch command syntax:

```
BAT_BRCH  SID  ALL  DOTEST(1)...DOTEST(10)  TOLVAL(1)...TOLVAL(7)
```

Python syntax:

```
ierr = brch(sid, all, dotest, tolval)
```

Fortran syntax:

```
CALL BRCHNW(SID, ALL, DOTEST, TOLVAL, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 — process all buses. ALL = 0 — process only buses in subsystem SID.
Integer DOTEST(10)	Is an array of ten elements. Each element determines whether a specific branch check should be applied. For DOTEST(I) = 1, the corresponding branch check is applied; for DOTEST(I) = 0, the check is bypassed (input; 0 by default). DOTEST(1) — check for small reactances. DOTEST(2) — check for large reactances. DOTEST(3) — check for large resistance to reactance ratios (R/X). DOTEST(4) — check for negative reactances. DOTEST(5) — check for large ratio of largest to smallest reactance at each bus. DOTEST(6) — check for high or negative charging. DOTEST(7) — check data of parallel transformers. DOTEST(8) — check for high tap ratios. DOTEST(9) — check for low tap ratios. DOTEST(10) — check for zero sequence non-transformer branch impedances of 0.0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real TOLVAL(7) Is an array of seven elements. Each element contains a limit or threshold value used in one of the branch checks (input).

TOLVAL(1) > 0.0	reactance lower limit (0.0005 by default).
TOLVAL(2) > 0.0	reactance upper limit (1.0 by default).
TOLVAL(3) > 0.0	R/X threshold (0.66667 by default).
TOLVAL(4) >= 10.0	Xmax/Xmin threshold (500.0 by default).
TOLVAL(5) >= 0.0	charging upper limit (5.0 by default).
TOLVAL(6) > 0.8	high tap ratio limit (1.1 by default).
TOLVAL(7) > 0.0 and < 1.2	low tap ratio limit (0.9 by default).

Integer IERR Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid SID value or subsystem SID is not defined.
IERR = 2	invalid ALL value.
IERR = 3	invalid DOTEST value.
IERR = 4	invalid TOLVAL value.
IERR = 5	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.43 BRCH_2

Use this API to tabulate those branches where impedances or other characteristics are such that they may be detrimental to the rate of convergence of one or more of the power flow solution activities (activity BRCH).

Batch command syntax:

```
BAT_BRCH_2  SID  ALL  DOTEST(1)...DOTEST(11)  TOLVAL(1)...TOLVAL(7)
```

Python syntax:

```
ierr = brch_2(sid, all, dotest, tolval)
```

Fortran syntax:

```
CALL BRCH_2(SID, ALL, DOTEST, TOLVAL, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Integer DOTEST(11)	Is an array of eleven elements. Each element determines whether a specific branch check should be applied. For DOTEST(I) = 1, the corresponding branch check is applied; for DOTEST(I) = 0, the check is bypassed (input; 0 by default). DOTEST(1) check for small reactances. DOTEST(2) check for large reactances. DOTEST(3) check for large resistance to reactance ratios (R/X). DOTEST(4) check for negative reactances. DOTEST(5) check for large ratio of largest to smallest reactance at each bus. DOTEST(6) check for high or negative charging. DOTEST(7) check data of parallel transformers. DOTEST(8) check for high tap ratios. DOTEST(9) check for low tap ratios. DOTEST(10) check for inconsistent transformer loss data values (copper loss and no-load loss). DOTEST(11) check for zero sequence non-transformer branch impedances of 0.0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real TOLVAL(7) Is an array of seven elements. Each element contains a limit or threshold value used in one of the branch checks (input).

TOLVAL(1) > 0.0	reactance lower limit (0.0005 by default).
TOLVAL(2) > 0.0	reactance upper limit (1.0 by default).
TOLVAL(3) > 0.0	R/X threshold (0.66667 by default).
TOLVAL(4) >= 10.0	Xmax/Xmin threshold (500.0 by default).
TOLVAL(5) >= 0.0	charging upper limit (5.0 by default).
TOLVAL(6) > 0.8	high tap ratio limit (1.1 by default).
TOLVAL(7) > 0.0 and < 1.2	low tap ratio limit (0.9 by default).

Integer IERR Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid SID value or subsystem SID is not defined.
IERR = 2	invalid ALL value.
IERR = 3	invalid DOTEST value.
IERR = 4	invalid TOLVAL value.
IERR = 5	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.44 BSNM

Use this API to change the bus numbers of specified network buses in the working case and retain a tabulation, in file form, of bus number changes made (activity BSNM). This API **must** be referenced with one OPT 0 reference followed by one of the following:

- one OPT 5, 11 or 12 reference.
- one or more OPT 1 references, followed by one OPT 7 reference.
- one or more OPT 2 references, followed by one OPT 7 reference.
- one or more OPT 3 references, followed by one OPT 7 reference.
- one or more OPT 4 references, followed by one OPT 7 reference.
- one OPT 15 reference for each area in the case containing at least one bus, in ascending area number order, followed by one OPT 25 reference.
- zero or more OPT 15 references in the same order as above, followed by one OPT 35 reference.

Batch command syntax:

```
BAT_BSNM  SID  ALL  OPT  STATUS  BUSRNG(1)  BUSRNG(2)  OLDNAM  TFILE
OUT  OFILE
```

Python syntax:

```
ierr = bsnm(sid, all, opt, status, busrng, oldnam, tfile, out, ofile)
```

Fortran syntax:

```
CALL BSNMNW(SID, ALL, OPT, STATUS, BUSRNG, OLDNAM, TFILE, OUT, OFILE,
IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all buses.
	ALL = 0	process only buses in subsystem SID.
Integer OPT	Is the method used to renumber the buses (input; no default allowed).	
	OPT = 0	initialize for renumbering.
	OPT = 1	renumber buses based on name/new number pairs.
	OPT = 11	renumber using name translation file.
	OPT = 2	renumber buses based on old number/new number pairs.
	OPT = 12	renumber buses using number translation file.
	OPT = 3	pack all buses in specified low/high pairs.
	OPT = 4	renumber buses based on the subsystem specified in the OPT 0 reference.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPT = 5	renumber all buses.
	OPT = 15	renumber all buses blocking by area.
	OPT = 25	apply blocking by area changes.
	OPT = 35	cancel blocking by area changes.
	OPT = 7	cleanup from renumbering.
Integer STATUS	STATUS has different values/meanings depending on the value of OPT (input; no default allowed when OPT is 4 or 15).	
	If OPT = 4:	
	STATUS = 0	new bus number range is provided.
	STATUS = 1	bus number offset is provided.
	If OPT = 15:	
	STATUS =	the number of the area where the bus number range is specified in BUSRNG. STATUS must be between 1 and AREAS (the highest area number allowed at the present size level of PSS®E).
Integer BUSRNG(2)	Is an array of two elements. BUSRNG has different values/meanings depending on the value of OPT (input; no defaults allowed except when the item is not used for the specified value of OPT).	
	If OPT = 1:	
	BUSRNG(1)	is the new bus number.
	If OPT = 2:	
	BUSRNG(1)	is the old bus number.
	BUSRNG(2)	is the new bus number.
	If OPT = 3:	
	BUSRNG(1)	is the bus number at which to start packing sequentially.
	BUSRNG(2)	is the bus number at which to end packing sequentially.
	If OPT = 4 and STATUS = 0:	
	BUSRNG(1)	is the starting bus number for the range into which the subsystem is to be renumbered.
	BUSRNG(2)	is the ending bus number for the range into which the subsystem is to be renumbered.
	If OPT = 4 and STATUS = 1:	
	BUSRNG(1)	is amount by which all buses in the specified subsystem are offset.
	If OPT = 5:	
	BUSRNG(1)	bus number at which to start renumbering all buses in the working case. Buses are re-sequenced in increments of 1.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	If OPT = 15: BUSRNG(1) is the starting bus number for the range into which the area is to be renumbered. BUSRNG(2) is the ending bus number for the range into which the area is to be renumbered.
Character OLDNAM*18	Is the extended bus name of the bus to be renumbered when OPT is 1 (input; no default allowed when OPT is 1).
Character TFILE*260	Is the bus name/number translation file. This file is only used if OPT = 11 or 12. Data in the translation file must be in the form specified in the *PSS®E Program Operation Manual*. (input; no default allowed when OPT is 11 or 12)
Integer OUT	Is the output option selection when OPT is 0 (input; 1 by default). OUT = 0 direct output to a data file specified by OFILE. OUT = 1 direct output to the report device. OUT = 2 no output.
Character OFILE*260	Is the Bus Renumbering Translation file for the output records of activity BSNM (input; no default allowed when OPT is 0 and OUT is 0).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 26 prerequisite requirements for API are not met. If OPT = 0: IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 invalid OUT value. IERR = 4 invalid OPT 0 call; bus renumbering already initialized. IERR = 25 error opening OFILE. If OPT = 1 or 2: IERR = 5 prior OPT is not 0. IERR = 6 bus not found. IERR = 7 invalid new bus number. IERR = 8 bus number is already assigned to another bus. If OPT = 11 or 12: IERR = 5 prior OPT is not 0. IERR = 6 bus not found. IERR = 7 invalid new bus number. IERR = 8 bus number is already assigned to another bus. IERR = 24 error opening TFILE.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

If OPT = 3:

IERR = 9	prior OPT not 0 or 3.
IERR = 10	invalid starting bus number.
IERR = 11	invalid ending bus number.
IERR = 12	starting bus number is greater than ending bus number.

If OPT = 4:

IERR = 7	invalid new bus number.
IERR = 8	bus number is already assigned to another bus.
IERR = 9	prior OPT is not 0 or 4.
IERR = 10	invalid starting bus number.
IERR = 11	invalid ending bus number.
IERR = 12	starting bus number is greater than ending bus number.
IERR = 13	invalid STATUS value.
IERR = 14	invalid offset value.
IERR = 15	not enough unused bus numbers in specified bus number range.

If OPT = 5:

IERR = 5	prior OPT is not 0.
IERR = 10	invalid starting bus number.
IERR = 16	not enough numbers between BUSRNG(1) and 999997.

If OPT = 15:

IERR = 9	prior OPT not 0 or 15.
IERR = 10	invalid starting bus number.
IERR = 11	invalid ending bus number.
IERR = 12	starting bus number is greater than ending bus number.
IERR = 17	area specified but no more areas left to process.
IERR = 18	area out of sequence.
IERR = 19	bus number range of an area overlaps that of another area.
IERR = 20	bus number range of an area is not large enough.

If OPT = 25:

IERR = 5	prior OPT not 15.
IERR = 21	not enough OPT 15 calls.

If OPT = 35:

IERR = 9	prior OPT not 15 or 0.

If OPT = 7:

IERR = 22	invalid OPT 7 call; bus renumbering is not initialized.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

If OPT = any other value:
IERR = 23 invalid OPT value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.45 BUS_INPUT

Use this API to set the bus input option setting to either *numbers* or *names* (activity OPTN).

Batch command syntax:

```
BAT_BUS_INPUT  IVAL
```

Python syntax:

```
ierr = bus_input(ival)
```

or:

```
ierr,ival = bus_input()
```

Fortran syntax:

```
CALL BUS_INPUT_API(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Is the value of the option setting (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 1 by default for input).	
	IVAL = 0	bus numbers.
	IVAL = 1	extended bus names.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE = 0	set the option setting to the value specified in IVAL.
	IOCODE = 1	return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid IVAL value.
	IERR = 2	invalid IOCODE value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.46 BUS_OUTPUT

Use this API to set the bus output option setting to either *numbers* or *names* (activity OPTN).

Batch command syntax:

```
BAT_BUS_OUTPUT  IVAL
```

Python syntax:

```
ierr = bus_output(ival)
```

or:

```
ierr,ival = bus_output()
```

Fortran syntax:

```
CALL BUS_OUTPUT_API(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Is the value of the option setting (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 1 by default for input).	
	IVAL = 0	bus numbers.
	IVAL = 1	extended bus names.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE = 0	set the option setting to the value specified in IVAL.
	IOCODE = 1	return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid IVAL value.
	IERR = 2	invalid IOCODE value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.47 BUS_SIZE_LEVEL

Use this API to set the value of the PSS®E size level option setting to a multiple of 1,000 between 1,000 and 150,000 (activity OPTN).

Batch command syntax:

```
BAT_BUS_SIZE_LEVEL  IVAL
```

Python syntax:

```
ierr = bus_size_level(ival)
```

or:

```
ierr,ival = bus_size_level()
```

Fortran syntax:

```
CALL BUS_SIZE_LEVEL_API(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Is the value of the option setting between 1000 and 150000 (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output if IOCODE is 1; 1000 by default for input).
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 — set the option setting to the value specified in IVAL. IOCODE = 1 — return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output). IERR = 0 — no error occurred. IERR = 1 — invalid IVAL value. IERR = 2 — invalid IOCODE value. IERR = 3 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.48 BUSN

Use this API to tabulate unused bus numbers within a specified bus number range (activity BUSN).

Batch command syntax:

```
BAT_BUSN  BUSLO  BUSHI
```

Python syntax:

```
ierr = busn(buslo, bushi)
```

Fortran syntax:

```
CALL BUSNAPI(BUSLO, BUSHI, IERR)
```

where:

Integer BUSLO	Is the low limit of bus number range (input; 1 by default).
Integer BUSHI	Is the high limit of bus number range (input; 999,997 by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error occurred.
	IERR = 1 — invalid starting bus number.
	IERR = 2 — starting bus number is greater than ending bus number.
	IERR = 3 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.49 CASE

Use this API to open a PSS®E Saved Case file and transfers its data into the PSS®E working case (activity CASE).

Batch command syntax:

```
BAT_CASE  SFILE
```

Python syntax:

```
ierr = case(sfile)
```

Fortran syntax:

```
CALL CASENW(SFILE, IERR)
```

where:

Character SFILE*260	Is the name of the PSS®E Saved Case File (input; no default allowed). If SFILE is "*", CASENW attempts to reopen the most recently accessed Saved Case File. If no Saved Case File was accessed in the current execution of PSS®E, CASENW returns an error code.
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 SFILE is blank.
	IERR = 2 error reading from SFILE.
	IERR = 3 error opening SFILE.
	IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.50 CHECK_POWERFLOW_DATA

Use this API routine to perform data checks on the selected categories of powerflow data for all buses in the working case or for all buses in a specified subsystem.

Batch command syntax:

```
BAT_CHECK_POWERFLOW_DATA  SID  ALL  OPT
```

Python syntax:

```
ierr = check_powerflow_data(sid, all, opt)
```

Fortran syntax:

```
CALL CHECK_POWERFLOW_DATA(SID, ALL, OPT, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Integer OPT	Is the powerflow data category to be checked (input; 0 by default). = 0, all data categories = 1, bus data = 2, load data = 3, plant data = 4, generator unit data = 5, induction machine data = 6, fixed bus shunt data = 7, switched shunt data = 8, non-transformer branch data = 9, two-winding transformer data = 10, three-winding transformer data = 11, transformer impedance table correction data = 12, multi-section line data = 13, two-terminal dc line data = 14, multi-terminal dc line data = 15, VSC dc line data = 16, FACTS device data = 17, GNE device data = 18, area interchange data = 19, owner data = 20, zone data
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 invalid OPT value. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.51 CLOSE_POWERFLOW

Removes the current powerflow working case from PSS®E's working memory. Thus, the PSS®E powerflow arrays are in the same state as they are immediately following PSS®E startup. The default PSS®E program option settings are unchanged.

Batch command syntax:

```
BAT_CLOSE_POWERFLOW
```

Python syntax:

```
ierr = close_powerflow()
```

Fortran syntax:

```
CALL CLOSE_POWERFLOW(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.52 CHECK_SEQUENCE_DATA

Use this API routine to perform data checks on the selected categories of sequence data for all buses in the working case or for all buses in a specified subsystem.

Batch command syntax:

```
BAT_CHECK_SEQUENCE_DATA  SID  ALL  OPT
```

Python syntax:

```
ierr = check_sequence_data(sid, all, opt)
```

Fortran syntax:

```
CALL CHECK_SEQUENCE_DATA(SID, ALL, OPT, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Integer OPT	Is the sequence data category to be checked (input; 0 by default). = 0, all data categories = 1, load data = 2, generator unit data = 3, induction machine data = 4, fixed bus shunt data = 5, switched shunt data = 6, non-transformer branch data = 7, two-winding transformer data = 8, three-winding transformer data = 9, zero sequence mutual data
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 invalid OPT value. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.53 CheckVoltageLimits

Use this API routine to tabulate those buses where voltage magnitude is beyond their normal or emergency voltage limits. This check may be performed for all buses in the working case or for all buses in a specified subsystem.

Batch command syntax:

```
BAT_CheckVoltageLimits  SID  ALL  OPT  THRSH
```

Python syntax:

```
ierr = CheckVoltageLimits(sid, all, opt, thrsh)
```

Fortran syntax:

```
CALL CheckVoltageLimits(SID, ALL, OPT, THRSH, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Integer OPT	Is the set of voltate limits to be used (input; 1 by default). = 1, normal limits = 2, emergency limits
Real THRSH	Is the voltage limit threshold (input; -0.05 ≤ THRSH ≤ 0.05; 0.0 by default). 0.0 log voltages above upper limit or below lower limit. > 0.0 log voltages beyond or within THRSH of a limit. < 0.0 log voltage outside a limit by more than (-THRSH).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 invalid OPT value. IERR = 4 invalid THRSH value. IERR = 5 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.54 CLOSE_REPORT

Sets the report output device to the standard output and sets the command line to request device selection for individual reporting activities (Activity CLOS).

Batch command syntax:

```
BAT_CLOSE_REPORT
```

Python syntax:

```
ierr = close_report()
```

Fortran syntax:

```
CALL CLOSE_REPORT_DEVICE(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.55 CMPR

Use this API to tabulate certain case totals, as contained in the working case, with those of a designated Saved Case (activity CMPR). Either system totals, area totals, owner totals, or zone totals may be compared. It **must** be referenced using one of the following procedures:

- One reference with APIOPT=1, followed by any number of references with APIOPT=2, followed by one reference with APIOPT=3.
- One reference with APIOPT=0. This automatically combines the processing of one APIOPT 1 reference, followed by one APIOPT 2 reference, followed by one APIOPT 3 reference.

Batch command syntax:

```
BAT_CMPR  SID  APIOPT  STATUS(1)...STATUS(4)  THRSH  CFILE
```

Python syntax:

```
ierr = cmpr(sid, apiopt, status, thrsh, cfile)
```

Fortran syntax:

```
CALL CMPRNW(SID, APIOPT, STATUS, THRSH, CFILE, IERR)
```

where:

Integer SID	Is a valid area, zone or owner subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; used only when APIOPT is 0 or 1; 0 by default).	
Integer APIOPT	Is the mode of operation in the API (input; 0 by default).	
	APIOPT = 0	initialize for case totals comparison, run the comparison specified by STATUS(3), and run post-processing clean up.
	APIOPT = 1	initialize system for system, area, owner, zone totals.
	APIOPT = 2	calculate and report system, area, owner, zone totals.
	APIOPT = 3	postprocessing housekeeping.
Integer STATUS(4)	Is an array of four elements (input). The value of each element is as follows.	
	STATUS(1)	is the tabulation method (used only when APIOPT is 0 or 1; 0 by default).
	STATUS(1) = 0	tabulate system totals.
	STATUS(1) = 1	tabulate area totals for area subsystem SID.
	STATUS(1) = 2	tabulate owner totals for owner subsystem SID.
	STATUS(1) = 3	tabulate zone totals for zone subsystem SID.
	STATUS(2)	is the units for the difference threshold (used only when APIOPT is 0 or 2; 0 by default).
	STATUS(2) = 0	threshold in engineering units.
	STATUS(2) = 1	threshold in percent.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

STATUS(3)	is the quantity to be compared (used only when APIOPT is 0 or 2; 1 by default).
STATUS(3) = 1	compare generation.
STATUS(3) = 2	compare load.
STATUS(3) = 3	compare losses.
STATUS(3) = 4	compare mismatch.
STATUS(3) = 5	compare interchange; not valid for STATUS(1) = 2.
STATUS(4)	is the file to restore into the working case upon completion (used only when APIOPT is 0 or 3; 0 by default).
STATUS(4) = 0	restore the working case.
STATUS(4) = 1	restore the comparison file to the working case.

Real THRSH — Is the difference threshold, in either engineering units or a percent, based on the value of STATUS(2) (input; used only when APIOPT is 0 or 2; 0.0 by default).

Character CFILE*260 — Is the Saved Case file to be compared against the working case (input; no default allowed when APIOPT is 0 or 1).

Integer IERR — Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid SID value or subsystem SID is not defined.
IERR = 2	invalid APIOPT value.
IERR = 3	invalid STATUS value.
IERR = 4	owner (STATUS(1)=2) interchange (STATUS(3)=5) combination is invalid.
IERR = 5	invalid THRSH value.
IERR = 6	CFILE is blank.
IERR = 7	error writing the working case to a temporary file.
IERR = 8	unexpected APIOPT value.
IERR = 9	error reading the comparison case.
IERR = 10	error opening the temporary file.
IERR = 11	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.56 CNTB

Use this API to tabulate the voltage setpoints and desired voltage bands of voltage-controlling equipment in the working case (activity CNTB). Optionally, new scheduled voltages may be specified.

Batch command syntax:

```
BAT_CNTB  SID  ALL  APIOPT  STATUS(1)...STATUS(3)  V(1)  V(2)
```

Python syntax:

```
ierr,next,bus,ckt,v = cntb(sid, all, apiopt, status, v)
```

Fortran syntax:

```
CALL CNTBNW(SID, ALL, APIOPT, STATUS, NEXT, BUS, CKT, V, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; used only when APIOPT is 1; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; used only when APIOPT is 1; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Integer APIOPT	Is the mode of operation in the API (input; 1 by default). APIOPT = 1 initialize. APIOPT = 2 specify voltage setpoint for controlled bus. APIOPT = 3 specify voltage band for load drop compensating transformer.
Integer STATUS(3)	Is an array of three elements (input). The interpretation of its values is dependent on the value specified as APIOPT, and is described below with the descriptions of the three values of APIOPT.
Integer NEXT	Is **returned** as the next APIOPT value expected by the API. NEXT = 1 done with this execution of CNTBNW. NEXT = 2 change voltage setpoint/schedule. BUS(1) **returned** as controlled bus for which voltage schedule may be changed. V(1) **returned** as present voltage setpoint/schedule.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	NEXT = 3	change VMIN, VMAX of load drop compensating transformer. BUS(1) **returned** as controlled bus for which voltage schedule may be changed. BUS(2) **returned** as from bus of the transformer. BUS(3) **returned** as to bus of the transformer. V(1) **returned** as present VMIN. V(2) **returned** as present VMAX.
Integer BUS(3)	Is a **returned** array of three elements described with NEXT (output).	
Character CKT*2	Is **returned** as the circuit ID of the transformer **returned** in BUS(2) and BUS(3) when NEXT **returned** as 3.	
Real V(2)	Is an array of two elements that is used both to supply input to the API and to pass data values from the API to the calling routine. Its role in passing input to the API is described below with the descriptions of the three values of APIOPT. Its content following a call of the API is dependent on the value specified for NEXT.	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 6	invalid APIOPT value.
	IERR = 8	prerequisite requirements for API are not met.

When APIOPT = 1, initialize.

Integer STATUS(3)	Is an array of three elements (input; 0 by default). They are used to control reporting and processing options.	
	STATUS(1) = 0	list subsystem controlled buses with conflicts.
	STATUS(1) = 1	list all subsystem controlled buses.
	STATUS(2) = 0	report at OPENed output device.
	STATUS(2) = 1	list in progress area and allow voltage schedule change.
	STATUS(3) = 0	process all controlling elements and controlled buses.
	STATUS(3) = 1	process active controlling elements and controlled buses.
Real V(2)	Is an array of two elements.	
	V(1)	VMIN, not used except for ECHO.
	V(2)	VMAX, not used except for ECHO.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	invalid STATUS value.
	IERR = 7	unexpected APIOPT value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

When APIOPT = 2, specify voltage setpoint.

Integer STATUS(3) Is an array of three elements (input; 0 by default). They are used to control reporting and processing options.

STATUS(1) = 0	no change.
STATUS(1) = 1	change voltage setpoints to new setpoint in V(1).
STATUS(2)	not used except for ECHO.
STATUS(3)	not used except for ECHO.

Real V(2) Is an array of two elements (input).

V(1)	new setpoint (present voltage setpoint of one of the setpoint mode voltage controlling devices controlling the voltage of this bus by default).
V(2)	VMAX not used except for ECHO.

Integer IERR Is the error code (output).

IERR = 0	no error occurred.
IERR = 3	invalid STATUS value.
IERR = 4	invalid new voltage setpoint value (V(1) <= 0.5 or V(1) >= 1.5).
IERR = 7	unexpected APIOPT value.

When APIOPT = 3, specify voltage band for load drop compensating transformer.

Integer STATUS(3) Is an array of three elements (input; 0 by default). They are used to control reporting and processing options.

STATUS(1) = 0	no change.
STATUS(1) = 1	change voltage band to new voltage band in V(1) and V(2).
STATUS(2)	not used except for ECHO.
STATUS(3)	not used except for ECHO.

Real V(2) Is an array of two elements (input).

V(1)	new VMIN (present VMIN by default).
V(2	new VMAX (present VMAX by default).

Integer IERR Is the error code (output).

IERR = 0	no error occurred.
IERR = 3	invalid STATUS value.
IERR = 5	VMIN is not less than VMAX.
IERR = 7	unexpected APIOPT value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.57 CONG

Use this API to convert generators from their power flow representation in preparation for switching studies and dynamic simulations (activity CONG).

Batch command syntax:

```
BAT_CONG  OPT
```

Python syntax:

```
ierr = cong(opt)
```

Fortran syntax:

```
CALL CONGAPI(OPT, IERR)
```

where:

Integer OPT	Is the machine impedance option (input; 0 by default).	
	OPT = 0	use ZSORCE and IMZSRC.
	OPT = 1	use fault analysis subtransient impedance and IMZPOS.
	OPT = 2	use fault analysis transient impedance and IMZPOS.
	OPT = 3	use fault analysis synchronous impedance and IMZPOS.
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid OPT value.
	IERR = 2	generators are already converted.
	IERR = 3	generator conversion failed due to a machine impedance of zero.
	IERR = 4	generator conversion failed due to the presence of in-service induction machines that are in the "stalled" or "tripped" state.
	IERR = 5	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.58 CONL

Use this API to convert the constant MVA load for a specified grouping of network loads to a specified mixture of the constant MVA, constant current, and constant admittance load characteristics (activity CONL).

Batch command syntax:

```
BAT_CONL  SID  ALL  APIOPT  STATUS(1)  STATUS(2)  LOADIN(1)...
LOADIN(2)  LOADIN(3)  LOADIN(4)
```

Python syntax:

```
ierr, rlods = conl(sid, all, apiopt, status, loadin)
```

Fortran syntax:

```
CALL CONLNW(SID, ALL, APIOPT, STATUS, LOADIN, RLODS, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; used only when APIOPT is 2; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; used only when APIOPT is 2; 0 by default).	
	ALL = 1	process all buses.
	ALL = 0	process only buses in subsystem SID.
Integer APIOPT	Is the mode of operation in the API (input; no default allowed).	
	APIOPT = 1	initialize for load conversion.
	APIOPT = 2	convert loads.
	APIOPT = 3	postprocessing housekeeping.
Integer STATUS(2)	Is an array of two elements (input). The value of each element is as follows.	
	STATUS(1)	is the method of conversion (used only when APIOPT is 1; 0 by default).
	STATUS(1) = 0	convert existing constant MVA load.
	STATUS(1) = 1	reconstruct loads using present voltage, then convert.
	STATUS(1) = 2	reconstruct loads using unity voltage, then convert.
	STATUS(2)	is a special OPF bus type code flag (used only when APIOPT is 2; 0 by default).
	STATUS(2) = 0	all buses in subsystem.
	STATUS(2) = 1	only Type 1 buses in subsystem.
	STATUS(2) = 2	only Type 2 and 3 buses in subsystem.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real LOADIN(4) Is an array of four elements (input; used only when APIOPT is 2; 0.0 by default). The value of each element is percent of the load being converted in this step to be represented by the corresponding load characteristic as follows.

LOADIN(1)	is the percent of active power load to be converted to the constant current characteristic.
LOADIN(2)	is the percent of active power load to be converted to the constant admittance characteristic.
LOADIN(3)	is the percent of reactive power load to be converted to the constant current characteristic.
LOADIN(4)	is the percent of reactive power load to be converted to the constant admittance characteristic.

Integer RLODS Is **returned** as the number of unconverted loads remaining in the working case. If the value is nonzero after a call with APIOPT = 2, and the remaining loads are to be converted, the API is called again with ALL = 1 and APIOPT = 2.

Integer IERR Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid SID value or subsystem SID is not defined.
IERR = 2	invalid ALL value.
IERR = 3	invalid APIOPT value.
IERR = 4	invalid STATUS value.
IERR = 5	unexpected APIOPT value.
IERR = 6	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.59 CONNECTIVITY_CHECK

Use this API to enable or disable the solution connectivity checking option setting (activity OPTN).

Batch command syntax:

```
BAT_CONNECTIVITY_CHECK  IVAL
```

Python syntax:

```
ierr = connectivity_check(ival)
```

or:

```
ierr,ival = connectivity_check()
```

Fortran syntax:

```
CALL CONNECTIVITY_CHECK_API(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Is the value of the option setting (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 1 by default for input).	
	IVAL = 0	no solution connectivity check before power flow solutions.
	IVAL = 1	run solution connectivity check before each power flow solution.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE = 0	set the option setting to the value specified in IVAL.
	IOCODE = 1	return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid IVAL value.
	IERR = 2	invalid IOCODE value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.60 CONTROL_AREA_INTERCHANGE

Use this API to set the area interchange control option setting to either *disabled*, *enabled with tie lines only* or *enabled with tie lines and loads* (activity OPTN).

Batch command syntax:

```
BAT_CONTROL_AREA_INTERCHANGE  IVAL
```

Python syntax:

```
ierr = control_area_interchange(ival)
```

or:

```
ierr,ival = control_area_interchange()
```

Fortran syntax:

```
CALL CONTROL_AREA_INTERCHANGE_API(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Is the value of the option setting (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 1 by default for input).	
	IVAL = 0	disabled.
	IVAL = 1	enabled with interchange determined from tie lines only.
	IVAL = 2	enabled with interchange determined from tie lines and loads.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE = 0	set the option setting to the value specified in IVAL.
	IOCODE = 1	return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid IVAL value.
	IERR = 2	invalid IOCODE value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.61 DC_TAP_ADJUSTMENT

Use this API to enable or disable the dc tap adjustment option setting (activity OPTN).

Batch command syntax:

```
BAT_DC_TAP_ADJUSTMENT  IVAL
```

Python syntax:

```
ierr = dc_tap_adjustment(ival)
```

or:

```
ierr,ival = dc_tap_adjustment()
```

Fortran syntax:

```
CALL DC_TAP_ADJUSTMENT_API(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Is the value of the option setting (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output if 0 by default for input).	
	IVAL = 0	disabled.
	IVAL = 1	enabled.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE = 0	set the option setting to the value specified in IVAL.
	IOCODE = 1	return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid IVAL value.
	IERR = 2	invalid IOCODE value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.62 DCCC

Use this API to run contingency case solutions using a linear network (dc) model (activity DCCC).

Batch command syntax:

```
BAT_DCCC  OPTIONS(1)...OPTIONS(6)  VALUES(1)...VALUES(3)  DFXFILE
```

Python syntax:

```
ierr = dccc(options, values, dfxfile)
```

Fortran syntax:

```
CALL DCCCAPI(OPTIONS, VALUES, DFXFILE, IERR)
```

where:

Integer OPTIONS(6)	Is an array of six elements specifying calculation and reporting options (input). The value of each element is as follows.	
	OPTIONS(1)	contingency case rating set (rating set program option setting by default).
	OPTIONS(1) = 1	RATEA.
	OPTIONS(1) = 2	RATEB.
	OPTIONS(1) = 3	RATEC.
	OPTIONS(2)	base case line flow code (0 by default).
	OPTIONS(2) = 0	dc base case.
	OPTIONS(2) = 1	ac base case.
	OPTIONS(3)	output format code (0 by default).
	OPTIONS(3) = 0	overload output.
	OPTIONS(3) = 1	full output.
	OPTIONS(4)	exclude cases with no overloads from the overload report (0 by default).
	OPTIONS(4) = 0	no.
	OPTIONS(4) = 1	yes.
	OPTIONS(5)	convert MVA ratings to estimated MW ratings (0 by default).
	OPTIONS(5) = 0	no.
	OPTIONS(5) = 1	yes.
	OPTIONS(6)	code to discontinue activity if MW mismatch exceeds TOLS(1) (0 by default).
	OPTIONS(6) = 0	discontinue.
	OPTIONS(6) = 1	continue.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real VALUES(3)	Is an array of three elements specifying tolerances and thresholds (input). The value of each element is as follows.	
	VALUES(1)	required MW mismatch tolerance for continuing (Newton solution convergence tolerance, TOLN, by default).
	VALUES(2)	percent of rating for reporting an element as overloaded (100.0 by default).
	VALUES(3)	minimum contingency case flow change for including an element in the overload report (0.0 by default).
Character DFXFILE*260	Is the name of Distribution Factor Data File (input; no default allowed).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid OPTIONS value.
	IERR = 2	invalid VALUES value.
	IERR = 3	no Distribution Factor Data File specified.
	IERR = 4	generators are converted.
	IERR = 5	buses in island(s) without a swing bus; use activity TREE.
	IERR = 6	largest mismatch exceeds mismatch tolerance.
	IERR = 7	error opening file DFXFILE.
	IERR = 8	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
	IERR = 9	file DFXFILE does not contain distribution factors; run DFAX with distribution factor option enabled.
	IERR = 10	no monitored branches or interfaces defined.
	IERR = 11	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.63 DCCOR

Use this API to apply corrective actions to the base case using linear programming methods. The corrective actions are to eliminate flow overloads with the objective of minimizing the control adjustments, including load shedding, MW generations dispatch, and phase shifter angle adjustments.

Batch command syntax:

```
BAT_DCCOR  OPTIONS(1)...OPTIONS(7)  VALUES(1)...VALUES(6)
LABELS(1)...LABELS(3)  DFXFILE
```

Python syntax:

```
ierr = dccor(options, values, labels, dfxfile)
```

Fortran syntax:

```
CALL DCCORAPI(OPTIONS, VALUES, LABELS, DFXFILE, IERR)
```

where:

Integer OPTIONS(7)	Is an array of seven elements specifying DC corrective actions solution options (input). The values are as follows:	
	OPTIONS(1)	rating set (rating set program option setting by default).
	OPTIONS(1) = 1	rating A
	OPTIONS(1) = 2	rating B
	OPTIONS(2) = 3	rating C
	OPTIONS(2)	generator control flag (1 by default).
	OPTIONS(2) = 0	disable.
	OPTIONS(2) = 1	enable.
	OPTIONS(3)	load control flag (0 by default).
	OPTIONS(3) = 0	disable.
	OPTIONS(3) = 1	enable.
	OPTIONS(4)	phase shifter control flag (1 by default).
	OPTIONS(4) = 0	disable.
	OPTIONS(4) = 1	enable.
	OPTIONS(5)	convert MVA ratings to estimated MW ratings (0 by default).
	OPTIONS(5) = 0	disable.
	OPTIONS(5) = 1	enable.
	OPTIONS(6)	code to discontinue activity if MW mismatch exceeds tolerance (VALUES(1) (0 by default).
	OPTIONS(6) = 0	disable (i.e. continue, do not discontinue activity).
	OPTIONS(6) = 1	enable (i.e. discontinue).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTIONS(7)	modify network data with corrective actions solutions (1 by default).
	OPTIONS(7) = 0	disable (i.e. do not modify network data).
	OPTIONS(7) = 1	enable (i.e. allow network data to be modified).
Real VALUES(6)	Is an array of six elements specifying DC corrective actions parameters (input). The values are as follows:	
	VALUES(1)	MW mismatch tolerance for continuing (Newton solution convergence tolerance, TOLN, by default).
	VALUES(2)	percent of rating set (100.0 by default).
	VALUES(3)	branch flow overload tolerance (0.1 by default).
	VALUES(4)	generator control weighting factor (1.0 by default).
	VALUES(5)	load control weighting factor (1.0 by default).
	VALUES(6)	phase shifter control weighting factor (1.0 by default).
Character LABELS(3)*12	Is an array of three elements specifying subsystem labels (input). The values are as follows:	
	LABELS(1)	label of generator control subsystem (blank by default; no default allowed if OPTIONS(2) is 1).
	LABELS(2)	label of load control subsystem (blank by default; no default allowed if OPTIONS(3) is 1).
	LABELS(3)	label of phase shifter control subsystem (blank by default; no default allowed if OPTIONS(4) is 1).
Character DFXFILE*260	Is the name of the Distribution Factor Data File (input; no default allowed).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid OPTIONS value.
IERR = 2	invalid VALUES value.
IERR = 3	generators are converted.
IERR = 4	buses in island(s) without a swing bus; use activity TREE.
IERR = 5	No controls.
IERR = 6	control subsystem not defined.
IERR = 7	Distribution Factor Data File not specified.
IERR = 8	largest mismatch exceeds tolerance.
IERR = 21	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
IERR = 22	monitored elements exceed limit when adding multi-section line members.
IERR = 51	error opening Distribution Factor Data File.
IERR = 52	prerequisite requirements for function not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.64 DCLF

Use this API to apply the dc analogy network solution algorithm to the network modeled in the working case (activity DCLF).

Batch command syntax:

```
BAT_DCLF  SID  ALL  STATUS(1)...STATUS(4)  BRNCH(1)...BRNCH(3) CKT
```

Python syntax:

```
ierr = dclf(sid, all, status, brnch, ckt)
```

Fortran syntax:

```
CALL DCLFAPI(SID, ALL, STATUS, BRNCH, CKT, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all buses.
	ALL = 0	process only buses in subsystem SID.
Integer STATUS(4)	Is an array of four elements (input). The value of each element is as follows.	
	STATUS(1)	is rating used in tabulating the report (default rating set program option setting by default).
	STATUS(1) = 1	RATEA.
	STATUS(1) = 2	RATEB.
	STATUS(1) = 3	RATEC.
	STATUS(2)	is the voltage that should be saved (0 by default).
	STATUS(2) = 0	original voltages.
	STATUS(2) = 1	base case dc power flow voltages.
	STATUS(2) = 2	change case dc power flow voltages.
	STATUS(3)	is status of the changed branch upon completion (0 by default).
	STATUS(3) = 0	leave unchanged.
	STATUS(3) = 1	save changed status.
	STATUS(4)	is action code if the MW mismatch exceeds 0.5 MW (1 by default).
	STATUS(4) = 0	discontinue.
	STATUS(4) = 1	continue.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer BRNCH(3) Is an array of three elements (input). The use of each element is as follows. The values can all be zero or defaulted if no branch is to be changed for the solution.

BRNCH(1)	is the from bus of the branch for which status is to be changed for the solution (0 by default).
BRNCH(2)	is the to bus of the branch for which status is to be changed for the solution (0 by default).
BRNCH(3)	is the third bus connected by a three-winding transformer, or zero for a nontransformer branch or a two-winding transformer (0 by default).

Character CKT*2 Is the circuit ID of the branch for which status is to be changed for the solution (input; ’1’ by default).

Integer IERR Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid SID value or subsystem SID is not defined.
IERR = 2	invalid ALL value.
IERR = 3	invalid STATUS value.
IERR = 4	bus not found.
IERR = 5	branch not found.
IERR = 6	three-winding transformer not found.
IERR = 7	multi-section line specified but multi-section line reporting option is disabled.
IERR = 8	closing a multi-section line is invalid in DCLF.
IERR = 9	returning a Type 4 bus to service is not allowed in DCLF.
IERR = 10	closing a three-winding transformer is not allowed in DCLF.
IERR = 11	generators are converted.
IERR = 12	buses in island(s) without a swing bus; use activity TREE.
IERR = 13	largest mismatch exceeds 0.5 MW.
IERR = 14	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.65 DFAX

Use this API to construct a Distribution Factor Data File (activity DFAX).

Batch command syntax:

```
BAT_DFAX  OPTIONS(1)  OPTIONS(2)  SUBFILE  MONFILE  CONFILE  DFXFILE
```

Python syntax:

```
ierr = dfax(options, subfile, monfile, confile, dfxfile)
```

Fortran syntax:

```
CALL DFAXAPI(OPTIONS, SUBFILE, MONFILE, CONFILE, DFXFILE, IERR)
```

where:

Integer OPTIONS(2)	Is an array of two elements specifying calculation options (input). The value of each element is as follows. OPTIONS(1) — distribution factor option flag (1 by default). OPTIONS(1) = 0 — do not calculate distribution factors (i.e., DFAX,AC). OPTIONS(1) = 1 — calculate distribution factors. OPTIONS(2) — monitored element sorting flag (0 by default). OPTIONS(2) = 0 — do not sort (i.e., leave in Monitored Element Description File order). OPTIONS(2) = 1 — sort.
Character SUBFILE*260	Is the name of the Subsystem Description File; blank for none (input; blank by default).
Character MONFILE*260	Is the name of Monitored Element Description File (input; no default allowed).
Character CONFILE*260	Is the name of Contingency Description Data File (input; no default allowed).
Character DFXFILE*260	Is the name of Distribution Factor Data File (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR

Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid OPTIONS value.
IERR = 2	generators are converted.
IERR = 3	buses in island(s) without a swing bus; use activity TREE.
IERR = 4	no Distribution Factor Data File specified.
IERR = 5	no Monitored Element Data input file specified.
IERR = 6	no Contingency Description Data file specified.
IERR = 7	fatal error reading input file.
IERR = 8	error opening output file DFXFILE.
IERR = 9	error opening input file SUBFILE.
IERR = 10	error opening input file MONFILE.
IERR = 11	error opening input file CONFILE.
IERR = 12	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.66 DFAX_CONTINGENCY

Use this API to combine one contingency in the first Distribution Factor file with one contingency in the second Distribution Factor file and so on, till specified contingency level is reached, to create multiple event contingencies and then export resulting contingencies to user defined report device.

Python syntax:

```
ierr = dfax_contingency(option, confile, dfxfile)
```

Fortran syntax:

```
CALL DFAX_CONTINGENCY(OPTION, CONFILE, DFXFILE, IERR)
```

where:

Integer OPTION	Is the level of multiple contingencies (from 1 to 3; 1 by default)
Character CONFILE*260	Is the name of Contingency Description Data File (input; blank by default).
Character DFXFILE(OPTION)*260	Is an array of OPTION names of Distribution Factor Data Files where contingencies are combined each other to build OPTION level contingencies (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 invalid OPTION value.
	IERR = 2 Input file is not a Distribution Factor Data File
	IERR = 3 error reading Distribution Factor Data File
	IERR = 51 error opening Contingency Description Data File
	IERR = 52 error opening Distribution Factor Data File

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.67 DFTI

Use this API to compare tie lines, as contained in the working case, with those of a designated Saved Case (activity DFTI). It **must** be referenced using one of the following procedures:

- One reference with APIOPT=1, followed by any number of references with APIOPT=2, followed by one reference with APIOPT=3.
- One reference with APIOPT=0. This automatically combines the processing of one APIOPT 1 reference, followed by one APIOPT 2 reference, followed by one APIOPT 3 reference.

Batch command syntax:

```
BAT_DFTI  SID  APIOPT  STATUS(1)...STATUS(5)  THRSH  CFILE
```

Python syntax:

```
ierr,seqflg = dfti(sid, apiopt, status, thrsh, cfile)
```

Fortran syntax:

```
CALL DFTIAPI(SID, APIOPT, STATUS, THRSH, CFILE, SEQFLG, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; used only when APIOPT is 0 or 1; 0 by default).	
Integer APIOPT	Is the mode of operation of DFTIAPI (input; 0 by default).	
	APIOPT = 0	initialize for case comparison, run the comparison specified by STATUS(4), and run post-processing clean up.
	APIOPT = 1	initialize for case comparison.
	APIOPT = 2	run a comparison.
	APIOPT = 3	post-processing clean up.
Integer STATUS(5)	Is an array of five elements specifying comparison options (input). The values are as follows:	
	STATUS(1)	defines the working case content following an APIOPT=0 or 3 reference (used only when APIOPT is 0, 1 or 3; 0 by default).
	STATUS(1) = 0	original working case.
	STATUS(1) = 1	Saved Case specified in CFILE.
	STATUS(2)	flag for method of considering buses in the two cases to be the same bus (input; used only when APIOPT is 0 or 1; 0 by default).
	STATUS(2) = 0	same bus numbers.
	STATUS(2) = 1	same extended bus names.
	STATUS(2) = 2	same bus numbers and extended bus names.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	STATUS(3)	flag for specifying THRESH (used only when APIOPT is 0 or 2; 0 by default).
	STATUS(3) = 0	engineering units.
	STATUS(3) = 1	percent.
	STATUS(4)	code for the comparison to be run (used only when APIOPT is 0 or 2; 1 by default):

= 1, branch status	= 2, line R, X, B
= 3, line shunts	= 4, line ratings
= 5, metered end	= 6, line length
= 7, transformers	= 8, flow MW or MVAR (from bus)
= 9, flow MW or MVAR (from & to)	= 10, line MW or MVAR losses
= 11, zero sequence R, X, B	= 12, zero sequence line shunts
= 13, transformer connection codes	= 14, flow MW (from bus)
= 15, flow MVAR (from bus)	= 16, flow MW (from and to buses)
= 17, flow MVAR (from and to buses)	= 18, line MW losses
= 19, line MVAR losses	

	STATUS(5)	line rating to be compared when STATUS(4) is 4 (used only when APIOPT is 0 or 2; 0 by default):
	= 0, all ratings	= 1, RATEA
	= 2, RATEB	= 3, RATEC
Real THRSH	Is the difference threshold (input; used only when APIOPT is 0 or 2; 0.0 by default).	
Character CFILE*260	Is the name of Saved Case File in which tie lines are to be compared to those of the working case (input; no default allowed when APIOPT is 0 or 1).	
Logical SEQFLG(2)	Is an array of two elements **returned** when APIOPT=0 or 1. The value of each is as follows:	
	SEQFLG(1)	is true if there is sequence data in the working case.
	SEQFLG(2)	is true if there is sequence data in the Saved Case in file CFILE.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid STATUS value.
	IERR = 3	CFILE is blank.
	IERR = 4	no ties to compare.
	IERR = 5	invalid APIOPT value.
	IERR = 6	unexpected APIOPT value.
	IERR = 7	invalid THRSH value.
	IERR = 8	error reading the comparison case.
	IERR = 9	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.68 DIFF

Use this API to compare specified power flow data and solution results, as contained in the working case, with those of a designated Saved Case (activity DIFF). It **must** be referenced first with APIOPT=1, followed by any number of references with APIOPT=2, followed by one reference with APIOPT=3:

Batch command syntax:

```
BAT_DIFF  SID  ALL  APIOPT  STATUS(1)...STATUS(4)
THRSH(1)...THRSH(3)  CFILE
```

Python syntax:

```
ierr,seqflg = diff(sid, all, apiopt, status, thrsh, cfile)
```

Fortran syntax:

```
CALL DIFFNW(SID, ALL, APIOPT, STATUS, THRSH, CFILE, SEQFLG, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; used only when APIOPT is 1; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; used only when APIOPT is 1; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Integer APIOPT	Is the mode of operation in the API (input; no default allowed). APIOPT = 1 initialize for case comparison. APIOPT = 2 run case comparison for specified data category. APIOPT = 3 postprocessing housekeeping.
Integer STATUS(4)	Is an array of four elements (input). STATUS has different values/meanings depending on the value of APIOPT. If APIOPT = 1, STATUS(1) is the case to leave in the working case (0 by default). STATUS(1) = 0 original working case. STATUS(1) = 1 saved case specified in CFILE. If APIOPT = 1, STATUS(2) is the method used to consider two buses the same for comparison (0 by default). STATUS(2) = 0 bus numbers must be the same. STATUS(2) = 1 bus names must be the same. STATUS(2) = 2 bus names and numbers must be the same. If APIOPT = 2, STATUS(1) is the method used for specifying difference thresholds (0 by default). STATUS(1) = 0 threshold in engineering units. STATUS(1) = 1 threshold as a percent.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

If APIOPT = 2, STATUS(2) is the comparison to be run (1 by default).

= 1, bus identifiers	= 2, bus type codes
= 3, machine status	= 4, generator MW
= 5, generator MW or MVAR	= 6, bus loads
= 7, bus shunts	= 8, switched shunts
= 9, voltage	= 10, voltage and angle
= 11, Mbase and Zsorce	= 12, Mbase, Rpos and X"
= 13, Mbase and Zneg	= 14, Mbase and Zzero
= 15, negative sequence loads	= 16, zero sequence loads
= 17, branch status	= 18, line R, X, B
= 19, line shunts	= 20, line ratings
= 21, metered end	= 22, transformers
= 23, flow MW or MVAR (from bus)	= 24, flow MW or MVAR (from & to)
= 25, line MW or MVAR losses	= 26, zero sequence R, X, B
= 27, zero sequence line shunts	= 28, connection codes
= 29, zero sequence mutuals	= 30, multi-section lines
= 31, multi-section metered end	= 32, load status
= 33, line lengths	= 34, generator MVAR
= 35, flow MW (from bus)	= 36, flow MVAR (from bus)
= 37, flow MW (from and to)	= 38, flow MVAR (from and to)
= 39, line MW losses	= 40, line MVAR losses
= 41, fixed bus shunt status	= 42, switched shunt status
= 43, scalable load flag	= 44, Mbase, Rpos and X’
= 45, Mbase, Rpos and Xs	

If APIOPT = 2, STATUS(3) is the load characteristic to be compared (0 by default).

STATUS(3) = 0	total nominal load.
STATUS(3) = 1	constant MVA.
STATUS(3) = 2	constant current.
STATUS(3) = 3	constant admittance.

If APIOPT = 2, STATUS(4) is the line rating set to be compared (0 by default).

STATUS(4) = 0	all ratings.
STATUS(4) = 1	RATEA.
STATUS(4) = 2	RATEB.
STATUS(4) = 3	RATEC.

If APIOPT = 3, STATUS(1) is the case to leave in the working case (0 by default).

STATUS(1) = 0	original working case.
STATUS(1) = 1	saved case specified in CFILE.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real THRSH(3)	Is an array of three elements (input; used only when APIOPT is 2). The value of each is as follows.	
	THRSH(1)	is the difference threshold (0.0 by default).
	THRSH(2)	is the voltage or tap ratio threshold (0.0 by default).
	THRSH(3)	is the angle threshold (0.0 by default).
Character CFILE*260	Is the saved case to be compared against the data currently in the working case (input; no default allowed when APIOPT is 1).	
Logical SEQFLG(2)	Is an array of two elements **returned** when APIOPT = 1. The value of each is as follows:	
	SEQFLG(1)	is true if there is sequence data in the working case.
	SEQFLG(2)	is true if there is sequence data in the Saved Case in file CFILE.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	invalid APIOPT value.
	IERR = 4	invalid STATUS value.
	IERR = 5	invalid THRSH value.
	IERR = 6	CFILE is blank.
	IERR = 7	unexpected APIOPT value.
	IERR = 8	error building the case to case translation files.
	IERR = 9	error reading the comparison case.
	IERR = 10	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.69 DSCN

Use this API to electrically disconnect a bus (activity DSCN).

Batch command syntax:

```
BAT_DSCN  BUS
```

Python syntax:

```
ierr = dscn(bus)
```

Fortran syntax:

```
CALL DSCNAPI(BUS, IERR)
```

where:

Integer BUS	Is the number of the bus to be disconnected (input; no default allowed).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	bus BUS not found.
	IERR = 2	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.70 ECDI

Use this API to place machines in a specified subsystem on- or off-line to satisfy a given subsystem minimum capacity; the in-service machines in the subsystem are then dispatched on the basis of incremental cost to meet a specified total subsystem generation (activity ECDI).

This API **must** be referenced using one of the following procedures:

- One reference with APIOPT=1, followed by one reference with APIOPT=2, followed by one or more references with APIOPT=3, followed by one reference with APIOPT=4.
- One reference with APIOPT=0. This automatically combines the processing of one APIOPT 1 reference, followed by one APIOPT 2 reference, followed by one APIOPT 3 reference, followed by one APIOPT 4 reference.

Batch command syntax:

```
BAT_ECDI  SID  ALL  APIOPT  CFILE  STATUS  VALUES(1)  VALUES(2)
```

Python syntax:

```
ierr = ecdi(sid, all, apiopt, cfile, status, values)
```

Fortran syntax:

```
CALL ECDIAPI(SID, ALL, APIOPT, CFILE, STATUS, VALUES, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; used only when APIOPT is 0 or 2; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; used only when APIOPT is 0 or 2; 1 by default).	
	ALL = 1	process all buses.
	ALL = 0	process only buses in subsystem SID.
Integer APIOPT	Is the mode of operation of the API (input; 0 by default).	
	APIOPT = 0	initialize, specify subsystem, process, and clean up.
	APIOPT = 1	initialize for dispatch.
	APIOPT = 2	specify subsystem.
	APIOPT = 3	apply calculation.
	APIOPT = 4	postprocessing housekeeping.
Character CFILE*260	Is the name of the Economic Dispatch Data File (input; no default allowed when APIOPT is 0 or 1).	
Integer STATUS	Is the unit commitment option (input; used only when APIOPT is 0 or 2; 0 by default).	
	STATUS = 0	start from current commitment profile.
	STATUS = 1	new commitment profile (start with all units off-line).
	STATUS = 2	unit commitment disabled.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real VALUES(2)	Is an array of two elements (input; used only when APIOPT is 0 or 3).	
	VALUES(1)	desired loading (present loading by default).
	VALUES(2)	desired minimum capacity of units being dispatched (0 by default).
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid APIOPT value.
	IERR = 3	invalid ALL value.
	IERR = 4	invalid STATUS value.
	IERR = 5	dispatch unable to meet desired power and capacity.
	IERR = 6	no dispatch data for any subsystem machines.
	IERR = 7	negative VALUES value.
	IERR = 8	unexpected APIOPT value.
	IERR = 9	error opening CFILE.
	IERR = 10	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.71 ECHO

Use this API to enable or disable response echoing (activity ECHO).

Batch command syntax:

```
BAT_ECHO  IDVFIL
```

Python syntax:

```
ierr = echo(idvfil)
```

Fortran syntax:

```
CALL ECHOAPI(IDVFIL, IERR)
```

where:

Character IDVFIL*260	Is the name of Response File to be constructed (blank to disable) (input; blank by default).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 error opening IDVFIL.
	IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.72 EEQV

Use this API to construct an electrical equivalent of a specified subsystem of the working case (activity EEQV).

Batch command syntax:

```
BAT_EEQV  SID  ALL  STATUS(1)...STATUS(6)  DVAL(1)  DVAL(2)
```

Python syntax:

```
ierr = eeqv(sid, all, status, dval)
```

Fortran syntax:

```
CALL EEQVNW(SID, ALL, STATUS, DVAL, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all buses.
	ALL = 0	process only buses in subsystem SID.
Integer STATUS(6)	Is an array of six elements (input). They are used to control equivalencing options. A value of 0 means the option is not enabled. A value of 1 means the option is enabled. They are as follows.	
	STATUS(1)	retain area boundary buses (0 by default).
	STATUS(2)	retain zone boundary buses (0 by default).
	STATUS(3)	suppress the equivalencing option for phase shifters (0 by default).
	STATUS(4)	retain controlled buses (0 by default).
	STATUS(5)	retain existing branches between retained buses (0 by default).
	STATUS(6)	net load and shunt at retained buses (0 by default).
Real DVAL(2)	Is an array of two elements (input). They are as follows.	
	DVAL(1)	minimum active or reactive power generation for retaining generator buses (0.0 by default).
	DVAL(2)	branch threshold tolerance (10.0 by default).
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	invalid STATUS value.
	IERR = 4	invalid DVAL value.
	IERR = 5	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.73 EQRD

Use this API to build an electrical equivalent of radial and, optionally, two-point Type 1 buses in a specified subsystem of the working case (activity EQRD).

Batch command syntax:

```
BAT_EQRD  SID  ALL  STATUS(1)...STATUS(8)
```

Python syntax:

```
ierr = eqrd(sid, all, status)
```

Fortran syntax:

```
CALL EQRDNW(SID, ALL, STATUS, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all buses.
	ALL = 0	process only buses in subsystem SID.
Integer STATUS(8)	Is an array of eight elements (input). They are used to control equivalencing options. A value of 0 means the option is not enabled. A value of 1 means the option is enabled. They are as follows.	
	STATUS(1)	build an electrical equivalent of radial buses only (0 by default).
	STATUS(2)	suppress the equivalencing option for transformers (0 by default).
	STATUS(3)	suppress the equivalencing option for zero impedance lines (0 by default).
	STATUS(4)	suppress the equivalencing option for controlled buses (0 by default).
	STATUS(5)	suppress the equivalencing option for mutually coupled branches (0 by default).
	STATUS(6)	suppress the equivalencing option for area boundary buses (0 by default).
	STATUS(7)	suppress the equivalencing option for zone boundary buses (0 by default).
	STATUS(8)	apply transformer impedance correction to zero sequence network (1 by default)
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred
	IERR = 1	invalid SID value or subsystem SID is not defined
	IERR = 2	invalid ALL value
	IERR = 3	invalid STATUS value

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IERR = 4	error writing the working case to a temporary file
IERR = 5	load table is full; restoring original working case
IERR = 6	fixed bus shunt table is full; restoring original working case
IERR = 7	error opening temporary file
IERR = 8	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.74 EXAM

Use this API to tabulate all power flow data pertaining to a specified bus (activity EXAM).

Batch command syntax:

```
BAT_EXAM  SID  ALL
```

Python syntax:

```
ierr = exam(sid, all)
```

Fortran syntax:

```
CALL EXAMNW(SID, ALL, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.75 EXTR

Use this API to purge specified buses and all branches connected to them from the working case (activity EXTR).

Batch command syntax:

```
BAT_EXTR  SID  ALL  STATUS(1)  STATUS(2)
```

Python syntax:

```
ierr = extr(sid, all, status)
```

Fortran syntax:

```
CALL EXTRNW(SID, ALL, STATUS, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; no default allowed when ALL is 0).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 0 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Integer STATUS(2)	Is an array of two elements (input). They are used to control subsystem removal options. A value of 0 means the option is not enabled. A value of 1 means the option is enabled. They are as follows. STATUS(1) remove generator entries for out-of-service plants (0 by default). STATUS(2) change codes of boundary buses (0 by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 invalid STATUS value. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.76 FACT

Use this API to factorize the network admittance matrix in preparation for switching studies and dynamic simulations (activity FACT).

Batch command syntax:

```
BAT_FACT
```

Python syntax:

```
ierr = fact()
```

Fortran syntax:

```
CALL FACTAPI(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	generators are not converted.
	IERR = 2	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.77 FDNS

Use this API to apply the fixed slope decoupled Newton-Raphson power flow calculation (activity FDNS).

Batch command syntax:

```
BAT_FDNS  OPTIONS(1)...OPTIONS(8)
```

Python syntax:

```
ierr = fdns(options)
```

Fortran syntax:

```
CALL FDNSAPI(OPTIONS, IERR)
```

where:

Integer OPTIONS(8)	Is an array of eight elements specifying solution options (input). The values are as follows:	
	OPTIONS(1)	tap adjustment flag (use tap adjustment option setting by default).
	OPTIONS(1) = 0	disable.
	OPTIONS(1) = 1	enable stepping adjustment.
	OPTIONS(1) = 2	enable direct adjustment.
	OPTIONS(2)	area interchange adjustment flag (use area interchange adjustment option setting by default).
	OPTIONS(2) = 0	disable.
	OPTIONS(2) = 1	enable using tie line flows only in calculating area interchange.
	OPTIONS(2) = 2	enable using tie line flows and loads in calculating area interchange.
	OPTIONS(3)	phase shift adjustment flag (use phase shift adjustment option setting by default).
	OPTIONS(3) = 0	disable.
	OPTIONS(3) = 1	enable.
	OPTIONS(4)	dc tap adjustment flag (use dc tap adjustment option setting by default).
	OPTIONS(4) = 0	disable.
	OPTIONS(4) = 1	enable.
	OPTIONS(5)	switched shunt adjustment flag (use switched shunt adjustment option setting by default).
	OPTIONS(5) = 0	disable.
	OPTIONS(5) = 1	enable.
	OPTIONS(5) = 2	enable continuous mode, disable discrete mode.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTIONS(6)	flat start flag (0 by default).
	OPTIONS(6) = 0	do not flat start.
	OPTIONS(6) = 1	flat start.
	OPTIONS(7)	var limit flag (99 by default).
	OPTIONS(7) = 0	apply var limits immediately.
	OPTIONS(7) = >0	apply var limits on iteration n (or sooner if mismatch gets small).
	OPTIONS(7) = -1	ignore var limits.
	OPTIONS(8)	non-divergent solution flag (use non-divergent solution option setting by default).
	OPTIONS(8) = 0	disable.
	OPTIONS(8) = 1	enable.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid OPTIONS value.
	IERR = 2	generators are converted.
	IERR = 3	buses in island(s) without a swing bus; use activity TREE.
	IERR = 4	bus type code and series element status inconsistencies.
	IERR = 5	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.78 FILE_OVERWRITE

Use this API to set the file overwrite option setting to either *ask first* or *overwrite* (activity OPTN).

Batch command syntax:

```
BAT_FILE_OVERWRITE  IVAL
```

Python syntax:

```
ierr = file_overwrite(ival)
```

or:

```
ierr,ival = file_overwrite()
```

Fortran syntax:

```
CALL FILE_OVERWRITE_API(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Is the value of the option setting (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0, else output; 1 by default for input).	
	IVAL = 0	ask first.
	IVAL = 1	overwrite.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE = 0	set the option setting to the value specified in IVAL.
	IOCODE = 1	return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid IVAL value.
	IERR = 2	invalid IOCODE value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.79 FIND

Use this API to tabulate a list of buses matching a partial extended bus name (activity FIND).

Batch command syntax:

```
BAT_FIND  NAME, BSKV
```

Python syntax:

```
ierr = find(name, bskv)
```

Fortran syntax:

```
CALL FINDAPI(NAME, BSKV, IERR)
```

where:

Character NAME	Is the bus name (up to 24 characters, including imbedded asterisks that are treated as wildcard characters) (input; blank by default).
Character BSKV	Is the bus base voltage (up to 12 characters) (input; blank by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 error decoding BSKV. IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.80 FLAT

This API routine is obsolete. It has been replaced by the API routine FLAT_2, and is implemented by a call to the FLAT_2 API routine.

Use this API to specify various levels of flat conditions in preparation for fault analysis calculations (activity FLAT).

Batch command syntax:

```
BAT_FLAT  OPTIONS(1)...OPTIONS(4)  VALUES(1)  VALUES(2)
```

Python syntax:

```
ierr = flat(options, values)
```

Fortran syntax:

```
CALL FLATAPI(OPTIONS, VALUES, IERR)
```

where:

Integer OPTIONS(4)	Is an array of four elements specifying boundary condition options (input). The values are as follows:	
	OPTIONS(1)	flat option (0 by default).
	OPTIONS(1) = 0	only change voltages (activity FLAT).
	OPTIONS(1) = 1	set classical fault analysis conditions (activity FLAT,CL).
	OPTIONS(1) = 2	set conditions in preparation for IEC standard 909 calculations (activity FLAT,IEC).
	OPTIONS(2)	tap ratio treatment flag (ignored if OPTIONS(1) is 0) (0 by default).
	OPTIONS(2) = 0	leave tap ratios unchanged.
	OPTIONS(2) = 1	set tap ratios to 1.0.
	OPTIONS(3)	charging treatment flag (ignored if OPTIONS(1) is 0) (0 by default).
	OPTIONS(3) = 0	leave charging unchanged.
	OPTIONS(3) = 1	set charging to 0.0.
	OPTIONS(4)	shunt treatment flag (ignored if OPTIONS(1) is 0) (0 by default).
	OPTIONS(4) = 0	leave shunts unchanged.
	OPTIONS(4) = 1	set line shunts to 0.0 in the positive (and hence negative) sequence, and fixed and switched shunts to 0.0 in all three sequence networks.
	OPTIONS(4) = 2	set line shunts, fixed shunts, and switched shunts to 0.0 in all sequence networks, and negative and zero sequence shunt loads to 0.0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real VALUES(2)	Is an array of two elements specifying boundary condition values (ignored unless OPTIONS(1) is 2) (input). The values are as follows:	
	VALUES(1)	voltage magnitude for all buses, or 0.0 to use present voltage magnitudes (0.0 by default).
	VALUES(2)	generator power factor, or 0.0 to leave generator Mvar unchanged (0.0 by default).
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid OPTIONS value.
	IERR = 2	invalid VALUES value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.81 FLAT_2

This API routine is the second release of the flat conditions function. Use this API routine to specify various levels of flat conditions in preparation for fault analysis calculations (activity FLAT)

Batch command syntax:

```
BAT_FLAT_2  OPTIONS(1)...OPTIONS(8)  VALUES(1)  VALUES(2)
```

Python syntax:

```
ierr = flat_2(options, values)
```

Fortran syntax:

```
CALL FLAT_2(OPTIONS, VALUES, IERR)
```

where:

Integer OPTIONS(8)	Is an array of eight elements specifying boundary condition options (input). The values are as follows:	
	OPTIONS(1)	flat option (0 by default).
	OPTIONS(1) = 0	only change voltages (activity FLAT).
	OPTIONS(1) = 1	set classical fault analysis conditions (activity FLAT,CL).
	OPTIONS(1) = 2	set conditions in preparation for IEC standard 909 calculations (activity FLAT,IEC).
	OPTIONS(2)	tap ratio treatment flag (ignored if OPTIONS(1) is 0) (0 by default).
	OPTIONS(2) = 0	leave tap ratios unchanged.
	OPTIONS(2) = 1	set tap ratios to 1.0.
	OPTIONS(3)	charging treatment flag (ignored if OPTIONS(1) is 0) (0 by default).
	OPTIONS(3) = 0	leave charging unchanged.
	OPTIONS(3) = 1	set charging to 0.0.
	OPTIONS(4)	fixed bus shunt treatment flag (ignored if OPTIONS(1) is 0) (0 by default).
	OPTIONS(4) = 0	leave fixed bus shunts unchanged.
	OPTIONS(4) = 1	set fixed bus shunts to 0.0 in the positive (and hence negative) sequence.
	OPTIONS(4) = 2	set fixed bus shunts to 0.0 in all sequence networks.
	OPTIONS(5)	switched shunt treatment flag (ignored if OPTIONS(1) is 0) (0 by default).
	OPTIONS(5) = 0	leave switched shunts unchanged.
	OPTIONS(5) = 1	set switched shunts to 0.0 in all sequence networks.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(6)	line shunt treatment flag (ignored if OPTIONS(1) is 0) (0 by default).
OPTIONS(6) = 0	leave line shunts unchanged.
OPTIONS(6) = 1	set line shunts to 0.0 in the positive (and hence negative) sequence.
OPTIONS(6) = 2	set line shunts to 0.0 in all sequence networks.
OPTIONS(7)	transformer magnetizing admittance treatment flag (ignored if OPTIONS(1) is 0) (0 by default).
OPTIONS(7) = 0	leave transformer magnetizing admittances unchanged.
OPTIONS(7) = 1	set transformer magnetizing admittances to 0.0 in the positive (and hence negative) sequence.
OPTIONS(8)	load treatment flag; this option applies to the constant power, current, and admittance load characteristic data (specified on Load Data records in the Power Flow Raw Data File), and to the exceptional negative and zero sequence load data (specified on Load Data records in the Sequence Data File) (ignored if OPTIONS(1) is 0) (0 by default).
OPTIONS(8) = 0	leave loads unchanged.
OPTIONS(8) = 1	set the constant power, current and admittance load to 0.0 in all sequence networks; model (GNEG + j BNEG) in the negative sequence, and (GZERO + j BZERO) in the zero sequence.
OPTIONS(8) = 2	set the constant power, current and admittance load to 0.0 in all sequence networks, and (GNEG + j BNEG) to 0.0 in the negative sequence; model (GZERO + j BZERO) in the zero sequence.
OPTIONS(8) = 3	set the constant power, current and admittance load to 0.0 in all sequence networks, (GNEG + j BNEG) to 0.0 in the negative sequence, and (GZERO + j BZERO) to 0.0 in the zero sequence.

Real VALUES(2) Is an array of two elements specifying boundary condition values (ignored unless OPTIONS(1) is 2) (input). The values are as follows:

VALUES(1)	voltage magnitude for all buses, or 0.0 to use present voltage magnitudes (0.0 by default).
VALUES(2)	generator power factor, or 0.0 to leave generator Mvar unchanged (0.0 by default).

Integer IERR Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid OPTIONS value.
IERR = 2	invalid VALUES value.
IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.82 FNSL

Use this API to apply the Newton-Raphson power flow calculation (activity FNSL).

Batch command syntax:

```
BAT_FNSL  OPTIONS(1)...OPTIONS(8)
```

Python syntax:

```
ierr = fnsl(options)
```

Fortran syntax:

```
CALL FNSLAPI(OPTIONS, IERR)
```

where:

Integer OPTIONS(8)	Is an array of eight elements specifying solution options (input). The values are as follows:	
	OPTIONS(1)	tap adjustment flag (use tap adjustment option setting by default).
	OPTIONS(1) = 0	disable.
	OPTIONS(1) = 1	enable stepping adjustment.
	OPTIONS(1) = 2	enable direct adjustment.
	OPTIONS(2)	area interchange adjustment flag (use area interchange adjustment option setting by default).
	OPTIONS(2) = 0	disable.
	OPTIONS(2) = 1	enable using tie line flows only in calculating area interchange.
	OPTIONS(2) = 2	enable using tie line flows and loads in calculating area interchange.
	OPTIONS(3)	phase shift adjustment flag (use phase shift adjustment option setting by default).
	OPTIONS(3) = 0	disable.
	OPTIONS(3) = 1	enable.
	OPTIONS(4)	dc tap adjustment flag (use dc tap adjustment option setting by default).
	OPTIONS(4) = 0	disable.
	OPTIONS(4) = 1	enable.
	OPTIONS(5)	switched shunt adjustment flag (use switched shunt adjustment option setting by default).
	OPTIONS(5) = 0	disable.
	OPTIONS(5) = 1	enable.
	OPTIONS(5) = 2	enable continuous mode, disable discrete mode.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTIONS(6)	flat start flag (0 by default).
	OPTIONS(6) = 0	do not flat start.
	OPTIONS(6) = 1	flat start.
	OPTIONS(7)	var limit flag (99 by default).
	OPTIONS(7) = 0	apply var limits immediately.
	OPTIONS(7) = >0	apply var limits on iteration n (or sooner if mismatch gets small).
	OPTIONS(7) = -1	ignore var limits.
	OPTIONS(8)	non-divergent solution flag (use non-divergent solution option setting by default).
	OPTIONS(8) = 0	disable.
	OPTIONS(8) = 1	enable.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid OPTIONS value.
	IERR = 2	generators are converted.
	IERR = 3	buses in island(s) without a swing bus; use activity TREE.
	IERR = 4	bus type code and series element status inconsistencies.
	IERR = 5	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.83 GCAP

Use this API to read a data file containing capability curve data and print a report of machine loading and limit data (activity GCAP). Optionally, machine reactive power limits in the working case may be updated.

Batch command syntax:

```
BAT_GCAP  SID  ALL  OPT  GFILE
```

Python syntax:

```
ierr = gcap(sid, all, opt, gfile)
```

Fortran syntax:

```
CALL GCAPNW(SID, ALL, OPT, GFILE, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Integer OPT	Is the update var limits option (input; 0 by default). OPT = 0 leave var limits unchanged. OPT = 1 update var limits in the working case.
Character GFILE*260	Is the Machine Capability Data File (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 invalid OPT value. IERR = 4 error opening GFILE. IERR = 5 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.84 GDIF

Use this API to calculate differences between the working case and a designated Saved Case (activity GDIF). Results may be programmatically accessed via the internally accessible fetch routines DIFTXT, DIFBUS, DIFBRN, DIF3WN, DIF2DC, DIFMDC and DIFFCT.

Usage is:

call to GDIFAPI with APIOPT = 0;

multiple calls to fetch routines;

optionally, repeat of above steps for different Saved Case and/or subsystem;

final call to GDIFAPI with APIOPT = 1.

Batch command syntax:

```
BAT_GDIF  SID  ALL  APIOPT  NAMARG  FILARG
```

Python syntax:

```
ierr = gdif(sid, all, apiopt, namarg, filarg)
```

Fortran syntax:

```
CALL GDIFAPI(SID, ALL, APIOPT, NAMARG, FILARG, IERR)
```

where:

Integer SID — Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; used only when APIOPT is 0; 0 by default).

Integer ALL — Is the *all buses* or specified subsystem flag (input; used only when APIOPT is 0; 1 by default).

- ALL = 1 — process all buses.
- ALL = 0 — process only buses in subsystem SID.

Integer APIOPT — Is the mode of operation of GDIFAPI (input; no default allowed).

- APIOPT = 0 — apply difference calculations and fill arrays used by fetch routines.
- APIOPT = 1 — deallocate arrays used by fetch routines.

Integer NAMARG — Is the flag indicating how to match up buses in the two cases (input; used only when APIOPT is 0; 0 by default).

- NAMARG = 0 — by bus number only.
- NAMARG = 1 — by extended bus name only.
- NAMARG = 2 — by both number and name.

Character FILARG*260 — Is the name of Saved Case File to be compared to the working case (input; no default allowed when APIOPT is 0).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	invalid APIOPT value.
	IERR = 4	invalid NAMARG value.
	IERR = 5	FILARG is blank.
	IERR = 6	error reading the comparison case.
	IERR = 7	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.85 GENDSP

Use this API to impose a contingency specified in the Distribution Factor Data file and apply the generation dispatch algorithm used in contingency analysis on the working case. The network can then be solved by power flow solutions.

Batch command syntax:

```
BAT_GENDSP  OPTIONS(1)  OPTIONS(2)  LABELS(1)  LABELS(2)  DFXFILE
THRFILE  INLFILE
```

Python syntax:

```
ierr = gendsp(options, labels, dfxfile, thrfile, inlfile)
```

Fortran syntax:

```
CALL GENDSPAPI(OPTIONS, LABELS, DFXFILE, THRFILE, INLFILE, IERR)
```

where:

Integer OPTIONS(2)	Is an array of two elements specifying options (input).The values are as follows:	
	OPTIONS(1)	dispatch mode (1 by default).
	OPTIONS(1) = 0	disable dispatch
	OPTIONS(1) = 1	subsystem machines (RESERVE)
	OPTIONS(1) = 2	subsystem machines (PMAX)
	OPTIONS(1) = 3	subsystem machines (INERTIA)
	OPTIONS(1) = 4	subsystem machines (GOVERNOR DROOP)
	OPTIONS(2)	study case (1 by default)
	OPTIONS(2) = 0	impose a contingency case on the working case
	OPTIONS(2) = 1	working case
Character LABELS(2)*12	Is an array of two elements specifying subsystem labels (input). The values are as follows:	
	LABELS(1)	label of generation dispatch subsystem (blank by default, no default allowed if OPTIONS(1) is from 1 to 4).
	LABELS(2)	contingency label (blank by default, no default allowed if OPTIONS(2) is 0).
Character DFXFILE*260	Is the name of the Distribution Factor Data File (input; no default allowed).	
Character THRFILE*260	Is the name of the Load Throwover Data File (blank by default).	
Character INLFILE*260	Is the name of the Unit Inertia and Governor Data File (blank by default).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error
IERR = 1	invalid OPTIONS value
IERR = 2	no dispatch subsystem specified
IERR = 3	no contingency specified
IERR = 4	no Distribution Factor Data File specified
IERR = 5	generators are converted
IERR = 6	too many islands in base case
IERR = 7	contingency not found in Distribution Factor Data File
IERR = 21	file is not in the form of a PSS®E-25 or later DFAX file; run DFAX
IERR = 23	no subsystem defined in Distribution Factor Data file
IERR = 24	subsystem not found
IERR = 51	error opening Distribution Factor Data File
IERR = 52	error opening Load Throwover Data File
IERR = 53	error opening Unit Inertia and Governor Data File
IERR = 54	prerequisite requirements for function not met

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.86 GENS

Use this API to tabulate the loading and voltage conditions at plant buses (activity GENS).

Batch command syntax:

```
BAT_GENS  SID  ALL  OPT  VREV
```

Python syntax:

```
ierr = gens(sid, all, opt, vrev)
```

Fortran syntax:

```
CALL GENSNW(SID, ALL, OPT, VREV, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all buses.
	ALL = 0	process only buses in subsystem SID.
Integer OPT	Is the output option (input; 4 by default).	
	OPT = 1	tabulate var limited plants with unequal var limits.
	OPT = 2	tabulate all var limited plants.
	OPT = 3	tabulate on-line plants.
	OPT = 4	tabulate all plants.
Integer VREV	Is the option to select the units in which voltages are tabulated (input; 0 by default).	
	VREV = 0	use default voltage units.
	VREV = 1	use alternate voltage units.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	invalid OPT value.
	IERR = 4	invalid VREV value.
	IERR = 5	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.87 GEOL

Use this API to tabulate the loading and voltage conditions at the generator terminals for online machines at Type 2 and 3 buses in the working case (activity GEOL).

Batch command syntax:

```
BAT_GEOL  SID  ALL  OPT
```

Python syntax:

```
ierr = geol(sid, all, opt)
```

Fortran syntax:

```
CALL GEOLNW(SID, ALL, OPT, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all buses.
	ALL = 0	process only buses in subsystem SID.
Integer OPT	Is the output option (input; 1 by default).	
	OPT = 1	tabulate overloaded machines only.
	OPT = 2	tabulate all machines.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	invalid OPT value.
	IERR = 4	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.88 GetContingencySavedCase

Use this API routine to place the working case in the form of a specified system condition as calculated during a previous run of one of the members of the the AC contingency calculation family. This function retrieves data from a ZIP Archive Ouptut File that was created during the contingency calculation.

Batch command syntax:

```
BAT_GetContingencySavedCase  PATHZIP  ISVFILE
```

Python syntax:

```
ierr = GetContingencySavedCase(pathzip, isvfile)
```

Fortran syntax:

```
CALL GetContingencySavedCase(PATHZIP, ISVFILE, IERR)
```

where:

Character PATHZIP*260	Is the name of the ZIP Archive Output File (input; no default allowed).
Character ISVFILE*260	Is the name of an Incremental Saved Case File (.isv file) contained in PATHZIP (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 — no error occurred.
	IERR = 1 — unable to pick up the base Saved Case File from the ZIP Archive File.
	IERR = 2 — error opening the Incremental Saved Case File.
	IERR = 3 — error closing the Incremental Saved Case File.
	IERR = 4 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.89 GIC

This API is used to calculate Geomagnetically Induced Currents (GIC) in electric power system network as a result of Geomagnetic Disturbance (GMD).

Batch command syntax:

```
BAT_GIC  SID  ALL  APIOPTNS(1)  APIOPTNS(2)  APIOPTNS(3)  APIOPTNS(4)
VALUES(1)....VALUES(5)  RPTOPTNS(1)....RPTOPTNS(7)  GICDATAFIL
GICADDFIL GICPURGFIL GICRNWKFIL
```

Python syntax:

```
ierr = gic(sid, all, apioptns, values, rptoptns, gicdatafil,
gicaddfil, gicpurgfil, gicrnwkfil)
```

Fortran syntax:

```
CALL GIC(SID, ALL, APIOPTNS, VALUES, RPTOPTNS, GICDATAFIL, GICADDFIL,
GICPURGFIL, GICRNWKFIL, IERR)
```

where:

Integer SID	Is a valid area subsystem identifier. Valid subsystem identifiers range from 0 to 11. Area subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the all areas or specified subsystem flag (input; 1 by default). ALL = 1 process all areas. ALL = 0 process only areas in area subsystem SID.
Integer APIOPTNS(4)	Is an array of 4 elements (input). The values are as follows. APIOPTNS (1) = Electric Field Type (0 by default) = 0 for Uniform = 1 for Non-Uniform (Note: only Uniform field type is implemented.) APIOPTNS (2) = Unit of Electric Field Magnitude (0 by default) = 0 for Volts/km = 1 for Volts/mile APIOPTNS (3) = Option to add GIC updates to base case (1 by default) = 0 add updates to Saved Case = 1 create RDCH for GIC updates APIOPTNS (4) = Number of intertie levels (0 by default) >=0 used only is subsystem is specified

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real VALUES(5)	Is an array of 5 elements (input). The values are as follows. VALUES (1) = electric field magnitude in units defined by APIOPTNS (1) > 0.0 (no default allowed) VALUES (2) = electric field direction in degrees >= 0.0 and <= 360.0 (no default allowed) VALUES (3) = substation grounding dc resistance in ohms (0.1 ohm by default) VALUES (4) = non-transformer branch X/R ratio, used to calculate branch DC resistance if R=0.0 in power flow data (30.0 by default) VALUES (5) = transformer winding X/R ratio, used to calculate winding DC resistance if R=0.0 in power flow data (30.0 by default)
Integer RPTOPTNS (7)	Is an array of 7 elements (input). The values are used to report different quantities in text report. RPTOPTNS (1) = what to report (-1 by default) = -1 to report all = 0 for no report >0 to report these many maximum values in each category RPTOPTNS (2) = report branch induced DC voltages (0 by default) 0=No 1=Yes RPTOPTNS (3) = report DC bus voltages (0 by default) 0=No 1=Yes (4) = report branch GIC flows (0 by default) 0=No 1=Yes (5) = report transformer GIC flows (0 by default) 0=No 1=Yes (6) = report substation GIC flows (0 by default) 0=No 1=Yes (7) = report transformer reactive power losses (1 by default) 0=No 1=Yes
Character GICDATAFIL*260	Is the GIC Data file (input; no default allowed).
Character GICADDFIL*260	Is the name of Saved Case or RDCH raw file that will be created adding transformer reactive power losses due to GIC flow as constant current loads (output; blank by default). APIOPTNS (2) determines type of file created.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character GICPURGFIL *260	Is the name of RDCH raw file that will be created which puts status as "out-of-service" for loads representing transformer reactive power losses due to GIC flow (output; blank by default).
Character GICRNWKFIL *260	Is the name of raw file that will be created which represents resistive network used to calculate GIC flows (output; blank by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred IERR = 1 invalid SID value or subsystem SID is not defined IERR = 2 invalid ALL value IERR = 3 topology error bus(es) with bus type code and series element status inconsistencies IERR = 4 invalid APIOPTNS, set to default value IERR = 5 invalid VALUES, no default allowed IERR = 6 invalid VALUES, set to default value IERR = 7 invalid RPTOPTNS, set to default value IERR = 8 electric Field Value must be greater than 0.0 IERR = 9 error opening GICDATAFIL IERR = 10 error opening GICADDFIL IERR = 11 error opening GICPURGFIL IERR = 12 error opening GICRNWKFIL IERR = 13 no substation data provided IERR = 14 no substation data specified for following GIC study subsystem buses IERR = 15 no buses in GIC study subsystem IERR = 16 no GIC data input file provided IERR = 17 no non-transformer branches in working case IERR = 18 no non-transformer branches in GIC study subsystem IERR = 19 prerequisite requirements for API are not met

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.90 GNESTATUS

Use this API to change the status of a GNE device.

Batch command syntax:

```
BAT_GneStatus  'NAME'  IVAL
```

Python syntax:

```
ierr = gnestatus(name, ival)
```

or:

```
ierr,ival = gnestatus(name)
```

Fortran syntax:

```
CALL GneStatus(NAME, IVAL, IOCODE, IERR)
```

where:

Character NAME*12	Is the name of the GNE device for which the status is to be changed (input; no default allowed).
Integer IVAL	Is the value of the status of the specified GNE device (input if Batch command or for Python if specified as an argument, or for FORTRAN if IOCODE is 0; else output no default allowed for input). IVAL = 0 out-of-service. IVAL = 1 in-service.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the status to the value specified in IVAL. IOCODE = 1 return the current value of the status in IVAL.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 GNE device not found. IERR = 2 invalid GNE status value. IERR = 3 invalid IOCODE value. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.91 GNET

Use this API to change in-service generation to negative MVA load at all Type 2 and 3 buses in the subsystem specified by the user (activity GNET).

Batch command syntax:

```
BAT_GNET  SID  ALL
```

Python syntax:

```
ierr = gnet(sid, all)
```

Fortran syntax:

```
CALL GNETNW(SID, ALL, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.92 IECS

This API is obsolete. It has been replaced by the API IECS_4, and is implemented by a call to the IECS_4 API. Use this API to apply a series of faults at various locations in the working case to compute short circuit currents according to IEC 60909 standard.

Batch command syntax:

```
BAT_IECS SID ALL STATUS(1)...STATUS(8) BRKTIM IECFIL FCDFIL
```

Python syntax:

```
ierr = iecs(sid, all, status, brktim, iecfil, fcdfil)
```

Fortran syntax:

```
CALL IECSAPI(SID, ALL, STATUS, BRKTIM, IECFIL, FCDFIL, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all buses.
	ALL = 0	process only buses in subsystem SID.
Integer STATUS(8)	Is an array of eight elements (input). The value of each element is as follows.	
	STATUS(1)	three phase faults (0 by default, no 3PH faults)
	STATUS(1) =	0 or 1.
	STATUS(2)	line to ground faults (0 by default, no LG faults)
	STATUS(2) =	0 or 1.
	STATUS(3)	line-line to ground faults (0 by default, no LLG faults)
	STATUS(3) =	0 or 1.
	STATUS(4)	line to line faults (0 by default, no LL faults)
	STATUS(4) =	0 or 1.
	STATUS(5)	output option (1 by default).
	STATUS(5) = 1	total fault currents
	STATUS(5) = 2	I"k contributions to N levels away current
	STATUS(5) = 3	total fault currents and I"k contributions to N levels away
	STATUS(6)	number of levels back for contributions output. Used if STATUS(5) = 2 or 3. (0 by default)

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	STATUS(7)	the location of fault (1 by default)
	STATUS(7) = 1	fault at network bus
	STATUS(7) = 2	fault at LV bus of Power Station Unit (PSU)
	STATUS(7) = 3	fault at auxiliary transformer (connected to PSU) LV bus
	STATUS(8)	report total fault currents in descending order of magnitude (AMPS) (0 by default)
	STATUS(8) =	0 or 1
Real BRKTIM	Is the breaker contact parting time in seconds (5/BaseFrequency by default).	
Character IECFIL*260	Is the name of IEC Fault Calculation Data File; blank for none (input; blank by default).	
Character FCDFIL*260	Is the name of Fault Control Data File; blank for none (input; blank by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	invalid STATUS value.
	IERR = 4	invalid BRKTIM value.
	IERR = 5	no faults specified.
	IERR = 6	error opening IECFIL.
	IERR = 7	error opening FCDFIL.
	IERR = 8	prerequisite requirements for API are not met

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.93 IECS_2

This API is obsolete. It has been replaced by the API IECS_4 and is implemented by a call to the IECS_4 API. This API executes the second release of the IEC short circuit calculations function. This API is used to apply a series of faults at various locations in the working case to compute short circuit currents according to IEC 60909 standard.

Batch command syntax:

```
BAT_IECS_2 SID ALL STATUS(1)...STATUS(14) BRKTIM IECFIL FCDFIL SCFILE
```

Python syntax:

```
ierr = iecs_2(sid, all, status, brktim, iecfil, fcdfil, scfile)
```

Fortran syntax:

```
CALL IECS_2(SID, ALL, STATUS, BRKTIM, IECFIL, FCDFIL, SCFILE, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the all buses or selected subsystem flag (input; 1 by default).	
	ALL = 1	process all buses
	ALL = 0	process only buses in subsystem SID
Integer STATUS(14)	Is an array of 14 elements (input). The values are as follows.	
	STATUS(1)	three phase faults, (0 by default)
	STATUS(1) = 0	omit
	STATUS(1) = 1	include
	STATUS(2)	line-to-ground faults (0 by default)
	STATUS(2) = 0	omit
	STATUS(2) = 1	include
	STATUS(3)	line-line-to-ground faults, (0 by default)
	STATUS(3) = 0	omit
	STATUS(3) = 1	include
	STATUS(4)	line-to-line faults, (0 by default)
	STATUS(4) = 0	omit
	STATUS(4) = 1	include
	STATUS(5)	output option (1 by default)
	STATUS(5) = 1	total fault currents
	STATUS(5) = 2	fault contributions to N levels away
	STATUS(5) = 3	total fault currents and fault contributions to N levels away
	STATUS(6)	number of levels back for contributions output (0 by default),
	STATUS(6) ≥ 0	used if STATUS(5) = 2 or 3

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

STATUS(7)	the location of fault (1 by default)
STATUS(7)=1	fault at network bus
STATUS(7)=2	fault at LV bus of Power Station Unit (PSU)
STATUS(7)=3	fault at auxiliary transformer (connected to PSU) LV bus
STATUS(8)	line-out faults (0 by default)
STATUS(8) = 0	omit
STATUS(8) = 1	include
STATUS(9)	line-end faults (0 by default)
STATUS(9) = 0	omit
STATUS(9) = 1	include
STATUS(10)	set transformer tap ratios and phase angles option (0 by default)
STATUS(10) = 0	leave tap ratios and phase angles unchanged
STATUS(10) = 1	set tap ratios 1.0 pu and phase angles to 0.0 deg
STATUS(11)	line charging option (1 by default)
STATUS(11) = 1	set to 0.0 in positive and negative sequences
STATUS(11) = 2	set to 0.0 in all sequences
STATUS(11) = 3	leave line charging unchanged
STATUS(12)	loads, fixed and switched shunts and and magnetizing admittance option (1 by default)
STATUS(12) = 1	set to 0.0 in the positive and negative sequences
STATUS(12) = 2	set to 0.0 in all sequences
STATUS(12) = 3	leave unchanged
STATUS(13)	dc line and FACTS device option (0 by default)
STATUS(13) = 0	blocked
STATUS(13) = 1	represent as load
STATUS(14)	zero sequence transformer impedance correction option (0 by default)
STATUS(14) = 0	ignore
STATUS(14) = 1	apply

Real BRKTIM	Is the breaker contact parting time in seconds (5/BaseFrequency by default).
Character IECFIL*260	Is the name of IEC Fault Calculation Data file; blank for none (input; blank by default).
Character FCDFIL*260	Is the name of Fault Control Data file; blank for none (input; blank by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character SCFILE*260	Is the IECS_2 Results (.sc) file; blank for none (output; blank by default).
Integer IERR	Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid SID value or subsystem SID is not defined.
IERR = 2	invalid ALL value.
IERR = 3	invalid STATUS value.
IERR = 4	invalid BRKTIM value.
IERR = 5	no faults specified.
IERR = 6	error opening IECFIL.
IERR = 7	error opening FCDFIL.
IERR = 8	prerequisite requirements for API are not met
IERR = 9	error opening SCFILE
IERR = 10	SCFILE error
IERR = 11	bus SC results not available.
IERR = 12	branch SC results not available.
IERR = 13	three winding transformer SC results not available.
IERR = 14	incorrect SCFILE.
IERR = 15	SCFILE not allowed.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.94 IECS_3

This API is obsolete. It has been replaced by the API IECS_4 and is implemented by a call to the IECS_4 API.

This API executes the third release of the IEC short circuit calculations function. This API is used to apply a series of faults at various locations in the working case to compute short circuit currents according to IEC 60909 standard.

Batch command syntax:

```
BAT_IECS_3 SID ALL STATUS(1)...STATUS(15) VALUES(1) VALUES(2) IECFIL
FCDFIL SCFILE
```

Python syntax:

```
ierr = iecs_3(sid, all, status, values, iecfil, fcdfil, scfile)
```

Fortran syntax:

```
CALL IECS_3(SID, ALL, STATUS, VALUES, IECFIL, FCDFIL, SCFILE, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the all buses or selected subsystem flag (input; 1 by default).	
	ALL = 1	process all buses
	ALL = 0	process only buses in subsystem SID
Integer STATUS(15)	Is an array of 15 elements (input). The values are as follows.	
	STATUS(1)	three phase faults, (0 by default)
	STATUS(1) = 0	omit
	STATUS(1) = 1	include
	STATUS(2)	line-to-ground faults (0 by default)
	STATUS(2) = 0	omit
	STATUS(2) = 1	include
	STATUS(3)	line-line-to-ground faults, (0 by default)
	STATUS(3) = 0	omit
	STATUS(3) = 1	include
	STATUS(4)	line-to-line faults, (0 by default)
	STATUS(4) = 0	omit
	STATUS(4) = 1	include
	STATUS(5)	output option (1 by default)
	STATUS(5) = 1	total fault currents
	STATUS(5) = 2	fault contributions to N levels away
	STATUS(5) = 3	total fault currents and fault contributions to N levels away

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

STATUS(6)	number of levels back for contributions output (0 by default),
STATUS(6) ≥ 0	used if STATUS(5) = 2 or 3
STATUS(7)	the location of fault (1 by default)
STATUS(7)=1	fault at network bus
STATUS(7)=2	fault at LV bus of Power Station Unit (PSU)
STATUS(7)=3	fault at auxiliary transformer (connected to PSU) LV bus
STATUS(8)	line-out faults (0 by default)
STATUS(8) = 0	omit
STATUS(8) = 1	include
STATUS(9)	line-end faults (0 by default)
STATUS(9) = 0	omit
STATUS(9) = 1	include
STATUS(10)	set transformer tap ratios and phase angles option (0 by default)
STATUS(10) = 0	leave tap ratios and phase angles unchanged
STATUS(10) = 1	set tap ratios 1.0 pu and phase angles to 0.0 deg
STATUS(11)	line charging option (1 by default)
STATUS(11) = 1	set to 0.0 in positive and negative sequences
STATUS(11) = 2	set to 0.0 in all sequences
STATUS(11) = 3	leave line charging unchanged
STATUS(12)	loads, fixed and switched shunts and and magnetizing admittance option (1 by default)
STATUS(12) = 1	set to 0.0 in the positive and negative sequences
STATUS(12) = 2	set to 0.0 in all sequences
STATUS(12) = 3	leave unchanged
STATUS(13)	dc line and FACTS device option (0 by default)
STATUS(13) = 0	blocked
STATUS(13) = 1	represent as load
STATUS(14)	zero sequence transformer impedance correction option (0 by default)
STATUS(14) = 0	ignore
STATUS(14) = 1	apply

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	STATUS(15) Is the option for Voltage Factor C STATUS(15) = 1 Voltage Factor C for maximum fault currents (default) STATUS(15) = 2 Voltage Factor C for minimum fault currents STATUS(15) = 3 Voltage Factor C as specified by 'vfactorc' value
Real VALUES(2)	Is the array of 2 elements (input). The values are as follows: VALUES(1) Is the breaker contact parting time in seconds (5/base Frequency by default) VALUES(2) Is the user specified voltage factor c (1.0 by default) This is used only when STATUS(15)=3
Character IECFIL*260	Is the name of IEC Fault Calculation Data file; blank for none (input; blank by default).
Character FCDFIL*260	Is the name of Fault Control Data file; blank for none (input; blank by default).
Character SCFILE*260	Is the IECS_3 Results (.sc) file; blank for none (output; blank by default).
Integer IERR	Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid SID value or subsystem SID is not defined.
IERR = 2	invalid ALL value.
IERR = 3	invalid STATUS value.
IERR = 4	invalid VALUES value.
IERR = 5	no faults specified.
IERR = 6	error opening IECFIL.
IERR = 7	error opening FCDFIL.
IERR = 8	prerequisite requirements for API are not met
IERR = 9	error opening SCFILE
IERR = 10	SCFILE error
IERR = 11	bus SC results not available.
IERR = 12	branch SC results not available.
IERR = 13	three winding transformer SC results not available.
IERR = 14	incorrect SCFILE.
IERR = 15	SCFILE not allowed.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.77 IECS_4

This API executes the fourth release of the IEC short circuit calculations function. This API is used to apply a series of faults at various locations in the working case to compute short circuit currents according to IEC 60909 standard.

Batch command syntax:

```
BAT_IECS_4 SID ALL STATUS(1)...STATUS(17) VALUES(1) VALUES(2) IECFIL
FCDFIL SCFILE
```

Python syntax:

```
ierr = iecs_4(sid, all, status, values, iecfil, fcdfil, scfile)
```

Fortran syntax:

```
CALL IECS_4(SID, ALL, STATUS, VALUES, IECFIL, FCDFIL, SCFILE, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the all buses or selected subsystem flag (input; 1 by default).	
	ALL = 1	process all buses
	ALL = 0	process only buses in subsystem SID
Integer STATUS(17)	Is an array of 17 elements (input). The values are as follows.	
	STATUS(1)	three phase faults, (0 by default)
	STATUS(1) = 0	omit
	STATUS(1) = 1	include
	STATUS(2)	line-to-ground faults (0 by default)
	STATUS(2) = 0	omit
	STATUS(2) = 1	include
	STATUS(3)	line-line-to-ground faults, (0 by default)
	STATUS(3) = 0	omit
	STATUS(3) = 1	include
	STATUS(4)	line-to-line faults, (0 by default)
	STATUS(4) = 0	omit
	STATUS(4) = 1	include
	STATUS(5)	output option (1 by default)
	STATUS(5) = 0	fault current summary table
	STATUS(5) = 1	total fault currents
	STATUS(5) = 2	fault contributions to N levels away
	STATUS(5) = 3	total fault currents and fault contributions to N levels away

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

STATUS(6)	number of levels back for contributions output (0 by default),
STATUS(6) ≥ 0	used if STATUS(5) = 2 or 3
STATUS(7)	the location of fault (0 by default)
STATUS(7)=0	fault at network bus
STATUS(7)=1	fault at LV bus of Power Station Unit (PSU)
STATUS(7)=2	fault at auxiliary transformer (connected to PSU) LV bus
STATUS(8)	line-out faults (0 by default)
STATUS(8) = 0	omit
STATUS(8) = 1	include
STATUS(9)	line-end faults (0 by default)
STATUS(9) = 0	omit
STATUS(9) = 1	include
STATUS(10)	transformer tap ratios and phase shift angles option (0 by default)
STATUS(10) = 0	leave tap ratios and phase shift angles unchanged
STATUS(10) = 1	set tap ratios to 1.0 pu and phase shift angles to 0 deg
STATUS(10) = 2	set tap ratios to 1.0 pu and phase shift angles unchanged
STATUS(10) = 3	set tap ratios unchanged and phase shift angles to 0 deg
STATUS(11)	line charging option (1 by default)
STATUS(11) = 0	leave unchanged
STATUS(11) = 1	set to 0.0 in the positive and negative sequences
STATUS(11) = 2	set to 0.0 in all sequences
STATUS(12)	line, fixed and switched shunts, and magnetizing admittance option (1 by default)
STATUS(12) = 0	leave unchanged
STATUS(12) = 1	set to 0.0 in the positive and negative sequences
STATUS(12) = 2	set to 0.0 in all sequences
STATUS(13)	dc line and FACTS device option (0 by default)
STATUS(13) = 0	blocked
STATUS(13) = 1	represent as load

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	STATUS(14)	zero sequence transformer impedance correction option (0 by default)
	STATUS(14) = 0	ignore
	STATUS(14) = 1	apply
	STATUS(15)	Is the option for Voltage Factor C (0 by default)
	STATUS(15) = 0	Voltage Factor C for maximum fault currents
	STATUS(15) = 1	Voltage Factor C for minimum fault currents
	STATUS(15) = 2	Voltage Factor C as specified by 'vfactorc' value
	STATUS(16)	load option (1 by default)
	STATUS(16) = 0	leave unchanged
	STATUS(16) = 1	set to 0.0 in the positive and negative sequences
	STATUS(16) = 2	set to 0.0 in all sequences
	STATUS(17)	fault analysis generator reactance option; ignored if generators are already "converted" (0 by default).
	STATUS(17) = 0	use subtransient reactance.
	STATUS(17) = 1	use transient reactance.
	STATUS(17) = 2	use synchronous reactance
Real VALUES(2)	Is the array of 2 elements (input). The values are as follows:	
	VALUES(1)	Is the breaker contact parting time in seconds (5/base Frequency by default)
	VALUES(2)	Is the user specified voltage factor c (1.0 by default) This is used only when STATUS(15)=2
Character IECFIL*260	Is the name of IEC Fault Calculation Data file; blank for none (input; blank by default).	
Character FCDFIL*260	Is the name of Fault Control Data file; blank for none (input; blank by default).	
Character SCFILE*260	Is the IECS_4 Results (.sc) file; blank for none (output; blank by default).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid SID value or subsystem SID is not defined.
IERR = 2	invalid ALL value.
IERR = 3	invalid STATUS value.
IERR = 4	invalid VALUES value.
IERR = 5	no faults specified.
IERR = 6	error opening IECFIL.
IERR = 7	error opening FCDFIL.
IERR = 8	prerequisite requirements for API are not met
IERR = 9	error opening SCFILE
IERR = 10	SCFILE error
IERR = 11	bus SC results not available.
IERR = 12	branch SC results not available.
IERR = 13	three winding transformer SC results not available.
IERR = 14	incorrect SCFILE.
IERR = 15	SCFILE not allowed.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.95 IECS_SCFILE

This API reads IECS short circuit results from SCFILE and produces same IECS output report that was generated when SCFILE was created. SCFILE is read into PSS®E memory to enable displaying IECS results onto slider diagrams. For this API to work properly, it is required that working case in PSS®E memory should be the same Saved case that was used when creating the SCFILE.

Batch command syntax:

```
BAT_IECS_SCFILE SCFILE
```

Python syntax:

```
ierr = iecs_scfile(scfile)
```

Fortran syntax:

```
CALL IECS_SCFILE(SCFILE,IERR)
```

where:

Character SCFILE*260	Is the IECS_3 Results (.sc) file (input; no default allowed):
Integer IERR	Is the error code
	IERR = 0, no error occurred
	IERR = 7, prerequisite requirements for function not met
	IERR = 9, error opening SCFILE
	IERR = 10, SCFILE error
	IERR = 11, bus SC results not available.
	IERR = 12, branch SC results not available.
	IERR = 13, three winding transformer SC results not available.
	IERR = 14, incorrect SCFILE.
	IERR = 15, SCFILE not allowed.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.96 ImNet

Use this API routine to change the status of induction machines that are in-service to out-of-service, and to replace the power flowing from the network into the machine with constant power load. The routine can be directed to process either all in-service induction machines, all in-service induction machines within a specified subsystem, or all in-service induction machines outside of a specified subsystem.

Batch command syntax:

```
BAT_ImNet  SID  ALL  INOUT
```

Python syntax:

```
ierr = imnet(sid, all, inout)
```

Fortran syntax:

```
CALL ImNet(SID, ALL, INOUT, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all induction machines.
	ALL = 0	process only induction machines within, or outside of, subsystem SID.
Integer INOUT	Is the include or exclude induction machines in subsystem SID option (input; 0 by default).	
	INOUT = 0	process only machines inside of subsystem SID.
	INOUT = 1	process only machines outside of subsystem SID.
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	invalid INOUT value.
	IERR = 4	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.97 ImOutage

Use this API routine to change the status of induction machines that are in-service, but that have been set to the "stalled" (for motors) or "tripped" (for generators) state by the power flow solution, to out-of-service.

Batch command syntax:

```
BAT_ImOutage  SID  ALL  MOTOPT  GENOPT
```

Python syntax:

```
ierr = imoutage(sid, all, motopt, genopt)
```

Fortran syntax:

```
CALL ImOutage(SID, ALL, MOTOPT, GENOPT, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all induction machines.
	ALL = 0	process only induction machines in subsystem SID.
Integer MOTOPT	Is the "stalled" induction motor option (input; 0 by default).	
	MOTOPT = 0	set to out-of-service.
	MOTOPT = 1	leave in-service.
Integer GENOPT	Is the "tripped" induction generator option (input; 0 by default).	
	GENOPT = 0	set to out-of-service.
	GENOPT = 1	leave in-service.
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	invalid MOTOPT value.
	IERR = 4	invalid GENOPT value.
	IERR = 5	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.98 IMPC

Use this API to calculate the impact of transaction events on MW flows using a linear network (dc) model (activity IMPC).

Batch command syntax:

```
BAT_IMPC  EVENT, DFXFILE, ATCFILE
```

Python syntax:

```
ierr = impc(event, dfxfile, atcfile)
```

Fortran syntax:

```
CALL IMPCAPI(EVENT, DFXFILE, ATCFILE, IERR)
```

where:

Integer EVENT	Is the transaction event number (input; no default allowed).
Character DFXFILE*260	Is the name of Distribution Factor Data File (input; no default allowed).
Character ATCFILE*260	Is the name of ATC Updates File; blank for none (input; blank by default).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 no transactions defined.
	IERR = 2 transaction event not defined.
	IERR = 3 no Distribution Factor Data File specified.
	IERR = 4 error opening ATC Updates File.
	IERR = 5 error opening file DFXFILE.
	IERR = 6 file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
	IERR = 8 no monitored branches or interfaces defined.
	IERR = 9 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.99 IMPLEMENT_TRANSFER

This API routine is obsolete. It has been replaced by the API routine IMPLEMENT_TRANSFER_2, and is implemented by a call to the IMPLEMENT_TRANSFER_2 API routine.

Use this API routine to apply a specified transfer using the same transfer dispatch methods that are available in the PV analysis calculation engine.

Batch command syntax:

```
BAT_IMPLEMENT_TRANSFER  OPTIONS(1)...OPTIONS(4)  RVAL  LABELS(1)
LABELS(2)  DFXFILE  ECDFILE
```

Python syntax:

```
ierr = implement_transfer(options, rval, labels, dfxfile, ecdfile)
```

Fortran syntax:

```
CALL IMPLEMENT_TRANSFER(OPTIONS, RVAL, LABELS, DFXFILE, ECDFILE,
IERR)
```

where:

Integer OPTIONS(4)	Is an array of four elements specifying transfer implementation options (input). The values are as follows:	
	OPTIONS(1)	study (source) system transfer dispatch method (1 by default).
	OPTIONS(1) = 1	buses and distribution factors from DFAX file for buses with positive MW machines.
	OPTIONS(1) = 2	buses and distribution factors from DFAX file for buses with positive MW constant MVA load.
	OPTIONS(1) = 3	buses and distribution factors from DFAX file for buses with either positive MW machines or positive MW constant MVA load.
	OPTIONS(1) = 4	subsystem buses with positive MW constant MVA load in proportion to their MW load.
	OPTIONS(1) = 5	subsystem buses with positive MW machines in proportion to their MW output.
	OPTIONS(1) = 6	subsystem buses with positive MW machines in proportion to their MBASEs.
	OPTIONS(1) = 7	subsystem buses with positive MW machines in proportion to their reserve (PMAXMC - PGENMC).
	OPTIONS(1) = 8	subsystem buses with positive MW machines via ECDI with unit commitment disabled.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTIONS(2)	opposing (sink) system transfer dispatch method (1 by default).
	OPTIONS(2) = 1	buses and distribution factors from DFAX file for buses with positive MW machines.
	OPTIONS(2) = 2	buses and distribution factors from DFAX file for buses with positive MW constant MVA load.
	OPTIONS(2) = 3	buses and distribution factors from DFAX file for buses with either positive MW machines or positive MW constant MVA load.
	OPTIONS(2) = 4	subsystem buses with positive MW constant MVA load in proportion to their MW load.
	OPTIONS(2) = 5	subsystem buses with positive MW machines in proportion to their MW output.
	OPTIONS(2) = 6	subsystem buses with positive MW machines in proportion to their MBASEs.
	OPTIONS(2) = 7	subsystem buses with positive MW machines in proportion to their reserve (PGENMC - PMINMC).
	OPTIONS(2) = 8	subsystem buses with positive MW machines via ECDI with unit commitment disabled
	OPTIONS(3)	generation plant limits flag for transfer methods 1, 3, 5, 6 and 7 (0 by default).
	OPTIONS(3) = 0	no limits.
	OPTIONS(3) = 1	honor machine active power limits.
	OPTIONS(4)	positive load flag for transfer methods 2, 3 and 4 (0 by default).
	OPTIONS(4) = 0	no limits.
	OPTIONS(4) = 1	enforce non-negative net active power constant MVA load.
Real RVAL	Is the transfer increment in MW (input; no default allowed).	
Character LABELS(2)*12	Is an array of two elements specifying subsystem labels (input). The values are as follows:	
	LABELS(1)	label of the study (source) system for which generation is to be increased and/or for which load is to be decreased (no default allowed).
	LABELS(2)	label of the opposing (sink) system for which generation is to be decreased and/or for which load is to be increased (no default allowed).
Character DFXFILE*260	Is the name of the Distribution Factor Data file (input; no default allowed).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character ECDFILE*260	Is the name of the Economic Dispatch Data file; blank for none (input; blank by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid RVAL value.
	IERR = 2	invalid OPTIONS value.
	IERR = 3	generators are converted.
	IERR = 4	buses in island(s) without a swing bus; use activity TREE.
	IERR = 5	no Distribution Factor Data File specified.
	IERR = 6	no Economic Dispatch Data File specified.
	IERR = 7	no participating dispatch buses in study system.
	IERR = 8	no participating dispatch buses in opposing system.
	IERR = 9	subsystem engine error.
	IERR = 10	Unable to implement transfer in study system.
	IERR = 11	Unable to implement transfer in opposing system.
	IERR = 21	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
	IERR = 22	monitored elements exceed limit when adding multi-section line members.
	IERR = 23	no subsystems defined.
	IERR = 24	subsystem not found.
	IERR = 25	specified subsystems overlap.
	IERR = 26	no in-service buses in subsystem WORLD.
	IERR = 50	error opening Economic Dispatch Data File.
	IERR = 52	error opening Distribution Factor Data File.
	IERR = 54	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.100 IMPLEMENT_TRANSFER_2

Use this API routine to apply a specified transfer using the same transfer dispatch methods that are available in the PV analysis calculation engine.

Batch command syntax:

```
BAT_IMPLEMENT_TRANSFER_2  OPTIONS(1)...OPTIONS(4)  RVAL(1)  RVAR(2)
LABELS(1)  LABELS(2)  DFXFILE  ECDFILE
```

Python syntax:

```
ierr = implement_transfer_2(options, rval, labels, dfxfile, ecdfile)
```

Fortran syntax:

```
CALL IMPLEMENT_TRANSFER_2(OPTIONS, RVAL, LABELS, DFXFILE, ECDFILE,
IERR)
```

where:

Integer OPTIONS(4)	Is an array of four elements specifying transfer implementation options (input). The values are as follows:	
	OPTIONS(1)	study (source) system transfer dispatch method (1 by default).
	OPTIONS(1) = 1	buses and distribution factors from DFAX file for buses with positive MW machines.
	OPTIONS(1) = 2	buses and distribution factors from DFAX file for buses with positive MW constant MVA load.
	OPTIONS(1) = 3	buses and distribution factors from DFAX file for buses with either positive MW machines or positive MW constant MVA load.
	OPTIONS(1) = 4	subsystem buses with positive MW constant MVA load in proportion to their MW load.
	OPTIONS(1) = 5	subsystem buses with positive MW machines in proportion to their MW output.
	OPTIONS(1) = 6	subsystem buses with positive MW machines in proportion to their MBASEs.
	OPTIONS(1) = 7	subsystem buses with positive MW machines in proportion to their reserve (PMAXMC - PGENMC).
	OPTIONS(1) = 8	subsystem buses with positive MW machines via ECDI with unit commitment disabled.
	OPTIONS(1) = 9	subsystem buses with positive MW machines via ECDI with unit commitment enabled.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(2)	opposing (sink) system transfer dispatch method (1 by default).
OPTIONS(2) = 1	buses and distribution factors from DFAX file for buses with positive MW machines.
OPTIONS(2) = 2	buses and distribution factors from DFAX file for buses with positive MW constant MVA load.
OPTIONS(2) = 3	buses and distribution factors from DFAX file for buses with either positive MW machines or positive MW constant MVA load.
OPTIONS(2) = 4	subsystem buses with positive MW constant MVA load in proportion to their MW load.
OPTIONS(2) = 5	subsystem buses with positive MW machines in proportion to their MW output.
OPTIONS(2) = 6	subsystem buses with positive MW machines in proportion to their MBASEs.
OPTIONS(2) = 7	subsystem buses with positive MW machines in proportion to their reserve (PGENMC - PMINMC).
OPTIONS(2) = 8	subsystem buses with positive MW machines via ECDI. with unit commitment disabled.
OPTIONS(2) = 9	subsystem buses with positive MW machines via ECDI with unit commitment enabled.
OPTIONS(3)	generation plant limits flag for transfer methods 1, 3, 5, 6 and 7 (0 by default).
OPTIONS(3) = 0	no limits.
OPTIONS(3) = 1	honor machine active power limits.
OPTIONS(4)	positive load flag for transfer methods 2, 3 and 4 (0 by default).
OPTIONS(4) = 0	no limits.
OPTIONS(4) = 1	enforce non-negative net active power constant MVA load.

Real RVAL(2)

Is an array of two elements specifying transfer implementation options (input). The values are as follows:

RVAL(1)	the transfer increment in MW (no default allowed).
RVAL(2)	power factor for load increases in dispatch methods 2, 3 and 4 (0.0 by default). Specify RVAL(2) as ≤ 0.0 to retain the original power factor at each load that is changed.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character LABELS(2)*12	Is an array of two elements specifying subsystem labels (input). The values are as follows:	
	LABELS(1)	label of the study (source) system for which generation is to be increased and/or for which load is to be decreased (no default allowed).
	LABELS(2)	label of the opposing (sink) system for which generation is to be decreased and/or for which load is to be increased (no default allowed).
Character DFXFILE*260	Is the name of the Distribution Factor Data file (input; no default allowed).	
Character ECDFILE*260	Is the name of the Economic Dispatch Data file; blank for none (input; blank by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid RVAL value.
	IERR = 2	invalid OPTIONS value.
	IERR = 3	generators are converted.
	IERR = 4	buses in island(s) without a swing bus; use activity TREE.
	IERR = 5	no Distribution Factor Data File specified.
	IERR = 6	no Economic Dispatch Data File specified.
	IERR = 7	no participating dispatch buses in study system.
	IERR = 8	no participating dispatch buses in opposing system.
	IERR = 9	subsystem engine error.
	IERR = 10	Unable to implement transfer in study system.
	IERR = 11	Unable to implement transfer in opposing system.
	IERR = 21	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
	IERR = 22	monitored elements exceed limit when adding multi-section line members.
	IERR = 23	no subsystems defined.
	IERR = 24	subsystem not found.
	IERR = 25	specified subsystems overlap.
	IERR = 26	no in-service buses in subsystem WORLD.
	IERR = 50	error opening Economic Dispatch Data File.
	IERR = 52	error opening Distribution Factor Data File.
	IERR = 54	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.101 INLF

This API routine is obsolete. It has been replaced by the API routine INLF_2, and is implemented by a call to the INLF_2 API routine.

Use this API to apply an inertial or governor response power flow calculation (activity INLF).

Batch command syntax:

```
BAT_INLF  OPTIONS(1)...OPTIONS(7)  IFILE
```

Python syntax:

```
ierr = inlf(options, ifile)
```

Fortran syntax:

```
CALL INLFAPI(OPTIONS, IFILE, IERR)
```

where:

Integer OPTIONS(7)	Is an array of seven elements specifying solution options (input). The values are as follows:
	OPTIONS(1) — solution type (0 by default).
	OPTIONS(1) = 0 — inertial.
	OPTIONS(1) = 1 — governor response.
	OPTIONS(2) — tap adjustment flag (tap adjustment option setting by default).
	OPTIONS(2) = 0 — disable.
	OPTIONS(2) = 1 — enable stepping adjustment.
	OPTIONS(2) = 2 — enable direct adjustment.
	OPTIONS(2) = -1 — solution type default (inertial => disable; governor => stepping).
	OPTIONS(3) — phase shift adjustment flag (phase shift adjustment option setting by default).
	OPTIONS(3) = 0 — disable.
	OPTIONS(3) = 1 — enable.
	OPTIONS(3) = -1 — solution type default (inertial => disable; governor => enable).
	OPTIONS(4) — dc tap adjustment flag (dc tap adjustment option setting by default).
	OPTIONS(4) = 0 — disable.
	OPTIONS(4) = 1 — enable.
	OPTIONS(4) = -1 — solution type default (inertial => disable; governor => enable).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
OPTIONS(5) = 0	disable.
OPTIONS(5) = 1	enable.
OPTIONS(5) = 2	enable continuous mode, disable discrete mode.
OPTIONS(5) = -1	solution type default (inertial => enable; governor => enable).
OPTIONS(6)	generator var limit flag (-1 (inertial) or 99 (governor) by default).
OPTIONS(6) = 0	apply var limits immediately.
OPTIONS(6) = >0	apply var limits on iteration n (or sooner if mismatch gets small).
OPTIONS(6) = -1	ignore var limits.
OPTIONS(7)	missing active power limits flag (only used in governor response power flow) (0 by default).
OPTIONS(7) = 0	pre-PSS®E-29 approach (0.0 to 1.0 pu for machines with no Unit Inertia and Governor Data File record read; working case values for machines with PMAX defaulted on ECDI data record).
OPTIONS(7) = 1	0.0 to 1.0 pu for both categories of machines.
OPTIONS(7) = 2	working case values for both categories of machines.

Character IFILE*260 — Is the name of Unit Inertia and Governor Data File; blank for none (input; blank by default).

Integer IERR — Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid OPTIONS value.
IERR = 2	generators are converted.
IERR = 3	bus type code and series element status inconsistencies.
IERR = 4	error opening IFILE.
IERR = 5	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.102 INLF_2

Use this API routine to run the second release of the inertial and governor response power flow calculation (activity INLF).

Batch command syntax:

```
BAT_INLF_2  OPTIONS(1)...OPTIONS(8)  IFILE
```

Python syntax:

```
ierr = inlf_2(options, ifile)
```

Fortran syntax:

```
CALL INLF_2(OPTIONS, IFILE, IERR)
```

where:

Integer OPTIONS(8)	Is an array of eight elements specifying solution options (input). The values are as follows:	
	OPTIONS(1)	solution type (0 by default).
	OPTIONS(1) = 0	inertial.
	OPTIONS(1) = 1	governor response.
	OPTIONS(2)	tap adjustment flag (tap adjustment option setting by default).
	OPTIONS(2) = 0	disable.
	OPTIONS(2) = 1	enable stepping adjustment.
	OPTIONS(2) = 2	enable direct adjustment.
	OPTIONS(2) = -1	solution type default (inertial => disable; governor => stepping).
	OPTIONS(3)	phase shift adjustment flag (phase shift adjustment option setting by default).
	OPTIONS(3) = 0	disable.
	OPTIONS(3) = 1	enable.
	OPTIONS(3) = -1	solution type default (inertial => disable; governor => enable).
	OPTIONS(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
	OPTIONS(4) = 0	disable.
	OPTIONS(4) = 1	enable.
	OPTIONS(4) = -1	solution type default (inertial => disable; governor => enable).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
OPTIONS(5) = 0	disable.
OPTIONS(5) = 1	enable.
OPTIONS(5) = 2	enable continuous mode, disable discrete mode.
OPTIONS(5) = -1	solution type default (inertial => enable; governor => enable).
OPTIONS(6)	generator var limit flag (-1 (inertial) or 99 (governor) by default).
OPTIONS(6) = 0	apply var limits immediately.
OPTIONS(6) = >0	apply var limits on iteration n (or sooner if mismatch gets small).
OPTIONS(6) = -1	ignore var limits.
OPTIONS(7)	induction motor treatment flag; applied when an induction motor fails to solve due to low terminal voltage (0 by default).
OPTIONS(7) = 0	stall.
OPTIONS(7) = 1	trip.
OPTIONS(8)	missing active power limits flag (only used in governor response power flow) (0 by default).
OPTIONS(8) = 0	pre-PSS®E-29 approach (0.0 to 1.0 pu for machines with no Unit Inertia and Governor Data File record read; working case values for machines with PMAX defaulted on ECDI data record).
OPTIONS(8) = 1	0.0 to 1.0 pu for both categories of machines.
OPTIONS(8) = 2	working case values for both categories of machines.

Character IFILE*260 — Is the name of Unit Inertia and Governor Data File; blank for none (input; blank by default).

Integer IERR — Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid OPTIONS value.
IERR = 2	generators are converted.
IERR = 3	bus type code and series element status inconsistencies.
IERR = 4	error opening IFILE.
IERR = 5	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.103 INTA

Use this API to summarize tie flows between an interchange area and all other areas in the working case (activity INTA).

Batch command syntax:

```
BAT_INTA  SID  ALL
```

Python syntax:

```
ierr = inta(sid, all)
```

Fortran syntax:

```
CALL INTANW(SID, ALL, IERR)
```

where:

Integer SID	Is a valid area subsystem identifier. Valid subsystem identifiers range from 0 to 11. Area subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all areas* or specified subsystem flag (input; 1 by default). ALL = 1 process all areas. ALL = 0 process only areas in area subsystem SID.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.104 INTZ

Use this API to summarize tie flows between each zone and all other zones in the working case (activity INTZ).

Batch command syntax:

```
BAT_INTZ  SID  ALL
```

Python syntax:

```
ierr = intz(sid, all)
```

Fortran syntax:

```
CALL INTZNW(SID, ALL, IERR)
```

where:

Integer SID	Is a valid zone subsystem identifier. Valid subsystem identifiers range from 0 to 11. Zone subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all zones* or specified subsystem flag (input; 1 by default). ALL = 1 process all zones. ALL = 0 process only zones in zone subsystem SID.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.105 JOIN

The API combines two buses into a single bus (activity JOIN).

Batch command syntax:

```
BAT_JOIN   RBUS   JBUS   STATUS
```

Python syntax:

```
ierr = join(rbus, jbus, status)
```

or:

```
ierr,idtrns = join(rbus, jbus, status, nid)
```

Fortran syntax:

```
CALL JOINAPI(RBUS, JBUS, STATUS, NID, IDTRNS, IERR)
```

where:

Integer RBUS	Is the retained bus number (input; no default allowed).
Integer JBUS	Is the eliminated bus number (input; no default allowed).
Integer STATUS	Is the line shunt option (input; 1 by default). STATUS(1) = 0 ignore line shunts of deleted branches. STATUS(1) = 1 add line shunts of deleted branches to shunt of retained bus.
Integer NID	Is the number of elements in the IDTRNS array; 0 to suppress setting IDTRNS (input; 0 by default).
Character IDTRNS*18	Is **returned** as the ID translation array of NID entries (output). Contains information on identifier changes of elements originally connected to bus JBUS. IDTRNS contents:

Columns	Content
1-2	LD for load.
	BR for non-transformer branch.
	IM for induction machine
	MC for machine.
	SH for fixed bus shunt
	T2 for two-winding transformer.
	T3 for three-winding transformer.
3-8	For BR and T2 records, bus number at the other end of the branch.
	For T3 records, bus number of one of the other buses connected by the transformer.
9-14	For T3 records, bus number of the other bus connected by the transformer.
15-16	Old identifier.
17-18	New identifier.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid STATUS value.
	IERR = 2	bus not found.
	IERR = 3	invalid NID value.
	IERR = 4	IDTRNS is not big enough.
	IERR = 5	JOIN processing error.
	IERR = 6	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.106 LAMP

Use this API to print power flow solution output, including loadings in amps, in a traditional power flow report format (activity LAMP). Bus related information is on the left side of the report and branch related information on the right side.

Batch command syntax:

```
BAT_LAMP  SID  ALL
```

Python syntax:

```
ierr = lamp(sid, all)
```

Fortran syntax:

```
CALL LAMPNW(SID, ALL, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all buses.
	ALL = 0	process only buses in subsystem SID.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.107 LINE_SHUNT_REPORTING

Use this API to enable or disable the line shunt reporting option setting (activity OPTN).

Batch command syntax:

```
BAT_LINE_SHUNT_REPORTING  IVAL
```

Python syntax:

```
ierr = line_shunt_reporting(ival)
```

or:

```
ierr,ival = line_shunt_reporting()
```

Fortran syntax:

```
CALL LINE_SHUNT_REPORTING_API(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Is the value of the option setting (input if Batch command or for Python if specified as an argument, or for FORTRAN if IOCODE is 0; else output 0 by default for input).	
	IVAL = 0	no special reporting of line shunts in POUT, LOUT and LAMP.
	IVAL = 1	include special reporting of line shunts in POUT, LOUT and LAMP.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE = 0	set the option setting to the value specified in IVAL.
	IOCODE = 1	return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid IVAL value.
	IERR = 2	invalid IOCODE value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.108 LINES_PER_PAGE

Use this API to specify the values for page length limits (activity OPTN) for the four output devices.

Batch command syntax:

```
BAT_LINES_PER_PAGE  IVAL(1)  IVAL(2)  IVAL(3)  IVAL(4)
```

Python syntax:

```
ierr = lines_per_page(ival)
```

or:

```
ierr,ival,devnam = lines_per_page()
```

Fortran syntax:

```
CALL LINES_PER_PAGE_API(IVAL, DEVNAM, IOCODE, IERR)
```

where:

Integer IVAL(4)	Is an array of four elements containing the value of the option setting (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0.)	
	IVAL(1)	lines per page for disk files (no change by default).
	IVAL(2)	lines per page for the report window (no change by default).
	IVAL(3)	lines per page for the first primary hard copy output device (no change by default).
	IVAL(4)	lines per page for the second primary hard copy output device (no change by default).
Character DEVNAM(4)*40	Is **returned** as an array of four elements set to the device names of the four output devices (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0).	
	DEVNAM(1)	'FILE OUTPUT'.
	DEVNAM(2)	'REPORT WINDOW'.
	DEVNAM(3)	name of first primary hard copy output device.
	DEVNAM(4)	name of second primary hard copy output device.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE = 0	set the option settings to the values specified in IVAL.
	IOCODE = 1	return the current values of the option settings in IVAL.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	negative IVAL value.
	IERR = 2	invalid IOCODE value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.109 LINES_PER_PAGE_ONE_DEVICE

Use this API to specify the values for page length limits (activity OPTN) for one of the four output devices.

Batch command syntax:

```
BAT_LINES_PER_PAGE_ONE_DEVICE  DEVICE  IVAL
```

Python syntax:

```
ierr = lines_per_page_one_device(device, ival)
```

or:

```
ierr,ival,devnam = lines_per_page_one_device(device)
```

Fortran syntax:

```
CALL LINES_PER_PAGE_ONE_DEVICE_API(DEVICE, IVAL, DEVNAM, IOCODE,
IERR)
```

where:

Integer DEVICE	Indicates which of the four output devices is to be processed (input; no default allowed). 1 for disk files. 2 for the report window. 3 for the first primary hard copy output device. 4 for the second primary hard copy output device.
Integer IVAL	Is the value of the lines per page option setting of the output device corresponding to DEVICE (input if Batch command or for Python if specified as an argument, or for FORTRAN if IOCODE is 0; else output no change by default for input).
Character DEVNAM*40	Is **returned** as the device name of the output device corresponding to DEVICE (output if IOCODE is 1, not used if batch command or IOCODE is 0).
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to the value specified in IVAL. IOCODE = 1 return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 negative IVAL value. IERR = 2 invalid IOCODE value. IERR = 3 invalid DEVICE value. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.110 LIST

Use this API routine to tabulate the power flow working case in a form suitable for problem data documentation (activity LIST).

Batch command syntax:

```
BAT_LIST  SID  ALL  OPT  VREV
```

Python syntax:

```
ierr = list(sid, all, opt, vrev)
```

Fortran syntax:

```
CALL LISTNW(SID, ALL, OPT, VREV, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Integer OPT	Is power flow data category to be tabulated (input; 1 by default). = 1, case summary = 2, bus data = 3, switched shunt data = 4, plant data = 5, generator unit data = 6, branch data (single entry) = 7, branch data (double entry) = 8, two-winding transformer impedance data = 9, line shunt data = 10, dc line data = 11, area interchange data = 12, full listing (single entry) = 13, full listing (double entry) = 14, impedance table correction data = 15, multi-section line data = 16, zone data = 17, inter-area transfer data = 18, load data = 19, owner data = 20, FACTS device sending end data = 21, two-winding transformer winding data = 22, three-winding transformer impedance data = 23, three-winding transformer winding data = 24, three-winding transformer control data = 25, three-winding transformer general data = 26, two-winding transformer control data = 27, FACTS device terminal end data = 28, fixed bus shunt data = 29, GNE device data = 30, induction machine circuit data = 31, induction machine miscellaneous data

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer VREV	Is the option to select the units used for displaying voltages (input; 0 by default).	
	VREV = 0	use default voltage units.
	VREV = 1	use alternate voltage units.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	invalid OPT value.
	IERR = 4	invalid VREV value.
	IERR = 5	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.111 ListContingencySavedCases

Use this API routine to obtain a report listing some or all of the system conditions preserved in a ZIP Archive Output File that was created during a previous run of one of the members of the the AC contingency calculation family.

Batch command syntax:

```
BAT_ListContingencySavedCases  PATHZIP  LABEL
```

Python syntax:

```
ierr = ListContingencySavedCases(pathzip, label)
```

Fortran syntax:

```
CALL ListContingencySavedCases(PATHZIP, LABEL, IERR)
```

where:

Character PATHZIP*260	Is the name of the ZIP Archive Output File (input; no default allowed).	
Character LABEL*32	Is a label indicating the system condition(s) that are to be tabulated (input; no default allowed).	
	blank	list all system conditions.
	contingency label	list all system conditions for the specified contingency label.
	'Base Case'	list all system conditions for the base case; use only for ZIP Archive Output Files written by the P-V and Q-V analysis functions.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	error opening the file Names.phy in the ZIP Archive Output File.
	IERR = 2	error reading the file Names.phy in the ZIP Archive Output File.
	IERR = 3	error closing the file Names.phy in the ZIP Archive Output File.
	IERR = 4	internal error.
	IERR = 5	internal error.
	IERR = 6	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.112 LLRF

Use this API to apply the line loading relief calculation using a linear network (dc) model (activity LLRF). It **must** first be called once with APIOPT set to 1, then once for each priority level for which the adjustment method is to be changed with APIOPT set to 2, and finally once with APIOPT set to 3.

Batch command syntax:

```
BAT_LLRF  APIOPT  OPTIONS(1)  OPTIONS(2)  RVAL  DFXFILE
```

Python syntax:

```
ierr = llrf(apiopt, options, rval, dfxfile)
```

Fortran syntax:

```
CALL LLRFAPI(APIOPT, OPTIONS, RVAL, DFXFILE, IERR)
```

where:

Integer APIOPT — Is the mode of operation (input; no default allowed).

APIOPT = 1	initialize.
APIOPT = 2	specify a priority level and the adjustment method to be applied; must be called once for each priority level, adjustment method pair to be specified.
APIOPT = 3	specify a line loading relief function, apply the required calculations, report results, and terminate.

Integer OPTIONS(2) — Is an array of two elements specifying options and input values to be used in the current calculation phase of LLRFAPI; ignored when APIOPT is 1 (input).

When APIOPT is 2, the elements of OPTIONS define a priority level and the adjustment method to be used in curtailing or restoring transactions.

OPTIONS(1)	priority level (no default allowed).
OPTIONS(2)	adjustment method (0 by default).
OPTIONS(2) = 0	first in, last out (FILO).
OPTIONS(2) = 1	decreasing order of distribution factor magnitude (DF-ORDER).
OPTIONS(2) = 2	distribution factor pro rata (DF-PRO RATA).
OPTIONS(2) = 3	pro rata based on the product of distribution factor and transaction schedule (DF-SCHED-PRO RATA).

When APIOPT is 3, the first element of OPTIONS defines the line loading relief function to be applied. For the *curtail transactions* and *restore transactions* functions, the second element of OPTIONS defines the monitored element to be used as the transaction adjustment target.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(1)	line loading relief function (0 by default).
OPTIONS(1) = 0	curtail transactions.
OPTIONS(1) = 1	restore transactions.
OPTIONS(1) = 2	report distribution factor matrix.
OPTIONS(2)	monitored element list index (1 by default when OPTIONS(1) is 0 or 1).

Real RVAL

Is a single value to be used in the current calculation phase of LLRFAPI; ignored when APIOPT is 2 (input).

When APIOPT is 1, RVAL is the distribution factor tolerance used to preclude those transactions with negligible effect on the target monitored element from the transaction adjustment functions (0.05 by default).

When APIOPT is 3, RVAL is the target element adjustment MW (10.0 by default).

Character DFXFILE*260

Is the name of Distribution Factor Data File when APIOPT is 1 (input; no default allowed when APIOPT is 1).

Integer IERR

Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	no transactions defined.
IERR = 2	no Distribution Factor Data File specified.
IERR = 3	invalid OPTIONS value.
IERR = 4	invalid RVAL value.
IERR = 5	unexpected APIOPT value.
IERR = 6	invalid APIOPT value.
IERR = 7	error opening file DFXFILE.
IERR = 8	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
IERR = 10	no monitored branches or interfaces defined.
IERR = 11	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.113 LOAD_REDUCTION

Use this API to tabulate the amount of load reduction in a specified subsystem due to the voltage at the bus to which the load is connected being below PQBRAK (for constant MVA load) or 0.5 (for constant current load.

Batch command syntax:

```
BAT_LOAD_REDUCTION  SID  ALL  AOZOPT
```

Python syntax:

```
ierr = load_reduction(sid, all, aozopt)
```

Fortran syntax:

```
CALL LOAD_REDUCTION(SID, ALL, AOZOPT, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all loads.
	ALL = 0	process only loads in subsystem SID.
Integer AOZOPT	Is the subsystem selection option when area, owner and/or zone are included in the subsystem selection criteria (input; 1 by default).	
	AOZOPT = 1	use the area, owner and/or zone assignments of the bus to which the load is connected.
	AOZOPT = 2	use the area, owner and/or zone assignments of the load.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	invalid AOZOPT value.
	IERR = 4	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.114 LOUT

Use this API to print the power flow solution results in a traditional power flow report format (activity LOUT). Bus related information is on the left side of the report and branch related information on the right side.

Batch command syntax:

```
BAT_LOUT  SID  ALL
```

Python syntax:

```
ierr = lout(sid, all)
```

Fortran syntax:

```
CALL LOUTNW(SID, ALL, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all buses.
	ALL = 0	process only buses in subsystem SID.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.115 LTAP

Use this API to insert a bus at a designated location along a line (activity LTAP).

Batch command syntax:

```
BAT_LTAP  FRMBUS  TOBUS  'CKT'  FRACTION  NEWNUM  'NEWNAM'  NEWKV
```

Python syntax:

```
ierr = ltap(frmbus, tobus, ckt, fraction, newnum, newnam, newkv)
```

Fortran syntax:

```
CALL LTAPAPI(FRMBUS, TOBUS, CKT, FRACTION, NEWNUM, NEWNAM, NEWKV,
IERR)
```

where:

Integer FRMBUS	Is one of the buses connected by the branch to be tapped (input; no default allowed).
Integer TOBUS	Is the other bus connected by the branch to be tapped (input; no default allowed).
Character CKT*2	Is the circuit identifier of the branch to be tapped (input; '1' by default).
Real FRACTION	New bus location relative to bus FRMBUS in per unit of total line length (input; 0.5 by default).
Integer NEWNUM	Is the number to be assigned to the new bus (input; no default allowed).
Character NEWNAM*12	Is the name to be assigned to the new bus (input; blank by default).
Real NEWKV	Is the base voltage to be assigned to the new bus (input; base voltage of bus FRMBUS by default).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 bus not found.
	IERR = 2 branch not found.
	IERR = 3 branch is a transformer.
	IERR = 4 invalid FRACTION value.
	IERR = 5 invalid NEWNUM value.
	IERR = 6 bus number is already assigned to another bus.
	IERR = 7 invalid NEWKV value.
	IERR = 8 bus, branch, or branch ownership tables full.
	IERR = 9 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.116 MACCC

This API routine is obsolete. It has been replaced by the API routine MACCC_2, and is implemented by a call to MACCC_2.

Use this API to run multiple level contingency analysis. User-specified and automatically-selected contingencies from built-in contingency ranking functions are evaluated individually and in combination with each other as overlapping outages up to the specified evaluation level.

Batch command syntax:

```
BAT_MACCC  TOL  OPTACC(1)...OPTACC(8)  OPTMAC(1)...OPTMAC(41)
OPTRNK(1)...OPTRNK(17)  VALUES(1)...VALUES(3)  LABELS(1)  LABELS(2)
DFXFILE  ACCFILE  THRFILE  INLFILE
```

Python syntax:

```
ierr = maccc(tol, optacc, optmac, optrnk, values, labels, dfxfile,
accfile, thrfile, inlfile)
```

Fortran syntax:

```
CALL MACCCAPI(TOL, OPTACC, OPTMAC, OPTRNK, VALUES, LABELS, DFXFILE,
ACCFILE, THRFILE, INLFILE, IERR)
```

where:

Real TOL	Is the mismatch tolerance (input; Newton solution convergence tolerance, TOLN, by default).
Integer OPTACC(8)	Is an array of eight elements specifying power flow solution options (input). The values are as follows:
	OPTACC(1) — tap adjustment flag (tap adjustment option setting by default).
	OPTACC(1) = 0 — disable.
	OPTACC(1) = 1 — enable stepping adjustment.
	OPTACC(1) = 2 — enable direct adjustment.
	OPTACC(2) — area interchange adjustment flag (area interchange adjustment option setting by default).
	OPTACC(2) = 0 — disable.
	OPTACC(2) = 1 — enable using tie line flows only in calculating area interchange.
	OPTACC(2) = 2 — enable using tie line flows and loads in calculating area interchange.
	OPTACC(3) — phase shift adjustment flag (phase shift adjustment option setting by default).
	OPTACC(3) = 0 — disable.
	OPTACC(3) = 1 — enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTACC(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
OPTACC(4) = 0	disable.
OPTACC(4) = 1	enable.
OPTACC(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
OPTACC(5) = 0	disable.
OPTACC(5) = 1	enable.
OPTACC(5) = 2	enable continuous mode, disable discrete mode.
OPTACC(6)	solution method flag (0 by default).
OPTACC(6) = 0	FDNS.
OPTACC(6) = 1	FNSL.
OPTACC(6) = 2	optimized FDNS.
OPTACC(7)	non-divergent solution flag (non-divergent solution option setting by default).
OPTACC(7) = 0	disable.
OPTACC(7) = 1	enable.
OPTACC(8)	dispatch mode (0 by default).
OPTACC(8) = 0	disable.
OPTACC(8) = 1	subsystem machines (reserve).
OPTACC(8) = 2	subsystem machines (pmax).
OPTACC(8) = 3	subsystem machines (inertia).
OPTACC(8) = 4	subsystem machines (governor droop).

Integer OPTMAC(41) Is an array of 41 elements to specifying contingency evaluation level and combinations (input). The values are as follows:

OPTMAC(1)	run multiple contingency analysis (1 by default).
OPTMAC(1) = 0	disable.
OPTMAC(1) = 1	enable.
OPTMAC(2)	contingency evaluation level (1<=OPTMAC(2)<=3; 1 by default).
OPTMAC(3-41)	each element represents a type of contingency combinations in the following order. =1, include the contingency combinations, otherwise =0 (0 by default).

{S, U, B, SS, SU, SB, US, UU, UB, BS, BU, BB, SSS, SSB, SSU, SBS, SBB, SBU, SUS, SUB, SUU, USS, USU, USB, UUS, UUB, UUU, UBS, UBU, UBB, BSS, BSU, BSB, SUS, SUB, SUU, BBS, BBU, BBB}, where S, U and B are user specified cases, single machine and single branch outages respectively.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer OPTRNK(17) Is an array of 17 elements to specifying ranking options (input). The values are as follows:

OPTRNK(1)	include ties from contingency subsystem (0 by default).
OPTRNK(1) = 0	disable.
OPTRNK(1) = 1	enable.
OPTRNK(2)	contingency rating (rating set program option setting by default).
OPTRNK(2) = 1	rate A.
OPTRNK(2) = 2	rate B.
OPTRNK(2) = 3	rate C.
OPTRNK(3)	convert ratings to MW (0 by default).
OPTRNK(3) = 0	disable.
OPTRNK(3) = 1	enable.
OPTRNK(4)	include machine contingencies from overload ranking (0 by default).
OPTRNK(4) = 0	disable.
OPTRNK(4) = 1	enable.
OPTRNK(5)	minimum of machine contingencies to evaluate (>=0; 1 by default).
OPTRNK(6)	maximum of machine contingencies to evaluate (>=0; 10 by default).
OPTRNK(7)	non-failure cutoff of machine contingencies (>=0; 5 by default).
OPTRNK(8)	include branch contingencies from overload ranking (0 by default).
OPTRNK(8) = 0	disable.
OPTRNK(8) = 1	enable.
OPTRNK(9)	minimum of branch contingencies from overload ranking to evaluate (>=0; 1 by default).
OPTRNK(10)	maximum of branch contingencies from overload ranking to evaluate (>=0; 10 by default).
OPTRNK(11)	non-failure cutoff of branch contingencies from overload ranking (>=0; 5 by default).
OPTRNK(12)	include branch contingencies from voltage depression ranking (0 by default).
OPTRNK(12) = 0	disable.
OPTRNK(12) = 1	enable.
OPTRNK(13)	minimum of branch contingencies from voltage depression ranking to evaluate (>=0; 1 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTRNK(14)	maximum of branch contingencies from voltage depression ranking to evaluate (>=0; 10 by default).
	OPTRNK(15)	non-failure cutoff of branch contingencies from voltage depression ranking (>=0; 5 by default).
	OPTRNK(16)	include all branches in voltage depression ranking (0 by default).
	OPTRNK(16) = 0	include only with non-zero rating.
	OPTRNK(16) = 1	all.
	OPTRNK(17)	include contingencies that create swingless island (0 by default).
	OPTRNK(17) = 0	disable.
	OPTRNK(17) = 1	enable.
Real VALUES(3)	Is an array of three elements to specifying ranking parameters (input).The values are as follows:	
	VALUES(1)	small reactance threshold (0<= VALUES(1)<=0.005; 0.0005 by default).
	VALUES(2)	percent of rating set (0<= VALUES(1)<=200.0; 100.0 by default).
	VALUES(3)	MW threshold for islanding contingencies (>=0.0; 0.0 by default).
Character LABELS(2)*12	Is an array of two elements specifying subsystem labels. The values are as follows:	
	LABELS (1)	label of generation dispatch subsystem (blank by default; no default allowed if OPTACC(8) is not 0).
	LABELS (2)	label of contingency subsystem (blank by default).
Character DFXFILE*260	Is the name of the Distribution Factor Data File (input; no default allowed).	
Character ACCFILE*260	Is the name of the Contingency Solution Output File (input; no default allowed).	
Character THRFILE*260	Is the name of the Load Throwover Data File (input; blank by default).	
Character INLFILE*260	Is the name of the Unit Inertia and Governor Data File (input; blank by default).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid TOL value.
IERR = 2	invalid OPTACC value.
IERR = 3	generators are converted.
IERR = 4	buses in island(s) without a swing bus; use activity TREE.
IERR = 5	largest mismatch exceeds mismatch tolerance.
IERR = 6	generation subsystem is not defined.
IERR = 7	too many islands in base case.
IERR = 8	invalid OPTMAC value.
IERR = 9	invalid OPTRNK value.
IERR = 10	invalid VALUES value.
IERR = 21	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
IERR = 22	monitored elements exceed limit when adding multi-section line members.
IERR = 50	error opening Unit Inertia and Governor Data File.
IERR = 51	error opening Contingency Solution Output File.
IERR = 52	error opening Distribution Factor Data File.
IERR = 53	error opening Load Throwover Data File.
IERR = 54	prerequisite requirements for function not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.117 MACCC_2

Use this API to run the second release of multiple level contingency analysis. User-specified and automatically-selected contingencies from built-in contingency ranking functions are evaluated individually and in combination with each other as overlapping outages up to the specified evaluation level.

Batch command syntax:

```
BAT_MACCC_2  TOL  OPTACC(1)...OPTACC(10)  OPTMAC(1)...OPTMAC(41)
OPTRNK(1)...OPTRNK(17)  VALUES(1)...VALUES(3)  LABELS(1)  LABELS(2)
DFXFILE  ACCFILE  THRFILE  INLFILE
```

Python syntax:

```
ierr = maccc_2(tol, optacc, optmac, optrnk, values, labels, dfxfile,
accfile, thrfile, inlfile)
```

Fortran syntax:

```
CALL MACCCAPI_2(TOL, OPTACC, OPTMAC, OPTRNK, VALUES, LABELS, DFXFILE,
ACCFILE, THRFILE, INLFILE, IERR)
```

where:

Real TOL	Is the mismatch tolerance (input; Newton solution convergence tolerance, TOLN, by default).
Integer OPTACC(10)	Is an array of ten elements specifying power flow solution options (input). The values are as follows:
	OPTACC(1) — tap adjustment flag (tap adjustment option setting by default).
	OPTACC(1) = 0 — disable.
	OPTACC(1) = 1 — enable stepping adjustment.
	OPTACC(1) = 2 — enable direct adjustment.
	OPTACC(2) — area interchange adjustment flag (area interchange adjustment option setting by default).
	OPTACC(2) = 0 — disable.
	OPTACC(2) = 1 — enable using tie line flows only in calculating area interchange.
	OPTACC(2) = 2 — enable using tie line flows and loads in calculating area interchange.
	OPTACC(3) — phase shift adjustment flag (phase shift adjustment option setting by default).
	OPTACC(3) = 0 — disable.
	OPTACC(3) = 1 — enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTACC(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
OPTACC(4) = 0	disable.
OPTACC(4) = 1	enable.
OPTACC(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
OPTACC(5) = 0	disable.
OPTACC(5) = 1	enable.
OPTACC(5) = 2	enable continuous mode, disable discrete mode.
OPTACC(6)	solution method flag (0 by default).
OPTACC(6) = 0	FDNS.
OPTACC(6) = 1	FNSL.
OPTACC(6) = 2	optimized FDNS.
OPTACC(7)	non-divergent solution flag (non-divergent solution option setting by default).
OPTACC(7) = 0	disable.
OPTACC(7) = 1	enable.
OPTACC(8)	induction motor treatment flag (applied when an induction motor fails to solve due to low terminal bus voltage, 0 by default)
OPTACC(8) = 0	stall.
OPTACC(8) = 1	trip.
OPTACC(9)	induction machine failure flag(0 by default)
OPTACC(9) = 0	treat contingency as non-converged if any induction machines are placed in the “stalled” or “tripped” state.
OPTACC(9) = 1	treat contingency as solved if it converges, even if any induction machines are placed in the “stalled” or “tripped” state.
OPTACC(10)	dispatch mode (0 by default).
OPTACC(10) = 0	disable.
OPTACC(10) = 1	subsystem machines (reserve).
OPTACC(10) = 2	subsystem machines (pmax).
OPTACC(10) = 3	subsystem machines (inertia).
OPTACC(10) = 4	subsystem machines (governor droop).

Integer OPTMAC(41)

Is an array of 41 elements to specifying contingency evaluation level and combinations (input). The values are as follows:

OPTMAC(1)	run multiple contingency analysis (1 by default).
OPTMAC(1) = 0	disable.
OPTMAC(1) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTMAC(2)	contingency evaluation level (1 by default)
OPTMAC(2)=1	N-1 contingency analysis
OPTMAC(2)=2 or -2	N-2 contingency analysis
OPTMAC(2)=3 or -3	N-3 contingency analysis

If it is 2 or 3, a newly ordered event list is built for the next level after solving AC power flow each case at current level; if it is -2 or -3, the event list at the first level is used for any higher levels.

OPTMAC(3-41)	each element represents a type of contingency combinations in the following order. =1, include the contingency combinations, otherwise =0 (0 by default).

{S, U, B, SS, SU, SB, US, UU, UB, BS, BU, BB, SSS, SSB, SSU, SBS, SBB, SBU, SUS, SUB, SUU, USS, USU, USB, UUS, UUB, UUU, UBS, UBU, UBB, BSS, BSU, BSB, SUS, SUB, SUU, BBS, BBU, BBB}, where S, U and B are user specified cases, single machine and single branch outages respectively.

Integer OPTRNK(17) Is an array of 17 elements to specifying ranking options (input). The values are as follows:

OPTRNK(1)	include ties from contingency subsystem (0 by default).
OPTRNK(1) = 0	disable.
OPTRNK(1) = 1	enable.
OPTRNK(2)	contingency rating (rating set program option setting by default).
OPTRNK(2) = 1	rate A.
OPTRNK(2) = 2	rate B.
OPTRNK(2) = 3	rate C.
OPTRNK(3)	convert ratings to MW (0 by default).
OPTRNK(3) = 0	disable.
OPTRNK(3) = 1	enable.
OPTRNK(4)	include machine contingencies from overload ranking (0 by default).
OPTRNK(4) = 0	disable.
OPTRNK(4) = 1	enable.
OPTRNK(5)	minimum of machine contingencies to evaluate (>=0; 1 by default).
OPTRNK(6)	maximum of machine contingencies to evaluate (>=0; 10 by default).
OPTRNK(7)	non-failure cutoff of machine contingencies (>=0; 5 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTRNK(8)	include branch contingencies from overload ranking (0 by default).
OPTRNK(8) = 0	disable.
OPTRNK(8) = 1	enable.
OPTRNK(9)	minimum of branch contingencies from overload ranking to evaluate (>=0; 1 by default).
OPTRNK(10)	maximum of branch contingencies from overload ranking to evaluate (>=0; 10 by default).
OPTRNK(11)	non-failure cutoff of branch contingencies from overload ranking (>=0; 5 by default).
OPTRNK(12)	include branch contingencies from voltage depression ranking (0 by default).
OPTRNK(12) = 0	disable.
OPTRNK(12) = 1	enable.
OPTRNK(13)	minimum of branch contingencies from voltage depression ranking to evaluate (>=0; 1 by default).
OPTRNK(14)	maximum of branch contingencies from voltage depression ranking to eval-uate (>=0; 10 by default).
OPTRNK(15)	non-failure cutoff of branch contingencies from voltage depression ranking (>=0; 5 by default).
OPTRNK(16)	include all branches in voltage depression ranking (0 by default).
OPTRNK(16) = 0	include only with non-zero rating.
OPTRNK(16) = 1	all.
OPTRNK(17)	include contingencies that create swingless island (0 by default).
OPTRNK(17) = 0	disable.
OPTRNK(17) = 1	enable.

Real VALUES(3) Is an array of three elements to specifying ranking parameters (input).The values are as follows:

VALUES(1)	small reactance threshold (0<= VALUES(1)<=0.005; 0.0005 by default).
VALUES(2)	percent of rating set (0<= VALUES(1)<=200.0; 100.0 by default).
VALUES(3)	MW threshold for islanding contingencies (>=0.0; 0.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character LABELS(2)*12	Is an array of two elements specifying subsystem labels. The values are as follows: LABELS (1) — label of generation dispatch subsystem (blank by default; no default allowed if OPTACC(8) is not 0). LABELS (2) — label of contingency subsystem (blank by default).
Character DFXFILE*260	Is the name of the Distribution Factor Data File (input; no default allowed).
Character ACCFILE*260	Is the name of the Contingency Solution Output File (input; no default allowed).
Character THRFILE*260	Is the name of the Load Throwover Data File (input; blank by default).
Character INLFILE*260	Is the name of the Unit Inertia and Governor Data File (input; blank by default).
Integer IERR	Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid TOL value.
IERR = 2	invalid OPTACC value.
IERR = 3	generators are converted.
IERR = 4	buses in island(s) without a swing bus; use activity TREE.
IERR = 5	largest mismatch exceeds mismatch tolerance.
IERR = 6	generation subsystem is not defined.
IERR = 7	too many islands in base case.
IERR = 8	in-service induction machines are in the “stalled” or “tripped” state.
IERR = 9	buses with bus type code and series element status inconsistencies.
IERR = 10	invalid OPTMAC value.
IERR = 11	invalid OPTRNK value.
IERR = 12	invalid VALUES value.
IERR = 21	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
IERR = 22	monitored elements exceed limit when adding multi-section line members.
IERR = 50	error opening Unit Inertia and Governor Data File.
IERR = 51	error opening Contingency Solution Output File.
IERR = 52	error opening Distribution Factor Data File.
IERR = 53	error opening Load Throwover Data File.
IERR = 54	prerequisite requirements for function not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.118 MACCC_PARALLEL

Use this API to run the multiple level contingency analysis in parallel when the number of contingency processors in Program Settings is greater than 1. User-specified and automatically-selected contingencies from built-in contingency ranking functions are evaluated individually and in combination with each other as overlapping outages up to the specified evaluation level.

Batch command syntax:

```
BAT_MACCC_PARALLEL  TOL  OPTACC(1)...OPTACC(10)
OPTMAC(1)...OPTMAC(41)  OPTRNK(1)...OPTRNK(17)  VALUES(1)...VALUES(3)
LABELS(1)  LABELS(2) DFXFILE  ACCFILE  THRFILE  INLFILE
```

Python syntax:

```
ierr = maccc_parallel(tol, optacc, optmac, optrnk, values, labels,
dfxfile, accfile, thrfile, inlfile)
```

Fortran syntax:

```
CALL MACCCAPI_PARALLEL(TOL, OPTACC, OPTMAC, OPTRNK, VALUES, LABELS,
DFXFILE, ACCFILE, THRFILE, INLFILE, IERR)
```

where:

Real TOL	Is the mismatch tolerance (input; Newton solution convergence tolerance, TOLN, by default).
Integer OPTACC(10)	Is an array of ten elements specifying power flow solution options (input). The values are as follows:

OPTACC(1)	tap adjustment flag (tap adjustment option setting by default).
OPTACC(1) = 0	disable.
OPTACC(1) = 1	enable stepping adjustment.
OPTACC(1) = 2	enable direct adjustment.
OPTACC(2)	area interchange adjustment flag (area interchange adjustment option setting by default).
OPTACC(2) = 0	disable.
OPTACC(2) = 1	enable using tie line flows only in calculating area interchange.
OPTACC(2) = 2	enable using tie line flows and loads in calculating area interchange.
OPTACC(3)	phase shift adjustment flag (phase shift adjustment option setting by default).
OPTACC(3) = 0	disable.
OPTACC(3) = 1	enable.
OPTACC(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
OPTACC(4) = 0	disable.
OPTACC(4) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTACC(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
OPTACC(5) = 0	disable.
OPTACC(5) = 1	enable.
OPTACC(5) = 2	enable continuous mode, disable discrete mode.
OPTACC(6)	solution method flag (0 by default).
OPTACC(6) = 0	FDNS.
OPTACC(6) = 1	FNSL.
OPTACC(6) = 2	optimized FDNS.
OPTACC(7)	non-divergent solution flag (non-divergent solution option setting by default).
OPTACC(7) = 0	disable.
OPTACC(7) = 1	enable.
OPTACC(10)	dispatch mode (0 by default).
OPTACC(10) = 0	disable.
OPTACC(10) = 1	subsystem machines (reserve).
OPTACC(10) = 2	subsystem machines (pmax).
OPTACC(10) = 3	subsystem machines (inertia).
OPTACC(10) = 4	subsystem machines (governor droop).

Integer OPTMAC(41)

Is an array of 41 elements to specifying contingency evaluation level and combinations (input). The values are as follows:

OPTMAC(1)	run multiple contingency analysis (1 by default).
OPTMAC(1) = 0	disable.
OPTMAC(1) = 1	enable.
OPTMAC(2)	contingency evaluation level.
OPTMAC(2)=1	N-1 contingency analysis
OPTMAC(2)=-2	N-2 contingency analysis
OPTMAC(2)=-3	N-3 contingency analysis When the option is -2, or -3, the event list at the first level is used for any higher levels. (1 by default).
OPTMAC(3-41)	each element represents a type of contingency combinations in the following order. =1, include the contingency combinations, otherwise =0 (0 by default).

{S, U, B, SS, SU, SB, US, UU, UB, BS, BU, BB, SSS, SSB, SSU, SBS, SBB, SBU, SUS, SUB, SUU, USS, USU, USB, UUS, UUB, UUU, UBS, UBU, UBB, BSS, BSU, BSB, SUS, SUB, SUU, BBS, BBU, BBB}, where S, U and B are user specified cases, single machine and single branch outages respectively.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer OPTRNK(17) Is an array of 17 elements to specifying ranking options (input). The values are as follows:

OPTRNK(1)	include ties from contingency subsystem (0 by default).
OPTRNK(1) = 0	disable.
OPTRNK(1) = 1	enable.
OPTRNK(2)	contingency rating (rating set program option setting by default).
OPTRNK(2) = 1	rate A.
OPTRNK(2) = 2	rate B.
OPTRNK(2) = 3	rate C.
OPTRNK(3)	convert ratings to MW (0 by default).
OPTRNK(3) = 0	disable.
OPTRNK(3) = 1	enable.
OPTRNK(4)	include machine contingencies from overload ranking (0 by default).
OPTRNK(4) = 0	disable.
OPTRNK(4) = 1	enable.
OPTRNK(5)	minimum of machine contingencies to evaluate (>=0; 1 by default).
OPTRNK(6)	maximum of machine contingencies to evaluate (>=0; 10 by default).
OPTRNK(7)	non-failure cutoff of machine contingencies (>=0; 5 by default).
OPTRNK(8)	include branch contingencies from overload ranking (0 by default).
OPTRNK(8) = 0	disable.
OPTRNK(8) = 1	enable.
OPTRNK(9)	minimum of branch contingencies from overload ranking to evaluate (>=0; 1 by default).
OPTRNK(10)	maximum of branch contingencies from overload ranking to evaluate (>=0; 10 by default).
OPTRNK(11)	non-failure cutoff of branch contingencies from overload ranking (>=0; 5 by default).
OPTRNK(12)	include branch contingencies from voltage depression ranking (0 by default).
OPTRNK(12) = 0	disable.
OPTRNK(12) = 1	enable.
OPTRNK(13)	minimum of branch contingencies from voltage depression ranking to evaluate (>=0; 1 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTRNK(14)	maximum of branch contingencies from voltage depression ranking to eval-uate (>=0; 10 by default).
	OPTRNK(15)	non-failure cutoff of branch contingencies from voltage depression ranking (>=0; 5 by default).
	OPTRNK(16)	include all branches in voltage depression ranking (0 by default).
	OPTRNK(16) = 0	include only with non-zero rating.
	OPTRNK(16) = 1	all.
	OPTRNK(17)	include contingencies that create swingless island (0 by default).
	OPTRNK(17) = 0	disable.
	OPTRNK(17) = 1	enable.
Real VALUES(3)	Is an array of three elements to specifying ranking parameters (input).The values are as follows:	
	VALUES(1)	small reactance threshold (0<= VALUES(1)<=0.005; 0.0005 by default).
	VALUES(2)	percent of rating set (0<= VALUES(1)<=200.0; 100.0 by default).
	VALUES(3)	MW threshold for islanding contingencies (>=0.0; 0.0 by default).
Character LABELS(2)*12	Is an array of two elements specifying subsystem labels. The values are as follows:	
	LABELS (1)	label of generation dispatch subsystem (blank by default; no default allowed if OPTACC(8) is not 0).
	LABELS (2)	label of contingency subsystem (blank by default).
Character DFXFILE*260	Is the name of the Distribution Factor Data File (input; no default allowed).	
Character ACCFILE*260	Is the name of the Contingency Solution Output File (input; no default allowed).	
Character THRFILE*260	Is the name of the Load Throwover Data File (input; blank by default).	
Character INLFILE*260	Is the name of the Unit Inertia and Governor Data File (input; blank by default).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid TOL value.
IERR = 2	invalid OPTACC value.
IERR = 3	generators are converted.
IERR = 4	buses in island(s) without a swing bus; use activity TREE.
IERR = 5	largest mismatch exceeds mismatch tolerance.
IERR = 6	generation subsystem is not defined.
IERR = 7	too many islands in base case.
IERR = 8	in-service induction machines are in the “stalled” or “tripped” state.
IERR = 9	buses with bus type code and series element status inconsistencies.
IERR = 10	invalid OPTMAC value.
IERR = 11	invalid OPTRNK value.
IERR = 12	invalid VALUES value.
IERR = 21	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
IERR = 22	monitored elements exceed limit when adding multi-section line members.
IERR = 50	error opening Unit Inertia and Governor Data File.
IERR = 51	error opening Contingency Solution Output File.
IERR = 52	error opening Distribution Factor Data File.
IERR = 53	error opening Load Throwover Data File.
IERR = 54	prerequisite requirements for function not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.119 MACCC_TRIP_COR

Use this API to run multiple AC contingency analysis with tripping simulations and corrective actions. The user-specified and automatically-selected contingencies are evaluated individually and in combination with each other as overlapping contingencies. When there are violations in post-tripping simulation solutions, the corrective actions analysis will be launched to eliminate the violations, with the objective of minimizing the controls adjustments, including load shedding, MW generation dispatch and phase shifter angle adjustments.

Batch command syntax:

```
BAT_MACCC_TRIP_COR  TOL OPTACC(1)...OPTACC(8) OPTMAC(1)...OPTMAC(41)
OPTRNK(1)...OPTRNK(17) OPTTRP(1)...OPTTRP(3) OPTCOR(1)...OPTCOR(5)
VALUES(1)...VALUES(8)  LABELS(1)...LABELS(5)  DFXFILE  ACCFILE
THRFILE  INLFILE TRPFILE
```

Python syntax:

```
ierr = maccc_trip_cor(tol, optacc, optmac, optrnk, opttrp, optcor,
values, labels, dfxfile, accfile, thrfile, inlfile, trpfile)
```

Fortran syntax:

```
CALL MACCC_TRIP_COR(TOL, OPTACC, OPTMAC, OPTRNK, OPTTRP, OPTCOR,
VALUES, LABELS, DFXFILE, ACCFILE, THRFILE, INLFILE, TRPFILE, IERR)
```

where:

Real TOL	Is the mismatch tolerance (input; Newton solution convergence tolerance, TOLN, by default).
Integer OPTACC(8)	Is an array of eight elements specifying power flow solution options (input). The values are as follows:

OPTACC(1)	tap adjustment flag (tap adjustment option setting by default).
OPTACC(1) = 0	disable.
OPTACC(1) = 1	enable stepping adjustment.
OPTACC(1) = 2	enable direct adjustment.
OPTACC(2)	area interchange adjustment flag (area interchange adjustment option setting by default).
OPTACC(2) = 0	disable.
OPTACC(2) = 1	enable using tie line flows only in calculating area interchange.
OPTACC(2) = 2	enable using tie line flows and loads in calculating area interchange.
OPTACC(3)	phase shift adjustment flag (phase shift adjustment option setting by default).
OPTACC(3) = 0	disable.
OPTACC(3) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTACC(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
OPTACC(4) = 0	disable.
OPTACC(4) = 1	enable.
OPTACC(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
OPTACC(5) = 0	disable.
OPTACC(5) = 1	enable.
OPTACC(5) = 2	enable continuous mode, disable discrete mode.
OPTACC(6)	solution method flag (0 by default).
OPTACC(6) = 0	FDNS.
OPTACC(6) = 1	FNSL.
OPTACC(6) = 2	optimized FDNS.
OPTACC(7)	non-divergent solution flag (non-divergent solution option setting by default).
OPTACC(7) = 0	disable.
OPTACC(7) = 1	enable.
OPTACC(8)	dispatch mode (0 by default).
OPTACC(8) = 0	disable.
OPTACC(8) = 1	subsystem machines (reserve).
OPTACC(8) = 2	subsystem machines (pmax).
OPTACC(8) = 3	subsystem machines (inertia).
OPTACC(8) = 4	subsystem machines (governor droop).

Integer OPTMAC(41) Is an array of 41 elements to specifying contingency evaluation level and combinations (input).

OPTMAC(1)	run multiple contingency analysis (1 by default).
OPTMAC(1) = 0	disable.
OPTMAC(1) = 1	enable.
OPTMAC(2)	contingency evaluation level (1<=OPTMAC(2)<=3; 1 by default).
OPTMAC(3-41)	each element represents a type of contingency combinations in the following order. =1, include the contingency combinations, otherwise =0 (0 by default).

{S, U, B, SS, SU, SB, US, UU, UB, BS, BU, BB, SSS, SSB, SSU, SBS, SBB, SBU, SUS, SUB, SUU, USS, USU, USB, UUS, UUB, UUU, UBS, UBU, UBB, BSS, BSU, BSB, SUS, SUB, SUU, BBS, BBU, BBB}, where S, U and B are user specified cases, single machine and single branch outages respectively.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer OPTRNK(17) Is an array of 17 elements to specifying ranking options (input). The values are as follows:

OPTRNK(1)	include ties from contingency subsystem (0 by default).
OPTRNK(1) = 0	disable.
OPTRNK(1) = 1	enable.
OPTRNK(2)	contingency rating (rating set program option setting by default).
OPTRNK(2) = 1	rate A.
OPTRNK(2) = 2	rate B.
OPTRNK(2) = 3	rate C.
OPTRNK(3)	convert ratings to MW (0 by default).
OPTRNK(3) = 0	disable.
OPTRNK(3) = 1	enable.
OPTRNK(4)	include machine contingencies from overload ranking (0 by default).
OPTRNK(4) = 0	disable.
OPTRNK(4) = 1	enable.
OPTRNK(5)	minimum of machine contingencies to evaluate (>=0; 1 by default).
OPTRNK(6)	maximum of machine contingencies to evaluate (>=0; 10 by default).
OPTRNK(7)	non-failure cutoff of machine contingencies (>=0; 5 by default).
OPTRNK(8)	include branch contingencies from overload ranking (0 by default).
OPTRNK(8) = 0	disable.
OPTRNK(8) = 1	enable.
OPTRNK(9)	minimum of branch contingencies from overload ranking to evaluate (>=0; 1 by default).
OPTRNK(10)	maximum of branch contingencies from overload ranking to evaluate (>=0; 10 by default).
OPTRNK(11)	non-failure cutoff of branch contingencies from overload ranking (>=0; 5 by default).
OPTRNK(12)	include branch contingencies from voltage depression ranking (0 by default).
OPTRNK(12) = 0	disable.
OPTRNK(12) = 1	enable.
OPTRNK(13)	minimum of branch contingencies from voltage depression ranking to evaluate (>=0; 1 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTRNK(14)	maximum of branch contingencies from voltage depression ranking to evaluate (>=0; 10 by default).
	OPTRNK(15)	non-failure cutoff of branch contingencies from voltage depression ranking (>=0; 5 by default).
	OPTRNK(16)	include all branches in voltage depression ranking (0 by default).
	OPTRNK(16) = 0	include only with non-zero rating.
	OPTRNK(16) = 1	all.
	OPTRNK(17)	include contingencies that create swingless island (0 by default).
	OPTRNK(17) = 0	disable.
	OPTRNK(17) = 1	enable.
Integer OPTTRP(3)	Is an array of three elements specifying tripping simulation options (input). The values are as follows:	
	OPTTRP(1)	tripping simulation flag (0 by default).
	OPTTRP(1) = 0	disable.
	OPTTRP(1) = 1	enable tripping simulation.
	OPTTRP(2)	number of tripping events per power flow (1 by default).
	OPTTRP(3)	number of power flows per contingency (1 by default).
Integer OPTCOR(5)	Is an array of five elements specifying corrective actions options (input). The values are as follows:	
	OPTCOR(1)	run corrective actions (0 by default)
	OPTCOR(1) = 0	disable.
	OPTCOR(1) = 1	enable.
	OPTCOR(2)	number of power flow solutions (from 0 to 10; 1 by default)
	OPTCOR(3)	generator control flag (1 by default).
	OPTCOR(3) = 0	disable.
	OPTCOR(3) = 1	enable.
	OPTCOR(4)	load control flag (0 by default).
	OPTCOR(4) = 0	disable.
	OPTCOR(4) = 1	enable.
	OPTCOR(5)	phase shifter control flag (1 by default).
	OPTCOR(5) = 0	disable.
	OPTCOR(5) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real VALUES(8)	Is an array of eight elements specifying ranking and corrective action parameters (input). The values are as follows:	
	VALUES(1)	small reactance threshold (0<=VALUES(1)<=0.005; 0.0005 by default).
	VALUES(2)	percent of rating (0<=VALUES(1)<=200.0; 100.0 by default).
	VALUES(3)	MW threshold for islanding contingencies (>=0.0; 0.0 by default).
	VALUES(4)	bus voltage violation tolerance (0.1 by default).
	VALUES(5)	branch flow overload tolerance (0.1 by default).
	VALUES(6)	generator control weighting factor (1.0 by default).
	VALUES(7)	load control weighting factor (1.0 by default).
	VALUES(8)	phase shifter control weighting factor (1.0 by default).
Character LABELS(5)*12	Is an array of five elements specifying subsystem labels (input). The values are as follows:	
	LABELS(1)	label of generation dispatch subsystem (blank by default; no default allowed if OPTACC(8) is not 0).
	LABELS(2)	label of contingency subsystem (blank by default).
	LABELS(3)	label of generator control subsystem (blank by default; no default allowed if OPTCOR(3) is 1).
	LABELS(4)	label of load control subsystem (blank by default; no default allowed if OPTCOR(4) is 1).
	LABELS(5)	label of phase shifter control subsystem (blank by default; no default allowed if OPTCOR(5) is 1).
Character DFXFILE*260	Is the name of the Distribution Factor Data File (input; no default allowed).	
Character ACCFILE*260	Is the name of the Contingency Solution Output File (input; no default allowed).	
Character THRFILE*260	Is the name of the Load Throwover Data File (input; blank by default).	
Character INLFILE*260	Is the name of the Unit Inertia and Governor Data File (input; blank by default).	
Character TRPFILE*260	Is the name of the Tripping Element Data File (input, blank for none; no default allowed if OPTTRP(1) is 1).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid TOL value.
IERR = 2	invalid OPTACC value.
IERR = 3	generators are converted.
IERR = 4	buses in island(s) without a swing bus; use activity TREE.
IERR = 5	largest mismatch exceeds mismatch tolerance.
IERR = 6	generation dispatch subsystem is not defined.
IERR = 7	too many islands in base case.
IERR = 8	invalid OPTMAC value.
IERR = 9	invalid OPTRNK value.
IERR = 10	invalid OPTTRP value.
IERR = 11	invalid OPTCOR value.
IERR = 12	invalid VALUES value.
IERR = 21	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
IERR = 22	monitored elements exceed limit when adding multi-section line members.
IERR = 49	error opening Tripping Element Data File.
IERR = 50	error opening Unit Inertia and Governor Data File.
IERR = 51	error opening Contingency Solution Output File.
IERR = 52	error opening Distribution Factor Data File.
IERR = 53	error opening Load Throwover Data File.
IERR = 54	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.120 MACCC_TRIP_COR_2

This API is obsolete. It has been replaced by API MACCC_TRIP_COR_3, and implemented by a call to MACCC_TRIP_COR_3

This API is the second release of function to run multiple AC contingency analysis with tripping simulations and corrective actions. The user-specified and automatically-selected contingencies are evaluated individually and in combination each other as overlapping contingencies. Following a post contingency solution, trip events are simulated if their tripping conditions are satisfied. Corrective actions are then specified automatically to eliminate the violations in each post tripping solution with the objective of minimizing the control adjustments including load shedding, MW generation dispatch, phase shifter angle adjustments, off-line generator controls, tap setting adjustments and switched shunt controls.

Batch command syntax:

```
BAT_MACCC_TRIP_COR_2 TOL  OPTACC(1)…OPTACC(8), OPTMAC(1)…OPTMAC(41),
OPTRNK(1)…OPTRNK(17), OPTTRP(1)…OPTTRP(3), OPTCOR(1)…OPTCOR(8),
VALUES(1)…VALUES(11),  LABELS(1)…LABELS(8), DFXFILE, ACCFILE,
THRFILE, INLFILE, TRPFILE
```

Python syntax:

```
ierr = maccc_trip_cor_2(tol, optacc, optmac, optrnk, opttrp, optcor,
values, labels, dfxfile, accfile, thrfile, inlfile, trpfile)
```

Fortran syntax:

```
MACCC_TRIP_COR_2(TOL, OPTACC, OPTMAC, OPTRNK, OPTTRP, OPTCOR, VALUES,
LABELS, DFXFILE, ACCFILE, THRFILE, INLFILE, TRPFILE, IERR)
```

where:

Real TOL	Is the mismatch tolerance (input, Newton solution convergence tolerance, TOLN, by default).	
Integer OPTACC(8)	Is an array of eight elements specifying power flow solution (input).	
	OPTACC(1)	tap adjustment flag (tap adjustment option setting by default).
	OPTACC(1) = 0	disable.
	OPTACC(1) = 1	enable stepping adjustment.
	OPTACC(1) = 2	enable direct adjustment.
	OPTACC(2)	area interchange adjustment flag (area interchange adjustment option setting by default).
	OPTACC(2) = 0	disable.
	OPTACC(2) = 1	enable using tie line flows only in calculating area interchange.
	OPTACC(2) = 2	enable using tie line flows and loads in calculating area interchange.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTACC(3)	phase shift adjustment flag (phase shift adjustment option setting by default).
	OPTACC(3) = 0	disable.
	OPTACC(3) = 1	enable.
	OPTACC(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
	OPTACC(4) = 0	disable.
	OPTACC(4) = 1	enable.
	OPTACC(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
	OPTACC(5) = 0	disable.
	OPTACC(5) = 1	enable.
	OPTACC(5) = 2	enable continuous, disable discrete.
	OPTACC(6)	solution method flag (0 by default).
	OPTACC(6) = 0	FDNS.
	OPTACC(6) = 1	FNSL.
	OPTACC(6) = 2	optimized FDNS.
	OPTACC(7)	non-divergent solution flag (non-divergent solution option setting by default).
	OPTACC(7) = 0	disable.
	OPTACC(7) = 1	enable.
	OPTACC(8) = dispatch mode (0 by default).	
	OPTACC(8) = 0	disable.
	OPTACC(8) = 1	subsystem machines (RESERVE).
	OPTACC(8) = 2	subsystem machines (PMAX).
	OPTACC(8) = 3	subsystem machines (INERTIA).
	OPTACC(8) = 4	subsystem machines (GOVERNOR DROOP).
Integer OPTMAC(41)	Is an array of 41 elements to specifying contingency evaluation level and combinations (input).	
	OPTMAC(1)	allow multiple contingency analysis (0 by default).
	OPTMAC(1) =0	disable.
	OPTMAC(1) =1	enable.
	OPTMAC(1) =2	enabled with failure cutoff disabled.
	OPTMAC(2)	contingency evaluation level (1 by default).
	OPTMAC(2) =1	N-1 contingency analysis.
	OPTMAC(2) =2	N-2 contingency analysis.
	OPTMAC(2) =3	N-3 contingency analysis.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTMAC(3-41)each element represents a type of contingency combinations in the following order. =1, include the contingency combinations, otherwise =0 (0 by default). {S, U, B, SS, SU, SB, US, UU, UB, BS, BU, BB, SSS, SSB, SSU, SBS, SBB, SBU, SUS, SUB, SUU, USS, USU, USB, UUS, UUB, UUU, UBS, UBU, UBB, BSS, BSU, BSB, SUS, SUB, SUU, BBS, BBU, BBB}, where S, U and B are user specified cases, single unit and single branch outages respectively.

Integer OPTRNK(17)

Is an array of 17 elements to specifying ranking options (input).

OPTRNK(1)	include ties from subsystem (0 by default).
OPTRNK(1) = 0	disable.
OPTRNK(1) = 1	enable.
OPTRNK(2)	contingency rating (rating set program option setting by default).
OPTRNK(2) = 1	rate A.
OPTRNK(2) = 2	rate B.
OPTRNK(2) = 3	rate C.
OPTRNK(3)	convert ratings to MW (0 by default).
OPTRNK(3) = 0	disable.
OPTRNK(3) = 1	enable.
OPTRNK(4)	machine overload contingencies (0 by default).
OPTRNK(4) = 0	disable.
OPTRNK(4) = 1	enable.
OPTRNK(5)	minimum of machine overload contingencies to evaluate (>=0, 1 by default).
OPTRNK(6)	maximum of machine overload contingencies to evaluate (>=0, 10 by default).
OPTRNK(7)	cutoff of machine overload contingencies to evaluate (>=0, 5 by default).
OPTRNK(8)	line overload contingencies (0 by default).
OPTRNK(8) = 0	disable.
OPTRNK(8) = 1	enable.
OPTRNK(9)	minimum of line overload contingencies to evaluate (>=0, 1 by default).
OPTRNK(10)	maximum of line overload contingencies to evaluate (>=0, 10 by default).
OPTRNK(11)	cutoff of line overload contingencies (>=0, 5 by default).
OPTRNK(12)	voltage collapse line contingencies (0 by default).
OPTRNK(12) = 0	disable.
OPTRNK(12) = 1	enable.
OPTRNK(13)	minimum of voltage collapse line contingencies to evaluate (>=0, 1 by default).
OPTRNK(14)	maximum of voltage collapse line contingencies to evaluate (>=0, 10 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTRNK(15)	cutoff of voltage collapse line contingencies (>=0, 5 be default).
	OPTRNK(16)	include all branches in voltage collapse ranking (0 by default).
	OPTRNK(16) = 0	include only with non-zero rating.
	OPTRNK(16) = 1	all.
	OPTRNK(17)	include contingencies that create swingless island (0 by default).
	OPTRNK(17) = 0	disable.
	OPTRNK(17) = 1	enable.
Integer OPTTRP(3)	Is an array of three elements specifying tripping simulation options:	
	OPTTRP(1)	tripping simulation flag (0 by default).
	OPTTRP(1) = 0	disable.
	OPTTRP(1) = 1	enable tripping simulation.
	OPTTRP(2)	number of tripping events per power flow (1 by default).
	OPTTRP(3)	number of power flows per contingency (1 by default).
Integer OPTCOR(8)	Is an array of eight elements specifying corrective actions options.	
	OPTCOR(1)	run corrective action (0 by default).
	OPTCOR(1) = 0	disable.
	OPTCOR(1) = 1	enable.
	OPTCOR(2)	number power flows (1 by default).
	OPTCOR(3)	generator control flag (1 by default).
	OPTCOR(3) = 0	disable.
	OPTCOR(3) = 1	enable.
	OPTCOR(4)	load control flag (0 by default).
	OPTCOR(4) = 0	disable.
	OPTCOR(4) = 1	enable.
	OPTCOR(5)	phase shifter control flag (1 by default).
	OPTCOR(5) = 0	disable.
	OPTCOR(5) = 1	enable.
	OPTCOR(6)	off-line generator control flag (0 by default).
	OPTCOR(6) = 0	disable.
	OPTCOR(6) = 1	enable.
	OPTCOR(7)	tap setting adjustment flag (0 by default).
	OPTCOR(7) = 0	disable.
	OPTCOR(7) = 1	enable.
	OPTCOR(8)	switched shunt control flag (0 by default).
	OPTCOR(8) = 0	disable.
	OPTCOR(8) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real VALUES(11)	Is the array of eleven elements to specifying ranking parameters and corrective actions parameters (input).	
	VALUES(1)	small reactance threshold, (0<= VALUES(1)<=0.005, 0.0005 by default).
	VALUES(2)	percent of flow rating, (0<= VALUES(2)<=200.0, 100.0 by default).
	VALUES(3)	MW threshold for islanding contingencies(>=0, 0.0 by default).
	VALUES(4)	bus voltage violation tolerance (0.1 by default).
	VALUES(5)	branch overload violation tolerance (0.1 by default).
	VALUES(6)	generator control weighting factor (1.0 by default).
	VALUES(7)	load control weighting factor (1.0 by default).
	VALUES(8)	phase shifter control weighting factor (1.0 by default).
	VALUES(9)	off-line generator control weighting factor (1.0 by default).
	VALUES(10)	tap setting adjustment weighting factor (1.0 by default).
	VALUES(11)	switched shunt control weighting factor (1.0 by default).
Character LABELS(8)*12	Is an array of eight elements specifying subsystem labels.	
	LABELS(1)	label of generation dispatch subsystem (blank by default, no default allowed if OPTACC(8) is not 0).
	LABELS(2)	label of contingency subsystem (blank by default).
	LABELS(3)	label of generator control subsystem (blank by default for none, no default allowed if OPTCOR(3) is 1).
	LABELS(4)	label of load control subsystem (blank by default, no default allowed if OPTCOR(4) is 1).
	LABELS(5)	label of phase shifter control subsystem (blank by default for none, no default allowed if OPTCOR(5) is 1).
	LABELS(6)	off-line generator control subsystem (blank by default; no default allowed if OPTCOR(6) is 1).
	LABELS(7)	tap setting adjustment subsystem (blank by default; no default allowed if OPTCOR(7) is 1).
	LABELS(8)	switched shunt control subsystem (blank by default; no default allowed if OPTCOR(8) is 1).
Character*260 DFXFILE	Is the name of Distribution Factor Data File (input, no default allowed).	
Character*260 ACCFILE	Is the name of Contingency Solution Output File (input, no default allowed).	
Character*260 THRFILE	Is the name of Load Throwover Data File (input, blank for none).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character*260 INLFILE	Is the name of the Unit Inertia and Governor Data File (input; blank for none).
Character TRPFILE*260	Is the name of the Tripping Element Data File (input, blank for none; no default allowed if OPTTRP(1) is 1).
Integer IERR	Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid TOL value.
IERR = 2	invalid OPTACC value.
IERR = 3	generators are converted.
IERR = 4	buses in island(s) without a swing bus; use activity TREE.
IERR = 5	largest mismatch exceeds mismatch tolerance.
IERR = 6	generation subsystem is not defined.
IERR = 7	too many islands in base case.
IERR = 8	invalid OPTMAC value.
IERR = 9	invalid OPTRNK value.
IERR = 10	invalid OPTTRP value.
IERR = 11	invalid OPTCOR value.
IERR = 12	invalid VALUES value.
IERR = 21	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
IERR = 22	monitored elements exceed limit when adding multisection line members.
IERR = 49	error opening Tripping Element Data File.
IERR = 50	error opening Unit Inertia Data File.
IERR = 51	error opening Contingency Solution Output File.
IERR = 52	error opening Distribution Factor Data File.
IERR = 53	error opening Load Throwover Data File.
IERR = 54	prerequisite requirements for function not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.121 MACCC_TRIP_COR_3

This API is the third release of function to run multiple AC contingency analysis with tripping simulations and corrective actions. The user-specified and automatically-selected contingencies are evaluated individually and in combination each other as overlapping contingencies. Following a post contingency solution, trip events are simulated if their tripping conditions are satisfied. Corrective actions are then specified automatically to eliminate the violations in each post tripping solution with the objective of minimizing the control adjustments including load shedding, MW generation dispatch, phase shifter angle adjustments, off-line generator controls, tap setting adjustments and switched shunt controls.

Batch command syntax:

```
BAT_MACCC_TRIP_COR_3 TOL  OPTACC(1)…OPTACC(10), OPTMAC(1)…OPTMAC(41),
OPTRNK(1)…OPTRNK(17), OPTTRP(1)…OPTTRP(3), OPTCOR(1)…OPTCOR(8),
VALUES(1)…VALUES(11),  LABELS(1)…LABELS(8), DFXFILE, ACCFILE, THRFILE,
INLFILE, TRPFILE
```

Python syntax:

```
ierr = maccc_trip_cor_3(tol, optacc, optmac, optrnk, opttrp, optcor,
values, labels, dfxfile, accfile, thrfile, inlfile, trpfile)
```

Fortran syntax:

```
MACCC_TRIP_COR_3(TOL, OPTACC, OPTMAC, OPTRNK, OPTTRP, OPTCOR, VALUES,
LABELS, DFXFILE, ACCFILE, THRFILE, INLFILE, TRPFILE, IERR)
```

where:

Real TOL	Is the mismatch tolerance (input, Newton solution convergence tolerance, TOLN, by default).	
Integer OPTACC(10)	Is an array of ten elements specifying power flow solution (input).	
	OPTACC(1)	tap adjustment flag (tap adjustment option setting by default).
	OPTACC(1) = 0	disable.
	OPTACC(1) = 1	enable stepping adjustment.
	OPTACC(1) = 2	enable direct adjustment.
	OPTACC(2)	area interchange adjustment flag (area interchange adjustment option setting by default).
	OPTACC(2) = 0	disable.
	OPTACC(2) = 1	enable using tie line flows only in calculating area interchange.
	OPTACC(2) = 2	enable using tie line flows and loads in calculating area interchange.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTACC(3)	phase shift adjustment flag (phase shift adjustment option setting by default).
OPTACC(3) = 0	disable.
OPTACC(3) = 1	enable.
OPTACC(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
OPTACC(4) = 0	disable.
OPTACC(4) = 1	enable.
OPTACC(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
OPTACC(5) = 0	disable.
OPTACC(5) = 1	enable.
OPTACC(5) = 2	enable continuous, disable discrete.
OPTACC(6)	solution method flag (0 by default).
OPTACC(6) = 0	FDNS.
OPTACC(6) = 1	FNSL.
OPTACC(6) = 2	optimized FDNS.
OPTACC(7)	non-divergent solution flag (non-divergent solution option setting by default).
OPTACC(7) = 0	disable.
OPTACC(7) = 1	enable.
OPTACC(8)	induction motor treatment flag (applied when an induction motor fails to solve due to low terminal bus voltage, 0 by default)
OPTACC(8) = 0	stall.
OPTACC(8) = 1	trip.
OPTACC(9)	induction machine failure flag(0 by default)
OPTACC(9) = 0	treat contingency as non-converged if any induction machines are placed in the “stalled” or “tripped” state.
OPTACC(9) = 1	treat contingency as solved if it converges, even if any induction machines are placed in the “stalled” or “tripped” state.
OPTACC(10) = dispatch mode (0 by default).	
OPTACC(10) = 0	disable.
OPTACC(10) = 1	subsystem machines (RESERVE).
OPTACC(10) = 2	subsystem machines (PMAX).
OPTACC(10) = 3	subsystem machines (INERTIA).
OPTACC(10) = 4	subsystem machines (GOVERNOR DROOP).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer OPTMAC(41)	Is an array of 41 elements to specifying contingency evaluation level and combinations (input).
	OPTMAC(1) — allow multiple contingency analysis (0 by default).
	OPTMAC(1) =0 — disable.
	OPTMAC(1) =1 — enable.
	OPTMAC(1) =2 — enabled with failure cutoff disabled.
	OPTMAC(2) — contingency evaluation level (1 by default)
	OPTMAC(2)=1 — N-1 contingency analysis
	OPTMAC(2)=2 or -2 — N-2 contingency analysis
	OPTMAC(2)=3 or -3 — N-3 contingency analysis
	If it is 2 or 3, a newly ordered event list is built for the next level after solving AC power flow each case at current level; if it is -2 or -3, the event list at the first level is used for any higher levels.
	OPTMAC(3-41) — each element represents a type of contingency combinations in the following order. =1, include the contingency combinations, otherwise =0 (0 by default).
	{S, U, B, SS, SU, SB, US, UU, UB, BS, BU, BB, SSS, SSB, SSU, SBS, SBB, SBU, SUS, SUB, SUU, USS, USU, USB, UUS, UUB, UUU, UBS, UBU, UBB, BSS, BSU, BSB, SUS, SUB, SUU, BBS, BBU, BBB}, where S, U and B are user specified cases, single unit and single branch outages respectively.
Integer OPTRNK(17)	Is an array of 17 elements to specifying ranking options (input).
	OPTRNK(1) — include ties from subsystem (0 by default).
	OPTRNK(1) = 0 — disable.
	OPTRNK(1) = 1 — enable.
	OPTRNK(2) — contingency rating (rating set program option setting by default).
	OPTRNK(2) = 1 — rate A.
	OPTRNK(2) = 2 — rate B.
	OPTRNK(2) = 3 — rate C.
	OPTRNK(3) — convert ratings to MW (0 by default).
	OPTRNK(3) = 0 — disable.
	OPTRNK(3) = 1 — enable.
	OPTRNK(4) — machine overload contingencies (0 by default).
	OPTRNK(4) = 0 — disable.
	OPTRNK(4) = 1 — enable.
	OPTRNK(5) — minimum of machine overload contingencies to evaluate (>=0, 1 by default).
	OPTRNK(6) — maximum of machine overload contingencies to evaluate (>=0, 10 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTRNK(7)	cutoff of machine overload contingencies to evaluate (>=0, 5 by default).
	OPTRNK(8)	line overload contingencies (0 by default).
	OPTRNK(8) = 0	disable.
	OPTRNK(8) = 1	enable.
	OPTRNK(9)	minimum of line overload contingencies to evaluate (>=0, 1 by default).
	OPTRNK(10)	maximum of line overload contingencies to evaluate (>=0, 10 by default).
	OPTRNK(11)	cutoff of line overload contingencies (>=0, 5 by default).
	OPTRNK(12)	voltage collapse line contingencies (0 by default).
	OPTRNK(12) = 0	disable.
	OPTRNK(12) = 1	enable.
	OPTRNK(13)	minimum of voltage collapse line contingencies to evaluate (>=0, 1 by default).
	OPTRNK(14)	maximum of voltage collapse line contingencies to evaluate (>=0, 10 by default).
	OPTRNK(15)	cutoff of voltage collapse line contingencies (>=0, 5 be default).
	OPTRNK(16)	include all branches in voltage collapse ranking (0 by default).
	OPTRNK(16) = 0	include only with non-zero rating.
	OPTRNK(16) = 1	all.
	OPTRNK(17)	include contingencies that create swingless island (0 by default).
	OPTRNK(17) = 0	disable.
	OPTRNK(17) = 1	enable.
Integer OPTTRP(3)	Is an array of three elements specifying tripping simulation options:	
	OPTTRP(1)	tripping simulation flag (0 by default).
	OPTTRP(1) = 0	disable.
	OPTTRP(1) = 1	enable tripping simulation.
	OPTTRP(2)	number of tripping events per power flow (1 by default).
	OPTTRP(3)	number of power flows per contingency (1 by default).
Integer OPTCOR(8)	Is an array of eight elements specifying corrective actions options.	
	OPTCOR(1)	run corrective action (0 by default).
	OPTCOR(1) = 0	disable.
	OPTCOR(1) = 1	enable.
	OPTCOR(2)	number power flows (1 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTCOR(3)	generator control flag (1 by default).
OPTCOR(3) = 0	disable.
OPTCOR(3) = 1	enable.
OPTCOR(4)	load control flag (0 by default).
OPTCOR(4) = 0	disable.
OPTCOR(4) = 1	enable.
OPTCOR(5)	phase shifter control flag (1 by default).
OPTCOR(5) = 0	disable.
OPTCOR(5) = 1	enable.
OPTCOR(6)	off-line generator control flag (0 by default).
OPTCOR(6) = 0	disable.
OPTCOR(6) = 1	enable.
OPTCOR(7)	tap setting adjustment flag (0 by default).
OPTCOR(7) = 0	disable.
OPTCOR(7) = 1	enable.
OPTCOR(8)	switched shunt control flag (0 by default).
OPTCOR(8) = 0	disable.
OPTCOR(8) = 1	enable.

Real VALUES(11) Is the array of eleven elements to specifying ranking parameters and corrective actions parameters (input).

VALUES(1)	small reactance threshold, (0<= VALUES(1)<=0.005, 0.0005 by default).
VALUES(2)	percent of flow rating, (0<= VALUES(2)<=200.0, 100.0 by default).
VALUES(3)	MW threshold for islanding contingencies(>=0, 0.0 by default).
VALUES(4)	bus voltage violation tolerance (0.1 by default).
VALUES(5)	branch overload violation tolerance (0.1 by default).
VALUES(6)	generator control weighting factor (1.0 by default).
VALUES(7)	load control weighting factor (1.0 by default).
VALUES(8)	phase shifter control weighting factor (1.0 by default).
VALUES(9)	off-line generator control weighting factor (1.0 by default).
VALUES(10)	tap setting adjustment weighting factor (1.0 by default).
VALUES(11)	switched shunt control weighting factor (1.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character LABELS(8)*12	Is an array of eight elements specifying subsystem labels.	
	LABELS(1)	label of generation dispatch subsystem (blank by default, no default allowed if OPTACC(8) is not 0).
	LABELS(2)	label of contingency subsystem (blank by default).
	LABELS(3)	label of generator control subsystem (blank by default for none, no default allowed if OPTCOR(3) is 1).
	LABELS(4)	label of load control subsystem (blank by default, no default allowed if OPTCOR(4) is 1).
	LABELS(5)	label of phase shifter control subsystem (blank by default for none, no default allowed if OPTCOR(5) is 1).
	LABELS(6)	off-line generator control subsystem (blank by default; no default allowed if OPTCOR(6) is 1).
	LABELS(7)	tap setting adjustment subsystem (blank by default; no default allowed if OPTCOR(7) is 1).
	LABELS(8)	switched shunt control subsystem (blank by default; no default allowed if OPTCOR(8) is 1).
Character*260 DFXFILE	Is the name of Distribution Factor Data File (input, no default allowed).	
Character*260 ACCFILE	Is the name of Contingency Solution Output File (input, no default allowed).	
Character*260 THRFILE	Is the name of Load Throwover Data File (input, blank for none).	
Character*260 INLFILE	Is the name of the Unit Inertia and Governor Data File (input; blank for none).	
Character TRPFILE*260	Is the name of the Tripping Element Data File (input, blank for none; no default allowed if OPTTRP(1) is 1).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid TOL value.
IERR = 2	invalid OPTACC value.
IERR = 3	generators are converted.
IERR = 4	buses in island(s) without a swing bus; use activity TREE.
IERR = 5	largest mismatch exceeds mismatch tolerance.
IERR = 6	generation subsystem is not defined.
IERR = 7	too many islands in base case.
IERR = 8	in-service induction machines are in the “stalled” or “tripped” state.
IERR = 9	buses with bus type code and series element status inconsistencies.
IERR = 10	invalid OPTMAC value.
IERR = 11	invalid OPTRNK value.
IERR = 12	invalid OPTTRP value.
IERR = 13	invalid OPTCOR value.
IERR = 14	invalid VALUES value.
IERR = 21	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
IERR = 22	monitored elements exceed limit when adding multisection line members.
IERR = 49	error opening Tripping Element Data File.
IERR = 50	error opening Unit Inertia Data File.
IERR = 51	error opening Contingency Solution Output File.
IERR = 52	error opening Distribution Factor Data File.
IERR = 53	error opening Load Throwover Data File.
IERR = 54	prerequisite requirements for function not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.122 MACCC_WITH_COR

Use this API to run multiple AC contingency analysis with corrective actions. The user-specified and automatically-selected contingencies are evaluated individually and in combination with each other as overlapping contingencies. The corrective actions analysis will be launched for the contingencies with flow and voltage violations to eliminate the violations with the objective of minimizing the controls adjustments including load shedding, MW generation dispatch and phase shifter angle adjustments.

Batch command syntax:

```
BAT_MACCC_WITH_COR  TOL  OPTACC(1)...OPTACC(8) OPTMAC(1)...OPTMAC(41)
OPTRNK(1)...OPTRNK(17)  OPTCOR(1)...OPTCOR(5) VALUES(1)...VALUES(8)
LABELS(1)...LABELS(5)  DFXFILE  ACCFILE  THRFILE  INLFILE
```

Python syntax:

```
ierr = maccc_with_cor(tol, optacc, optmac, optrnk, optcor, values,
labels, dfxfile, accfile, thrfile, inlfile)
```

Fortran syntax:

```
CALL MACCC_WITH_COR(TOL, OPTACC, OPTMAC, OPTRNK, OPTCOR, VALUES,
LABELS, DFXFILE, ACCFILE, THRFILE, INLFILE, IERR)
```

where:

Real TOL	Is the mismatch tolerance (input; Newton solution convergence tolerance, TOLN, by default).
Integer OPTACC(8)	Is an array of eight elements specifying power flow solution options (input). The values are as follows:

OPTACC(1)	tap adjustment flag (tap adjustment option setting by default).
OPTACC(1) = 0	disable.
OPTACC(1) = 1	enable stepping adjustment.
OPTACC(1) = 2	enable direct adjustment.
OPTACC(2)	area interchange adjustment flag (area interchange adjustment option setting by default).
OPTACC(2) = 0	disable.
OPTACC(2) = 1	enable using tie line flows only in calculating area interchange.
OPTACC(2) = 2	enable using tie line flows and loads in calculating area interchange.
OPTACC(3)	phase shift adjustment flag (phase shift adjustment option setting by default).
OPTACC(3) = 0	disable.
OPTACC(3) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTACC(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
OPTACC(4) = 0	disable.
OPTACC(4) = 1	enable.
OPTACC(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
OPTACC(5) = 0	disable.
OPTACC(5) = 1	enable.
OPTACC(5) = 2	enable continuous mode, disable discrete mode.
OPTACC(6)	solution method flag (0 by default).
OPTACC(6) = 0	FDNS.
OPTACC(6) = 1	FNSL.
OPTACC(6) = 2	optimized FDNS.
OPTACC(7)	non-divergent solution flag (non-divergent solution option setting by default).
OPTACC(7) = 0	disable.
OPTACC(7) = 1	enable.
OPTACC(8)	dispatch mode (0 by default).
OPTACC(8) = 0	disable.
OPTACC(8) = 1	subsystem machines (reserve).
OPTACC(8) = 2	subsystem machines (pmax).
OPTACC(8) = 3	subsystem machines (inertia).
OPTACC(8) = 4	subsystem machines (governor droop).

Integer OPTMAC(41) Is an array of 41 elements specifying contingency evaluation level and combinations (input). The values are as follows:

OPTMAC(1)	run multiple contingency analysis (1 by default).
OPTMAC(1) = 0	disable.
OPTMAC(1) = 1	enable.
OPTMAC(2)	contingency evaluation level (1<=OPTMAC(2)<=3; 1 by default).
OPTMAC(3-41)	each element represents a type of contingency combinations in the following order. =1, include the contingency combinations, otherwise =0 (0 by default).

{S, U, B, SS, SU, SB, US, UU, UB, BS, BU, BB, SSS, SSB, SSU, SBS, SBB, SBU, SUS, SUB, SUU, USS, USU, USB, UUS, UUB, UUU, UBS, UBU, UBB, BSS, BSU, BSB, SUS, SUB, SUU, BBS, BBU, BBB}, where S, U and B are user specified cases, single machine and single branch outages respectively.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer OPTRNK(17) Is an array of 17 elements specifying contingency ranking options (input). The values are as follows:

OPTRNK(1)	include ties from contingency subsystem (0 by default).
OPTRNK(1) = 0	disable.
OPTRNK(1) = 1	enable.
OPTRNK(2)	contingency rating (rating set program option setting by default).
OPTRNK(2) = 1	rate A.
OPTRNK(2) = 2	rate B.
OPTRNK(2) = 3	rate C.
OPTRNK(3)	convert ratings to MW (0 by default).
OPTRNK(3) = 0	disable.
OPTRNK(3) = 1	enable.
OPTRNK(4)	include machine contingencies from overload ranking (0 by default).
OPTRNK(4) = 0	disable.
OPTRNK(4) = 1	enable.
OPTRNK(5)	minimum of machine contingencies to evaluate (>=0; 1 by default).
OPTRNK(6)	maximum of machine contingencies to evaluate (>=0; 10 by default).
OPTRNK(7)	non-failure cutoff of machine contingencies (>=0).
OPTRNK(8)	include branch contingencies from overload ranking (0 by default).
OPTRNK(8) = 0	disable.
OPTRNK(8) = 1	enable.
OPTRNK(9)	minimum of branch contingencies from overload ranking to evaluate (>=0; 1 by default).
OPTRNK(10)	maximum of branch contingencies from overload ranking to evaluate (>=0; 10 by default).
OPTRNK(11)	non-failure cutoff of branch contingencies from overload ranking (>=0; 5 by default).
OPTRNK(12)	include branch contingencies from voltage depression ranking (0 by default).
OPTRNK(12) = 0	disable.
OPTRNK(12) = 1	enable.
OPTRNK(13)	minimum of branch contingencies from voltage depression ranking to evaluate (>=0; 1 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTRNK(14)	maximum of branch contingencies from voltage depression ranking to evaluate (>=0; 10 by default).
	OPTRNK(15)	non-failure cutoff of branch contingencies from voltage depression ranking (>=0; 5 by default).
	OPTRNK(16)	include all branches in voltage depression ranking (0 by default).
	OPTRNK(16) = 0	include only with non-zero rating.
	OPTRNK(16) = 1	all.
	OPTRNK(17)	include contingencies that create swingless island (0 by default).
	OPTRNK(17) = 0	disable.
	OPTRNK(17) = 1	enable.
Integer OPTCOR(5)	Is an array of five elements specifying corrective actions options (input). The values are as follows:	
	OPTCOR(1)	run corrective actions (0 by default).
	OPTCOR(1) = 0	disable.
	OPTCOR(1) = 1	enable.
	OPTCOR(2)	number of power flow solutions (from 0 to 10; 1 by default).
	OPTCOR(3)	generator control flag (1 by default).
	OPTCOR(3) = 0	disable.
	OPTCOR(3) = 1	enable.
	OPTCOR(4)	load control flag (0 by default).
	OPTCOR(4) = 0	disable.
	OPTCOR(4) = 1	enable.
	OPTCOR(5)	phase shifter control flag (1 by default).
	OPTCOR(5) = 0	disable.
	OPTCOR(5) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real VALUES(8)	Is an array of eight elements specifying ranking and corrective actions parameters (input). The values are as follows:	
	VALUES(1)	small reactance threshold (0<=VALUES(1)<=0.005; 0.0005 by default).
	VALUES(2)	percent of rating set (0<=VALUES(1)<=200.0; 100.0 by default).
	VALUES(3)	MW threshold for islanding contingencies (>=0.0; 0.0 by default).
	VALUES(4)	bus voltage violation tolerance (0.1 by default).
	VALUES(5)	branch flow overload tolerance (0.1 by default).
	VALUES(6)	generator control weighting factor (1.0 by default).
	VALUES(7)	load control weighting factor (1.0 by default).
	VALUES(8)	phase shifter control weighting factor (1.0 by default).
Character LABELS(5)*12	Is an array of five elements specifying subsystem labels (input). The values are as follows:	
	LABELS(1)	label of generation dispatch subsystem (blank by default; no default allowed if OPTACC(8) is not 0).
	LABELS(2)	label of contingency subsystem (blank by default).
	LABELS(3)	label of generator control subsystem (blank by default; no default allowed if OPTCOR(3) is 1).
	LABELS(4)	label of load control subsystem (blank by default; no default allowed if OPTCOR(4) is 1).
	LABELS(5)	label of phase shifter control subsystem (blank by default; no default allowed if OPTCOR(5) is 1).
Character DFXFILE*260	Is the name of the Distribution Factor Data File (input; no default allowed).	
Character ACCFILE*260	Is the name of the Contingency Solution Output File (input; no default allowed).	
Character THRFILE*260	Is the name of the Load Throwover Data File (input; blank by default).	
Character INLFILE*260	Is the name of the Unit Inertia and Governor Data File (input; blank by default).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR

Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid TOL value.
IERR = 2	invalid OPTACC value.
IERR = 3	generators are converted.
IERR = 4	buses in island(s) without a swing bus; use activity TREE.
IERR = 5	largest mismatch exceeds mismatch tolerance.
IERR = 6	generation dispatch subsystem is not defined.
IERR = 7	too many islands in base case.
IERR = 8	invalid OPTMAC value.
IERR = 9	invalid OPTRNK value.
IERR = 10	invalid OPTCOR value.
IERR = 11	invalid VALUES value.
IERR = 21	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
IERR = 22	monitored elements exceed limit when adding multi-section line members.
IERR = 50	error opening Unit Inertia and Governor Data File.
IERR = 51	error opening Contingency Solution Output File.
IERR = 52	error opening Distribution Factor Data File.
IERR = 53	error opening Load Throwover Data File.
IERR = 54	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.123 MACCC_WITH_COR_2

The API is obsolete. It has been replaced by MACCC_WITH_COR_3, and implemented by a call to MACCC_WITH_COR_3.

This API is the second release of function to run multiple AC contingency analysis with corrective actions. The user-specified and automatically-selected contingencies are evaluated individually and in combination each other as overlapping contingencies. Corrective actions are specified automatically to eliminate the violations in each post contingency solution with the objective of minimizing the controls adjustments including load shedding, MW generation dispatch, phase shifter angle adjustments, off-line generator controls, tap setting adjustments and switched shunt controls.

Batch command syntax:

```
BAT_MACCC_WITH_COR_2 TOL  OPTACC(1)…OPTACC(8), OPTMAC(1)… OPTMAC(41),
OPTRNK(1)…OPTRNK(17), OPTCOR(1)…OPTCOR(8), VALUES(1)…VALUES(11),
LABELS(1)…LABELS(8), DFXFILE, ACCFILE, THRFILE, INLFILE
```

Python syntax:

```
ierr = maccc_with_cor_2(tol, optacc, optmac, optrnk, optcor, values,
labels, dfxfile, accfile, thrfile, inlfile)
```

Fortran syntax:

```
Call MACCC_WITH_COR_2(TOL, OPTACC, OPTMAC, OPTRNK, OPTCOR, VALUES,
LABELS, DFXFILE, ACCFILE, THRFILE, INLFILE, IERR)
```

where:

Real TOL	Is the mismatch tolerance (input, Newton solution convergence tolerance, TOLN, by default).	
Integer OPTACC(8)	Is an array of eight elements specifying power flow solution (input).	
	OPTACC(1)	tap adjustment flag (tap adjustment option setting by default).
	OPTACC(1) = 0	disable.
	OPTACC(1) = 1	enable stepping adjustment.
	OPTACC(1) = 2	enable direct adjustment.
	OPTACC(2)	area interchange adjustment flag (area interchange adjustment option setting by default).
	OPTACC(2) = 0	disable.
	OPTACC(2) = 1	enable using tie line flows only in calculating area interchange.
	OPTACC(2) = 2	enable using tie line flows and loads in calculating area interchange.
	OPTACC(3)	phase shift adjustment flag (phase shift adjustment option setting by default).
	OPTACC(3) = 0	disable.
	OPTACC(3) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTACC(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
	OPTACC(4) = 0	disable.
	OPTACC(4) = 1	enable.
	OPTACC(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
	OPTACC(5) = 0	disable.
	OPTACC(5) = 1	enable.
	OPTACC(5) = 2	enable continuous, disable discrete.
	OPTACC(6)	solution method flag (0 by default).
	OPTACC(6) = 0	FDNS.
	OPTACC(6) = 1	FNSL.
	OPTACC(6) = 2	optimized FDNS.
	OPTACC(7)	non-divergent solution flag (non-divergent solution option setting by default).
	OPTACC(7) = 0	disable.
	OPTACC(7) = 1	enable.
	OPTACC(8)	dispatch mode (0 by default).
	OPTACC(8) = 0	disable.
	OPTACC(8) = 1	subsystem machines (RESERVE).
	OPTACC(8) = 2	subsystem machines (PMAX).
	OPTACC(8) = 3	subsystem machines (INERTIA).
	OPTACC(8) = 4	subsystem machines (GOVERNOR DROOP).
Integer OPTMAC(41)	Is the array of 41 elements to specifying contingency evaluation level and combinations (input).	
	OPTMAC(1)	allow multiple contingency analysis (0 by default).
	OPTMAC(1) =0	disable.
	OPTMAC(1) =1	enable.
	OPTMAC(1) =2	enabled with failure cutoff disabled.
	OPTMAC(2)	contingency evaluation level (1 by default).
	OPTMAC(2) =1	N-1 contingency analysis.
	OPTMAC(2) =2	N-2 contingency analysis.
	OPTMAC(2) =3	N-3 contingency analysis.
	OPTMAC(3-41)	each element represents a type of contingency combinations in the following order. =1, include the contingency combinations, otherwise =0 (0 by default). {S, U, B, SS, SU, SB, US, UU, UB, BS, BU, BB, SSS, SSB, SSU, SBS, SBB, SBU, SUS, SUB, SUU, USS, USU, USB, UUS, UUB, UUU, UBS, UBU, UBB, BSS, BSU, BSB, SUS, SUB, SUU, BBS, BBU, BBB}, where S, U and B are user specified cases, single unit and single branch outages respectively.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer OPTRNK(17)	Is the array of 17 elements to specifying ranking options (input).	
	OPTRNK(1)	include ties from subsystem (0 by default).
	OPTRNK(1) = 0	disable.
	OPTRNK(1) = 1	enable.
	OPTRNK(2)	contingency rating (rating set program option setting by default).
	OPTRNK(2) = 1	rate A.
	OPTRNK(2) = 2	rate B.
	OPTRNK(2) = 3	rate C.
	OPTRNK(3)	convert ratings to MW (0 by default).
	OPTRNK(3) = 0	disable.
	OPTRNK(3) = 1	enable.
	OPTRNK(4)	machine overload contingencies (0 by default).
	OPTRNK(4) = 0	disable.
	OPTRNK(4) = 1	enable.
	OPTRNK(5)	minimum of machine overload contingencies to evaluate (>=0, 1 by default).
	OPTRNK(6)	maximum of machine overload contingencies to evaluate (>=0, 10 by default).
	OPTRNK(7)	cutoff of machine overload contingencies to evaluate (>=0, 5 by default).
	OPTRNK(8)	line overload contingencies (0 by default).
	OPTRNK(8) = 0	disable.
	OPTRNK(8) = 1	enable.
	OPTRNK(9)	minimum of line overload contingencies to evaluate (>=0, 1 by default).
	OPTRNK(10)	maximum of line overload contingencies to evaluate (>=0, 10 by default).
	OPTRNK(11)	cutoff of line overload contingencies (>=0, 5 by default).
	OPTRNK(12)	voltage collapse line contingencies (0 by default).
	OPTRNK(12) = 0	disable.
	OPTRNK(12) = 1	enable.
	OPTRNK(13)	minimum of voltage collapse line contingencies to evaluate (>=0, 1 by default).
	OPTRNK(14)	maximum of voltage collapse line contingencies to evaluate (>=0, 10 by default).
	OPTRNK(15)	cutoff of voltage collapse line contingencies (>=0, 5 be default).
	OPTRNK(16)	include all branches in voltage collapse ranking (0 by default).
	OPTRNK(16) = 0	include only with non-zero rating.
	OPTRNK(16) = 1	all.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTRNK(17)	include contingencies that create swingless island (0 by default).
	OPTRNK(17) = 0	disable.
	OPTRNK(17) = 1	enable.
Integer OPTCOR(8)	Is an array of eight elements specifying corrective actions options.	
	OPTCOR(1)	run corrective action (0 by default).
	OPTCOR(1) = 0	disable.
	OPTCOR(1) = 1	enable.
	OPTCOR(2)	number power flows (1 by default).
	OPTCOR(3)	generator control flag (1 by default).
	OPTCOR(3) = 0	disable.
	OPTCOR(3) = 1	enable.
	OPTCOR(4)	load control flag (0 by default).
	OPTCOR(4) = 0	disable.
	OPTCOR(4) = 1	enable.
	OPTCOR(5)	phase shifter control flag (1 by default).
	OPTCOR(5) = 0	disable.
	OPTCOR(5) = 1	enable.
	OPTCOR(6)	off-line generator control flag (0 by default).
	OPTCOR(6) = 0	disable.
	OPTCOR(6) = 1	enable.
	OPTCOR(7)	tap setting adjustment flag (0 by default).
	OPTCOR(7) = 0	disable.
	OPTCOR(7) = 1	enable.
	OPTCOR(8)	switched shunt control flag (0 by default).
	OPTCOR(8) = 0	disable.
	OPTCOR(8) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real VALUES(11)	Is an array of eleven elements to specifying ranking parameters and corrective actions parameters (input).	
	VALUES(1)	small reactance threshold, (0<= VALUES(1)<=0.005, 0.0005 by default).
	VALUES(2)	percent of flow rating, (0<= VALUES(2)<=200.0, 100.0 by default).
	VALUES(3)	MW threshold for islanding contingencies(>=0, 0.0 by default).
	VALUES(4)	bus voltage violation tolerance (0.1 by default).
	VALUES(5)	branch overload violation tolerance (0.1 by default).
	VALUES(6)	generator control weighting factor (1.0 by default).
	VALUES(7)	load control weighting factor (1.0 by default).
	VALUES(8)	phase shifter control weighting factor (1.0 by default).
	VALUES(9)	off-line generator control weighting factor (1.0 by default).
	VALUES(10)	tap setting adjustment weighting factor (1.0 by default).
	VALUES(11)	switched shunt control weighting factor (1.0 by default).
Character LABELS(8)*12	Is an array of eight elements specifying subsystem labels.	
	LABELS (1)	label of generation dispatch subsystem (blank by default, no default allowed if OPTACC(8) is 1).
	LABELS (2)	label of contingency subsystem (blank by default).
	LABELS (3)	label of generator control subsystem (blank by default, no default allowed if OPTCOR(3) is 1).
	LABELS (4)	label of load control subsystem (blank by default, no default allowed if OPTCOR(4) is 1).
	LABELS (5)	label of phase shifter control subsystem (blank by default, no default allowed if OPTCOR(5) is 1).
	LABELS(6)	off-line generator control subsystem (blank by default; no default allowed if OPTCOR(6) is 1).
	LABELS(7)	tap setting adjustment subsystem (blank by default; no default allowed if OPTCOR(7) is 1).
	LABELS(8)	switched shunt control subsystem (blank by default; no default allowed if OPTCOR(8) is 1).
Character*260 DFXFILE	Is the name of Distribution Factor Data File (input, no default allowed).	
Character*260 ACCFILE	Is the name of Contingency Solution Output File (input, no default allowed).	
Character*260 THRFILE	Is the name of Load Throwover Data File (input, blank for none).	
Character*260 INLFILE	Is the name of Unit Inertia and Governor Data File (input, blank for none).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid TOL value.
IERR = 2	invalid OPTACC value.
IERR = 3	generators are converted.
IERR = 4	buses in island(s) without a swing bus; use activity TREE.
IERR = 5	largest mismatch exceeds mismatch tolerance.
IERR = 6	generation subsystem is not defined.
IERR = 7	too many islands in base case.
IERR = 8	invalid OPTMAC value.
IERR = 9	invalid OPTRNK value.
IERR = 10	invalid OPTCOR value.
IERR = 11	invalid VALUES value.
IERR = 21	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
IERR = 22	monitored elements exceed limit when adding multisection line members.
IERR = 50	error opening Unit Inertia Data File.
IERR = 51	error opening Contingency Solution Output File.
IERR = 52	error opening Distribution Factor Data File.
IERR = 53	error opening Load Throwover Data File.
IERR = 54	prerequisite requirements for function not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.124 MACCC_WITH_COR_3

This API is the third release of function to run multiple AC contingency analysis with corrective actions. The user-specified and automatically-selected contingencies are evaluated individually and in combination each other as overlapping contingencies. Corrective actions are specified automatically to eliminate the violations in each post contingency solution with the objective of minimizing the controls adjustments including load shedding, MW generation dispatch, phase shifter angle adjustments, off-line generator controls, tap setting adjustments and switched shunt controls.

Batch command syntax:

```
BAT_MACCC_WITH_COR_3 TOL  OPTACC(1)…OPTACC(10), OPTMAC(1)… OPTMAC(41),
OPTRNK(1)…OPTRNK(17), OPTCOR(1)…OPTCOR(8), VALUES(1)…VALUES(11),
LABELS(1)…LABELS(8), DFXFILE, ACCFILE, THRFILE, INLFILE
```

Python syntax:

```
ierr = maccc_with_cor_3(tol, optacc, optmac, optrnk, optcor, values,
labels, dfxfile, accfile, thrfile, inlfile)
```

Fortran syntax:

```
Call MACCC_WITH_COR_3(TOL, OPTACC, OPTMAC, OPTRNK, OPTCOR, VALUES,
LABELS, DFXFILE, ACCFILE, THRFILE, INLFILE, IERR)
```

where:

Real TOL	Is the mismatch tolerance (input, Newton solution convergence tolerance, TOLN, by default).	
Integer OPTACC(10)	Is an array of ten elements specifying power flow solution (input).	
	OPTACC(1)	tap adjustment flag (tap adjustment option setting by default).
	OPTACC(1) = 0	disable.
	OPTACC(1) = 1	enable stepping adjustment.
	OPTACC(1) = 2	enable direct adjustment.
	OPTACC(2)	area interchange adjustment flag (area interchange adjustment option setting by default).
	OPTACC(2) = 0	disable.
	OPTACC(2) = 1	enable using tie line flows only in calculating area interchange.
	OPTACC(2) = 2	enable using tie line flows and loads in calculating area interchange.
	OPTACC(3)	phase shift adjustment flag (phase shift adjustment option setting by default).
	OPTACC(3) = 0	disable.
	OPTACC(3) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTACC(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
OPTACC(4) = 0	disable.
OPTACC(4) = 1	enable.
OPTACC(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
OPTACC(5) = 0	disable.
OPTACC(5) = 1	enable.
OPTACC(5) = 2	enable continuous, disable discrete.
OPTACC(6)	solution method flag (0 by default).
OPTACC(6) = 0	FDNS.
OPTACC(6) = 1	FNSL.
OPTACC(6) = 2	optimized FDNS.
OPTACC(7)	non-divergent solution flag (non-divergent solution option setting by default).
OPTACC(7) = 0	disable.
OPTACC(7) = 1	enable.
OPTACC(8)	induction motor treatment flag (applied when an induction motor fails to solve due to low terminal bus voltage, 0 by default)
OPTACC(8) = 0	stall.
OPTACC(8) = 1	trip.
OPTACC(9)	induction machine failure flag(0 by default)
OPTACC(9) = 0	treat contingency as non-converged if any induction machines are placed in the “stalled” or “tripped” state.
OPTACC(9) = 1	treat contingency as solved if it converges, even if any induction machines are placed in the “stalled” or “tripped” state.
OPTACC(10)	dispatch mode (0 by default).
OPTACC(10) = 0	disable.
OPTACC(10) = 1	subsystem machines (RESERVE).
OPTACC(10) = 2	subsystem machines (PMAX).
OPTACC(10) = 3	subsystem machines (INERTIA).
OPTACC(10) = 4	subsystem machines (GOVERNOR DROOP).

Integer OPTMAC(41) — Is the array of 41 elements to specifying contingency evaluation level and combinations (input).

OPTMAC(1)	allow multiple contingency analysis (0 by default).
OPTMAC(1) =0	disable.
OPTMAC(1) =1	enable.
OPTMAC(1) =2	enabled with failure cutoff disabled.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTMAC(2)	contingency evaluation level (1 by default)
	OPTMAC(2)=1	N-1 contingency analysis
	OPTMAC(2)=2 or -2	N-2 contingency analysis
	OPTMAC(2)=3 or -3	N-3 contingency analysis
	If it is 2 or 3, a newly ordered event list is built for the next level after solving AC power flow each case at current level; if it is -2 or -3, the event list at the first level is used for any higher levels.	
	OPTMAC(3-41)	each element represents a type of contingency combinations in the following order. =1, include the contingency combinations, otherwise =0 (0 by default).
	{S, U, B, SS, SU, SB, US, UU, UB, BS, BU, BB, SSS, SSB, SSU, SBS, SBB, SBU, SUS, SUB, SUU, USS, USU, USB, UUS, UUB, UUU, UBS, UBU, UBB, BSS, BSU, BSB, SUS, SUB, SUU, BBS, BBU, BBB}, where S, U and B are user specified cases, single unit and single branch outages respectively.	
Integer OPTRNK(17)	Is the array of 17 elements to specifying ranking options (input).	
	OPTRNK(1)	include ties from subsystem (0 by default).
	OPTRNK(1) = 0	disable.
	OPTRNK(1) = 1	enable.
	OPTRNK(2)	contingency rating (rating set program option setting by default).
	OPTRNK(2) = 1	rate A.
	OPTRNK(2) = 2	rate B.
	OPTRNK(2) = 3	rate C.
	OPTRNK(3)	convert ratings to MW (0 by default).
	OPTRNK(3) = 0	disable.
	OPTRNK(3) = 1	enable.
	OPTRNK(4)	machine overload contingencies (0 by default).
	OPTRNK(4) = 0	disable.
	OPTRNK(4) = 1	enable.
	OPTRNK(5)	minimum of machine overload contingencies to evaluate (>=0, 1 by default).
	OPTRNK(6)	maximum of machine overload contingencies to evaluate (>=0, 10 by default).
	OPTRNK(7)	cutoff of machine overload contingencies to evaluate (>=0, 5 by default).
	OPTRNK(8)	line overload contingencies (0 by default).
	OPTRNK(8) = 0	disable.
	OPTRNK(8) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTRNK(9)	minimum of line overload contingencies to evaluate (>=0, 1 by default).
	OPTRNK(10)	maximum of line overload contingencies to evaluate (>=0, 10 by default).
	OPTRNK(11)	cutoff of line overload contingencies (>=0, 5 by default).
	OPTRNK(12)	voltage collapse line contingencies (0 by default).
	OPTRNK(12) = 0	disable.
	OPTRNK(12) = 1	enable.
	OPTRNK(13)	minimum of voltage collapse line contingencies to evaluate (>=0, 1 by default).
	OPTRNK(14)	maximum of voltage collapse line contingencies to evaluate (>=0, 10 by default).
	OPTRNK(15)	cutoff of voltage collapse line contingencies (>=0, 5 be default).
	OPTRNK(16)	include all branches in voltage collapse ranking (0 by default).
	OPTRNK(16) = 0	include only with non-zero rating.
	OPTRNK(16) = 1	all.
	OPTRNK(17)	include contingencies that create swingless island (0 by default).
	OPTRNK(17) = 0	disable.
	OPTRNK(17) = 1	enable.
Integer OPTCOR(8)	Is an array of eight elements specifying corrective actions options.	
	OPTCOR(1)	run corrective action (0 by default).
	OPTCOR(1) = 0	disable.
	OPTCOR(1) = 1	enable.
	OPTCOR(2)	number power flows (1 by default).
	OPTCOR(3)	generator control flag (1 by default).
	OPTCOR(3) = 0	disable.
	OPTCOR(3) = 1	enable.
	OPTCOR(4)	load control flag (0 by default).
	OPTCOR(4) = 0	disable.
	OPTCOR(4) = 1	enable.
	OPTCOR(5)	phase shifter control flag (1 by default).
	OPTCOR(5) = 0	disable.
	OPTCOR(5) = 1	enable.
	OPTCOR(6)	off-line generator control flag (0 by default).
	OPTCOR(6) = 0	disable.
	OPTCOR(6) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTCOR(7)	tap setting adjustment flag (0 by default).
	OPTCOR(7) = 0	disable.
	OPTCOR(7) = 1	enable.
	OPTCOR(8)	switched shunt control flag (0 by default).
	OPTCOR(8) = 0	disable.
	OPTCOR(8) = 1	enable.
Real VALUES(11)	Is an array of eleven elements to specifying ranking parameters and corrective actions parameters (input).	
	VALUES(1)	small reactance threshold, (0<= VALUES(1)<=0.005, 0.0005 by default).
	VALUES(2)	percent of flow rating, (0<= VALUES(2)<=200.0, 100.0 by default).
	VALUES(3)	MW threshold for islanding contingencies(>=0, 0.0 by default).
	VALUES(4)	bus voltage violation tolerance (0.1 by default).
	VALUES(5)	branch overload violation tolerance (0.1 by default).
	VALUES(6)	generator control weighting factor (1.0 by default).
	VALUES(7)	load control weighting factor (1.0 by default).
	VALUES(8)	phase shifter control weighting factor (1.0 by default).
	VALUES(9)	off-line generator control weighting factor (1.0 by default).
	VALUES(10)	tap setting adjustment weighting factor (1.0 by default).
	VALUES(11)	switched shunt control weighting factor (1.0 by default).
Character LABELS(8)*12	Is an array of eight elements specifying subsystem labels.	
	LABELS (1)	label of generation dispatch subsystem (blank by default, no default allowed if OPTACC(8) is 1).
	LABELS (2)	label of contingency subsystem (blank by default).
	LABELS (3)	label of generator control subsystem (blank by default, no default allowed if OPTCOR(3) is 1).
	LABELS (4)	label of load control subsystem (blank by default, no default allowed if OPTCOR(4) is 1).
	LABELS (5)	label of phase shifter control subsystem (blank by default, no default allowed if OPTCOR(5) is 1).
	LABELS(6)	off-line generator control subsystem (blank by default; no default allowed if OPTCOR(6) is 1).
	LABELS(7)	tap setting adjustment subsystem (blank by default; no default allowed if OPTCOR(7) is 1).
	LABELS(8)	switched shunt control subsystem (blank by default; no default allowed if OPTCOR(8) is 1).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character*260 DFXFILE	Is the name of Distribution Factor Data File (input, no default allowed).
Character*260 ACCFILE	Is the name of Contingency Solution Output File (input, no default allowed).
Character*260 THRFILE	Is the name of Load Throwover Data File (input, blank for none).
Character*260 INLFILE	Is the name of Unit Inertia and Governor Data File (input, blank for none).
Integer IERR	Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid TOL value.
IERR = 2	invalid OPTACC value.
IERR = 3	generators are converted.
IERR = 4	buses in island(s) without a swing bus; use activity TREE.
IERR = 5	largest mismatch exceeds mismatch tolerance.
IERR = 6	generation subsystem is not defined.
IERR = 7	too many islands in base case.
IERR = 8	in-service induction machines are in the “stalled” or “tripped” state.
IERR = 9	buses with bus type code and series element status inconsistencies.
IERR = 10	invalid OPTMAC value.
IERR = 11	invalid OPTRNK value.
IERR = 12	invalid OPTCOR value.
IERR = 13	invalid VALUES value.
IERR = 21	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
IERR = 22	monitored elements exceed limit when adding multisection line members.
IERR = 50	error opening Unit Inertia Data File.
IERR = 51	error opening Contingency Solution Output File.
IERR = 52	error opening Distribution Factor Data File.
IERR = 53	error opening Load Throwover Data File.
IERR = 54	prerequisite requirements for function not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.125 MACCC_WITH_TRIP

The API is obsolete. It has been replaced by MACCC_WITH_TRIP_2, and implemented by a call to the MACCC_WITH_TRIP_2.

Use this API to run multiple level contingency analysis with tripping simulation.

Batch command syntax:

```
BAT_MACCC_WITH_TRIP TOL OPTACC(1)...OPTACC(8) OPTMAC(1)... OPTMAC(41)
OPTRNK(1)... OPTRNK(17) OPTTRP(1)...OPTTRP(3) VALUES(1)... VALUES(3)
LABELS(1) LABELS(2) DFXFILE ACCFILE THRFILE INLFILE TRPFILE
```

Python syntax:

```
ierr = maccc_with_trip (tol, optacc, optmac, optrnk, opttrp, values,
labels, dfxfile, accfile, thrfile, inlfile, trpfile)
```

Fortran syntax:

```
CALL MACCC_WITH_TRIP(TOL, OPTACC, OPTMAC, OPTRNK, OPTTRP, VALUES,
LABELS, DFXFILE, ACCFILE, THRFILE, INLFILE, TRPFILE, IERR)
```

where:

Real TOL	Is the mismatch tolerance (input, Newton solution convergence tolerance, TOLN, by default).
Integer OPTACC(8)	Is an array of eight elements specifying power flow solution options (input). The values are as follows:

OPTACC(1)	tap adjustment flag (tap adjustment option setting by default).
OPTACC(1) = 0	disable.
OPTACC(1) = 1	enable stepping adjustment.
OPTACC(1) = 2	enable direct adjustment.
OPTACC(2)	area interchange adjustment flag (area interchange adjustment option setting by default).
OPTACC(2) = 0	disable.
OPTACC(2) = 1	enable using tie line flows only in calculating area interchange.
OPTACC(2) = 2	enable using tie line flows and loads in calculating area interchange.
OPTACC(3)	phase shift adjustment flag (phase shift adjustment option setting by default).
OPTACC(3) = 0	disable.
OPTACC(3) = 1	enable.
OPTACC(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
OPTACC(4) = 0	disable.
OPTACC(4) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTACC(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
OPTACC(5) = 0	disable.
OPTACC(5) = 1	enable.
OPTACC(5) = 2	enable continuous mode, disable discrete mode.
OPTACC(6)	solution method flag (0 by default).
OPTACC(6) = 0	FDNS.
OPTACC(6) = 1	FNSL.
OPTACC(6) = 2	optimized FDNS.
OPTACC(7)	non-divergent solution flag (non-divergent solution option setting by default).
OPTACC(7) = 0	disable.
OPTACC(7) = 1	enable.
OPTACC(8)	dispatch mode (0 by default).
OPTACC(8) = 0	disable.
OPTACC(8) = 1	subsystem machines (reserve).
OPTACC(8) = 2	subsystem machines (pmax).
OPTACC(8) = 3	subsystem machines (inertia).
OPTACC(8) = 4	subsystem machines (governor droop).

Integer OPTMAC(41) Is an array of 41 elements to specifying contingency evaluation level and combinations (input). The values are as follows:

OPTMAC(1)	run multiple contingency analysis (1 by default).
OPTMAC(1) = 0	disable.
OPTMAC(1) = 1	enable.
OPTMAC(2)	contingency evaluation level (1<=OPTMAC(2)<=3; 1 by default).
OPTMAC(3-41)	each element represents a type of contingency combinations in the following order. =1, include the contingency combinations, otherwise =0 (0 by default).

{S, U, B, SS, SU, SB, US, UU, UB, BS, BU, BB, SSS, SSB, SSU, SBS, SBB, SBU, SUS, SUB, SUU, USS, USU, USB, UUS, UUB, UUU, UBS, UBU, UBB, BSS, BSU, BSB, SUS, SUB, SUU, BBS, BBU, BBB}, where S, U and B are user specified cases, single machine and single branch outages respectively.

Integer OPTRNK(17) Is an array of 17 elements specifying ranking options (input). The values are as follows:

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTRNK(1)	include ties from contingency subsystem (0 by default).
OPTRNK(1) = 0	disable.
OPTRNK(1) = 1	enable.
OPTRNK(2)	contingency rating (rating set program option setting by default).
OPTRNK(2) = 1	rate A.
OPTRNK(2) = 2	rate B.
OPTRNK(2) = 3	rate C.
OPTRNK(3)	convert ratings to MW (0 by default).
OPTRNK(3) = 0	disable.
OPTRNK(3) = 1	enable.
OPTRNK(4)	include machine contingencies from overload ranking (0 by default).
OPTRNK(4) = 0	disable.
OPTRNK(4) = 1	enable.
OPTRNK(5)	minimum of machine contingencies to evaluate (>=0; 1 by default).
OPTRNK(6)	maximum of machine contingencies to evaluate (>=0; 10 by default).
OPTRNK(7)	non-failure cutoff of machine contingencies (>=0; 5 by default).
OPTRNK(8)	include branch contingencies from overload ranking (0 by default).
OPTRNK(8) = 0	disable.
OPTRNK(8) = 1	enable.
OPTRNK(9)	minimum of branch contingencies from overload ranking to evaluate (>=0; 1 by default).
OPTRNK(10)	maximum of branch contingencies from overload ranking to evaluate (>=0; 10 by default).
OPTRNK(11)	non-failure cutoff of branch contingencies from overload ranking (>=0; 5 by default).
OPTRNK(12)	include branch contingencies from voltage depression ranking (0 by default).
OPTRNK(12) = 0	disable.
OPTRNK(12) = 1	enable.
OPTRNK(13)	minimum of branch contingencies from voltage depression ranking to evaluate (>=0; 1 by default).
OPTRNK(14)	maximum of branch contingencies from voltage depression ranking to evaluate (>=0; 10 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTRNK(15)	non-failure cutoff of branch contingencies from voltage depression ranking (>=0; 5 by default).
	OPTRNK(16)	include all branches in voltage depression ranking (0 by default).
	OPTRNK(16) = 0	include only with non-zero rating.
	OPTRNK(16) = 1	all.
	OPTRNK(17)	include contingencies that create swingless island (0 by default).
	OPTRNK(17) = 0	disable.
	OPTRNK(17) = 1	enable.
Integer OPTTRP(3)	Is an array of three elements specifying tripping simulation options (input). The values are as follows:	
	OPTTRP(1)	tripping simulation flag (0 by default).
	OPTTRP(1) = 0	disable.
	OPTTRP(1) = 1	enable.
	OPTTRP(2)	number of tripping events per power flow (1 by default).
	OPTTRP(3)	number of power flows per contingency (1 by default).
Real VALUES(3)	Is an array of three elements specifying ranking parameters (input). The values are as follows:	
	VALUES(1)	small reactance threshold (0<= VALUES(1)<=0.005; 0.0005 by default).
	VALUES(2)	percent of rating (0<= VALUES(1)<=200.0; 100.0 by default).
	VALUES(3)	MW threshold for islanding contingencies (>=0; 0.0 by default).
Character LABELS(2)*12	Is an array of two elements specifying subsystem labels (input). The values are as follows:	
	LABELS (1)	label of generation dispatch subsystem (blank by default; no default allowed if OPTACC(8) is not 0).
	LABELS (2)	label of contingency subsystem (blank by default).
Character DFXFILE*260	Is the name of the Distribution Factor Data File (input; no default allowed).	
Character ACCFILE*260	Is the name of the Contingency Solution Output File (input; no default allowed).	
Character THRFILE*260	Is the name of the Load Throwover Data File (input; blank by default).	
Character INLFILE*260	Is the name of the Unit Inertia and Governor Data File (input; blank by default).	
Character TRPFILE*260	Is the name of the Tripping Element Data File (input, blank for none; no default allowed if OPTTRP(1) is 1).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid TOL value.
IERR = 2	invalid OPTACC value.
IERR = 3	generators are converted.
IERR = 4	buses in island(s) without a swing bus; use activity TREE.
IERR = 5	largest mismatch exceeds mismatch tolerance.
IERR = 6	generation subsystem is not defined.
IERR = 7	too many islands in base case.
IERR = 8	invalid OPTMAC value.
IERR = 9	invalid OPTRNK value.
IERR = 10	invalid VALUES value.
IERR = 11	invalid OPTTRP value.
IERR = 21	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
IERR = 22	monitored elements exceed limit when adding multi-section line members.
IERR = 49	error opening Tripping Element Data File.
IERR = 50	error opening Unit Inertia and Governor Data File.
IERR = 51	error opening Contingency Solution Output File.
IERR = 52	error opening Distribution Factor Data File.
IERR = 53	error opening Load Throwover Data File.
IERR = 54	prerequisite requirements for function not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.126 MACCC_WITH_TRIP_2

Use this API to run the second version of multiple level contingency analysis with tripping simulation.

Batch command syntax:

```
BAT_MACCC_WITH_TRIP_2 TOL OPTACC(1)...OPTACC(10) OPTMAC(1)...
OPTMAC(41) OPTRNK(1)... OPTRNK(17) OPTTRP(1)...OPTTRP(3) VALUES(1)...
VALUES(3) LABELS(1) LABELS(2) DFXFILE ACCFILE THRFILE INLFILE TRPFILE
```

Python syntax:

```
ierr = maccc_with_trip_2 (tol, optacc, optmac, optrnk, opttrp, values,
labels, dfxfile, accfile, thrfile, inlfile, trpfile)
```

Fortran syntax:

```
CALL MACCC_WITH_TRIP_2(TOL, OPTACC, OPTMAC, OPTRNK, OPTTRP, VALUES,
LABELS, DFXFILE, ACCFILE, THRFILE, INLFILE, TRPFILE, IERR)
```

where:

Real TOL	Is the mismatch tolerance (input, Newton solution convergence tolerance, TOLN, by default).
Integer OPTACC(10)	Is an array of ten elements specifying power flow solution options (input). The values are as follows:
	OPTACC(1) — tap adjustment flag (tap adjustment option setting by default).
	OPTACC(1) = 0 — disable.
	OPTACC(1) = 1 — enable stepping adjustment.
	OPTACC(1) = 2 — enable direct adjustment.
	OPTACC(2) — area interchange adjustment flag (area interchange adjustment option setting by default).
	OPTACC(2) = 0 — disable.
	OPTACC(2) = 1 — enable using tie line flows only in calculating area interchange.
	OPTACC(2) = 2 — enable using tie line flows and loads in calculating area interchange.
	OPTACC(3) — phase shift adjustment flag (phase shift adjustment option setting by default).
	OPTACC(3) = 0 — disable.
	OPTACC(3) = 1 — enable.
	OPTACC(4) — dc tap adjustment flag (dc tap adjustment option setting by default).
	OPTACC(4) = 0 — disable.
	OPTACC(4) = 1 — enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTACC(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
OPTACC(5) = 0	disable.
OPTACC(5) = 1	enable.
OPTACC(5) = 2	enable continuous mode, disable discrete mode.
OPTACC(6)	solution method flag (0 by default).
OPTACC(6) = 0	FDNS.
OPTACC(6) = 1	FNSL.
OPTACC(6) = 2	optimized FDNS.
OPTACC(7)	non-divergent solution flag (non-divergent solution option setting by default).
OPTACC(7) = 0	disable.
OPTACC(7) = 1	enable.
OPTACC(8)	induction motor treatment flag (applied when an induction motor fails to solve due to low terminal bus voltage, 0 by default)
OPTACC(8) = 0	stall.
OPTACC(8) = 1	trip.
OPTACC(9)	induction machine failure flag(0 by default)
OPTACC(9) = 0	treat contingency as non-converged if any induction machines are placed in the “stalled” or “tripped” state.
OPTACC(9) = 1	treat contingency as solved if it converges, even if any induction machines are placed in the “stalled” or “tripped” state.
OPTACC(10)	dispatch mode (0 by default).
OPTACC(10) = 0	disable.
OPTACC(10) = 1	subsystem machines (reserve).
OPTACC(10) = 2	subsystem machines (pmax).
OPTACC(10) = 3	subsystem machines (inertia).
OPTACC(10) = 4	subsystem machines (governor droop).

Integer OPTMAC(41) Is an array of 41 elements to specifying contingency evaluation level and combinations (input). The values are as follows:

OPTMAC(1)	run multiple contingency analysis (1 by default).
OPTMAC(1) = 0	disable.
OPTMAC(1) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTMAC(2)	contingency evaluation level (1 by default)
OPTMAC(2)=1	N-1 contingency analysis
OPTMAC(2)=2 or -2	N-2 contingency analysis
OPTMAC(2)=3 or -3	N-3 contingency analysis

If it is 2 or 3, a newly ordered event list is built for the next level after solving AC power flow each case at current level; if it is -2 or -3, the event list at the first level is used for any higher levels.

OPTMAC(3-41)	each element represents a type of contingency combinations in the following order. =1, include the contingency combinations, otherwise =0 (0 by default).

{S, U, B, SS, SU, SB, US, UU, UB, BS, BU, BB, SSS, SSB, SSU, SBS, SBB, SBU, SUS, SUB, SUU, USS, USU, USB, UUS, UUB, UUU, UBS, UBU, UBB, BSS, BSU, BSB, SUS, SUB, SUU, BBS, BBU, BBB}, where S, U and B are user specified cases, single machine and single branch outages respectively.

Integer OPTRNK(17)

Is an array of 17 elements specifying ranking options (input). The values are as follows:

OPTRNK(1)	include ties from contingency subsystem (0 by default).
OPTRNK(1) = 0	disable.
OPTRNK(1) = 1	enable.
OPTRNK(2)	contingency rating (rating set program option setting by default).
OPTRNK(2) = 1	rate A.
OPTRNK(2) = 2	rate B.
OPTRNK(2) = 3	rate C.
OPTRNK(3)	convert ratings to MW (0 by default).
OPTRNK(3) = 0	disable.
OPTRNK(3) = 1	enable.
OPTRNK(4)	include machine contingencies from overload ranking (0 by default).
OPTRNK(4) = 0	disable.
OPTRNK(4) = 1	enable.
OPTRNK(5)	minimum of machine contingencies to evaluate (>=0; 1 by default).
OPTRNK(6)	maximum of machine contingencies to evaluate (>=0; 10 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTRNK(7)	non-failure cutoff of machine contingencies (>=0; 5 by default).
OPTRNK(8)	include branch contingencies from overload ranking (0 by default).
OPTRNK(8) = 0	disable.
OPTRNK(8) = 1	enable.
OPTRNK(9)	minimum of branch contingencies from overload ranking to evaluate (>=0; 1 by default).
OPTRNK(10)	maximum of branch contingencies from overload ranking to evaluate (>=0; 10 by default).
OPTRNK(11)	non-failure cutoff of branch contingencies from overload ranking (>=0; 5 by default).
OPTRNK(12)	include branch contingencies from voltage depression ranking (0 by default).
OPTRNK(12) = 0	disable.
OPTRNK(12) = 1	enable.
OPTRNK(13)	minimum of branch contingencies from voltage depression ranking to evaluate (>=0; 1 by default).
OPTRNK(14)	maximum of branch contingencies from voltage depression ranking to evaluate (>=0; 10 by default).
OPTRNK(15)	non-failure cutoff of branch contingencies from voltage depression ranking (>=0; 5 by default).
OPTRNK(16)	include all branches in voltage depression ranking (0 by default).
OPTRNK(16) = 0	include only with non-zero rating.
OPTRNK(16) = 1	all.
OPTRNK(17)	include contingencies that create swingless island (0 by default).
OPTRNK(17) = 0	disable.
OPTRNK(17) = 1	enable.

Integer OPTTRP(3) Is an array of three elements specifying tripping simulation options (input). The values are as follows:

OPTTRP(1)	tripping simulation flag (0 by default).
OPTTRP(1) = 0	disable.
OPTTRP(1) = 1	enable.
OPTTRP(2)	number of tripping events per power flow (1 by default).
OPTTRP(3)	number of power flows per contingency (1 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real VALUES(3)	Is an array of three elements specifying ranking parameters (input). The values are as follows: VALUES(1) small reactance threshold (0<= VALUES(1)<=0.005; 0.0005 by default). VALUES(2) percent of rating (0<= VALUES(1)<=200.0; 100.0 by default). VALUES(3) MW threshold for islanding contingencies (>=0; 0.0 by default).
Character LABELS(2)*12	Is an array of two elements specifying subsystem labels (input). The values are as follows: LABELS (1) label of generation dispatch subsystem (blank by default; no default allowed if OPTACC(8) is not 0). LABELS (2) label of contingency subsystem (blank by default).
Character DFXFILE*260	Is the name of the Distribution Factor Data File (input; no default allowed).
Character ACCFILE*260	Is the name of the Contingency Solution Output File (input; no default allowed).
Character THRFILE*260	Is the name of the Load Throwover Data File (input; blank by default).
Character INLFILE*260	Is the name of the Unit Inertia and Governor Data File (input; blank by default).
Character TRPFILE*260	Is the name of the Tripping Element Data File (input, blank for none; no default allowed if OPTTRP(1) is 1).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid TOL value. IERR = 2 invalid OPTACC value. IERR = 3 generators are converted. IERR = 4 buses in island(s) without a swing bus; use activity TREE. IERR = 5 largest mismatch exceeds mismatch tolerance. IERR = 6 generation subsystem is not defined. IERR = 7 too many islands in base case. IERR = 8 in-service induction machines are in the “stalled” or “tripped” state. IERR = 9 buses with bus type code and series element status inconsistencies.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IERR = 10	invalid OPTMAC value.
IERR = 11	invalid OPTRNK value.
IERR = 12	invalid VALUES value.
IERR = 13	invalid OPTTRP value.
IERR = 21	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
IERR = 22	monitored elements exceed limit when adding multi-sec-tion line members.
IERR = 49	error opening Tripping Element Data File.
IERR = 50	error opening Unit Inertia and Governor Data File.
IERR = 51	error opening Contingency Solution Output File.
IERR = 52	error opening Distribution Factor Data File.
IERR = 53	error opening Load Throwover Data File.
IERR = 54	prerequisite requirements for function not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.127 MACCC_WITH_TRIP_PARALLEL

Use this API to run the parallel version of multiple level AC contingency calculation function with a post-contingency tripping function (use MACCC_WITH_TRIP_2 to run the function sequentially).

Batch command syntax:

```
BAT_MACCC_WITH_TRIP_PARALLEL TOL OPTACC(1)...OPTACC(8) OPTMAC(1)...
OPTMAC(41)

OPTRNK(1)... OPTRNK(17) OPTTRP(1)...OPTTRP(3) VALUES(1)... VALUES(3)

LABELS(1) LABELS(2) DFXFILE ACCFILE THRFILE INLFILE TRPFILE
```

Python syntax:

```
ierr = maccc_with_trip_parallel (tol, optacc, optmac, optrnk, opttrp,
values, labels, dfxfile, accfile, thrfile, inlfile, trpfile)
```

Fortran syntax:

```
CALL MACCC_WITH_TRIP_PARALLEL(TOL, OPTACC, OPTMAC, OPTRNK, OPTTRP,
VALUES, LABELS, DFXFILE, ACCFILE, THRFILE, INLFILE, TRPFILE, IERR)
```

where:

Real TOL	Is the mismatch tolerance (input, Newton solution convergence tolerance, TOLN, by default).
Integer OPTACC(10)	Is an array of ten elements specifying power flow solution options (input). The values are as follows:

OPTACC(1)	tap adjustment flag (tap adjustment option setting by default).
OPTACC(1) = 0	disable.
OPTACC(1) = 1	enable stepping adjustment.
OPTACC(1) = 2	enable direct adjustment.
OPTACC(2)	area interchange adjustment flag (area interchange adjustment option setting by default).
OPTACC(2) = 0	disable.
OPTACC(2) = 1	enable using tie line flows only in calculating area interchange.
OPTACC(2) = 2	enable using tie line flows and loads in calculating area interchange.
OPTACC(3)	phase shift adjustment flag (phase shift adjustment option setting by default).
OPTACC(3) = 0	disable.
OPTACC(3) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTACC(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
OPTACC(4) = 0	disable.
OPTACC(4) = 1	enable.
OPTACC(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
OPTACC(5) = 0	disable.
OPTACC(5) = 1	enable.
OPTACC(5) = 2	enable continuous mode, disable discrete mode.
OPTACC(6)	solution method flag (0 by default).
OPTACC(6) = 0	FDNS.
OPTACC(6) = 1	FNSL.
OPTACC(6) = 2	optimized FDNS.
OPTACC(7)	non-divergent solution flag (non-divergent solution option setting by default).
OPTACC(7) = 0	disable.
OPTACC(7) = 1	enable.
OPTACC(8)	induction motor treatment flag (applied when an induction motor fails to solve due to low terminal bus voltage, 0 by default)
OPTACC(8) = 0	stall.
OPTACC(8) = 1	trip.
OPTACC(9)	induction machine failure flag(0 by default)
OPTACC(9) = 0	treat contingency as non-converged if any induction machines are placed in the “stalled” or “tripped” state.
OPTACC(9) = 1	treat contingency as solved if it converges, even if any induction machines are placed in the “stalled” or “tripped” state.
OPTACC(10)	dispatch mode (0 by default).
OPTACC(10) = 0	disable.
OPTACC(10) = 1	subsystem machines (reserve).
OPTACC(10) = 2	subsystem machines (pmax).
OPTACC(10) = 3	subsystem machines (inertia).
OPTACC(10) = 4	subsystem machines (governor droop).

Integer OPTMAC(41) Is an array of 41 elements to specifying contingency evaluation level and combinations (input). The values are as follows:

OPTMAC(1)	run multiple contingency analysis (1 by default).
OPTMAC(1) = 0	disable.
OPTMAC(1) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTMAC(2)	contingency evaluation level (1 by default)
OPTMAC(2)=1	N-1 contingency analysis
OPTMAC(2)=2 or -2	N-2 contingency analysis
OPTMAC(2)=3 or -3	N-3 contingency analysis

If it is 2 or 3, a newly ordered event list is built for the next level after solving AC power flow each case at current level; if it is -2 or -3, the event list at the first level is used for any higher levels.

OPTMAC(3-41)	each element represents a type of contingency combinations in the following order. =1, include the contingency combinations, otherwise =0 (0 by default).

{S, U, B, SS, SU, SB, US, UU, UB, BS, BU, BB, SSS, SSB, SSU, SBS, SBB, SBU, SUS, SUB, SUU, USS, USU, USB, UUS, UUB, UUU, UBS, UBU, UBB, BSS, BSU, BSB, SUS, SUB, SUU, BBS, BBU, BBB}, where S, U and B are user specified cases, single machine and single branch outages respectively.

Integer OPTRNK(17)

Is an array of 17 elements specifying ranking options (input). The values are as follows:

OPTRNK(1)	include ties from contingency subsystem (0 by default).
OPTRNK(1) = 0	disable.
OPTRNK(1) = 1	enable.
OPTRNK(2)	contingency rating (rating set program option setting by default).
OPTRNK(2) = 1	rate A.
OPTRNK(2) = 2	rate B.
OPTRNK(2) = 3	rate C.
OPTRNK(3)	convert ratings to MW (0 by default).
OPTRNK(3) = 0	disable.
OPTRNK(3) = 1	enable.
OPTRNK(4)	include machine contingencies from overload ranking (0 by default).
OPTRNK(4) = 0	disable.
OPTRNK(4) = 1	enable.
OPTRNK(5)	minimum of machine contingencies to evaluate (>=0; 1 by default).
OPTRNK(6)	maximum of machine contingencies to evaluate (>=0; 10 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTRNK(7)	non-failure cutoff of machine contingencies (>=0; 5 by default).
OPTRNK(8)	include branch contingencies from overload ranking (0 by default).
OPTRNK(8) = 0	disable.
OPTRNK(8) = 1	enable.
OPTRNK(9)	minimum of branch contingencies from overload ranking to evaluate (>=0; 1 by default).
OPTRNK(10)	maximum of branch contingencies from overload ranking to evaluate (>=0; 10 by default).
OPTRNK(11)	non-failure cutoff of branch contingencies from overload ranking (>=0; 5 by default).
OPTRNK(12)	include branch contingencies from voltage depression ranking (0 by default).
OPTRNK(12) = 0	disable.
OPTRNK(12) = 1	enable.
OPTRNK(13)	minimum of branch contingencies from voltage depression ranking to evaluate (>=0; 1 by default).
OPTRNK(14)	maximum of branch contingencies from voltage depression ranking to evaluate (>=0; 10 by default).
OPTRNK(15)	non-failure cutoff of branch contingencies from voltage depression ranking (>=0; 5 by default).
OPTRNK(16)	include all branches in voltage depression ranking (0 by default).
OPTRNK(16) = 0	include only with non-zero rating.
OPTRNK(16) = 1	all.
OPTRNK(17)	include contingencies that create swingless island (0 by default).
OPTRNK(17) = 0	disable.
OPTRNK(17) = 1	enable.

Integer OPTTRP(3) Is an array of three elements specifying tripping simulation options (input). The values are as follows:

OPTTRP(1)	tripping simulation flag (0 by default).
OPTTRP(1) = 0	disable.
OPTTRP(1) = 1	enable.
OPTTRP(2)	number of tripping events per power flow (1 by default).
OPTTRP(3)	number of power flows per contingency (1 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real VALUES(3)	Is an array of three elements specifying ranking parameters (input). The values are as follows: VALUES(1) small reactance threshold (0<= VALUES(1)<=0.005; 0.0005 by default). VALUES(2) percent of rating (0<= VALUES(1)<=200.0; 100.0 by default). VALUES(3) MW threshold for islanding contingencies (>=0; 0.0 by default).
Character LABELS(2)*12	Is an array of two elements specifying subsystem labels (input). The values are as follows: LABELS (1) label of generation dispatch subsystem (blank by default; no default allowed if OPTACC(8) is not 0). LABELS (2) label of contingency subsystem (blank by default).
Character DFXFILE*260	Is the name of the Distribution Factor Data File (input; no default allowed).
Character ACCFILE*260	Is the name of the Contingency Solution Output File (input; no default allowed).
Character THRFILE*260	Is the name of the Load Throwover Data File (input; blank by default).
Character INLFILE*260	Is the name of the Unit Inertia and Governor Data File (input; blank by default).
Character TRPFILE*260	Is the name of the Tripping Element Data File (input, blank for none; no default allowed if OPTTRP(1) is 1).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid TOL value. IERR = 2 invalid OPTACC value. IERR = 3 generators are converted. IERR = 4 buses in island(s) without a swing bus; use activity TREE. IERR = 5 largest mismatch exceeds mismatch tolerance. IERR = 6 generation subsystem is not defined. IERR = 7 too many islands in base case. IERR = 8 in-service induction machines are in the “stalled” or “tripped” state. IERR = 9 buses with bus type code and series element status inconsistencies.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IERR = 10	invalid OPTMAC value.
IERR = 11	invalid OPTRNK value.
IERR = 12	invalid VALUES value.
IERR = 13	invalid OPTTRP value.
IERR = 21	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
IERR = 22	monitored elements exceed limit when adding multi-sec-tion line members.
IERR = 49	error opening Tripping Element Data File.
IERR = 50	error opening Unit Inertia and Governor Data File.
IERR = 51	error opening Contingency Solution Output File.
IERR = 52	error opening Distribution Factor Data File.
IERR = 53	error opening Load Throwover Data File.
IERR = 54	prerequisite requirements for function not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.128 MATRIX_GROWTH_FACTOR

Use this API to specify the value of the matrix growth factor option setting (activity OPTN).

Batch command syntax:

```
BAT_MATRIX_GROWTH_FACTOR  RVAL
```

Python syntax:

```
ierr = matrix_growth_factor(rval)
```

or:

```
ierr,rval = matrix_growth_factor()
```

Fortran syntax:

```
CALL MATRIX_GROWTH_FACTOR_API(RVAL, IOCODE, IERR)
```

where:

Real RVAL	Is the value of the option setting (input if Batch command or for Python if specified as an argument, or for FORTRAN if IOCODE is 0; else output 2.0 by default).
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to the value specified in RVAL. IOCODE = 1 return the current value of the option setting in RVAL.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid RVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.129 MBID2DC

Use this API to change the name of the specified two-terminal dc line (activity MBID).

Batch command syntax:

```
BAT_MBID2DC  DC  DCNEW
```

Python syntax:

```
ierr = mbid2dc(dc, dcnew)
```

Fortran syntax:

```
CALL MBID2DC(NUM, NUMNEW, IERR)

CALL MBID2DC_II(NUM, NUMNEW, IERR)

CALL MBID2DC_IC(NUM, NAMENEW, IERR)

CALL MBID2DC_CI(NAME, NUMNEW, IERR)

CALL MBID2DC_CC(NAME, NAMENEW, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the two-terminal dc line for which the dc line name is to be changed; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Character*12 or Integer DCNEW	Is the quoted new name to be assigned to the two-terminal dc line; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the two-terminal dc line for which the dc line name is to be changed (input; no default allowed).
Integer NUMNEW	Is the new number to be assigned to the two-terminal dc line as a left-justified name (input; no default allowed).
Character*12 NAME	Is the name of the two-terminal dc line for which the dc line name is to be changed (input; no default allowed).
Character*12 NAMENEW	Is the new dc line name to be assigned to the two-terminal dc line (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 dc line DC not found. IERR = 2 dc line DCNEW already exists. IERR = 3 a non-blank new identifier must be specified. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.130 MBID3WND

Use this API to change the identifier of the specified three-winding transformer (activity MBID).

Batch command syntax:

```
BAT_MBID3WND  FRMBUS  TOBUS1  TOBUS2  'ID'  'IDNEW'
```

Python syntax:

```
ierr = mbid3wnd(frmbus, tobus1, tobus2, id, idnew)
```

Fortran syntax:

```
CALL MBID3WND(FRMBUS, TOBUS1, TOBUS2, ID, IDNEW, IERR)
```

where:

Integer FRMBUS	Is one of the buses connected by the three-winding transformer (input; no default allowed).
Integer TOBUS1	Is another bus connected by the three-winding transformer (input; no default allowed).
Integer TOBUS2	Is the other bus connected by the three-winding transformer (input; no default allowed).
Character ID*2	Is the present circuit identifier of the three-winding transformer for which the identifier is to be changed (input; '1' by default).
Character IDNEW*2	Is the new identifier to be assigned to the three-winding transformer (input; no default allowed).
Integer IERR	Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	bus not found.
IERR = 2	circuit ID bus FRMBUS to TOBUS1 to TOBUS2 not found.
IERR = 3	circuit IDNEW bus FRMBUS to TOBUS1 to TOBUS2 already exists.
IERR = 4	invalid new identifier; cannot start with &, * or @.
IERR = 5	a non-blank new identifier must be specified.
IERR = 6	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.131 MBIDATRN

Use this API to change the identifier of the specified inter-area transfer (activity MBID).

Batch command syntax:

```
BAT_MBIDATRN  FRMAREA  TOAREA  'ID'  'IDNEW'
```

Python syntax:

```
ierr = mbidatrn(frmarea, toarea, id, idnew)
```

Fortran syntax:

```
CALL MBIDATRN(FRMAREA, TOAREA, ID, IDNEW, IERR)
```

where:

Integer FRMAREA	Is one of the areas involved in the transfer (input; no default allowed).
Integer TOAREA	Is the other area involved in the transfer (input; no default allowed).
Character ID*1	Is the present inter-area transfer identifier (input; '1' by default).
Character IDNEW*1	Is the new identifier to be assigned to the inter-area transfer (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 — no error occurred.
	IERR = 1 — invalid area number.
	IERR = 2 — inter-area transfer ID from area FRMAREA to TOAREA not found.
	IERR = 3 — inter-area transfer ID from area FRMAREA to TOAREA already exists.
	IERR = 4 — invalid new identifier--must be 0 through 9 or A through Z.
	IERR = 5 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.132 MBIDBRN

Use this API to change the identifier of the specified non-transformer branch or two-winding transformer (activity MBID).

Batch command syntax:

```
BAT_MBIDBRN  FRMBUS  TOBUS  'ID'  'IDNEW'
```

Python syntax:

```
ierr = mbidbrn(frmbus, tobus, id, idnew)
```

Fortran syntax:

```
CALL MBIDBRN(FRMBUS, TOBUS, ID, IDNEW, IERR)
```

where:

Integer FRMBUS	Is one of the buses connected by the branch (input; no default allowed).
Integer TOBUS	Is the other bus connected by the branch (input; no default allowed).
Character ID*2	Is the present circuit identifier of the branch for which the identifier is to be changed (input; '1' by default).
Character IDNEW*2	Is the new identifier to be assigned to the branch (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 bus not found.
	IERR = 2 circuit ID from FRMBUS to TOBUS not found.
	IERR = 3 circuit ID from FRMBUS to TOBUS already exists.
	IERR = 4 invalid new identifier; cannot start with &, * or @.
	IERR = 5 a non-blank new identifier must be specified.
	IERR = 6 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.133 MBIDFACTS

Use this API to change the name of the specified FACTS device (activity MBID).

Batch command syntax:

```
BAT_MBIDFACTS  FD  FDNEW
```

Python syntax:

```
ierr = mbidfacts(fd, fdnew)
```

Fortran syntax:

```
CALL MBIDFACTS(NUM, NUMNEW, IERR)

CALL MBIDFACTS_II(NUM, NUMNEW, IERR)

CALL MBIDFACTS_IC(NUM, NAMENEW, IERR)

CALL MBIDFACTS_CI(NAME, NUMNEW, IERR)

CALL MBIDFACTS_CC(NAME, NAMENEW, IERR)
```

where:

Character*12 or Integer FD	Is the quoted name of the FACTS device for which the dc line name is to be changed; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Character*12 or Integer FDNEW	Is the quoted new name to be assigned to the FACTS device; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the FACTS device for which the dc line name is to be changed (input; no default allowed).
Integer NUMNEW	Is the new number to be assigned to the FACTS device as a left-justified name (input; no default allowed).
Character*12 NAME	Is the name of the FACTS device for which the dc line name is to be changed (input; no default allowed).
Character*12 NAMENEW	Is the new dc line name to be assigned to the FACTS device (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 FACTS device FD not found. IERR = 2 FACTS device FDNEW already exists. IERR = 3 a non-blank new identifier must be specified. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.134 MBIDGNE

Use this API routine to change the GNE device name of the specified GNE device (activity MBID).

Batch command syntax:

```
BAT_MBIDGNE  'NAME'  'NAMNEW'
```

Python syntax:

```
ierr = mbidgne(name, namnew)
```

Fortran syntax:

```
CALL MBIDGNE(NAME, NAMNEW, IERR)
```

where:

Character NAME*12	Is the name of the GNE device for which the GNE device name is to be changed (input; no default allowed).
Character NAMNEW*12	Is the new name to be assigned to the GNE device (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 GNE device 'NAME' not found. IERR = 2 GNE device 'NEWNAM' already exists. IERR = 3 a non-blank new identifier must be specified. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.135 MbidIndMac

Use this API routine to change the identifier of the specified induction machine (activity MBID).

Batch command syntax:

```
BAT_MbidIndMac  FRMBUS  'ID'  'IDNEW'
```

Python syntax:

```
ierr = mbidindmac(frmbus, id, idnew)
```

Fortran syntax:

```
CALL MbidIndMac(FRMBUS, ID, IDNEW, IERR)
```

where:

Integer FRMBUS	Is the bus to which induction machine ID is connected (input; no default allowed).
Character ID*2	Is the present induction machine identifier of the induction machine for which the identifier is to be changed (input; '1' by default).
Character IDNEW*2	Is the new identifier to be assigned to the induction machine (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 bus not found. IERR = 2 induction machine ID at bus FRMBUS not found. IERR = 3 induction machine ID at bus FRMBUS already exists. IERR = 4 a non-blank new identifier must be specified. IERR = 5 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.136 MoveIndMac

Use this API routine to disconnect the specified induction machine from the bus to which it is currently connected, and reconnect it to a designated bus (activity MOVE).

Batch command syntax:

```
BAT_MoveIndMac  FRMBUS  'ID'  NEWBUS  'NEWID'
```

Python syntax:

```
ierr = moveindmac(frmbus, id, newbus, newid)
```

Fortran syntax:

```
CALL MoveIndMac(FRMBUS, ID, NEWBUS, NEWID, IERR)
```

where:

Integer FRMBUS	Is the bus to which induction machine ID is connected (input; no default allowed).
Character ID*2	Is the machine identifier of the induction machine to be moved (input; '1' by default).
Integer NEWBUS	Is the number of the bus to which the induction machine is to be moved (input; ID by default).
Character NEWID*2	Is the new induction machine identifier to be assigned to the relocated induction machine (input; no default allowed).
Integer IERR	Is the error code (output).

IERR = 0	no error occurred
IERR = 1	bus not found
IERR = 2	induction machine not found
IERR = 3	FRMBUS and NEWBUS are the same bus
IERR = 4	induction machine NEWID at bus NEWBUS already exists
IERR = 5	invalid new identifier; must not be blank
IERR = 6	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.137 MoveIndMacs

Use this API routine to disconnect all of the induction machines at the specified bus, and reconnect them to a designated bus (activity MOVE). Bus TOBUS must not have any induction machines connected to it at the time this API routine is called.

Batch command syntax:

```
BAT_MoveIndMacs  FRMBUS  NEWBUS
```

Python syntax:

```
ierr = moveindmacs(frmbus, newbus)
```

Fortran syntax:

```
CALL MoveIndMacs(FRMBUS, NEWBUS, IERR)
```

where:

Integer FRMBUS	Is the bus at which one or more induction machines currently exist (input; no default allowed).
Integer NEWBUS	Is the bus to which the induction machines on bus FRMBUS are to be moved (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 — no error occurred.
	IERR = 1 — bus not found.
	IERR = 2 — no induction machines at bus FRMBUS.
	IERR = 3 — FRMBUS and NEWBUS are the same bus.
	IERR = 4 — NEWBUS already has induction machines (move individual machines).
	IERR = 5 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.138 MBIDLOAD

Use this API to change the identifier of the specified load (activity MBID).

Batch command syntax:

```
BAT_MBIDLOAD  FRMBUS  'ID'  'IDNEW'
```

Python syntax:

```
ierr = mbidload(frmbus, id, idnew)
```

Fortran syntax:

```
CALL MBIDLOAD(FRMBUS, ID, IDNEW, IERR)
```

where:

Integer FRMBUS	Is the bus to which the load ID is connected (input; no default allowed).
Character ID*2	Is the present load identifier of the load for which the identifier is to be changed (input; '1' by default).
Character IDNEW*2	Is the new identifier to be assigned to the load (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 bus not found. IERR = 2 load ID at bus FRMBUS not found. IERR = 3 load ID at bus FRMBUS already exists. IERR = 4 a non-blank new identifier must be specified. IERR = 5 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.139 MBIDMAC

Use this API to change the identifier of the specified machine (activity MBID).

Batch command syntax:

```
BAT_MBIDMAC  FRMBUS  'ID'  'IDNEW'
```

Python syntax:

```
ierr = mbidmac(frmbus, id, idnew)
```

Fortran syntax:

```
CALL MBIDMAC(FRMBUS, ID, IDNEW, IERR)
```

where:

Integer FRMBUS	Is the bus to which the machine ID is connected (input; no default allowed).
Character ID*2	Is the present machine identifier of the machine for which the identifier is to be changed (input; '1' by default).
Character IDNEW*2	Is the new identifier to be assigned to the machine (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 bus not found. IERR = 2 machine ID at bus FRMBUS not found. IERR = 3 machine ID at bus FRMBUS already exists. IERR = 4 a non-blank new identifier must be specified. IERR = 5 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.140 MBIDMDC

Use this API to change the name of the specified multi-terminal dc line (activity MBID).

Batch command syntax:

```
BAT_MBIDMDC  DC  DCNEW
```

Python syntax:

```
ierr = mbidmdc(dc, dcnew)
```

Fortran syntax:

```
CALL MBIDMDC(NUM, NUMNEW, IERR)

CALL MBIDMDC_II(NUM, NUMNEW, IERR)

CALL MBIDMDC_IC(NUM, NAMENEW, IERR)

CALL MBIDMDC_CI(NAME, NUMNEW, IERR)

CALL MBIDMDC_CC(NAME, NAMENEW, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the multi-terminal dc line for which the dc line name is to be changed; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Character*12 or Integer DCNEW	Is the quoted new name to be assigned to the multi-terminal dc line; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the multi-terminal dc line for which the dc line name is to be changed (input; no default allowed).
Integer NUMNEW	Is the new number to be assigned to the multi-terminal dc line as a left-justified name (input; no default allowed).
Character*12 NAME	Is the name of the multi-terminal dc line for which the dc line name is to be changed (input; no default allowed).
Character*12 NAMENEW	Is the new dc line name to be assigned to the multi-terminal dc line (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 dc line DC not found. IERR = 2 dc line DCNEW already exists. IERR = 3 a non-blank new identifier must be specified. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.141 MBIDMSL

Use this API to change the identifier of the specified multi-section line (activity MBID).

Batch command syntax:

```
BAT_MBIDMSL  FRMBUS  TOBUS  'ID'  'IDNEW'
```

Python syntax:

```
ierr = mbidmsl(frmbus, tobus, id, idnew)
```

Fortran syntax:

```
CALL MBIDMSL(FRMBUS, TOBUS, ID, IDNEW, IERR)
```

where:

Integer FRMBUS	Is one of the endpoint buses connected by the multi-section line (input; no default allowed).
Integer TOBUS	Is the other endpoint bus connected by the multi-section line (input; no default allowed).
Character ID*2	Is the present multi-section identifier of the multi-section line for which the identifier is to be changed (input; '&1' by default).
Character IDNEW*2	Is the new identifier to be assigned to the multi-section line (input; no default allowed).
Integer IERR	Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	bus not found.
IERR = 2	multi-section line ID from FRMBUS to TOBUS not found.
IERR = 3	multi-section line ID from FRMBUS to TOBUS already exists.
IERR = 4	invalid new identifier; must start with &.
IERR = 5	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.142 MBIDSHUNT

Use this API to change the identifier of the specified fixed bus shunt (activity MBID).

Batch command syntax:

```
BAT_MBIDSHUNT  FRMBUS  'ID'  'IDNEW'
```

Python syntax:

```
ierr = mbidshunt(frmbus, id, idnew)
```

Fortran syntax:

```
CALL MBIDSHUNT(FRMBUS, ID, IDNEW, IERR)
```

where:

Integer FRMBUS	Is the bus to which the shunt ID is connected (input; no default allowed).
Character ID*2	Is the present shunt identifier of the shunt for which the identifier is to be changed (input; '1' by default).
Character IDNEW*2	Is the new identifier to be assigned to the shunt (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 bus not found. IERR = 2 shunt ID at bus FRMBUS not found. IERR = 3 shunt ID at bus FRMBUS already exists. IERR = 4 a non-blank new identifier must be specified. IERR = 5 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.143 MBIDVSC

Use this API to change the VSC dc line name of the specified VSC dc line (activity MBID).

Batch command syntax:

```
BAT_MBIDVSC  'NAME'  'NAMNEW'
```

Python syntax:

```
ierr = mbidvsc(name, namnew)
```

Fortran syntax:

```
CALL MBIDVSC(NAME, NAMNEW, IERR)
```

where:

Character NAME*12	Is the name of the VSC dc line for which the VSC dc line name is to be changed (input; no default allowed).
Character NAMNEW*12	Is the new name to be assigned to the VSC dc line (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 dc line 'NAME' not found.
	IERR = 2 dc line 'NEWNAM' already exists.
	IERR = 3 a non-blank new identifier must be specified.
	IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.144 MCRE

Use this API to read a Machine Impedance Data File and add the data specified in it to the working case (activity MCRE).

Batch command syntax:

```
BAT_MCRE  STATUS(1)  STATUS(2)  IFILE
```

Python syntax:

```
ierr = mcre(status, ifile)
```

Fortran syntax:

```
CALL MCREAPI(STATUS, IFILE, IERR)
```

where:

Integer STATUS(2)	Is an array of two elements specifying MCRE options (input)	
	STATUS(1)	new machine status option (0 by default).
	STATUS(1) = 0	set from STAT field on input record.
	STATUS(1) = 1	set from status of other machines at the bus.
	STATUS(2)	option for treatment of machines with no data record specified and other machines at the bus with record specified (0 by default).
	STATUS(2) = 0	place out-of-service.
	STATUS(2) = 1	delete.
Character IFILE*260	Is the filename of Machine Impedance Data File (input; no default allowed).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid STATUS value.
	IERR = 2	generators are converted.
	IERR = 3	error opening IFILE.
	IERR = 4	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.145 MODR

Use this API to uniformly increase or decrease the line resistances of in-service nontransformer branches (activity MODR). It **must** first be called once with APIOPT set to 1, then once for each subsystem to be processed with APIOPT set to 2, and finally once with APIOPT set to 3.

Batch command syntax:

```
BAT_MODR  SID  ALL  APIOPT  IRAT  RLOAD  FMAX
```

Python syntax:

```
ierr,nchng = modr(sid, all, apiopt, irat, rload, fmax)
```

Fortran syntax:

```
CALL MODRNW(SID, ALL, APIOPT, IRAT, RLOAD, FMAX, NCHNG, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; used only when APIOPT is 2; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; used only when APIOPT is 2; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Integer APIOPT	Is the mode of operation of the API (input; no default allowed). APIOPT = 1 initialize system for line resistance modification. APIOPT = 2 modify line resistances. APIOPT = 3 postprocessing housekeeping.
Integer IRAT	Is the rating set on which the present values of branch resistances are based (input; used only when APIOPT is 1; default rating set program option setting by default). IRAT = 1 RATEA. IRAT = 2 RATEB. IRAT = 3 RATEC.
Real RLOAD	Is the base line loading in percent. 0.0001 ≤ RLOAD ≤ 2000.0 (input; used only when APIOPT is 1; 100.0 by default).
Real FMAX	Is the line resistance scaling factor in pu 0.0 ≤ FMAX ≤ 0.1 (input; used only when APIOPT is 1; 0.01 by default).
Integer NCHNG	Is **returned** as the number of branch resistances changed.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 invalid APIOPT value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IERR = 4	invalid IRAT value.
IERR = 5	invalid RLOAD value.
IERR = 6	invalid FMAX value.
IERR = 7	unexpected APIOPT value.
IERR = 8	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.146 MOVE3WND

Use this API to disconnect the third bus of a specified three-winding transformer from the bus to which it is currently connected, and reconnect it to a designated bus (activity MOVE).

Batch command syntax:

```
BAT_MOVE3WND  FRMBUS  TOBUS1  TOBUS2  'CKT'  NEWTOBUS2  'NEWCKT'
```

Python syntax:

```
ierr = move3wnd(frmbus, tobus1, tobus2, ckt, newtobus2, newckt)
```

Fortran syntax:

```
CALL MOVE3WND(FRMBUS, TOBUS1, TOBUS2, CKT, NEWTOBUS2, NEWCKT, IERR)
```

where:

Integer FRMBUS	Is one of the buses connected by the three-winding transformer; the transformer will remain connected to this bus (input; no default allowed).
Integer TOBUS1	Is another bus connected by the three-winding transformer; the transformer will remain connected to this bus (input; no default allowed).
Integer TOBUS2	Is the third bus connected by the three-winding transformer; this connection will be severed and the transformer connected to bus NEWTOBUS2 (input; no default allowed).
Character CKT*2	Is the circuit identifier of the three-winding transformer to be rerouted (input; '1' by default).
Integer NEWTOBUS2	Is the number of the bus to which the three-winding transformer is to be connected (input; no default allowed).
Character NEWCKT*2	Is the new circuit identifier to be assigned to the rerouted three-winding transformer (input; CKT by default).
Integer IERR	Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	bus not found.
IERR = 2	three-winding transformer not found.
IERR = 3	new connection bus must not be the present from or to buses.
IERR = 4	circuit NEWCKT from FRMBUS to TOBUS1 to NEWTOBUS2 already exists.
IERR = 5	invalid new identifier; cannot start with &, * or @.
IERR = 6	invalid new identifier; must not be blank.
IERR = 7	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.147 MOVEBRN

Use this API to disconnect the to bus of a specified non-transformer branch or two-winding transformer from the bus to which it is currently connected, and reconnect it to a designated bus (activity MOVE).

Batch command syntax:

```
BAT_MOVEBRN  FRMBUS  TOBUS  'CKT'  NEWTOBUS  'NEWCKT'
```

Python syntax:

```
ierr = movebrn(frmbus, tobus, ckt, newtobus, newckt)
```

Fortran syntax:

```
CALL MOVEBRN(FRMBUS, TOBUS, CKT, NEWTOBUS, NEWCKT, IERR)
```

where:

Integer FRMBUS	Is one of the buses connected by the branch; the branch will remain connected to this bus (input; no default allowed).
Integer TOBUS	Is the other bus connected by the branch; this connection will be severed and the branch connected to bus NEWTOBUS (input; no default allowed).
Character CKT*2	Is the circuit identifier of the branch to be rerouted (input; '1' by default).
Integer NEWTOBUS	Is the number of the bus to which the branch is to be connected (input; no default allowed).
Character NEWCKT*2	Is the new circuit identifier to be assigned to the rerouted branch (input; CKT by default).
Integer IERR	Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	bus not found.
IERR = 2	circuit CKT from FRMBUS to TOBUS not found.
IERR = 3	new connection bus must not be the present from or to bus.
IERR = 4	circuit NEWCKT from FRMBUS to NEWTOBUS already exists.
IERR = 5	invalid new identifier; cannot start with &, * or @.
IERR = 6	invalid new identifier; must not be blank.
IERR = 7	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.148 MOVELOAD

Use this API to disconnect the specified load from the bus to which it is currently connected, and reconnect it to a designated bus (activity MOVE).

Batch command syntax:

```
BAT_MOVELOAD  FRMBUS  'ID'  NEWBUS  'NEWID'
```

Python syntax:

```
ierr = moveload(frmbus, id, newbus, newid)
```

Fortran syntax:

```
CALL MOVELOAD(FRMBUS, ID, NEWBUS, NEWID, IERR)
```

where:

Integer FRMBUS	Is the bus to which the load ID is connected (input; no default allowed).
Character ID*2	Is the load identifier of the load to be moved (input; '1' by default).
Integer NEWBUS	Is the number of the bus to which the load is to be moved (input; no default allowed).
Character NEWID*2	Is the new load identifier to be assigned to the relocated load (input; ID by default).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 bus not found.
	IERR = 2 load not found.
	IERR = 3 FRMBUS and NEWBUS are the same bus.
	IERR = 4 load NEWID at bus NEWBUS already exists.
	IERR = 5 invalid new identifier; must not be blank.
	IERR = 6 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.149 MOVELOADS

Use this API to disconnect all of the load from the specified bus, and reconnect it to a designated bus (activity MOVE). Bus TOBUS must not have any load connected to it at the time this API is called.

Batch command syntax:

```
BAT_MOVELOADS  FRMBUS  NEWBUS
```

Python syntax:

```
ierr = moveloads(frmbus, newbus)
```

Fortran syntax:

```
CALL MOVELOADS(FRMBUS, NEWBUS, IERR)
```

where:

Integer FRMBUS	Is the bus on which one or more loads currently exist (input; no default allowed).
Integer NEWBUS	Is the bus to which the loads on bus FRMBUS are to be moved (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 — no error occurred.
	IERR = 1 — bus not found.
	IERR = 2 — no loads at bus FRMBUS.
	IERR = 3 — FRMBUS and NEWBUS are the same bus.
	IERR = 4 — NEWBUS already has loads (move individual loads).
	IERR = 5 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.150 MOVEMAC

Use this API to disconnect the specified machine from the bus to which it is currently connected, and reconnect it to a designated bus (activity MOVE).

Batch command syntax:

```
BAT_MOVEMAC  FRMBUS  'ID'  NEWBUS  'NEWID'
```

Python syntax:

```
ierr = movemac(frmbus, id, newbus, newid)
```

Fortran syntax:

```
CALL MOVEMAC(FRMBUS, ID, NEWBUS, NEWID, IERR)
```

where:

Integer FRMBUS	Is the bus to which the machine ID is connected (input; no default allowed).
Character ID*2	Is the machine identifier of the machine to be moved (input; '1' by default).
Integer NEWBUS	Is the number of the bus to which the machine is to be moved (input; ID by default).
Character NEWID*2	Is the new machine identifier to be assigned to the relocated machine (input; no default allowed).
Integer IERR	Is the error code (output).

IERR = 0	no error occurred
IERR = 1	bus not found
IERR = 2	machine not found
IERR = 3	FRMBUS and NEWBUS are the same bus
IERR = 4	machine NEWID at bus NEWBUS already exists
IERR = 5	plant table full (move machine to an existing plant)
IERR = 6	invalid new identifier; must not be blank
IERR = 7	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.151 MOVEPLNT

Use this API to disconnect all of the machines from the specified bus, and reconnect it to a designated bus (activity MOVE). Bus TOBUS must not have any plant data associated with it at the time this API is called.

Batch command syntax:

```
BAT_MOVEPLNT  FRMBUS  NEWBUS
```

Python syntax:

```
ierr = moveplnt(frmbus, newbus)
```

Fortran syntax:

```
CALL MOVEPLNT(FRMBUS, NEWBUS, IERR)
```

where:

Integer FRMBUS	Is the bus on which plant data currently exists (input; no default allowed).
Integer NEWBUS	Is the bus to which the plant on bus FRMBUS is to be moved (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 — no error occurred.
	IERR = 1 — bus not found.
	IERR = 2 — no plant at bus FRMBUS.
	IERR = 3 — FRMBUS and NEWBUS are the same bus.
	IERR = 4 — NEWBUS already has plant data (move individual machines).
	IERR = 5 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.152 MOVESHUNT

Use this API to disconnect the specified fixed shunt from the bus to which it is currently connected, and reconnect it to a designated bus (activity MOVE).

Batch command syntax:

```
BAT_MOVESHUNT  FRMBUS  'ID'  NEWBUS  'NEWID'
```

Python syntax:

```
ierr = moveshunt(frmbus, id, newbus, newid)
```

Fortran syntax:

```
CALL MOVESHUNT(FRMBUS, ID, NEWBUS, NEWID, IERR)
```

where:

Integer FRMBUS	Is the bus to which the shunt ID is connected (input; no default allowed).
Character ID*2	Is the shunt identifier of the fixed shunt to be moved (input; '1' by default).
Integer NEWBUS	Is the number of the bus to which the shunt is to be moved (input; no default allowed).
Character NEWID*2	Is the new shunt identifier to be assigned to the relocated fixed shunt (input; ID by default).
Integer IERR	Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	bus not found.
IERR = 2	fixed shunt not found.
IERR = 3	FRMBUS and NEWBUS are the same bus.
IERR = 4	shunt NEWID at bus NEWBUS already exists.
IERR = 5	invalid new identifier; must not be blank.
IERR = 6	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.153 MOVESHUNTS

Use this API to disconnect all of the fixed shunts from the specified bus and reconnect them to a designated bus (activity MOVE). Bus TOBUS must not have any fixed shunt connected to it at the time this API is called.

Batch command syntax:

```
BAT_MOVESHUNTS  FRMBUS  NEWBUS
```

Python syntax:

```
ierr = moveshunts(frmbus, newbus)
```

Fortran syntax:

```
CALL MOVESHUNTS(FRMBUS, NEWBUS, IERR)
```

where:

Integer FRMBUS	Is the bus on which one or more fixed shunts currently exist (input; no default allowed).
Integer NEWBUS	Is the bus to which the fixed shunts on bus FRMBUS are to be moved (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 — no error occurred.
	IERR = 1 — bus not found.
	IERR = 2 — no fixed shunts at bus FRMBUS.
	IERR = 3 — FRMBUS and NEWBUS are the same bus.
	IERR = 4 — NEWBUS already has fixed shunts (move individual shunts).
	IERR = 5 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.154 MOVESWS

Use this API to disconnect the switched shunt load from the specified bus and reconnect it to a designated bus (activity MOVE).

Batch command syntax:

```
BAT_MOVESWS  FRMBUS  NEWBUS
```

Python syntax:

```
ierr = movesws(frmbus, newbus)
```

Fortran syntax:

```
CALL MOVESWS(FRMBUS, NEWBUS, IERR)
```

where:

Integer FRMBUS	Is the bus on which a switched shunt currently exists (input; no default allowed).
Integer NEWBUS	Is the bus to which the switched shunt on bus FRMBUS is to be moved (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 bus not found. IERR = 2 no switched shunt at bus FRMBUS. IERR = 3 FRMBUS and NEWBUS are the same bus. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.155 MSLV

Use this API to apply the modified Gauss-Seidel power flow calculation (activity MSLV).

Batch command syntax:

```
BAT_MSLV  OPTIONS(1)...OPTIONS(6)
```

Python syntax:

```
ierr = mslv(options)
```

Fortran syntax:

```
CALL MSLVAPI(OPTIONS, IERR)
```

where:

Integer OPTIONS(6)	Is an array of six elements specifying solution options (input). The values are as follows:	
	OPTIONS(1)	tap adjustment flag (use tap adjustment option setting by default).
	OPTIONS(1) = 0	disable.
	OPTIONS(1) = 1	enable.
	OPTIONS(2)	area interchange adjustment flag (use area interchange adjustment option setting by default).
	OPTIONS(2) = 0	disable.
	OPTIONS(2) = 1	enable using tie line flows only in calculating area interchange.
	OPTIONS(2) = 2	enable using tie line flows and loads in calculating area interchange.
	OPTIONS(3)	flat start flag (0 by default).
	OPTIONS(3) = 0	do not flat start.
	OPTIONS(3) = 1	flat start.
	OPTIONS(4)	dc tap adjustment flag (use dc tap adjustment option setting by default).
	OPTIONS(4) = 0	disable.
	OPTIONS(4) = 1	enable.
	OPTIONS(5)	switched shunt adjustment flag (use switched shunt adjustment option setting by default).
	OPTIONS(5) = 0	disable.
	OPTIONS(5) = 1	enable.
	OPTIONS(5) = 2	enable continuous mode, disable discrete mode.
	OPTIONS(6)	var limit flag (0 by default).
	OPTIONS(6) = 0	apply var limits.
	OPTIONS(6) = 1	ignore var limits.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid OPTIONS value.
	IERR = 2	generators are converted.
	IERR = 3	buses in island(s) without a swing bus; use activity TREE.
	IERR = 4	bus type code and series element status inconsistencies.
	IERR = 5	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.156 MSUM

Use this API to print a summary of mileage by owner (activity ALOC).

Batch command syntax:

```
BAT_MSUM
```

Python syntax:

```
ierr = msum()
```

Fortran syntax:

```
CALL MSUMAPI(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 4	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.157 MTDC

Use this API to produce a report of bus voltages and flows for each in-service multi-terminal dc line (activity MTDC).

Batch command syntax:

```
BAT_MTDC
```

Python syntax:

```
ierr = mtdc()
```

Fortran syntax:

```
CALL MTDCAPI(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	no unblocked multi-terminal dc lines in this case.
	IERR = 2	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.158 MULTISECTION_REPORTING

Use this API to enable or disable the multi-section line reporting option setting (activity OPTN).

Batch command syntax:

```
BAT_MULTISECTION_REPORTING  IVAL
```

Python syntax:

```
ierr = multisection_reporting(ival)
```

or:

```
ierr,ival = multisection_reporting()
```

Fortran syntax:

```
CALL MULTISECTION_LINE_REPORTING_API(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Is the value of the option setting (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 1 by default for input).	
	IVAL = 0	off.
	IVAL = 1	on.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE = 0	set the option setting to the value specified in IVAL.
	IOCODE = 1	return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid IVAL value.
	IERR = 2	invalid IOCODE value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.159 MWMI

Use this API to apply the MAPP MW-mile calculation (activity MWMI).

Batch command syntax:

```
BAT_MWMI  SID  ALL  STATUS(1)...STATUS(4)  VALUES(1)...VALUES(3)
INPFILE  TRNFILE  OUTFILE
```

Python syntax:

```
ierr = mwmi(sid, all, status, values, inpfile, trnfile, outfile)
```

Fortran syntax:

```
CALL MWMIAPI(SID, ALL, STATUS, VALUES, INPFILE, TRNFILE, OUTFILE,
IERR)
```

where:

Integer SID	Is a valid area subsystem identifier. Valid subsystem identifiers range from 0 to 11. Area subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all areas* or specified subsystem flag (input; 1 by default).	
	ALL = 1	all areas are in the pool.
	ALL = 0	only areas in area subsystem SID are in the pool.
Integer STATUS(4)	Is an array of four elements (input). The value of each element is as follows.	
	STATUS(1)	is rating used (default rating set program option setting by default).
	STATUS(1) = 1	RATEA.
	STATUS(1) = 2	RATEB.
	STATUS(1) = 3	RATEC.
	STATUS(2)	is the option to list input data (0 by default).
	STATUS(2) = 0	no.
	STATUS(2) = 1	yes.
	STATUS(3)	is the option to include buyer and seller branches in the detailed branch output file (0 by default).
	STATUS(3) = 0	no.
	STATUS(3) = 1	yes.
	STATUS(4)	is action code if the MW mismatch exceeds 0.5 MW (0 by default).
	STATUS(4) = 0	discontinue.
	STATUS(4) = 1	continue.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real VALUES(3)	Is an array of three elements specifying solution parameters (input).	
	VALUES(1)	percent load reduction for calculating capacity for serving load (10.0 by default).
	VALUES(2)	MW for each transaction; used when no Transactions Data Input File is specified (10.0 by default).
	VALUES(3)	distribution factor threshold for detailed branch output (0.0 by default)..
Character INPFILE*260	Is the name of MWMI Data Input File (input; no default allowed).	
Character TRNFILE*260	Is the name of the Transactions Raw Data File; blank for none (input; blank by default).	
Character OUTFILE*260	Is the name of the Detailed Branch Output File; blank for none (input; blank by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	invalid STATUS value.
	IERR = 4	invalid VALUES value.
	IERR = 5	INPFILE is blank.
	IERR = 6	generators are converted.
	IERR = 7	buses in island(s) without a swing bus; use activity TREE.
	IERR = 8	largest mismatch exceeds 0.5 MW.
	IERR = 9	error opening INPFILE.
	IERR = 10	error opening OUTFILE.
	IERR = 11	error opening TRNFILE.
	IERR = 12	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.160 MWOMWMSF

Use this API to tabulate generation (MW) on MW-mile shift factors for a specified transaction event (activity ALOC).

Batch command syntax:

```
BAT_MWOMWMSF  EVENT
```

Python syntax:

```
ierr = mwomwmsf(event)
```

Fortran syntax:

```
CALL MWOMWMSF(EVENT, IERR)
```

where:

Integer EVENT	Is the transaction event number (input; no default allowed).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	no transactions defined.
	IERR = 2	invalid transaction event number.
	IERR = 3	transaction event not defined.
	IERR = 4	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.161 MWOMWOSF

Use this API to tabulate generation (MW) on MW-ohm shift factors for a specified transaction event (activity ALOC).

Batch command syntax:

```
BAT_MWOMWOSF  EVENT
```

Python syntax:

```
ierr = mwomwosf(event)
```

Fortran syntax:

```
CALL MWOMWOSF(EVENT, IERR)
```

where:

Integer EVENT	Is the transaction event number (input; no default allowed).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	no transactions defined.
	IERR = 2	invalid transaction event number.
	IERR = 3	transaction event not defined.
	IERR = 4	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.162 N11_ACCC

The API is obsolete. It has been replaced by N11_ACCC_2 and is implemented by a call to N11_ACCC_2.

This API is used to perform N-1-1 contingency analysis. Each pair of N-1-1 contingencies is simulated in the way that the first N-1 contingency is simulated by a full AC load flow solution with the power flow options for the first N-1 contingency; the post solution of the first N-1 contingency is used as the base case for the second N-1 contingency, the second N-1 contingency is imposed and solved by a full AC load flow solution with power flow options for the second N-1 contingency.

Batch command syntax:

```
BAT_N11_ACCC OPTIONS(1)…OPTIONS(18) VALUES(1) VALUES(2) LABEL DFXFILE
ACCFILE CONFILE INLFILE THRFILE
```

Python syntax:

```
ierr = n11_accc(options,values,label,dfxfile,accfile,confile,inlfile,
thrfile)
```

Fortran syntax:

```
Call N11_ACCC(OPTIONS,VALUES,LABEL,DFXFILE,ACCFILE,CONFILE,INLFILE,
THRFILE,IERR)
```

where:

Integer OPTIONS(18)	Is an integer array of eighteen elements specifying power flow options

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(1),	tap adjustment flag for the first N-1 contingency (tap adjustment option setting by default).
OPTIONS(1) = 0,	disable
OPTIONS(1) = 1,	enable stepping adjustment
OPTIONS(1) = 2,	enable direct adjustment
OPTIONS(2),	area interchange adjustment flag for the first N-1 contingency (area interchange adjustment option setting by default)
OPTIONS(2) = 0,	disable
OPTIONS(2) = 1,	enable using tie line flows only in calculating area interchange
OPTIONS(2) = 2,	enable using tie line flows and loads in calculating area interchange
OPTIONS(3),	phase shift adjustment flag for the first N-1 contingency (phase shift adjustment option setting by default)
OPTIONS(3) = 0,	disable
OPTIONS(3) = 1,	enable
OPTIONS(4),	dc tap adjustment flag for the first N-1 contingency (dc tap adjustment option setting by default).
OPTIONS(4) = 0,	disable
OPTIONS(4) = 1,	enable

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(5),	switched shunt adjustment flag for the first N-1 contingency (switched shunt adjustment option setting by default)
OPTIONS(5)= 0,	disable
OPTIONS(5)= 1,	enable all
OPTIONS(5)=2,	enable continuous, disable discrete.
OPTIONS(6)	induction motor treatment flag for the first N-1 contingency (applied when an induction motor fails to solve due to low terminal bus voltage, 0 by default)
OPTIONS(6) = 0,	stall.
OPTIONS(6) = 1,	trip.
OPTIONS(7),	solution method flag (0 by default).
OPTIONS(7)= 0,	Fixed slope decoupled Newton-Raphson method
OPTIONS(7)= 1,	Full Newton-Raphson method
OPTIONS(7)=2,	Optimized fixed slope decoupled Newton-Raphson method
OPTIONS(8),	non-divergent solution flag (non-divergent solution option setting by default).
OPTIONS(8)=0,	Disable
OPTIONS(8)=1,	Enable
OPTIONS(9),	tap adjustment flag for the second N-1 contingency (tap adjustment option setting by default).
OPTIONS(9)= 0,	disable.
OPTIONS(9)= 1,	enable stepping adjustment.
OPTIONS(9)= 2,	enable direct adjustment.
OPTIONS(10),	area interchange adjustment flag for the second N-1 contingency (area interchange adjustment option setting by default).
OPTIONS(10) = 0,	disable.
OPTIONS(10) = 1,	enable using tie line flows only in calculating area interchange.
OPTIONS(10) = 2,	enable using tie line flows and loads in calculating area interchange.
OPTIONS(11),	phase shift adjustment flag for the second N-1 contingency (phase shift adjustment option setting by default).
OPTIONS(11) = 0,	disable.
OPTIONS(11) = 1,	enable.
OPTIONS(12),	dc tap adjustment flag for the second N-1 contingency (dc tap adjustment option setting by default).
OPTIONS(12) = 0,	disable.
OPTIONS(12) = 1,	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(13), switched shunt adjustment flag for the second N-1 contingency (switched shunt adjustment option setting by default).

OPTIONS(13) = 0, disable.

OPTIONS(13) = 1, enable.

OPTIONS(13) =2, enable continuous, disable discrete.

OPTIONS(14) induction motor treatment flag for the second N-1 contingency (applied when an induction motor fails to solve due to low terminal bus voltage, 0 by default)

OPTIONS(14) = 0, stall.

OPTIONS(14) = 1, trip.

OPTIONS(15) = 1, subsystem machines (RESERVE).

OPTIONS(15) = 2, subsystem machines (PMAX).

OPTIONS(15) = 3, subsystem machines (INERTIA).

OPTIONS(15) = 4, subsystem machines (GOVERNOR DROOP).

OPTIONS(16), flag to skip the second N-1 contingencies if the first N-1 contingency is not clean

OPTIONS(16)=0, No

OPTIONS(16)=1, Yes

OPTIONS(17),Rating set for the first N-1 contingency case

OPTIONS(17)=1, RATE A

OPTIONS(17)=2, RATE B

OPTIONS(17)=3, RATE C

OPTIONS(18), Rating set for the second N-1 contingency case

OPTIONS(18)=1, RATE A

OPTIONS(18)=2, RATE B

OPTIONS(18)=3, RATE C

Real Values(2) VALUES Is a real array of 2 elements

VALUES (1)= Mismatch tolerance (TOLN by default)

VALUES (2)= Percent of rating for the first N-1 contingency and second N-1 contingency case (100.0 by default)

Character LABEL*32 The label of generation dispatch system, required if the dispatch mode of OPTIONS(15) is not zero. The dispatch mode and the system provide for a default dispatch method for contingencies causing unbalance between load demand and generation output.

Character DFXFILE*260 Is the name of the Distribution Factor Data File (input, no default allowed).

Character ACCFILE*260 Is the name of contingency analysis output file (input, no default allowed)

Character CONFILE*260 Is the name of Contingency Description Data File for the second N-1 contingencies (input, blank for none)

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character THRFILE*260	Is the name of Load Throwover Data File (input, blank for none).	
Character INLFILE*260	Is the name of Unit Inertia Data File (input, blank for none).	
Integer IERR	Is the error code (OUTPUT)	
	IERR = 0	No error
	IERR=1,	Invalid OPTIONS value
	IERR=2,	Invalid VALUES value
	IERR=3,	Generators are converted
	IERR=4,	Buses in island(s) without a swing bus--use activity TREE
	IERR=5,	Largest mismatch exceeds tolerance
	IERR=6,	Generation subsystem not specified
	IERR=7,	Too many islands in base case.
	IERR=8,	in-service induction machines are in the "stalled" or "tripped" state.
	IERR=9,	buses with bus type code and series element status inconsistencies.
	IERR=10,	No Distribution Factor Data input file specified
	IERR=11,	No Contingency Analysis Output File specified.
	IERR=12,	Can not read the Contingency Description Data File for the second N-1 contingencies
	IERR=13,	No second N-1 contingencies are specified
	IERR=14	Invalid base/contingency case flag combination.
	IERR=21,	File is not in the form of a PSS/E-25 or later DFAX file--run DFAX
	IERR=22,	Monitored elements exceed limit when adding multi-section line members
	IERR=51,	Error opening distribution factor data file
	IERR=52,	Error opening load throwover data file
	IERR=53,	Error opening inertia data file
	IERR=54,	Prerequisite requirements for function not met

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.163 N11_ACCC_2

This API is the second release of N-1-1 contingency analysis solution. This API is used to perform N-1-1 contingency analysis. Each pair of N-1-1 contingencies is simulated in the way that the first N-1 contingency is simulated by a full AC load flow solution with the power flow options for the first N-1 contingency; the post solution of the first N-1 contingency is used as the base case for the second N-1 contingency, the second N-1 contingency is imposed and solved by a full AC load flow solution with power flow options for the second N-1 contingency.

Batch command syntax:

```
BAT_N11_ACCC_2 OPTIONS(1)…OPTIONS(20) VALUES(1) VALUES(2) LABEL
DFXFILE ACCFILE CONFILE INLFILE THRFILE
```

Python syntax:

```
ierr =
n11_accc_2(options,values,label,dfxfile,accfile,confile,inlfile,
thrfile)
```

Fortran syntax:

```
Call N11_ACCC_2(OPTIONS,VALUES,LABEL,DFXFILE,ACCFILE,CONFILE,INLFILE,
THRFILE,IERR)
```

where:

Integer OPTIONS(20)	Is an integer array of twenty elements specifying power flow options	
	OPTIONS(1)	tap adjustment flag for the first N-1 contingency (tap adjustment option setting by default).
	OPTIONS(1) = 0,	disable
	OPTIONS(1) = 1,	enable stepping adjustment
	OPTIONS(1) = 2,	enable direct adjustment
	OPTIONS(2)	area interchange adjustment flag for the first N-1 contingency (area interchange adjustment option setting by default)
	OPTIONS(2) = 0,	disable
	OPTIONS(2) = 1,	enable using tie line flows only in calculating area interchange
	OPTIONS(2) = 2,	enable using tie line flows and loads in calculating area interchange
	OPTIONS(3)	phase shift adjustment flag for the first N-1 contingency (phase shift adjustment option setting by default)
	OPTIONS(3) = 0,	disable
	OPTIONS(3) = 1,	enable
	OPTIONS(4)	dc tap adjustment flag for the first N-1 contingency (dc tap adjustment option setting by default).
	OPTIONS(4) = 0,	disable
	OPTIONS(4) = 1,	enable

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(5)	switched shunt adjustment flag for the first N-1 contingency (switched shunt adjustment option setting by default)
OPTIONS(5)= 0,	disable
OPTIONS(5)= 1,	enable all
OPTIONS(5)=2,	enable continuous, disable discrete.
OPTIONS(6)	induction motor treatment flag for the first N-1 contingency (applied when an induction motor fails to solve due to low terminal bus voltage, 0 by default)
OPTIONS(6) = 0,	stall.
OPTIONS(6) = 1,	trip.
OPTIONS(7)	solution method flag (0 by default).
OPTIONS(7)= 0,	Fixed slope decoupled Newton-Raphson method
OPTIONS(7)= 1,	Full Newton-Raphson method
OPTIONS(7)=2,	Optimized fixed slope decoupled Newton-Raphson method
OPTIONS(8)	non-divergent solution flag (non-divergent solution option setting by default).
OPTIONS(8)=0,	Disable
OPTIONS(8)=1,	Enable
OPTIONS(9)	tap adjustment flag for the second N-1 contingency (tap adjustment option setting by default).
OPTIONS(9)= 0,	disable.
OPTIONS(9)= 1,	enable stepping adjustment.
OPTIONS(9)= 2,	enable direct adjustment.
OPTIONS(10)	area interchange adjustment flag for the second N-1 contingency (area interchange adjustment option setting by default).
OPTIONS(10) = 0,	disable.
OPTIONS(10) = 1,	enable using tie line flows only in calculating area interchange.
OPTIONS(10) = 2,	enable using tie line flows and loads in calculating area interchange.
OPTIONS(11)	phase shift adjustment flag for the second N-1 contingency (phase shift adjustment option setting by default).
OPTIONS(11) = 0,	disable.
OPTIONS(11) = 1,	enable.
OPTIONS(12)	dc tap adjustment flag for the second N-1 contingency (dc tap adjustment option setting by default).
OPTIONS(12) = 0,	disable.
OPTIONS(12) = 1,	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(13), switched shunt adjustment flag for the second N-1 contingency (switched shunt adjustment option setting by default).

OPTIONS(13) = 0, disable.

OPTIONS(13) = 1, enable.

OPTIONS(13) =2, enable continuous, disable discrete.

OPTIONS(14) induction motor treatment flag for the second N-1 contingency (applied when an induction motor fails to solve due to low terminal bus voltage, 0 by default)

OPTIONS(14) = 0, stall.

OPTIONS(14) = 1, trip.

OPTIONS(15) generation dispatch method

OPTIONS(15) = 1, subsystem machines (RESERVE).

OPTIONS(15) = 2, subsystem machines (PMAX).

OPTIONS(15) = 3, subsystem machines (INERTIA).

OPTIONS(15) = 4, subsystem machines (GOVERNOR DROOP).

OPTIONS(16) flag to skip the second N-1 contingencies if the first N-1 contingency is not clean

OPTIONS(16)=0, No

OPTIONS(16)=1, Yes

OPTIONS(17) Rating set for the first N-1 contingency case

OPTIONS(17)=1, rate A

OPTIONS(17)=2, rate B

OPTIONS(17)=3, rate C

OPTIONS(18) Rating set for the second N-1 contingency case

OPTIONS(18)=1, rate A

OPTIONS(18)=2, rate B

OPTIONS(18)=3, rate C

OPTIONS(19) voltage limit for the first N-1 contingency case

OPTIONS(19)=1, normal

OPTIONS(19)=2, emergency

OPTIONS(20) voltage limit for the second N-1 contingency case

OPTIONS(20)=1, normal

OPTIONS(20)=2, emergency

Real Values(2) is a real array of 2 elements

VALUES (1)= Mismatch tolerance (TOLN by default)

VALUES (2)= Percent of rating for the first N-1 contingency and second N-1 contingency case (100.0 by default)

Character LABEL*32 The label of generation dispatch system, required if the dispatch mode of OPTIONS(15) is not zero. The dispatch mode and the system provide for a default dispatch method for contingencies causing unbalance between load demand and generation output.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character DFXFILE*260	Is the name of the Distribution Factor Data File (input, no default allowed).
Character ACCFILE*260	Is the name of contingency analysis output file (input, no default allowed)
Character CONFILE*260	Is the name of Contingency Description Data File for the second N-1 contingencies (input, blank for none)
Character THRFILE*260	Is the name of Load Throwover Data File (input, blank for none).
Character INLFILE*260	Is the name of Unit Inertia Data File (input, blank for none).
Integer IERR	Is the error code (OUTPUT)

IERR=0	No error
IERR=1	Invalid OPTIONS value
IERR=2	Invalid VALUES value
IERR=3	Generators are converted
IERR=4	Buses in island(s) without a swing bus--use activity TREE
IERR=5	Largest mismatch exceeds tolerance
IERR=6	Generation subsystem not specified
IERR=7	Too many islands in base case.
IERR=8	in-service induction machines are in the "stalled" or "tripped" state.
IERR=9	buses with bus type code and series element status inconsistencies.
IERR=10	No Distribution Factor Data input file specified
IERR=11	No Contingency Analysis Output File specified.
IERR=12	Can not read the Contingency Description Data File for the second N-1 contingencies
IERR=13	No second N-1 contingencies are specified
IERR=14	Invalid base/contingency case flag combination.
IERR=21	File is not in the form of a PSS/E-25 or later DFAX file--run DFAX
IERR=22	Monitored elements exceed limit when adding multi-section line members
IERR=51	Error opening distribution factor data file
IERR=52	Error opening load throwover data file
IERR=53	Error opening inertia data file
IERR=54	Prerequisite requirements for function not met

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.164 N11_ACCC_PSCOPF

The API is obsolete. It has been replaced by N11_ACCC_PSCOPF_2 and is implemented by a call to N11_ACCC_PSCOPF_2

The API is used to perform N-1-1 contingency analysis in either corrective mode or security constrained optimal power flow mode (SCOPF). In corrective mode, for each pair of N-1 contingencies a full AC load flow is performed for the first N-1 contingency; the post contingency solution is checked against the operation limits and corrective actions are specified to mitigate the limit violations; the post-corrective action solution is used as the base case for the secondary contingency and a full AC load flow is performed to solve the second N-1 contingency.

In the SCOPF mode, a primary contingency is imposed and solved with a full AC load flow solution. A secure control is conducted on the system condition existing after the primary contingency with the secondary contingencies, control adjustments are specified by security constrained optimal power flow to ensure security of the system after the primary contingency. The system condition after the primary contingency and adjustments is used as the base case for secondary contingency; a full AC load flow solution is then performed to solve a secondary N-1 contingency.

Batch command syntax:

```
BAT_N11_ACCC_PSCOPF OPTIONS(1)…OPTIONS(18) OPTCOR(1)…OPTCOR(9)
VALUES(1)…VALUES(10)  LABELS(1)…LABELS(7) DFXFILE ACCFILE  CONFILE
INLFILE THRFILE
```

Python syntax:

```
ierr = n11_accc_pscopf(options,optcor,values,labels,dfxfile,accfile,
confile,inlfile,thrfile)
```

Fortran syntax:

```
CALL N11_ACCC_PSCOPF(OPTIONS, OPTCOR, VALUES, LABELS, DFXFILE,
ACCFILE,  CONFILE, INLFILE, THRFILE, IERR)
```

where:

Integer Options (18)	Is an integer array of eighteen elements specifying power flow options

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(1),	tap adjustment flag for the first N-1 contingency (tap adjustment option setting by default).
OPTIONS(1) = 0,	disable
OPTIONS(1) = 1,	enable stepping adjustment
OPTIONS(1) = 2,	enable direct adjustment
OPTIONS(2),	area interchange adjustment flag for the first N-1 contingency (area interchange adjustment option setting by default)
OPTIONS(2) = 0,	disable
OPTIONS(2) = 1,	enable using tie line flows only in calculating area interchange
OPTIONS(2) = 2,	enable using tie line flows and loads in calculating area interchange
OPTIONS(3),	phase shift adjustment flag for the first N-1 contingency (phase shift adjustment option setting by default)
OPTIONS(3) = 0,	disable
OPTIONS(3) = 1,	enable
OPTIONS(4),	dc tap adjustment flag for the first N-1 contingency (dc tap adjustment option setting by default).
OPTIONS(4) = 0,	disable
OPTIONS(4) = 1,	enable

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(5),	switched shunt adjustment flag for the first N-1 contingency (switched shunt adjustment option setting by default)
OPTIONS(5)= 0,	disable
OPTIONS(5)= 1,	enable all
OPTIONS(5)=2,	enable continuous, disable discrete.
OPTIONS(6)	induction motor treatment flag for the first N-1 contingency (applied when an induction motor fails to solve due to low terminal bus voltage, 0 by default)
OPTIONS(6) = 0,	stall.
OPTIONS(6) = 1,	trip.
OPTIONS(7),	solution method flag (0 by default).
OPTIONS(7)= 0,	Fixed slope decoupled Newton-Raphson method
OPTIONS(7)= 1,	Full Newton-Raphson method
OPTIONS(7)=2,	Optimized fixed slope decoupled Newton-Raphson method
OPTIONS(8),	non-divergent solution flag (non-divergent solution option setting by default).
OPTIONS(8)=0,	Disable
OPTIONS(8)=1,	Enable
OPTIONS(9),	tap adjustment flag for the second N-1 contingency (tap adjustment option setting by default).
OPTIONS(9)= 0,	disable.
OPTIONS(9)= 1,	enable stepping adjustment.
OPTIONS(9)= 2,	enable direct adjustment.
OPTIONS(10),	area interchange adjustment flag for the second N-1 contingency (area interchange adjustment option setting by default).
OPTIONS(10) = 0,	disable.
OPTIONS(10) = 1,	enable using tie line flows only in calculating area interchange.
OPTIONS(10) = 2,	enable using tie line flows and loads in calculating area interchange.
OPTIONS(11),	phase shift adjustment flag for the second N-1 contingency (phase shift adjustment option setting by default).
OPTIONS(11) = 0,	disable.
OPTIONS(11) = 1,	enable.
OPTIONS(12),	dc tap adjustment flag for the second N-1 contingency (dc tap adjustment option setting by default).
OPTIONS(12) = 0,	disable.
OPTIONS(12) = 1,	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTIONS(13),	switched shunt adjustment flag for the second N-1 contingency (switched shunt adjustment option setting by default).
	OPTIONS(13) = 0,	disable.
	OPTIONS(13) = 1,	enable.
	OPTIONS(13) = 2,	enable continuous, disable discrete.
	OPTIONS(14)	induction motor treatment flag for the second N-1 contingency (applied when an induction motor fails to solve due to low terminal bus voltage, 0 by default)
	OPTIONS(14) = 0,	stall.
	OPTIONS(14) = 1,	trip.
	OPTIONS(15), dispatch mode (0 by default).	
	OPTIONS(15) = 0,	disable.
	OPTIONS(15) = 1,	subsystem machines (RESERVE).
	OPTIONS(15) = 2,	subsystem machines (PMAX).
	OPTIONS(15) = 3,	subsystem machines (INERTIA).
	OPTIONS(15) = 4,	subsystem machines (GOVERNOR DROOP).
	OPTIONS(16),	flag to skip the second N-1 contingencies if the first N-1 contingency is not clean
	OPTIONS(16)=0,	No
	OPTIONS(16)=1,	Yes
	OPTIONS(17),	Rating set for the first N-1 contingency case
	OPTIONS(17)=1,	RATE A
	OPTIONS(17)=2,	RATE B
	OPTIONS(17)=3,	RATE C
	OPTIONS(18),	Rating set for the second N-1 contingency case
	OPTIONS(18)=1,	RATE A
	OPTIONS(18)=2,	RATE B
	OPTIONS(18)=3,	RATE C
Integer OPTCOR(9)	OPTCOR	Is an integer array of nine elements specifying control options
	OPTCOR(1)=	System adjustment mode
	OPTCOR(1)=0,	local adjustment only
	OPTCOR(1)=1,	Corrective action mode
	OPTCOR(1)=2,	Preventive security constrained optimal power flow mode

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTCOR(2)=	maximum iteration number of PSCOPF, required if OPTCOR(1) is 2
OPTCOR(3)=	maximum number of a contingency that does not cause limit violations and instability issue to be tested in the iterative process of PSCOPF, required if OPTCOR(1) is 2.
OPTCOR(4),	generator control flag (1 by default).
OPTCOR(4)=0,	disable generation MW dispatch in the corrective action mode or preventive security constrained optimal power flow mode
OPTCOR(4)=1,	enable
OPTCOR(5),	load control flag (0 by default).
OPTCOR(5)= 0,	disable load shedding control in the corrective action mode or preventive security constrained optimal power flow mode
OPTCOR(5)= 1,	enable
OPTCOR(6),	phase shifter control flag (1 by default).
OPTCOR(6)=0,	disable phase shifter control in the corrective action mode or preventive security constrained optimal power flow mode
OPTCOR(6)=1,	enable
OPTCOR(7),	off-line generator control flag (0 by default).
OPTCOR(7)=0,	disable generation MW dispatch of off-line generators in the corrective action mode or preventive security constrained optimal power flow mode
OPTCOR(7)=1,	enable
OPTCOR(8),	tap setting adjustment flag (0 by default).
OPTCOR(8)=0,	disable tap setting adjustment in the corrective action mode or preventive security constrained optimal power flow mode
OPTCOR(8)=1,	enable
OPTCOR(9),	switched shunt control flag (0 by default).
OPTCOR(9)=0,	disable switch shunt control in the corrective action mode or preventive security constrained optimal power flow mode
OPTCOR(9)=1,	enable

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real VALUES(10)	VALUES Is a real array of ten elements	
	VALUES(1)=	Mismatch tolerance (TOLN by default)
	VALUES(2)=	Percentage of rating for the base case (100.0 bye default)
	VALUES(3)=	Voltage tolerance in linear programming
	VALUES(4)=	Flow tolerance in linear programming
	VALUES(5)=	Weighting factor of generation MW dispatch (1.0 bye default)
	VALUES(6)=	Weighting factor of load control (1.0 bye default)
	VALUES(7)=	Weighting factor of phase shifter control (1.0 bye default)
	VALUES(8)=	Weighting factor of generation MW dispatch of off-line generators (1.0 bye default)
	VALUES(9)=	Weighting factor of tap setting adjustment (1.0 by default)
	VALUES(10)=	Weighting factor of switched shunt control (1.0 by default)
		Character LABELS(7)*12 is an array of 7 elements specifiying subsystem labels (input).
	VALUESE(10)=	Weighting factor of switched shunt control (1.0 bye default)
Character LABELS(7)*32	is an array of 7 elements specifying subsystem labels (input)	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

LABELS(1)=	The label of generation dispatch system, required if the dispatch mode of OPTIONS(15) is not zero. The dispatch mode and the system provide for a default dispatch method for contingencies causing unbalance between load demand and generation output (blank for none).
LABELS(2)=	The label of generation dispatch control system. MW injections of the generators within the system are available for dispatch in the corrective action mode or preventive security constrained optimal power flow mode (blank for none).
LABELS(3)=	The label of load control system. Loads in the load control system are available for curtailment in the corrective action mode or preventive security constrained optimal power flow mode (blank for none).
LABELS(4)=	The label of phase shifter adjustment system. The phase angles of phase shifters in the system are regulated either by corrective action or preventive security constrained optimal power flow (blank for none).
LABELS(5)=	The label of generation MW control of off-line generator control system. MW injections of the off-line generators within the system are available for dispatch in the corrective action mode or preventive security constrained optimal power flow mode (blank for none).
LABELS(6)=	The label of tap adjustment system. The tap positions of transformers in voltage control mode are regulated either by corrective action or preventive security constrained optimal power flow (blank for none).
LABELS(7)=	The label of switched shunt control system. The admittances of switched shunts in voltage control mode are regulated either by corrective action or preventive security constrained optimal power flow (blank for none).

Character DFXFILE*260	Is the name of the Distribution Factor Data File (input, no default allowed).
Character ACCFILE*260	Is the name of contingency analysis output file (input, no default allowed)
Character CONFILE*260	Is the name of Contingency Description Data File for the second N-1 contingencies (input, blank for none)
Character THRFILE*260	Is the name of Load Throwover Data File (input, blank for none).
Character INLFILE*260	Is the name of Unit Inertia Data File (input, blank for none).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code	
	IERR = 0	No error
	IERR=1,	Invalid OPTIONS value
	IERR=2,	Invalid VALUES value
	IERR=3,	Generators are converted
	IERR=4,	Buses in island(s) without a swing bus--use activity TREE
	IERR=5,	Largest mismatch exceeds tolerance
	IERR=6,	Generation subsystem not specified
	IERR=7,	Too many islands in base case.
	IERR=8,	in-service induction machines are in the "stalled" or "tripped" state.
	IERR=9,	buses with bus type code and series element status inconsistencies.
	IERR=10,	No Distribution Factor Data input file specified
	IERR=11,	No Contingency Analysis Output File specified.
	IERR=12,	Can not read the Contingency Description Data File for the second N-1 contingencies
	IERR=13,	No second N-1 contingencies are specified
	IERR=14	Invalid base/contingency case flag combination.
	IERR=15,	Invalid OPTCOR value
	IERR=21,	File is not in the form of a PSS/E-25 or later DFAX file--run DFAX
	IERR=22,	Monitored elements exceed limit when adding multi-section line members
	IERR=51,	Error opening distribution factor data file
	IERR=52,	Error opening load throwover data file
	IERR=53,	Error opening inertia data file
	IERR=54,	Prerequisite requirements for function not met

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.165 N11_ACCC_PSCOPF_2

The API is the second release of N-1-1 contingency analysis solution in either corrective mode or security constrained optimal power flow mode (SCOPF). In corrective mode, for each pair of N-1 contingencies a full AC load flow is performed for the first N-1 contingency; the post contingency solution is checked against the operation limits and corrective actions are specified to mitigate the limit violations; the post-corrective action solution is used as the base case for the secondary contingency and a full AC load flow is performed to solve the second N-1 contingency.

In the SCOPF mode, a primary contingency is imposed and solved with a full AC load flow solution. A secure control is conducted on the system condition existing after the primary contingency with the secondary contingencies, control adjustments are specified by security constrained optimal power flow to ensure security of the system after the primary contingency. The system condition after the primary contingency and adjustments is used as the base case for secondary contingency; a full AC load flow solution is then performed to solve a secondary N-1 contingency.

Batch command syntax:

```
BAT_N11_ACCC_PSCOPF_2 OPTIONS(1)…OPTIONS(20) OPTCOR(1)…OPTCOR(9)
VALUES(1)…VALUES(10)  LABELS(1)…LABELS(7) DFXFILE ACCFILE  CONFILE
INLFILE THRFILE
```

Python syntax:

```
ierr =
n11_accc_pscopf_2(options,optcor,values,labels,dfxfile,accfile,
confile,inlfile,thrfile)
```

Fortran syntax:

```
CALL N11_ACCC_PSCOPF_2(OPTIONS, OPTCOR, VALUES, LABELS, DFXFILE,
ACCFILE,  CONFILE, INLFILE, THRFILE, IERR)
```

where:

Integer Options (20)	Is an integer array of twenty elements specifying power flow options

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(1)	tap adjustment flag for the first N-1 contingency (tap adjustment option setting by default).
OPTIONS(1) = 0,	disable
OPTIONS(1) = 1,	enable stepping adjustment
OPTIONS(1) = 2,	enable direct adjustment
OPTIONS(2)	area interchange adjustment flag for the first N-1 contingency (area interchange adjustment option setting by default)
OPTIONS(2) = 0,	disable
OPTIONS(2) = 1,	enable using tie line flows only in calculating area interchange
OPTIONS(2) = 2,	enable using tie line flows and loads in calculating area interchange
OPTIONS(3)	phase shift adjustment flag for the first N-1 contingency (phase shift adjustment option setting by default)
OPTIONS(3) = 0,	disable
OPTIONS(3) = 1,	enable
OPTIONS(4)	dc tap adjustment flag for the first N-1 contingency (dc tap adjustment option setting by default).
OPTIONS(4) = 0,	disable
OPTIONS(4) = 1,	enable

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(5)	switched shunt adjustment flag for the first N-1 contingency (switched shunt adjustment option setting by default)
OPTIONS(5)= 0,	disable
OPTIONS(5)= 1,	enable all
OPTIONS(5)=2,	enable continuous, disable discrete.
OPTIONS(6)	induction motor treatment flag for the first N-1 contingency (applied when an induction motor fails to solve due to low terminal bus voltage, 0 by default)
OPTIONS(6) = 0,	stall.
OPTIONS(6) = 1,	trip.
OPTIONS(7)	solution method flag (0 by default).
OPTIONS(7)= 0,	Fixed slope decoupled Newton-Raphson method
OPTIONS(7)= 1,	Full Newton-Raphson method
OPTIONS(7)=2,	Optimized fixed slope decoupled Newton-Raphson method
OPTIONS(8)	non-divergent solution flag (non-divergent solution option setting by default).
OPTIONS(8)=0,	Disable
OPTIONS(8)=1,	Enable
OPTIONS(9)	tap adjustment flag for the second N-1 contingency (tap adjustment option setting by default).
OPTIONS(9)= 0,	disable.
OPTIONS(9)= 1,	enable stepping adjustment.
OPTIONS(9)= 2,	enable direct adjustment.
OPTIONS(10)	area interchange adjustment flag for the second N-1 contingency (area interchange adjustment option setting by default).
OPTIONS(10) = 0,	disable.
OPTIONS(10) = 1,	enable using tie line flows only in calculating area interchange.
OPTIONS(10) = 2,	enable using tie line flows and loads in calculating area interchange.
OPTIONS(11)	phase shift adjustment flag for the second N-1 contingency (phase shift adjustment option setting by default).
OPTIONS(11) = 0,	disable.
OPTIONS(11) = 1,	enable.
OPTIONS(12)	dc tap adjustment flag for the second N-1 contingency (dc tap adjustment option setting by default).
OPTIONS(12) = 0,	disable.
OPTIONS(12) = 1,	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTIONS(13)	switched shunt adjustment flag for the second N-1 contingency (switched shunt adjustment option setting by default).
	OPTIONS(13) = 0,	disable.
	OPTIONS(13) = 1,	enable.
	OPTIONS(13) = 2,	enable continuous, disable discrete.
	OPTIONS(14)	induction motor treatment flag for the second N-1 contingency (applied when an induction motor fails to solve due to low terminal bus voltage, 0 by default)
	OPTIONS(14) = 0,	stall.
	OPTIONS(14) = 1,	trip.
	OPTIONS(15), dispatch mode (0 by default).	
	OPTIONS(15) = 0,	disable.
	OPTIONS(15) = 1,	subsystem machines (RESERVE).
	OPTIONS(15) = 2,	subsystem machines (PMAX).
	OPTIONS(15) = 3,	subsystem machines (INERTIA).
	OPTIONS(15) = 4,	subsystem machines (GOVERNOR DROOP).
	OPTIONS(16)	flag to skip the second N-1 contingencies if the first N-1 contingency is not clean
	OPTIONS(16)=0,	No
	OPTIONS(16)=1,	Yes
	OPTIONS(17)	Rating set for the first N-1 contingency case
	OPTIONS(17)=1,	rate A
	OPTIONS(17)=2,	rate B
	OPTIONS(17)=3,	rate C
	OPTIONS(18)	Rating set for the second N-1 contingency case
	OPTIONS(18)=1,	rate A
	OPTIONS(18)=2,	rate B
	OPTIONS(18)=3,	rate C
	OPTIONS(19)	voltage limit for the first N-1 contingency case
	OPTIONS(19)=1,	normal
	OPTIONS(19)=2,	emergency
	OPTIONS(20)	voltage limit for the second N-1 contingency case
	OPTIONS(20)=1,	normal
	OPTIONS(20)=2,	emergency
Integer OPTCOR(9)		Is an integer array of nine elements specifying control options
	OPTCOR(1)	System adjustment mode
	OPTCOR(1)=0,	local adjustment only
	OPTCOR(1)=1,	Corrective action mode
	OPTCOR(1)=2,	Preventive security constrained optimal power flow mode

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTCOR(2)	maximum iteration number of PSCOPF, required if OPTCOR(1) is 2
OPTCOR(3)	maximum number of a contingency that does not cause limit violations and instability issue to be tested in the iterative process of PSCOPF, required if OPTCOR(1) is 2.
OPTCOR(4)	generator control flag (1 by default).
OPTCOR(4)=0,	disable generation MW dispatch in the corrective action mode or preventive security constrained optimal power flow mode
OPTCOR(4)=1,	enable
OPTCOR(5)	load control flag (0 by default).
OPTCOR(5)= 0,	disable load shedding control in the corrective action mode or preventive security constrained optimal power flow mode
OPTCOR(5)= 1,	enable
OPTCOR(6)	phase shifter control flag (1 by default).
OPTCOR(6)=0,	disable phase shifter control in the corrective action mode or preventive security constrained optimal power flow mode
OPTCOR(6)=1,	enable
OPTCOR(7)	off-line generator control flag (0 by default).
OPTCOR(7)=0,	disable generation MW dispatch of off-line generators in the corrective action mode or preventive security constrained optimal power flow mode
OPTCOR(7)=1,	enable
OPTCOR(8)	tap setting adjustment flag (0 by default).
OPTCOR(8)=0,	disable tap setting adjustment in the corrective action mode or preventive security constrained optimal power flow mode
OPTCOR(8)=1,	enable
OPTCOR(9)	switched shunt control flag (0 by default).
OPTCOR(9)=0,	disable switch shunt control in the corrective action mode or preventive security constrained optimal power flow mode
OPTCOR(9)=1,	enable

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real VALUES(10)		is a real array of ten elements
	VALUES(1)=	Mismatch tolerance (TOLN by default)
	VALUES(2)=	Percentage of rating for the base case (100.0 bye default)
	VALUES(3)=	Voltage tolerance in linear programming
	VALUES(4)=	Flow tolerance in linear programming
	VALUES(5)=	Weighting factor of generation MW dispatch (1.0 bye default)
	VALUES(6)=	Weighting factor of load control (1.0 bye default)
	VALUES(7)=	Weighting factor of phase shifter control (1.0 bye default)
	VALUES(8)=	Weighting factor of generation MW dispatch of off-line generators (1.0 bye default)
	VALUES(9)=	Weighting factor of tap setting adjustment (1.0 by default)
	VALUES(10)=	Weighting factor of switched shunt control (1.0 by default)
Character LABELS(7)* 32		is an array of 7 elements specifiying subsystem labels (input).
	LABELS(1)=	The label of generation dispatch system, required if the dispatch mode of OPTIONS(15) is not zero. The dispatch mode and the system provide for a default dispatch method for contingencies causing unbalance between load demand and generation output (blank for none).
	LABELS(2)=	The label of generation dispatch control system. MW injections of the generators within the system are available for dispatch in the corrective action mode or preventive security constrained optimal power flow mode (blank for none).
	LABELS(3)=	The label of load control system. Loads in the load control system are available for curtailment in the corrective action mode or preventive security constrained optimal power flow mode (blank for none).
	LABELS(4)=	The label of phase shifter adjustment system. The phase angles of phase shifters in the system are regulated either by corrective action or preventive security constrained optimal power flow (blank for none).
	LABELS(5)=	The label of generation MW control of off-line generator control system. MW injections of the off-line generators within the system are available for dispatch in the corrective action mode or preventive security constrained optimal power flow mode (blank for none).
	LABELS(6)=	The label of tap adjustment system. The tap positions of transformers in voltage control mode are regulated either by corrective action or preventive security constrained optimal power flow (blank for none).
	LABELS(7)=	The label of switched shunt control system. The admittances of switched shunts in voltage control mode are regulated either by corrective action or preventive security constrained optimal power flow (blank for none).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character DFXFILE*260	Is the name of the Distribution Factor Data File (input, no default allowed).
Character ACCFILE*260	Is the name of contingency analysis output file (input, no default allowed)
Character CONFILE*260	Is the name of Contingency Description Data File for the second N-1 contingencies (input, blank for none)
Character THRFILE*260	Is the name of Load Throwover Data File (input, blank for none).
Character INLFILE*260	Is the name of Unit Inertia Data File (input, blank for none).
Integer IERR	Is the error code

IERR=0	No error
IERR=1	Invalid OPTIONS value
IERR=2	Invalid VALUES value
IERR=3	Generators are converted
IERR=4	Buses in island(s) without a swing bus--use activity TREE
IERR=5	Largest mismatch exceeds tolerance
IERR=6	Generation subsystem not specified
IERR=7	Too many islands in base case.
IERR=8	in-service induction machines are in the "stalled" or "tripped" state.
IERR=9	buses with bus type code and series element status inconsistencies.
IERR=10	No Distribution Factor Data input file specified
IERR=11	No Contingency Analysis Output File specified.
IERR=12	Can not read the Contingency Description Data File for the second N-1 contingencies
IERR=13	No second N-1 contingencies are specified
IERR=14	Invalid base/contingency case flag combination.
IERR=15	Invalid OPTCOR value
IERR=21	File is not in the form of a PSS/E-25 or later DFAX file--run DFAX
IERR=22	Monitored elements exceed limit when adding multi-section line members
IERR=51	Error opening distribution factor data file
IERR=52	Error opening load throwover data file
IERR=53	Error opening inertia data file
IERR=54	Prerequisite requirements for function not met

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.166 NETG

Use this API to change the in-service generation to negative MVA load at all Type 2 and 3 buses **except** those in the subsystem specified by the user (activity NETG).

Batch command syntax:

```
BAT_NETG  SID  ALL
```

Python syntax:

```
ierr = netg(sid, all)
```

Fortran syntax:

```
CALL NETGNW(SID, ALL, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses outside of subsystem SID.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.167 NEW_DIMENSION

Use this API to reset PSS®E to an increased bus size level.

Batch command syntax:

```
BAT_NEW_DIMENSION  BUSES
```

Python syntax:

```
ierr = new_dimension(buses)
```

Fortran syntax:

```
CALL NEW_DIMENSION(BUSES, IERR)
```

where:

Integer BUSES	Is the maximum number of buses allowed at the new size level (input; no default allowed). If BUSES is less than the current maximum number of buses, PSS®E's capacity limits are not changed.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid BUSES value. IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.168 NEWCAS

This API routine is obsolete. It has been replaced by the API routine NEWCASE_2, and is implemented by a call to NEWCASE_2.

Use this API to initialize a new power flow case. All existing working case data will be lost upon execution of this API.

Batch command syntax:

```
BAT_NEWCAS  BASEMVA  TITL1  TITL2
```

Python syntax:

```
ierr = newcas(basemva, titl1, titl2)
```

Fortran syntax:

```
CALL NEWCASNW(BASEMVA, TITL1, TITL2, IERR)
```

where:

Real BASEMVA	Is the new system base MVA (input). Valid base MVA values range from 1.0 to 10,000.0 (100.0 by default).
Character TITL1*60	Is the first line of the new case title (input; blank by default).
Character TITL2*60	Is the second line of the new case title (input; blank by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid BASEMVA value. IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.169 NEWCASE_2

This API is the second release of the function used to initialize a new power flow case. All existing working case data will be lost upon execution of this API.

Batch command syntax:

```
BAT_NEWCASE_2  OPTIONS(1)  OPTIONS(2)  BASEMVA  BASEFREQ  TITL1
TITL2
```

Python syntax:

```
ierr = newcase_2(options, basemva, basefreq, titl1, titl2)
```

Fortran syntax:

```
CALL NEWCASE_2(OPTIONS, BASEMVA, BASEFREQ, TITL1, TITL2, IERR)
```

where:

Integer OPTIONS(2)	Is an array of two elements specifying units for ratings (used to set the percent loading program option settings) (input; present program option setting by default). The values are as follows: OPTIONS(1) units for transformer ratings: OPTIONS(1) > 0 current expressed as MVA. OPTIONS(1) ≤ 0 MVA. OPTIONS(2) units for ratings of non-transformer branches: OPTIONS(2) > 0 current expressed as MVA. OPTIONS(2) ≤ 0 MVA.
Real BASEMVA	Is the system base MVA. Valid base MVA values range from 1.0 to 10,000.0 (input; 100.0 by default).
Real BASEFREQ	Is the system base frequency. Valid base frequency values range from greater than 0.0 to less than 100.0 (input; present base frequency option setting by default).
Character TITL1*60	Is the first line of the new case title (input; blank by default).
Character TITL2*60	Is the second line of the new case title (input; blank by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid BASEMVA value. IERR = 2 invalid BASEFREQ value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.170 NEWSEQ

Use this API to initialize the fault analysis data arrays, set them to default values, and set the flag indicating that sequence data is present in the working case (activity RESQ).

Batch command syntax:

```
BAT_NEWSEQ
```

Python syntax:

```
ierr = newseq()
```

Fortran syntax:

```
CALL NEWSEQ(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.171 NEWTON_TOLERANCE

Use this API to specify the value of the default Newton-Raphson convergence tolerance option setting (activity OPTN).

Batch command syntax:

```
BAT_NEWTON_TOLERANCE  RVAL
```

Python syntax:

```
ierr = newton_tolerance(rval)
```

or:

```
ierr,rval = newton_tolerance()
```

Fortran syntax:

```
CALL NEWTON_TOLERANCE_API(RVAL, IOCODE, IERR)
```

where:

Real RVAL	Is the value of the option setting (input if Batch command or for Python if specified as an argument, or for FORTRAN if IOCODE is 0; else output 0.1 by default).
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to the value specified in RVAL. IOCODE = 1 return the current value of the option setting in RVAL.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid RVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.172 NON_DIVERGENT

Use this API to enable or disable the non-divergent Newton power flow solution option setting (activity OPTN).

Batch command syntax:

```
BAT_NON_DIVERGENT  IVAL
```

Python syntax:

```
ierr = non_divergent(ival)
```

or:

```
ierr,ival = non_divergent()
```

Fortran syntax:

```
CALL NON_DIVERGENT_API(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Is the value of the option setting (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 1 by default for input).	
	IVAL = 0	disable non-divergent Newton solution option.
	IVAL = 1	enable non-divergent Newton solution option.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE = 0	set the option setting to the value specified in IVAL.
	IOCODE = 1	return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid IVAL value.
	IERR = 2	invalid IOCODE value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.173 NON_TRANS_PERCENT_UNITS

Use this API to set the non-transformer branch percent units option setting to either *MVA* or *current expressed as MVA* (activity OPTN).

Batch command syntax:

```
BAT_NON_TRANS_PERCENT_UNITS  IVAL
```

Python syntax:

```
ierr = non_trans_percent_units(ival)
```

or:

```
ierr,ival = non_trans_percent_units()
```

Fortran syntax:

```
CALL NON_TRANS_PERCENT_UNITS_API(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Is the value of the option setting (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 1 by default for input).	
	IVAL = 0	MVA.
	IVAL = 1	current expressed as MVA.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE = 0	set the option setting to the value specified in IVAL.
	IOCODE = 1	return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid IVAL value.
	IERR = 2	invalid IOCODE value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.174 NSOL

Use this API to apply the decoupled Newton-Raphson power flow calculation (activity NSOL).

Batch command syntax:

```
BAT_NSOL  OPTIONS(1)  OPTIONS(2)  OPTIONS(3)  OPTIONS(4)  OPTIONS(5)
OPTIONS(6)  OPTIONS(7)
```

Python syntax:

```
ierr = nsol(options)
```

Fortran syntax:

```
CALL NSOLAPI(OPTIONS, IERR)
```

where:

Integer OPTIONS(7)	Is an array of seven elements specifying solution options (input). The values are as follows:	
	OPTIONS(1)	tap adjustment flag (use tap adjustment option setting by default).
	OPTIONS(1) = 0	disable.
	OPTIONS(1) = 1	enable stepping adjustment.
	OPTIONS(1) = 2	enable direct adjustment.
	OPTIONS(2)	area interchange adjustment flag (use area interchange adjustment option setting by default).
	OPTIONS(2) = 0	disable.
	OPTIONS(2) = 1	enable using tie line flows only in calculating area interchange.
	OPTIONS(2) = 2	enable using tie line flows and loads in calculating area interchange.
	OPTIONS(3)	phase shift adjustment flag (use phase shift adjustment option setting by default).
	OPTIONS(3) = 0	disable.
	OPTIONS(3) = 1	enable.
	OPTIONS(4)	dc tap adjustment flag (use dc tap adjustment option setting by default).
	OPTIONS(4) = 0	disable.
	OPTIONS(4) = 1	enable.
	OPTIONS(5)	switched shunt adjustment flag (use switched shunt adjustment option setting by default).
	OPTIONS(5) = 0	disable.
	OPTIONS(5) = 1	enable.
	OPTIONS(5) = 2	enable continuous mode, disable discrete mode.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTIONS(6)	flat start flag (0 by default).
	OPTIONS(6) = 0	do not flat start.
	OPTIONS(6) = 1	flat start.
	OPTIONS(7)	var limit flag (99 by default).
	OPTIONS(7) = 0	apply var limits immediately.
	OPTIONS(7) = >0	apply var limits on iteration n (or sooner if mismatch gets small).
	OPTIONS(7) = -1	ignore var limits.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid OPTIONS value.
	IERR = 2	generators are converted.
	IERR = 3	buses in island(s) without a swing bus; use activity TREE.
	IERR = 4	bus type code and series element status inconsistencies.
	IERR = 5	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.175 NUMBER_THREADS

Use this API to specify the number of processors available for contingency analysis (activity OPTN).

Batch command syntax:

```
BAT_NUMBER_THREADS IVAL
```

Python syntax:

```
ierr = number_threads(ival)
```

or:

```
ierr,ival = number_threads()
```

Fortran syntax:

```
CALL NUMBER_THREADS_API(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Is the value of the option setting between 1 and the number of processors of the computer (input if Batch command or for Python if specified as an argument, or for FORTRAN if IOCODE is 0; 1 by default for input).
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to the value specified in IVAL. IOCODE = 1 return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid IVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.176 OPENUCTEFILE

Use this API to open an existing UCTE file and translate it to a PSSE RAW file.

Batch command syntax:

```
BAT_OPENUCTEFILE IFILE UFILE RFILE
```

Python syntax:

```
openuctefile(ifile, ufile, rfile)
```

where:

Character IFILE*260	Is the name of an existing UCTE translation options input file (input).
Character UFILE*260	Is the name of an existing UCTE data input file (input).
Character RFILE*260	Is the name of the target PSSE RAW output file (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.177 ORDR

Use this API to calculate a sparsity preserving ordering of buses in preparation for the processing of network matrices (activity ORDR).

Batch command syntax:

```
BAT_ORDR  OPT
```

Python syntax:

```
ierr = ordr(opt)
```

Fortran syntax:

```
CALL ORDRAPI(OPT, IERR)
```

where:

Integer OPT	Is the flag for ordering option (input; 0 by default).	
	OPT = 0	assume all branches between in-service buses are in-service.
	OPT = 1	ignore out-of-service branches in calculating the bus ordering.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid OPT value.
	IERR = 2	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.178 OTDF

Use this API to tabulate vectors of distribution factors using a linear network (dc) model (activity OTDF).

Batch command syntax:

```
BAT_OTDF  DFXFILE
```

Python syntax:

```
ierr = otdf(dfxfile)
```

Fortran syntax:

```
CALL OTDFAPI(DFXFILE, IERR)
```

where:

Character DFXFILE*260	Is the name of Distribution Factor Data File (input; no default allowed).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	no Distribution Factor Data File specified.
	IERR = 2	generators are converted.
	IERR = 3	buses in island(s) without a swing bus; use activity TREE.
	IERR = 4	error opening file DFXFILE.
	IERR = 5	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
	IERR = 6	file DFXFILE does not contain distribution factors; run DFAX with distribution factor option enabled.
	IERR = 7	no monitored branches or interfaces defined.
	IERR = 8	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.179 OUTPUT_Y_MATRIX

Use this API to display the network admittance matrix for a subsystem of the working case in the form of a list of matrix terms.

Batch command syntax:

```
BAT_OUTPUT_Y_MATRIX  SID  ALL  TIES  OUT  OFILE
```

Python syntax:

```
ierr = output_y_matrix(sid, all, ties, out, ofile)
```

Fortran syntax:

```
CALL OUTPUT_Y_MATRIX(SID, ALL, TIES, OUT, OFILE, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 process the complete case. ALL = 0 process only rows corresponding to buses in subsystem SID.
Integer TIES	Is an option for the omission or inclusion of off-diagonal terms corresponding to ties from the specified subsystem; ignored if ALL is 1 (input; 0 by default). TIES = 0 omit ties. TIES = 1 include ties.
Integer OUT	Is the output destination selection (input; 1 by default). OUT = 0 direct output to a data file specified by OFILE. OUT = 1 direct output to the progress window.
Character OFILE*260	Is the name of the file to which data records are to be written (input; no default allowed when OUT is 0).
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 invalid TIES value. IERR = 4 invalid OUT value. IERR = 5 error opening OFILE. IERR = 6 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.180 OUTS

Use this API to tabulate those components in the working case that are removed from service (activity OUTS).

Batch command syntax:

```
BAT_OUTS  SID  ALL
```

Python syntax:

```
ierr = outs(sid, all)
```

Fortran syntax:

```
CALL OUTSNW(SID, ALL, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.181 OWNM

This API routine is obsolete. It has been replaced by the API routine OWNM_3, and is implemented by a call to OWNM_3.

Use this API routine to reassign the buses, load, machines, and branches in a specified subsystem of the working case from their original owner to a designated owner (activity OWNM).

Batch command syntax:

```
BAT_OWNM  SID  ALL  STATUS(1)...STATUS(7)  IONEW
```

Python syntax:

```
ierr = ownm(sid, all, status, ionew)
```

Fortran syntax:

```
CALL OWNMAPI(SID, ALL, STATUS, IONEW, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all buses.
	ALL = 0	process only buses in subsystem SID.
Integer STATUS(7)	Is an array of seven elements (input). They are used to control reassignment options.	
	STATUS(1) = 1	change bus owners to IONEW; otherwise = 0 (0 by default).
	STATUS(2) = 1	change load owners to IONEW; otherwise = 0 (0 by default).
	STATUS(3) = 1	change machine owners to IONEW; otherwise = 0 (0 by default).
	STATUS(4) = 1	change branch owners to IONEW; otherwise = 0 (0 by default).
	STATUS(5) = 1	change FACTS device owners to IONEW; otherwise = 0 (0 by default).
	STATUS(6) = 1	change VSC dc line owners to IONEW; otherwise = 0 (0 by default).
	STATUS(7)	branch, series FACTS device, VSC dc line option (1 by default.
	STATUS(7) = 1	subsystem branches only.
	STATUS(7) = 2	subsystem ties only.
	STATUS(7) = 3	both subsystem branches and ties.
Integer IONEW	Is the new owner to which the specified equipment items are to be assigned (input). IONEW must be between 1 and 9999 (1 by default).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	invalid STATUS value.
	IERR = 4	invalid new owner number.
	IERR = 5	owner table is full; cannot add owner to the working case.
	IERR = 6	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.182 OWNM_2

This API routine is obsolete. It has been replaced by the API routine OWNM_3, and is implemented by a call to OWNM_3.

This API routine is the second release of the owner renumbering function. Use this API routine to reassign the buses, loads, machines, branches, FACTS devices, VSC dc lines, and/or GNE devices in a specified subsystem of the working case from their original owner to a designated owner (activity OWNM).

Batch command syntax:

```
BAT_OWNM_2  SID  ALL  STATUS(1)...STATUS(8)  IONEW
```

Python syntax:

```
ierr = ownm_2(sid, all, status, ionew)
```

Fortran syntax:

```
CALL OWNMAPI_2(SID, ALL, STATUS, IONEW, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all buses.
	ALL = 0	process only buses in subsystem SID.
Integer STATUS(8)	Is an array of eight elements (input). They are used to control reassignment options.	
	STATUS(1) = 1	change bus owners to IONEW; 0 to leave bus ownership unchanged (0 by default).
	STATUS(2) = 1	change load owners to IONEW; 0 to leave load ownership unchanged (0 by default).
	STATUS(3) = 1	change machine owners to IONEW; 0 to leave machine ownership unchanged (0 by default).
	STATUS(4) = 1	change branch owners to IONEW; 0 to leave branch ownership unchanged (0 by default).
	STATUS(5) = 1	change FACTS device owners to IONEW; 0 to leave FACTS device ownership unchanged (0 by default).
	STATUS(6) = 1	change VSC dc line owners to IONEW; 0 to leave VSC dc line ownership unchanged (0 by default).
	STATUS(7) = 1	change GNE device owners to IONEW; 0 to leave GNE device ownership unchanged (0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	STATUS(8)	branch, series FACTS device, VSC dc line, and series GNE device option (1 by default).
	STATUS(8) = 1	subsystem branches only.
	STATUS(8) = 2	subsystem ties only.
	STATUS(8) = 3	both subsystem branches and ties.
Integer IONEW	Is the new owner to which the specified equipment items are to be assigned (input). IONEW must be between 1 and 9999 (1 by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	invalid STATUS value.
	IERR = 4	invalid new owner number.
	IERR = 5	owner table is full; cannot add owner to the working case.
	IERR = 6	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.183 OWNM_3

This API routine is the third release of the owner renumbering function. Use this API routine to reassign the buses, loads, machines, induction machines, branches, FACTS devices, VSC dc lines, and/or GNE devices in a specified subsystem of the working case from their original owner to a designated owner (activity OWNM).

Batch command syntax:

```
BAT_OWNM_3  SID  ALL  STATUS(1)...STATUS(9)  IONEW
```

Python syntax:

```
ierr = ownm_3(sid, all, status, ionew)
```

Fortran syntax:

```
CALL OWNM_3(SID, ALL, STATUS, IONEW, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all buses.
	ALL = 0	process only buses in subsystem SID.
Integer STATUS(9)	Is an array of nine elements (input). They are used to control reassignment options.	
	STATUS(1) = 1	change bus owners to IONEW; 0 to leave bus ownership unchanged (0 by default).
	STATUS(2) = 1	change load owners to IONEW; 0 to leave load ownership unchanged (0 by default).
	STATUS(3) = 1	change machine owners to IONEW; 0 to leave machine ownership unchanged (0 by default).
	STATUS(4) = 1	change induction machine owners to IONEW; 0 to leave induction machine ownership unchanged (0 by default).
	STATUS(5) = 1	change branch owners to IONEW; 0 to leave branch ownership unchanged (0 by default).
	STATUS(6) = 1	change FACTS device owners to IONEW; 0 to leave FACTS device ownership unchanged (0 by default).
	STATUS(7) = 1	change VSC dc line owners to IONEW; 0 to leave VSC dc line ownership unchanged (0 by default).
	STATUS(8) = 1	change GNE device owners to IONEW; 0 to leave GNE device ownership unchanged (0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	STATUS(9)	branch, series FACTS device, VSC dc line, and series GNE device option (1 by default).
	STATUS(9) = 1	subsystem branches only.
	STATUS(9) = 2	subsystem ties only.
	STATUS(9) = 3	both subsystem branches and ties.
Integer IONEW	Is the new owner to which the specified equipment items are to be assigned (input). IONEW must be between 1 and 9999 (1 by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	invalid STATUS value.
	IERR = 4	invalid new owner number.
	IERR = 5	owner table is full; cannot add owner to the working case.
	IERR = 6	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.184 OWNR

Use this API to tabulate owner totals by owner (activity OWNR).

Batch command syntax:

```
BAT_OWNR  SID  ALL
```

Python syntax:

```
ierr = ownr(sid, all)
```

Fortran syntax:

```
CALL OWNRNW(SID, ALL, IERR)
```

where:

Integer SID	Is a valid owner subsystem identifier. Valid subsystem identifiers range from 0 to 11. Owner subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all owners* or specified subsystem flag (input; 1 by default). ALL = 1 process all owners. ALL = 0 process only owners in owner subsystem SID.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.185 PATH

The directory path specification activity PATH allows the user to specify a directory name used whenever a simple filename preceded by an ampersand is specified to PSSE. E.g. &myfile will be opened in PATHNAME\myfile

Batch command syntax:

```
BAT_PATH  PATHNAME
```

Python syntax:

```
ierr = path(pathname)
```

Fortran syntax:

```
CALL PATHAPI(PATHNAME, IERR)
```

where:

Character PATHNAME*260	Is the name of path to be used (blank to disable) (input; blank by default).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 error setting path PATHNAME.
	IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.186 PBUS_DELETE

Use this API to remove a bus from the set of buses that are participating in a transaction event.

Batch command syntax:

```
BAT_PBUS_DELETE  EVENT  IBUS
```

Python syntax:

```
ierr = pbus_delete(event, ibus)
```

Fortran syntax:

```
CALL PBUS_DELETE(EVENT, IBUS, IERR, NERR)
```

where:

Integer EVENT	Is the transaction event number.
Integer IBUS	Is the number of the participating bus.
Integer IERR	Is the error code. IERR = 0 no error occurred. IERR = 1 invalid transaction event number. IERR = 2 transaction event not found. IERR = 3 prerequisite requirements for API are not met.
Integer NERR	Is the number of error entries in PBUF02 (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.187 PHASE_SHIFT_ADJUSTMENT

Use this API to enable or disable the phase shift adjustment option setting (activity OPTN).

Batch command syntax:

```
BAT_PHASE_SHIFT_ADJUSTMENT  IVAL
```

Python syntax:

```
ierr = phase_shift_adjustment(ival)
```

or:

```
ierr,ival = phase_shift_adjustment()
```

Fortran syntax:

```
CALL PHASE_SHIFT_ADJUSTMENT_API(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Is the value of the option setting (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 1 by default for input).	
	IVAL = 0	disabled.
	IVAL = 1	enabled.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE = 0	set the option setting to the value specified in IVAL.
	IOCODE = 1	return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid IVAL value.
	IERR = 2	invalid IOCODE value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.188 POLY

Use this API to calculate interchange limits of a study system against two opposing systems using a linear network (dc) model (activity POLY).

Batch command syntax:

```
BAT_POLY  OPTIONS(1)...OPTIONS(15)  VALUES(1)...VALUES(6)
LABELS(1)...LABELS(3)  DFXFILE  RESFILE
```

Python syntax:

```
ierr = poly(options, values, labels, dfxfile, resfile)
```

Fortran syntax:

```
CALL POLYAPI(OPTIONS, VALUES, LABELS, DFXFILE, RESFILE, IERR)
```

where:

Integer OPTIONS(15)	Is an array of fifteen elements specifying calculation and reporting options (input). The values are as follows:	
	OPTIONS(1)	base case rating set (rating set program option setting by default).
	OPTIONS(1) = 1	RATEA.
	OPTIONS(1) = 2	RATEB.
	OPTIONS(1) = 3	RATEC.
	OPTIONS(2)	contingency case rating set (rating set program option setting by default).
	OPTIONS(2) = 1	RATEA.
	OPTIONS(2) = 2	RATEB.
	OPTIONS(2) = 3	RATEC.
	OPTIONS(3)	base case line flow code (0 by default).
	OPTIONS(3) = 0	dc base case.
	OPTIONS(3) = 1	ac base case.
	OPTIONS(4)	phase shifter code (0 by default).
	OPTIONS(4) = 0	locked in base case shift solutions.
	OPTIONS(4) = 1	regulating in base case shift solutions.
	OPTIONS(5)	code for the treatment of base case line flow constraints in contingency cases (0 by default).
	OPTIONS(5) = 0	ignore.
	OPTIONS(5) = 1	include.
	OPTIONS(6)	list study system buses (0 by default).
	OPTIONS(6) = 0	no.
	OPTIONS(6) = 1	yes.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(7)	list opposing system buses (0 by default).
OPTIONS(7) = 0	no.
OPTIONS(7) = 1	yes.
OPTIONS(8)	list study system tie lines (0 by default).
OPTIONS(8) = 0	no.
OPTIONS(8) = 1	yes.
OPTIONS(9)	add study system tie lines to monitored line list (0 by default).
OPTIONS(9) = 0	no.
OPTIONS(9) = 1	yes.
OPTIONS(10)	output format code (0 by default).
OPTIONS(10) = 0	summary output.
OPTIONS(10) = 1	full output.
OPTIONS(11)	convert MVA ratings to estimated MW ratings (0 by default).
OPTIONS(11) = 0	no.
OPTIONS(11) = 1	yes.
OPTIONS(12)	discontinue activity if MW mismatch exceeds VALUES(1) (0 by default).
OPTIONS(12) = 0	discontinue.
OPTIONS(12) = 1	continue.
OPTIONS(13)	graphical output option (0 by default).
OPTIONS(13) = 0	plot combined case only.
OPTIONS(13) = 1	plot combined case only.
OPTIONS(13) = 2	plot for all system conditions.
OPTIONS(14)	option for plotting constraints (0 by default).
OPTIONS(14) = 0	plot all constraints.
OPTIONS(14) = 1	plot only constraints forming the boundary of the feasible region.
OPTIONS(15)	graphical output format (0 by default).
OPTIONS(15) = 0	single page per case.
OPTIONS(15) = 1	two pages per case.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real VALUES(6)	Is an array of six elements specifying tolerances and thresholds (input).	
	VALUES(1)	required MW mismatch tolerance for continuing (Newton solution convergence tolerance, TOLN, by default).
	VALUES(2)	percent of rating for reporting an element as overloaded (100.0 by default).
	VALUES(3)	maximum opposing system generation shift (10000.0 by default).
	VALUES(4)	maximum opposing system delta interchange for plotting (1000.0 by default).
	VALUES(5)	minimum distribution factor magnitude (0.0 by default).
	VALUES(6)	study system generation shift (100.0 by default).
Character LABELS(3)*12	Is an array of three elements specifying subsystem labels (input).	
	LABELS(1)	study system label (no default allowed).
	LABELS(2)	first opposing system label (no default allowed).
	LABELS(3)	second opposing system label (no default allowed).
Character DFXFILE*260	Is the name of Distribution Factor Data File (input; no default allowed).	
Character RESFILE*260	Is the name of POLY Results file (input; no default allowed).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error.
IERR = 1	invalid OPTIONS value.
IERR = 2	invalid VALUES value.
IERR = 3	no Distribution Factor Data File specified.
IERR = 4	same label specified for at least two subsystems.
IERR = 5	no POLY Results Output File specified.
IERR = 6	generators are converted.
IERR = 7	buses in island(s) without a swing bus; use activity TREE.
IERR = 8	largest mismatch exceeds mismatch tolerance.
IERR = 9	error opening file DFXFILE.
IERR = 10	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
IERR = 11	file DFXFILE does not contain distribution factors; run DFAX with distribution factor option enabled.
IERR = 12	no monitored branches or interfaces defined.
IERR = 13	no subsystems defined.
IERR = 14	subsystem not found.
IERR = 15	specified subsystems overlap.
IERR = 16	no in-service buses in subsystem WORLD.
IERR = 17	no tie lines from study system.
IERR = 18	only one subsystem defined.
IERR = 19	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.189 POUT

Use this API to print the power flow solution results with boundary condition and flow information on the left side of the report and other information on the right side (activity POUT).

Batch command syntax:

```
BAT_POUT  SID  ALL
```

Python syntax:

```
ierr = pout(sid, all)
```

Fortran syntax:

```
CALL POUTNW(SID, ALL, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.190 POWER_OUTPUT

Use this API to set the power output option setting to either *MVA* or *KVA* (activity OPTN).

Batch command syntax:

```
BAT_POWER_OUTPUT  IVAL
```

Python syntax:

```
ierr = power_output(ival)
```

or:

```
ierr,ival = power_output()
```

Fortran syntax:

```
CALL POWER_OUTPUT_API(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Is the value of the option setting (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 1 by default for input).	
	IVAL = 0	kVA.
	IVAL = 1	MVA.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE = 0	set the option setting to the value specified in IVAL.
	IOCODE = 1	return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid IVAL value.
	IERR = 2	invalid IOCODE value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.191 PP_ACCC

This API routine is obsolete. It has been replaced by the API routine ACCC_SINGLE_RUN_REPORT_2, and is implemented by a call to ACCC_SINGLE_RUN_REPORT_2.

This API, the AC Contingency Report function, reports the results of the AC Contingency Calculation function.

Batch command syntax:

```
BAT_PP_ACCC  STATUS(1)...STATUS(5)  INTVAL(1)...INTVAL(5)
REALVAL(1)...REALVAL(6) RFILE
```

Python syntax:

```
ierr = pp_accc(status, intval, realval, rfile)
```

Fortran syntax:

```
CALL PP_ACCC(STATUS, INTVAL, REALVAL, RFILE, IERR)
```

where:

Integer STATUS(5)	Is an array of five elements (input). The values are as follows.	
	STATUS(1)	contingency case rating (rating set program option setting by default).
	STATUS(1) = 0	rate A.
	STATUS(1) = 1	rate B.
	STATUS(1) = 2	rate C.
	STATUS(2)	report format (3 by default).
	STATUS(2) = 0	spreadsheet overload report.
	STATUS(2) = 1	spreadsheet loading table.
	STATUS(2) = 2	available capacity table.
	STATUS(2) = 3	non-spreadsheet overload report.
	STATUS(2) = 4	non-spreadsheet loading table.
	STATUS(2) = 5	non-converged networks report.
	STATUS(3)	exclude interfaces from report (0 by default).
	STATUS(3) = 0	no.
	STATUS(3) = 1	yes.
	STATUS(4)	run voltage limit check (0 by default).
	STATUS(4) = 0	no.
	STATUS(4) = 1	yes.
	STATUS(5)	exclude cases with no overloads from non-spreadsheet overload report (0 by default).
	STATUS(5) = 0	no.
	STATUS(5) = 1	yes.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer INTVAL(5)	Is an array of five elements (input). The values are as follows.	
	INTVAL(1)	number of low voltage range violations (0 by default).
	INTVAL(2)	number of high voltage range violations (0 by default).
	INTVAL(3)	number of voltage deviation violations (0 by default).
	INTVAL(4)	number of buses in the largest disconnected island (0 by default).
	INTVAL(5)	maximum number of elements in available capacity table (no limit by default).
Real REALVAL(6)	Is an array of six elements (input). The values are as follows.	
	REALVAL(1)	percent of flow rating (100.0 by default).
	REALVAL(2)	minimum contingency case flow change for overload report (0.0 by default).
	REALVAL(3)	cutoff threshold for available capacity table (99999.0 by default).
	REALVAL(4)	minimum contingency case voltage change for range violations (0.0 by default).
	REALVAL(5)	bus mismatch converged tolerance (MW or Mvar) (0.5 by default).
	REALVAL(6)	system mismatch converged tolerance (MVA) (5.0 by default).
Character RFILE*260	Is the Contingency Solution Output File (input; no default allowed).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid STATUS value.
	IERR = 2	invalid INTVAL value.
	IERR = 3	invalid REALVAL value.
	IERR = 4	error opening RFILE.
	IERR = 5	error reading RFILE.
	IERR = 6	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.192 PP_ACCC_MULTI_CASE

This API routine is obsolete. It has been replaced by the API routine ACCC_MULTIPLE_RUN_REPORT, and is implemented by a call to ACCC_MULTIPLE_RUN_REPORT.

This API, the AC Contingency Multiple Run Report function, reports the results of up to twenty-two executions of the AC Contingency Calculation function.

Batch command syntax:

```
BAT_PP_ACCC_MULTI_CASE  OPTIONS(1)...OPTIONS(11)
VALUES(1)...VALUES(6)  NFILES  ACFILES(1)...ACFILES(NFILES)
```

Python syntax:

```
ierr = pp_accc_multi_case(options, values, nfiles, acfiles)
```

Fortran syntax:

```
CALL PP_ACCC_MULTI_CASE(OPTIONS, VALUES, NFILES, ACFILES, IERR)
```

where:

Integer OPTIONS(11)	Is an array of eleven elements defining reporting options (input). The values are as follows.	
	OPTIONS(1)	column headings code (1 by default).
	OPTIONS(1) = 1	Contingency Solution Output file names.
	OPTIONS(1) = 2	Saved Case file names as contained in the Contingency Solution Output Files.
	OPTIONS(2)	rating set (rating set program option setting by default).
	OPTIONS(2) = 1	rate A.
	OPTIONS(2) = 2	rate B.
	OPTIONS(2) = 3	rate C.
	OPTIONS(3)	print monitored elements summary report (0 by default).
	OPTIONS(3) = 0	no.
	OPTIONS(3) = 1	yes.
	OPTIONS(4)	print missing monitored elements report (0 by default).
	OPTIONS(4) = 0	no.
	OPTIONS(4) = 1	yes.
	OPTIONS(5)	print missing monitored voltage buses report (0 by default).
	OPTIONS(5) = 0	no.
	OPTIONS(5) = 1	yes.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(6)	print contingency legend (2 by default).
OPTIONS(6) = 0	no.
OPTIONS(6) = 1	complete master legend report only.
OPTIONS(6) = 2	reduced legend with each table.
OPTIONS(6) = 3	both master legend report and legend with each table.
OPTIONS(7)	print missing contingencies report (0 by default).
OPTIONS(7) = 0	no.
OPTIONS(7) = 1	yes.
OPTIONS(8)	print non-converged contingencies report (0 by default).
OPTIONS(8) = 0	no.
OPTIONS(8) = 1	yes.
OPTIONS(9)	print loading violation reports (0 by default).
OPTIONS(9) = 0	no.
OPTIONS(9) = 1	base case and worst case contingency violations only.
OPTIONS(9) = 2	base case and all contingency violations only.
OPTIONS(9) = 3	base case, worst case contingency, and all contingency violations.
OPTIONS(10)	print voltage violation reports (0 by default).
OPTIONS(10) = 0	no.
OPTIONS(10) = 1	base case and worst case contingency violations only.
OPTIONS(10) = 2	base case and all contingency violations only.
OPTIONS(10) = 3	base case, worst case contingency, and all contingency violations.
OPTIONS(11)	option for processing interfaces in loading violation reports (1 by default).
OPTIONS(11) = 0	exclude interfaces from loading violation checking and reporting.
OPTIONS(11) = 1	check and report interface loading violations.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real VALUES(6)	Is an array of six elements specifying reporting parameters (input). The values are as follows.	
	VALUES(1)	percent of flow rating for reporting (100.0 by default).
	VALUES(2)	percent of flow rating for counting in worst case contingency violations report (≥ VALUES(1)) (100.0 by default).
	VALUES(3)	minimum contingency case flow change for reporting and counting (0.0 by default).
	VALUES(4)	minimum contingency case voltage change for reporting and counting in voltage range violations (0.0 by default).
	VALUES(5)	bus mismatch converged tolerance (MW or Mvar) (0.5 by default).
	VALUES(6)	system mismatch converged tolerance (MVA) (5.0 by default).
Integer NFILES	Is the number of Contingency Solution Output Files to be processed (1 through 22) (input; no default allowed).	
Character ACFILES(NFILES)*260	Is an array of NFILES names of Contingency Solution Output Files (input; no defaults allowed).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid OPTIONS value.
	IERR = 2	invalid VALUES value.
	IERR = 3	invalid NFILES value.
	IERR = 4	the percent for counting is less than the percent for reporting.
	IERR = 5	file is not in the form of a current Contingency Solution Output File.
	IERR = 6	error opening a Contingency Solution Output File.
	IERR = 7	error reading a Contingency Solution Output File.
	IERR = 8	error opening a temporary file.
	IERR = 9	error opening a subfile in the temporary file.
	IERR = 10	error reading or writing the temporary file.
	IERR = 11	error closing a subfile in the temporary file.
	IERR = 12	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.193 PRINT_OUTAGED_BRANCHES

Use this API to enable or disable the print out-of-service branches option setting (activity OPTN).

Batch command syntax:

```
BAT_PRINT_OUTAGED_BRANCHES  IVAL
```

Python syntax:

```
ierr = print_outaged_branches(ival)
```

or:

```
ierr,ival = print_outaged_branches()
```

Fortran syntax:

```
CALL PRINT_OUTAGED_BRANCHES_API(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Is the value of the option setting (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 1 by default for input).	
	IVAL = 0	do not print out-of-service branches.
	IVAL = 1	print out-of-service branches.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE = 0	set the option setting to the value specified in IVAL.
	IOCODE = 1	return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid IVAL value.
	IERR = 2	invalid IOCODE value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.194 PRINT_WINDING_BUSES

Use this API to enable or disable the print winding to buses of three-winding transformers option setting (activity OPTN).

Batch command syntax:

```
BAT_PRINT_WINDING_BUSES  IVAL
```

Python syntax:

```
ierr = print_winding_buses(ival)
```

or:

```
ierr,ival = print_winding_buses()
```

Fortran syntax:

```
CALL PRINT_WINDING_BUSES_API(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Is the value of the option setting (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 1 by default for input).	
	IVAL = 0	do not print winding to buses.
	IVAL = 1	print winding to buses.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE = 0	set the option setting to the value specified in IVAL.
	IOCODE = 1	return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid IVAL value.
	IERR = 2	invalid IOCODE value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.195 PROGRESS_OUTPUT

Use this API to specify the progress output device.

Batch command syntax:

```
BAT_PROGRESS_OUTPUT  ISLCT  FILARG  OPTIONS(1)  OPTIONS(2)
```

Python syntax:

```
ierr = progress_output(islct, filarg, options)
```

Fortran syntax:

```
CALL OPEN_PROGRESS_DEVICE(ISLCT, FILARG, OPTIONS, IERR)
```

where:

Integer ISLCT	Is the virtual device selector (input).	
	ISLCT = 1	standard destination. In the PSS®E GUI standard output for progress output is the Progress View.
	ISLCT = 2	direct output to a file.
	ISLCT = 3	direct output to a printer.
	ISLCT = 4	direct output to progress device.
	ISLCT = 5	direct output to report device.
	ISLCT = 6	no output.
Character FILARG*260	Is the printer name or filename; ignored if ISLCT is not 2 or 3 (input).	
Integer OPTIONS(2)	Is an array of two elements specifying open and printing options; ignored if ISLCT is not 2 or 3 (input). The values are as follows:	
	OPTIONS(1)	file and printer open options:
	OPTIONS(1) = 0	open with carriage control format and, for files, for overwrite of existing files
	OPTIONS(1) = 1	open with list format.
	OPTIONS(1) = 2	open file for append (file output only).
	OPTIONS(1) = 2	set printer orientation to portrait mode (printer output only).
	The OPTIONS(1) values may be summed to select multiple options.	
	OPTIONS(1) = 4	open printer in immediate print mode (printer output only).
	OPTIONS(2)	number of copies to print (printer output only).
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	prerequisite requirements for API are not met.
	IERR = 2	failed to open device.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.196 PROMPT_OUTPUT

Use this API to specify the prompt output device.

Batch command syntax:

```
BAT_PROMPT_OUTPUT  ISLCT  FILARG  OPTIONS(1)  OPTIONS(2)
```

Python syntax:

```
ierr = prompt_output(islct, filarg, options)
```

Fortran syntax:

```
CALL OPEN_PROMPT_DEVICE(ISLCT, FILARG, OPTIONS, IERR)
```

where:

Integer ISLCT	Is the virtual device selector (input).	
	ISLCT = 1	standard destination. In the PSS®E GUI standard output for prompts is the input window generated by user input requests.
	ISLCT = 2	direct output to a file.
	ISLCT = 3	direct output to a printer.
	ISLCT = 4	direct output to progress device.
	ISLCT = 5	direct output to report device.
	ISLCT = 6	no output.
Character FILARG*260	Is the printer name or filename; ignored if ISLCT is not 2 or 3 (input).	
Integer OPTIONS(2)	Is an array of two elements specifying open and printing options; ignored if ISLCT is not 2 or 3 (input). The values are as follows:	
	OPTIONS(1)	file and printer open options:
	OPTIONS(1) = 0	open with carriage control format and, for files, for overwrite of existing files
	OPTIONS(1) = 1	open with list format.
	OPTIONS(1) = 2	open file for append (file output only).
	OPTIONS(1) = 2	set printer orientation to portrait mode (printer output only).
	OPTIONS(1) = 4	open printer in immediate print mode (printer output only).
	The OPTIONS(1) values may be summed to select multiple options.	
	OPTIONS(2)	number of copies to print (printer output only).
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	prerequisite requirements for API are not met.
	IERR = 2	failed to open device.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.197 PRTI

Use this API to print the 16 line long title (activity PRTI).

Batch command syntax:

```
BAT_PRTI
```

Python syntax:

```
ierr = prti()
```

Fortran syntax:

```
CALL PRTIAPI(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.198 PSCOPF

The API is obsolete. It has been replaced by PSCOPF_2 and is implemented by a call to PSCOPF_2.

This API is used to find the settings of controls for a system to comply with base case and contingency case operation limits. The API is implemented with preventive security constrained optimal power flow method (PSCOPF). The objective of the PSCOPF is to minimize control adjustments subject to operation limits constraints. The available controls are generation MW outputs of on-line and off-line generators, loads, phase shifters, tap settings and switched shunts.

Batch command syntax:

```
BAT_PSCOPF OPTIONS(1)…OPTIONS(26) VALUES(1)…VALUES(11)
LABELS(1)…LABELS(7) DFXFILE INLFILE THRFILE
```

Python syntax:

```
ierr = pscopf(options,values,labels,dfxfile,inlfile,thrfile)
```

Fortran syntax:

```
CALL PSCOPFAPI(OPTIONS,VALUES, LABELS,DFXFILE,INLFILE,THRFILE,IERR)
```

where:

Integer OPTIONS(26)	Is an integer array of twenty six elements specifying power flow options	
	OPTIONS(1),	tap adjustment flag for the base case (tap adjustment option setting by default).
	OPTIONS(1) = 0,	disable
	OPTIONS(1) = 1,	enable stepping adjustment
	OPTIONS(1) = 2,	enable direct adjustment
	OPTIONS(2),	area interchange adjustment flag for the base case (area interchange adjustment option setting by default).
	OPTIONS(2) = 0,	disable
	OPTIONS(2) = 1,	enable using tie line flows only in calculating area interchange.
	OPTIONS(2) = 2,	enable using tie line flows and loads in calculating area interchange
	OPTIONS(3),	phase shift adjustment flag for the base case (phase shift adjustment option setting by default).
	OPTIONS(3) = 0,	disable
	OPTIONS(3) = 1,	enable
	OPTIONS(4),	dc tap adjustment flag for the base case (dc tap adjustment option setting by default).
	OPTIONS(4) = 0,	disable
	OPTIONS(4) = 1,	enable

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(5),	switched shunt adjustment flag for the base case (switched shunt adjustment option setting by default).
OPTIONS(5)= 0,	disable
OPTIONS(5)= 1,	enable all
OPTIONS(5)=2,	enable continuous, disable discrete.
OPTIONS(6)	induction motor treatment flag (applied when an induction motor fails to solve due to low terminal bus voltage, 0 by default)
OPTIONS(6) = 0,	stall.
OPTIONS(6) = 1,	trip.
OPTIONS(7),	solution method flag (0 by default).
OPTIONS(7)= 0,	Fixed slope decoupled Newton-Raphson method
OPTIONS(7)= 1,	Full Newton-Raphson method
OPTIONS(7)=2,	Optimized fixed slope decoupled Newton-Raphson method
OPTIONS(8),	non-divergent solution flag (non-divergent solution option setting by default).
OPTIONS(8)=0,	Disable
OPTIONS(8)=1,	Enable
OPTIONS(9),	tap adjustment flag for the first N-1 contingency (tap adjustment option setting by default).
OPTIONS(9)= 0,	disable.
OPTIONS(9)= 1,	enable stepping adjustment.
OPTIONS(9)= 2,	enable direct adjustment.
OPTIONS(10),	area interchange adjustment flag for the contingency case (area interchange adjustment option setting by default).
OPTIONS(10) = 0,	disable.
OPTIONS(10) = 1,	enable using tie line flows only in calculating area interchange.
OPTIONS(10) = 2,	enable using tie line flows and loads in calculating area interchange.
OPTIONS(11),	phase shift adjustment flag for the contingency case (phase shift adjustment option setting by default).
OPTIONS(11) = 0,	disable.
OPTIONS(11) = 1,	enable.
OPTIONS(12),	dc tap adjustment flag for the contingency case (dc tap adjustment option setting by default).
OPTIONS(12) = 0,	disable.
OPTIONS(12) = 1,	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(13), switched shunt adjustment flag for the contingency case (switched shunt adjustment option setting by default).

OPTIONS(13) = 0, disable.

OPTIONS(13) = 1, enable.

OPTIONS(13) = 2, enable continuous, disable discrete.

OPTIONS(14) induction motor treatment flag for the contingency case(applied when an induction motor fails to solve due to low terminal bus voltage, 0 by default)

OPTIONS(14) = 0, stall.

OPTIONS(14) = 1, trip.

OPTIONS(15), dispatch mode (0 by default).

OPTIONS(15) = 0, disable.

OPTIONS(15) = 1, subsystem machines (RESERVE).

OPTIONS(15) = 2, subsystem machines (PMAX).

OPTIONS(15) = 3, subsystem machines (INERTIA).

OPTIONS(15) = 4, subsystem machines (GOVERNOR DROOP).

OPTIONS(16), Rating set for the base case

OPTIONS(16)=1, RATE A

OPTIONS(16)=2, RATE B

OPTIONS(16)=3, RATE C

OPTIONS(17), Rating set for the contingency case

OPTIONS(17)=1, RATE A

OPTIONS(17)=2, RATE B

OPTIONS(17)=3, RATE C

OPTIONS(18), maximum iteration number of PSCOPF

OPTIONS(18)=1, RATE A

OPTIONS(18)=2, RATE B

OPTIONS(18)=3, RATE C

OPTIONS(19)= maximum number of a contingency that does not cause limit violations and instability issue to be tested in the iterative process of PSCOPF

OPTIONS(20)= Handle initial violations in the base case

OPTIONS(20)=0, Ignore the initial violations in the base case from the PSCOPF analysis

OPTIONS(20)=1, Clean the initial violations in the base case

OPTIONS(21), generator control flag (1 by default).

OPTIONS(21)=0, disable generation MW dispatch

OPTIONS(21)=1, enable

OPTIONS(22), load control flag (0 by default).

OPTIONS(22)= 0, disable load shedding control

OPTIONS(22)= 1, enable

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTIONS(23),	phase shifter control flag (1 by default).
	OPTIONS(23)=0,	disable phase shifter control
	OPTIONS(23)=1,	enable
	OPTIONS(24), off-line generator control flag (0 by default).	
	OPTIONS(24)=0,	disable generation MW dispatch of off-line generators
	OPTIONS(24)=1,	enable
	OPTIONS(25), tap setting adjustment flag (0 by default).	
	OPTIONS(25)=0,	disable tap setting adjustment
	OPTIONS(25)=1,	enable
	OPTIONS(26),	switched shunt control flag (0 by default).
	OPTIONS(26)=0,	disable switch shunt control
	OPTIONS(26)=1,	enable
Real VALUES (11)	VALUES Is a real array of eleven elements	
	VALUES(1)=	Mismatch tolerance (TOLN by default)
	VALUES(2)=	Percentage of rating for the base case (100.0 by default)
	VALUES(3)=	Percentage of rating for the contingency cases (100.0 by default)
	VALUES(4)=	Voltage tolerance in linear programming
	VALUES(5)=	Flow tolerance in linear programming
	VALUES(6)=	Weighting factor of generation MW dispatch (1.0 by default)
	VALUES(7)=	Weighting factor of load control (1.0 by default)
	VALUES(8)=	Weighting factor of phase shifter control (1.0 by default)
	VALUES(9)=	Weighting factor of generation MW dispatch of off-line generators (1.0 by default)
	VALUES(10)=	Weighting factor of tap setting adjustment (1.0 by default)
	VALUES(11)=	Weighting factor of switched shunt control (1.0 by default)

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character LABELS(7)*32	Is a character array of seven elements	
	LABELS(1)=	The label of generation dispatch system, required if the dispatch mode of OPTIONS(15) is not zero. The dispatch mode and the system provide for a default dispatch method for contingencies causing unbalance between load demand and generation output (blank for none).
	LABELS(2)=	The label of generation dispatch control system. MW injections of the generators within the system are available for dispatch in the corrective action mode or preventive security constrained optimal power flow mode (blank for none).
	LABELS(3)=	The label of load control system. Loads in the load control system are available for curtailment in the corrective action mode or preventive security constrained optimal power flow mode (blank for none).
	LABELS(4)=	The label of phase shifter adjustment system. The phase angles of phase shifters in the system are regulated either by corrective action or preventive security constrained optimal power flow (blank for none).
	LABELS(5)=	The label of generation MW control of off-line generator control system. MW injections of the off-line generators within the system are available for dispatch in the corrective action mode or preventive security constrained optimal power flow mode (blank for none).
	LABELS(6)=	The label of tap adjustment system. The tap positions of transformers in voltage control mode are regulated either by corrective action or preventive security constrained optimal power flow (blank for none).
	LABELS(7)=	The label of switched shunt control system. The admittances of switched shunts in voltage control mode are regulated either by corrective action or preventive security constrained optimal power flow (blank for none).
Character DFXFILE*260	Is the name of the Distribution Factor Data File (input, no default allowed).	
Character THRFILE*260	Is the name of Load Throwover Data File (input, blank for none).	
Character INLFILE*260	Is the name of Unit Inertia Data File (input, blank for none).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code	
	IERR=-2,	PSCOPF terminates with active constraints
	IERR=-1,	PSCOPF reaches the maximum iteration
	IERR = 0	No error
	IERR=1,	Invalid OPTIONS value
	IERR=2,	Invalid VALUES value
	IERR=3,	Generators are converted
	IERR=4,	Buses in island(s) without a swing bus--use activity TREE
	IERR=5,	Largest mismatch exceeds tolerance
	IERR=6,	Generation subsystem not specified
	IERR=7,	Too many islands in base case.
	IERR=8,	in-service induction machines are in the "stalled" or "tripped" state.
	IERR=9,	buses with bus type code and series element status inconsistencies.
	IERR=10,	No Distribution Factor Data input file specified
	IERR=11,	No controls are specified
	IERR=12,	No constraints are specified
	IERR=13,	Invalid base/contingency case flag combination.
	IERR=21,	File is not in the form of a PSS/E-25 or later DFAX file--run DFAX
	IERR=22,	Monitored elements exceed limit when adding multi-section line members
	IERR=51,	Error opening distribution factor data file
	IERR=52,	Error opening load throwover data file
	IERR=53,	Error opening inertia data file
	IERR=54,	Prerequisite requirements for function not met

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.199 PSCOPF_2

This API is the second release of Preventive Security Constrained Optimal Power Flow solution (PSCOPF). This API is used to find the settings of controls for a system to comply with base case and contingency case operation limits. The API is implemented with preventive security constrained optimal power flow method (PSCOPF). The objective of the PSCOPF is to minimize control adjustments subject to operation limits constraints. The available controls are generation MW outputs of on-line and off-line generators, loads, phase shifters, tap settings and switched shunts.

Batch command syntax:

```
BAT_PSCOPF_2 OPTIONS(1)…OPTIONS(28) VALUES(1)…VALUES(11)
LABELS(1)…LABELS(7) DFXFILE INLFILE THRFILE
```

Python syntax:

```
ierr = pscopf_2(options,values,labels,dfxfile,inlfile,thrfile)
```

Fortran syntax:

```
CALL PSCOPFAPI_2(OPTIONS,VALUES, LABELS,DFXFILE,INLFILE,THRFILE,IERR)
```

where:

Integer OPTIONS(28)	Is an integer array of twenty eight elements specifying power flow options	
	OPTIONS(1)	tap adjustment flag for the base case (tap adjustment option setting by default).
	OPTIONS(1) = 0,	disable
	OPTIONS(1) = 1,	enable stepping adjustment
	OPTIONS(1) = 2,	enable direct adjustment
	OPTIONS(2)	area interchange adjustment flag for the base case (area interchange adjustment option setting by default).
	OPTIONS(2) = 0,	disable
	OPTIONS(2) = 1,	enable using tie line flows only in calculating area interchange.
	OPTIONS(2) = 2,	enable using tie line flows and loads in calculating area interchange
	OPTIONS(3)	phase shift adjustment flag for the base case (phase shift adjustment option setting by default).
	OPTIONS(3) = 0,	disable
	OPTIONS(3) = 1,	enable
	OPTIONS(4)	dc tap adjustment flag for the base case (dc tap adjustment option setting by default).
	OPTIONS(4) = 0,	disable
	OPTIONS(4) = 1,	enable

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(5)	switched shunt adjustment flag for the base case (switched shunt adjustment option setting by default).
OPTIONS(5)= 0,	disable
OPTIONS(5)= 1,	enable all
OPTIONS(5)=2,	enable continuous, disable discrete.
OPTIONS(6)	induction motor treatment flag (applied when an induction motor fails to solve due to low terminal bus voltage, 0 by default)
OPTIONS(6) = 0,	stall.
OPTIONS(6) = 1,	trip.
OPTIONS(7)	solution method flag (0 by default).
OPTIONS(7)= 0,	Fixed slope decoupled Newton-Raphson method
OPTIONS(7)= 1,	Full Newton-Raphson method
OPTIONS(7)=2,	Optimized fixed slope decoupled Newton-Raphson method
OPTIONS(8)	non-divergent solution flag (non-divergent solution option setting by default).
OPTIONS(8)=0,	Disable
OPTIONS(8)=1,	Enable
OPTIONS(9)	tap adjustment flag for the first N-1 contingency (tap adjustment option setting by default).
OPTIONS(9)= 0,	disable.
OPTIONS(9)= 1,	enable stepping adjustment.
OPTIONS(9)= 2,	enable direct adjustment.
OPTIONS(10)	area interchange adjustment flag for the contingency case (area interchange adjustment option setting by default).
OPTIONS(10) = 0,	disable.
OPTIONS(10) = 1,	enable using tie line flows only in calculating area interchange.
OPTIONS(10) = 2,	enable using tie line flows and loads in calculating area interchange.
OPTIONS(11)	phase shift adjustment flag for the contingency case (phase shift adjustment option setting by default).
OPTIONS(11) = 0,	disable.
OPTIONS(11) = 1,	enable.
OPTIONS(12)	dc tap adjustment flag for the contingency case (dc tap adjustment option setting by default).
OPTIONS(12) = 0,	disable.
OPTIONS(12) = 1,	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(13)	switched shunt adjustment flag for the contingency case (switched shunt adjustment option setting by default).
OPTIONS(13) = 0,	disable.
OPTIONS(13) = 1,	enable.
OPTIONS(13) = 2,	enable continuous, disable discrete.
OPTIONS(14)	induction motor treatment flag for the contingency case(applied when an induction motor fails to solve due to low terminal bus voltage, 0 by default)
OPTIONS(14) = 0,	stall.
OPTIONS(14) = 1,	trip.
OPTIONS(15)	dispatch mode (0 by default).
OPTIONS(15) = 0,	disable.
OPTIONS(15) = 1,	subsystem machines (RESERVE).
OPTIONS(15) = 2,	subsystem machines (PMAX).
OPTIONS(15) = 3,	subsystem machines (INERTIA).
OPTIONS(15) = 4,	subsystem machines (GOVERNOR DROOP).
OPTIONS(16)	Rating set for the base case
OPTIONS(16)=1,	rate A
OPTIONS(16)=2,	rate B
OPTIONS(16)=3,	rate C
OPTIONS(17)	Rating set for the contingency case
OPTIONS(17)=1,	rate A
OPTIONS(17)=2,	rate B
OPTIONS(17)=3,	rate C
OPTIONS(18)	base case voltage limit
OPTIONS(18)=1,	normal
OPTIONS(18)=2,	emergency
OPTIONS(19)	contingency case voltage limit
OPTIONS(19)=1,	normal
OPTIONS(19)=2,	emergency
OPTIONS(20)	maximum iteration number of PSCOPF
OPTIONS(21)	maximum number of a contingency that does not cause limit violations and instability issue to be tested in the iterative process of PSCOPF
OPTIONS(22)	Handle initial violations in the base case
OPTIONS(22)=0,	Ignore the initial violations in the base case from the PSCOPF analysis
OPTIONS(22)=1	Clean the initial violations in the base case
OPTIONS(23)	generator control flag (1 by default).
OPTIONS(23)=0,	disable generation MW dispatch
OPTIONS(23)=1,	enable

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTIONS(24)	load control flag (0 by default).
	OPTIONS(24)= 0,	disable load shedding control
	OPTIONS(24)= 1,	enable
	OPTIONS(25)	phase shifter control flag (1 by default).
	OPTIONS(25)=0,	disable phase shifter control
	OPTIONS(25)=1,	enable
	OPTIONS(26)	off-line generator control flag (0 by default).
	OPTIONS(26)=0,	disable generation MW dispatch of off-line generators
	OPTIONS(26)=1,	enable
	OPTIONS(27)	tap setting adjustment flag (0 by default).
	OPTIONS(27)=0,	disable tap setting adjustment
	OPTIONS(27)=1,	enable
	OPTIONS(28)	switched shunt control flag (0 by default).
	OPTIONS(28)=0,	disable switch shunt control
	OPTIONS(28)=1,	enable
Real VALUES (11)	VALUES Is a real array of eleven elements	
	VALUES(1)=	Mismatch tolerance (TOLN by default)
	VALUES(2)=	Percentage of rating for the base case (100.0 by default)
	VALUES(3)=	Percentage of rating for the contingency cases (100.0 by default)
	VALUES(4)=	Voltage tolerance in linear programming
	VALUES(5)=	Flow tolerance in linear programming
	VALUES(6)=	Weighting factor of generation MW dispatch (1.0 by default)
	VALUES(7)=	Weighting factor of load control (1.0 by default)
	VALUES(8)=	Weighting factor of phase shifter control (1.0 by default)
	VALUES(9)=	Weighting factor of generation MW dispatch of off-line generators (1.0 by default)
	VALUES(10)=	Weighting factor of tap setting adjustment (1.0 by default)
	VALUES(11)=	Weighting factor of switched shunt control (1.0 by default)

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character LABELS(7)*32	Is a character array of seven elements	
	LABELS(1)	The label of generation dispatch system, required if the dispatch mode of OPTIONS(15) is not zero. The dispatch mode and the system provide for a default dispatch method for contingencies causing unbalance between load demand and generation output (blank for none).
	LABELS(2)	The label of generation dispatch control system. MW injections of the generators within the system are available for dispatch in the corrective action mode or preventive security constrained optimal power flow mode (blank for none).
	LABELS(3)	The label of load control system. Loads in the load control system are available for curtailment in the corrective action mode or preventive security constrained optimal power flow mode (blank for none).
	LABELS(4)	The label of phase shifter adjustment system. The phase angles of phase shifters in the system are regulated either by corrective action or preventive security constrained optimal power flow (blank for none).
	LABELS(5)	The label of generation MW control of off-line generator control system. MW injections of the off-line generators within the system are available for dispatch in the corrective action mode or preventive security constrained optimal power flow mode (blank for none).
	LABELS(6)	The label of tap adjustment system. The tap positions of transformers in voltage control mode are regulated either by corrective action or preventive security constrained optimal power flow (blank for none).
	LABELS(7)	The label of switched shunt control system. The admittances of switched shunts in voltage control mode are regulated either by corrective action or preventive security constrained optimal power flow (blank for none).
Character DFXFILE*260	Is the name of the Distribution Factor Data File (input, no default allowed).	
Character THRFILE*260	Is the name of Load Throwover Data File (input, blank for none).	
Character INLFILE*260	Is the name of Unit Inertia Data File (input, blank for none).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code	
	IERR=-2	PSCOPF terminates with active constraints
	IERR=-1	PSCOPF reaches the maximum iteration
	IERR= 0	No error
	IERR=1	Invalid OPTIONS value
	IERR=2	Invalid VALUES value
	IERR=3	Generators are converted
	IERR=4	Buses in island(s) without a swing bus--use activity TREE
	IERR=5	Largest mismatch exceeds tolerance
	IERR=6	Generation subsystem not specified
	IERR=7	Too many islands in base case.
	IERR=8	in-service induction machines are in the "stalled" or "tripped" state.
	IERR=9	buses with bus type code and series element status inconsistencies.
	IERR=10	No Distribution Factor Data input file specified
	IERR=11	No controls are specified
	IERR=12	No constraints are specified
	IERR=13	Invalid base/contingency case flag combination.
	IERR=21	File is not in the form of a PSS/E-25 or later DFAX file--run DFAX
	IERR=22	Monitored elements exceed limit when adding multi-section line members
	IERR=51	Error opening distribution factor data file
	IERR=52	Error opening load throwover data file
	IERR=53	Error opening inertia data file
	IERR=54	Prerequisite requirements for function not met

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.200 PSEB

Use this API to convert a PSEB Command File into a PSS®E Response File (activity PSEB).

Batch command syntax:

```
BAT_PSEB  INPFILE  OUTFILE
```

Python syntax:

```
ierr,respfil = pseb(inpfile, outfile)
```

Fortran syntax:

```
CALL PSEBAPI(INPFILE, OUTFILE, RESPFIL, IERR)
```

where:

Character INPFILE*260	Is the name of the PSEB Command input file (input; no default allowed).
Character OUTFILE*260	Is the name of output Response File; blank to have PSEB assign the filename (input; blank by default).
Character RESPFIL*260	Is the pathname of the Response File constructed by PSEB (output).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 error opening INPFILE. IERR = 2 error opening OUTFILE. IERR = 3 errors detected while reading the PSEB Command Data input file. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.201 PSSEHALT

This API routine is obsolete. It has been replaced by the API routine PSSEHALT_2, and is implemented by a call to PSSEHALT_2.

Use this API to end the operation of PSS®E, closes all associated files and returns to the calling application.

Batch command syntax:

```
BAT_PSSEHALT  DELETE
```

Python syntax:

```
ierr = pssehalt(delete)
```

Fortran syntax:

```
CALL HALTAPI(DELETE, IERR)
```

where:

Integer DELETE	Is the delete working files flag (input).	
	DELETE = 1	delete working files.
	DELETE =	anything except 1, keep working files.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.202 PSSEHALT_2

Use this API to end the operation of PSS®E, closes all associated files and returns to the calling application.

Batch command syntax:

```
BAT_PSSEHALT_2
```

Python syntax:

```
ierr = pssehalt_2()
```

Fortran syntax:

```
CALL HALTAPI_2(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.203 PURG

Use this API routine to delete specified outaged equipment items from the working case (activity PURG).

Batch command syntax:

```
BAT_PURG  SID  ALL  OPT  STATUS
```

Python syntax:

```
ierr = purg(sid, all, opt, status)
```

Fortran syntax:

```
CALL PURGNW(SID, ALL, OPT, STATUS, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Integer OPT	Is the equipment type to remove (input; 1 by default). = 1, ac lines = 2, switched shunts = 3, machines = 4, plants = 5, two-terminal dc lines = 6, multi-terminal dc lines = 7, loads = 8, FACTS devices = 9, three-winding transformers =10, VSC dc lines =11, fixed bus shunts =12, GNE devices =13, induction machines
Integer STATUS	Is the option for the treatment of out-of-service ties to other systems (input; 0 by default). STATUS = 0 keep out-of-service ties. STATUS = 1 remove out-of-service ties.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 invalid OPT value. IERR = 4 invalid STATUS value. IERR = 5 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.204 PURG2DC

Use this API to delete the specified two-terminal dc line from the working case (activity PURG).

Batch command syntax:

```
BAT_PURG2DC  DC
```

Python syntax:

```
ierr = purg2dc(dc)
```

Fortran syntax:

```
CALL PURG2DC(NUM, IERR)

CALL PURG2DC_I(NUM, IERR)

CALL PURG2DC_C(NAME, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the two-terminal dc line to be removed; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the two-terminal dc line to be removed (input; no default allowed).
Character*12 NAME	Is the name of the two-terminal dc line to be removed (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 dc line not found. IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.205 PURG3WND

Use this API to delete the specified three-winding transformer from the working case (activity PURG).

Batch command syntax:

```
BAT_PURG3WND  FRMBUS  TOBUS1  TOBUS2  'CKT'
```

Python syntax:

```
ierr = purg3wnd(frmbus, tobus1, tobus2, ckt)
```

Fortran syntax:

```
CALL PURG3WND(FRMBUS, TOBUS1, TOBUS2, CKT, IERR)
```

where:

Integer FRMBUS	Is one of the buses connected by the three-winding transformer (input; no default allowed).
Integer TOBUS1	Is another bus connected by the three-winding transformer (input; no default allowed).
Integer TOBUS2	Is the other bus connected by the three-winding transformer (input; no default allowed).
Character CKT*2	Is the circuit identifier of the three-winding transformer to be deleted (input; '1' by default).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 bus not found.
	IERR = 2 three-winding transformer not found.
	IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.206 PURGAREA

Use this API to delete areas with no equipment assigned to them from the working case.

Batch command syntax:

```
BAT_PURGAREA  SID  ALL
```

Python syntax:

```
ierr = purgarea(sid, all)
```

Fortran syntax:

```
CALL PURGAREA(SID, ALL, IERR)
```

where:

Integer SID	Is a valid area subsystem identifier. Valid subsystem identifiers range from 0 to 11. Area subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all areas* or specified subsystem flag (input; 1 by default). ALL = 1 delete all areas with no equipment assigned to them. ALL = 0 delete only areas in area subsystem SID with no equipment assigned to them.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.207 PURGATRN

Use this API to delete the specified inter-area transfer from the working case (activity PURG).

Batch command syntax:

```
BAT_PURGATRN  FRMAREA  TOAREA  'ID'
```

Python syntax:

```
ierr = purgatrn(frmarea, toarea, id)
```

Fortran syntax:

```
CALL PURGATRN(FRMAREA, TOAREA, ID, IERR)
```

where:

Integer FRMAREA	Is one of the areas involved in the transfer (input; no default allowed).
Integer TOAREA	Is the other area involved in the transfer (input; no default allowed).
Character ID*1	Is the inter-area transfer identifier (input; '1' by default).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 area not found.
	IERR = 2 inter-area transfer not found.
	IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.208 PURGBRN

Use this API to delete the specified non-transformer branch or two-winding transformer from the working case (activity PURG).

Batch command syntax:

```
BAT_PURGBRN  FRMBUS  TOBUS  'CKT'
```

Python syntax:

```
ierr = purgbrn(frmbus, tobus, ckt)
```

Fortran syntax:

```
CALL PURGBRN(FRMBUS, TOBUS, CKT, IERR)
```

where:

Integer FRMBUS	Is one of the buses connected by the branch (input; no default allowed).
Integer TOBUS	Is the other bus connected by the branch (input; no default allowed).
Character CKT*2	Is the circuit identifier of the branch to be deleted (input; '1' by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 bus not found. IERR = 2 branch not found. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.209 PURGFACTS

Use this API to delete the specified FACTS device from the working case (activity PURG).

Batch command syntax:

```
BAT_PURGFACTS  FD
```

Python syntax:

```
ierr = purgfacts(fd)
```

Fortran syntax:

```
CALL PURGFACTS(NUM, IERR)

CALL PURGFACTS_I(NUM, IERR)

CALL PURGFACTS_C(NAME, IERR)
```

where:

Character*12 or Integer FD	Is the quoted name of the FACTS device to be removed; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the FACTS device to be removed (input; no default allowed).
Character*12 NAME	Is the name of the FACTS device to be removed (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 FACTS device not found. IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.210 PURGGNE

Use this API routine to delete the specified GNE device from the working case (activity PURG).

Batch command syntax:

```
BAT_PURGGNE  'NAME'
```

Python syntax:

```
ierr = purggne(name)
```

Fortran syntax:

```
CALL PURGGNE(NAME, IERR)
```

where:

Character NAME*12	Is the name of the GNE device to be removed (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 GNE device not found.
	IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.211 PurgIndMac

Use this API routine to delete the specified induction machine from the working case (activity PURG).

Batch command syntax:

```
BAT_PurgIndMac  FRMBUS  'ID'
```

Python syntax:

```
ierr = purgindmac(frmbus, id)
```

Fortran syntax:

```
CALL PurgIndMac(FRMBUS, ID, IERR)
```

where:

Integer FRMBUS	Is the bus to which induction machine ID is connected (input; no default allowed).
Character ID*2	Is the identifier of the induction machine to be removed (input; '1' by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 bus not found. IERR = 2 no induction machines at bus. IERR = 3 induction machine not found. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.212 PurgIndMacs

Use this API routine to delete all induction machines from the specified bus in the working case (activity PURG).

Batch command syntax:

```
BAT_PurgIndMacs  FRMBUS
```

Python syntax:

```
ierr = purgindmacs(frmbus)
```

Fortran syntax:

```
CALL PurgIndMacs(FRMBUS, IERR)
```

where:

Integer FRMBUS	Is the bus from which all induction machines are to be removed (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 bus not found.
	IERR = 2 no induction machines at bus.
	IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.213 PURGLOAD

Use this API to delete the specified load from the working case (activity PURG).

Batch command syntax:

```
BAT_PURGLOAD  FRMBUS  'ID'
```

Python syntax:

```
ierr = purgload(frmbus, id)
```

Fortran syntax:

```
CALL PURGLOAD(FRMBUS, ID, IERR)
```

where:

Integer FRMBUS	Is the bus to which the load ID is connected (input; no default allowed).
Character ID*2	Is the identifier of the load to be removed (input; '1' by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 bus not found. IERR = 2 no loads at bus. IERR = 3 load not found. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.214 PURGLOADS

Use this API to delete all loads from the specified bus in the working case (activity PURG).

Batch command syntax:

```
BAT_PURGLOADS  FRMBUS
```

Python syntax:

```
ierr = purgloads(frmbus)
```

Fortran syntax:

```
CALL PURGLOADS(FRMBUS, IERR)
```

where:

Integer FRMBUS	Is the bus from which all loads are to be removed (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 bus not found.
	IERR = 2 no loads at bus.
	IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.215 PURGMAC

Use this API to delete the specified machine from the working case (activity PURG).

Batch command syntax:

```
BAT_PURGMAC  FRMBUS  'ID'
```

Python syntax:

```
ierr = purgmac(frmbus, id)
```

Fortran syntax:

```
CALL PURGMAC(FRMBUS, ID, IERR)
```

where:

Integer FRMBUS	Is the bus to which the machine ID is connected (input; no default allowed).
Character ID*2	Is the identifier of the machine to be removed (input; '1' by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 bus not found. IERR = 2 no machines at bus. IERR = 3 machine not found. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.216 PURGMDC

Use this API to delete the specified multi-terminal dc line from the working case (activity PURG).

Batch command syntax:

```
BAT_PURGMDC  DC
```

Python syntax:

```
ierr = purgmdc(dc)
```

Fortran syntax:

```
CALL PURGMDC(NUM, IERR)

CALL PURGMDC_I(NUM, IERR)

CALL PURGMDC_C(NAME, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the multi-terminal dc line to be removed; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the multi-terminal dc line to be removed (input; no default allowed).
Character*12 NAME	Is the name of the multi-terminal dc line to be removed (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 dc line not found. IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.217 PURGMSL

Use this API to delete the specified multi-section line grouping from the working case (activity PURG).

Batch command syntax:

```
BAT_PURGMSL  FRMBUS  TOBUS  'CKT'
```

Python syntax:

```
ierr = purgmsl(frmbus, tobus, ckt)
```

Fortran syntax:

```
CALL PURGMSL(FRMBUS, TOBUS, CKT, IERR)
```

where:

Integer FRMBUS	Is one of the endpoint buses of the multi-section line (input; no default allowed).
Integer TOBUS	Is the other endpoint bus of the multi-section line (input; no default allowed).
Character CKT*2	Is the circuit identifier of the multi-section line to be deleted; the first character must be an ampersand (input; '&1' by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error occurred.
	IERR = 1 — bus not found.
	IERR = 2 — multi-section line not found.
	IERR = 3 — invalid multi-section line identifier (first character not &).
	IERR = 4 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.218 PURGMUT

Use this API to delete the specified zero sequence mutual coupling from the working case (activity PURG).

Batch command syntax:

```
BAT_PURGMUT  FRMBUS1  TOBUS1  'CKT1'  FRMBUS2  TOBUS2  'CKT2'
```

Python syntax:

```
ierr = purgmut(frmbus1, tobus1, ckt1, frmbus2, tobus2, ckt2)
```

Fortran syntax:

```
CALL PURGMUT(FRMBUS1, TOBUS1, CKT1, FRMBUS2, TOBUS2, CKT2, IERR)
```

where:

Integer FRMBUS1	Is one of the buses connected by the first branch of the mutual coupling (input; no default allowed).
Integer TOBUS1	Is the other bus connected by the first branch of the mutual coupling (input; no default allowed).
Character CKT1*2	Is the circuit identifier of the first branch of the mutual coupling (input; '1' by default).
Integer FRMBUS2	Is one of the buses connected by the second branch of the mutual coupling (input; no default allowed).
Integer TOBUS2	Is the other bus connected by the second branch of the mutual coupling (input; no default allowed).
Character CKT2*2	Is the circuit identifier of the second branch of the mutual coupling (input; '1' by default).
Integer IERR	Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	bus not found.
IERR = 2	branch not found.
IERR = 3	mutual coupling not found.
IERR = 4	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.219 PURGOWNER

Use this API to delete owners with no equipment assigned to them from the working case.

Batch command syntax:

```
BAT_PURGOWNER  SID  ALL
```

Python syntax:

```
ierr = purgowner(sid, all)
```

Fortran syntax:

```
CALL PURGOWNER(SID, ALL, IERR)
```

where:

Integer SID	Is a valid owner subsystem identifier. Valid subsystem identifiers range from 0 to 11. Owner subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all owners* or specified subsystem flag (input; 1 by default).	
	ALL = 1	delete all owners with no equipment assigned to them.
	ALL = 0	delete only owners in owner subsystem SID with no equipment assigned to them.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.220 PURGPLNT

Use this API to delete the plant and machine data at the specified bus from the working case (activity PURG).

Batch command syntax:

```
BAT_PURGPLNT  FRMBUS
```

Python syntax:

```
ierr = purgplnt (frmbus)
```

Fortran syntax:

```
CALL PURGPLNT (FRMBUS, IERR)
```

where:

Integer FRMBUS	Is the bus from which the plant is to be removed (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 0 no error occurred.
	IERR = 1 bus not found.
	IERR = 2 no machines at bus.
	IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.221 PURGSHUNT

Use this API to delete the specified fixed bus shunt from the working case (activity PURG).

Batch command syntax:

```
BAT_PURGSHUNT  FRMBUS  'ID'
```

Python syntax:

```
ierr = purgshunt(frmbus, id)
```

Fortran syntax:

```
CALL PURGSHUNT(FRMBUS, ID, IERR)
```

where:

Integer FRMBUS	Is the bus to which the shunt ID is connected (input; no default allowed).
Character ID*2	Is the identifier of the shunt to be removed (input; '1' by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 bus not found. IERR = 2 no fixed shunts at bus. IERR = 3 shunt not found. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.222 PURGSHUNTS

Use this API to delete all fixed shunts from the specified bus in the working case (activity PURG).

Batch command syntax:

```
BAT_PURGSHUNTS  FRMBUS
```

Python syntax:

```
ierr = purgshunts(frmbus)
```

Fortran syntax:

```
CALL PURGSHUNTS(FRMBUS, IERR)
```

where:

Integer FRMBUS	Is the bus from which all fixed shunts are to be removed (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 bus not found.
	IERR = 2 no fixed shunts at bus.
	IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.223 PURGSWS

Use this API to delete the specified switched shunt from the working case (activity PURG).

Batch command syntax:

```
BAT_PURGSWS  FRMBUS
```

Python syntax:

```
ierr = purgsws(frmbus)
```

Fortran syntax:

```
CALL PURGSWS(FRMBUS, IERR)
```

where:

Integer FRMBUS	Is the bus from which the switched shunt is to be removed (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 bus not found.
	IERR = 2 no switched shunt at bus.
	IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.224 PURGVSC

Use this API to delete the specified VSC dc line from the working case (activity PURG).

Batch command syntax:

```
BAT_PURGVSC  'NAME'
```

Python syntax:

```
ierr = purgvsc(name)
```

Fortran syntax:

```
CALL PURGVSC(NAME, IERR)
```

where:

Character NAME*12	Is the name of the VSC dc line to be removed (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 dc line not found.
	IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.225 PURGZONE

Use this API to delete zones with no equipment assigned to them from the working case.

Batch command syntax:

```
BAT_PURGZONE  SID  ALL
```

Python syntax:

```
ierr = purgzone(sid, all)
```

Fortran syntax:

```
CALL PURGZONE(SID, ALL, IERR)
```

where:

Integer SID	Is a valid zone subsystem identifier. Valid subsystem identifiers range from 0 to 11. Zone subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all zones* or specified subsystem flag (input; 1 by default).	
	ALL = 1	delete all zones with no equipment assigned to them.
	ALL = 0	delete only zones in zone subsystem SID with no equipment assigned to them.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.226 PURGE_MULTI_TERM_DC_BUS

Use this API to delete the specified dc bus from the specified multi-terminal dc line in the working case.

Batch command syntax:

```
BAT_PURGE_MULTI_TERM_DC_BUS  DC  IBUS
```

Python syntax:

```
ierr = purge_multi_term_dc_bus(dc, ibus)
```

Fortran syntax:

```
CALL PURGMDC_DC_BUS_API(NUM, IBUS, IERR, NERR)

CALL PURGMDC_DC_BUS_API_I(NUM, IBUS, IERR)

CALL PURGMDC_DC_BUS_API_C(NAME, IBUS, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the multi-terminal dc line from which a dc bus is to be removed; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the multi-terminal dc line from which a dc bus is to be removed (input; no default allowed).
Character*12 NAME	Is the name of the multi-terminal dc line from which a dc bus is to be removed (input; no default allowed).
Integer IBUS	Is the number of the dc bus to be removed (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred: IERR = 1 dc line not found. IERR = 2 invalid dc bus number. IERR = 3 dc bus not present in this multi-terminal dc line. IERR = 4 prerequisite requirements for API are not met. IERR = -1 data error, warning and information messages; one or more of: • dc bus no longer connected to converter as first dc bus. • dc bus no longer connected to converter as second dc bus. • dc link deleted.
Integer NERR	Is the number of error entries in PBUF02 (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.227 PURGE_MULTI_TERM_DC_CONVR

Use this API to delete the specified converter from the specified multi-terminal dc line in the working case.

Batch command syntax:

```
BAT_PURGE_MULTI_TERM_DC_CONVR  DC  IBUS
```

Python syntax:

```
ierr = purge_multi_term_dc_convr(dc, ibus)
```

Fortran syntax:

```
CALL PURGMDC_CONVERTER_API  (NUM, IBUS, IERR, NERR)

CALL PURGMDC_CONVERTER_API_I(NUM, IBUS, IERR)

CALL PURGMDC_CONVERTER_API_C(NAME, IBUS, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the multi-terminal dc line from which a converter is to be removed; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the multi-terminal dc line from which a converter is to be removed (input; no default allowed).
Character*12 NAME	Is the name of the multi-terminal dc line from which a converter is to be removed (input; no default allowed).
Integer IBUS	Is the number of the ac bus from which the converter is to be removed (input).
Integer IERR	Is the error code (output). IERR = 0 — no error occurred. IERR > 0 — fatal error occurred: IERR = 1 — dc line not found. IERR = 2 — converter bus not found. IERR = 3 — bus is not a converter bus of this multi-terminal dc line. IERR = 4 — prerequisite requirements for API are not met. IERR = -1 — data error, warning and information messages; one or more of: • this converter bus is no longer its pole's voltage controlling converter. • the voltage setpoint of the new voltage controlling converter is not positive. • the voltage setpoint of the new voltage controlling converter is less than the mode switch voltage.
Integer NERR	Is the number of error entries in PBUF02 (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.228 PURGE_MULTI_TERM_DC_LINK

Use this API to delete the specified dc link from the specified multi-terminal dc line in the working case.

Batch command syntax:

```
BAT_PURGE_MULTI_TERM_DC_LINK  DC  IBUS  JBUS  'CKT'
```

Python syntax:

```
ierr = purge_multi_term_dc_link(dc, ibus, jbus, ckt)
```

Fortran syntax:

```
CALL PURGMDC_DC_LINK_API  (NUM, IBUS, JBUS, CKT, IERR, NERR)

CALL PURGMDC_DC_LINK_API_I(NUM, IBUS, JBUS, CKT, IERR)

CALL PURGMDC_DC_LINK_API_C(NAME, IBUS, JBUS, CKT, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the multi-terminal dc line from which a dc link is to be removed; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the multi-terminal dc line from which a dc link is to be removed (input; no default allowed).
Character*12 NAME	Is the name of the multi-terminal dc line from which a dc link is to be removed (input; no default allowed).
Integer IBUS	Is one of the dc buses connected by the dc link to be removed (input).
Integer JBUS	Is the other dc bus connected by the dc link (input).
Character*1 CKT	Is the circuit identifier of the dc link to be deleted (input).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred: IERR = 1 dc line not found. IERR = 2 invalid dc bus number. IERR = 3 dc bus not present in this multi-terminal dc line. IERR = 4 circuit identifier is more than one character. IERR = 5 dc link not found in this multi-terminal dc line. IERR = 6 prerequisite requirements for API are not met.
Integer NERR	Is the number of error entries in PBUF02 (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.229 PV_ENGINE

This API routine is obsolete. It has been replaced by the API routine PV_ENGINE_1A, and is implemented by a call to the PV_ENGINE_1A API routine.

Use this API routine to run the first release of the PV analysis calculation engine. The same set of automatic adjustments is applied in base case and contingency case transfer increment solutions.

Batch command syntax:

```
BAT_PV_ENGINE  OPTIONS(1)...OPTIONS(16)  VALUES(1)...VALUES(6)
LABELS(1)  LABELS(2)  DFXFILE  THRFILE  ECDFILE  ACCFILE
```

Python syntax:

```
ierr = pv_engine(options, values, labels, dfxfile, thrfile, ecdfile,
accfile)
```

Fortran syntax:

```
CALL PV_ENGINE(OPTIONS, VALUES, LABELS, DFXFILE, THRFILE, ECDFILE,
ACCFILE, IERR)
```

where:

Integer OPTIONS(16) Is an array of sixteen elements specifying solution options (input). The values are as follows:

OPTIONS(1)	tap adjustment flag (tap adjustment option setting by default).
OPTIONS(1) = 0	disable.
OPTIONS(1) = 1	enable stepping adjustment.
OPTIONS(1) = 2	enable direct adjustment.
OPTIONS(2)	area interchange adjustment flag (area interchange adjustment option setting by default).
OPTIONS(2) = 0	disable.
OPTIONS(2) = 1	enable using tie line flows only in calculating area interchange.
OPTIONS(2) = 2	enable using tie line flows and loads in calculating area interchange.
OPTIONS(3)	phase shift adjustment flag (phase shift adjustment option setting by default).
OPTIONS(3) = 0	disable.
OPTIONS(3) = 1	enable.
OPTIONS(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
OPTIONS(4) = 0	disable.
OPTIONS(4) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
OPTIONS(5) = 0	disable.
OPTIONS(5) = 1	enable.
OPTIONS(5) = 2	enable continuous mode, disable discrete mode.
OPTIONS(6)	non-divergent solution flag (non-divergent solution option setting by default).
OPTIONS(6) = 0	disable.
OPTIONS(6) = 1	enable.
OPTIONS(7)	solution method flag (0 by default).
OPTIONS(7) = 0	FDNS.
OPTIONS(7) = 1	FNSL.
OPTIONS(7) = 2	optimized FDNS.
OPTIONS(8)	var limit code for the contingency case initial power flow solution (0 by default).
OPTIONS(8) = 0	apply immediately.
OPTIONS(8) = 1	initially ignore, then apply automatically.
OPTIONS(9)	var limit code for the subsequent transfer increment cases (0 by default).
OPTIONS(9) = 0	apply immediately.
OPTIONS(9) = 1	initially ignore, then apply automatically.
OPTIONS(10)	rating set (rating set program option setting by default).
OPTIONS(10) = 1	RATEA.
OPTIONS(10) = 2	RATEB.
OPTIONS(10) = 3	RATEC.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(11)	study (source) system transfer dispatch method (1 by default).
OPTIONS(11) = 1	buses and distribution factors from DFAX file for buses with positive MW machines.
OPTIONS(11) = 2	buses and distribution factors from DFAX file for buses with positive MW constant MVA load.
OPTIONS(11) = 3	buses and distribution factors from DFAX file for buses with either positive MW machines or positive MW constant MVA load.
OPTIONS(11) = 4	subsystem buses with positive MW constant MVA load in proportion to their MW load.
OPTIONS(11) = 5	subsystem buses with positive MW machines in proportion to their MW output.
OPTIONS(11) = 6	subsystem buses with positive MW machines in proportion to their MBASEs.
OPTIONS(11) = 7	subsystem buses with positive MW machines in proportion to their reserve (PMAXMC - PGENMC).
OPTIONS(11) = 8	subsystem buses with positive MW machines via ECDI with unit commitment disabled.
OPTIONS(12)	opposing (sink) system transfer dispatch method (1 by default).
OPTIONS(12) = 1	buses and distribution factors from DFAX file for buses with positive MW machines.
OPTIONS(12) = 2	buses and distribution factors from DFAX file for buses with positive MW constant MVA load.
OPTIONS(12) = 3	buses and distribution factors from DFAX file for buses with either positive MW machines or positive MW constant MVA load.
OPTIONS(12) = 4	subsystem buses with positive MW constant MVA load in proportion to their MW load.
OPTIONS(12) = 5	subsystem buses with positive MW machines in proportion to their MW output.
OPTIONS(12) = 6	subsystem buses with positive MW machines in proportion to their MBASEs.
OPTIONS(12) = 7	subsystem buses with positive MW machines in proportion to their reserve (PGENMC - PMINMC).
OPTIONS(12) = 8	subsystem buses with positive MW machines via ECDI with unit commitment disabled.
OPTIONS(13)	generation plant limits flag for transfer methods 1, 3, 5, 6 and 7 (0 by default).
OPTIONS(13) = 0	no limits.
OPTIONS(13) = 1	honor machine active power limits.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTIONS(14)	positive load flag for transfer methods 2, 3 and 4 (0 by default).
	OPTIONS(14) = 0	no limits.
	OPTIONS(14) = 1	enforce non-negative net active power constant MVA load.
	OPTIONS(15)	flag to check for low voltage at any monitored bus (0 by default).
	OPTIONS(15) = 0	disable check.
	OPTIONS(15) = 1	enable check.
	OPTIONS(16)	flag to check for excessive loading on any monitored branch (0 by default).
	OPTIONS(16) = 0	disable check.
	OPTIONS(16) = 1	enable check.
Real VALUES(6)	Is an array of six elements specifying solution values (input). The values are as follows:	
	VALUES(1)	mismatch tolerance in MW and Mvar (Newton solution convergence tolerance, TOLN, by default).
	VALUES(2)	initial transfer increment in MW (100.0 by default).
	VALUES(3)	transfer increment tolerance in MW (10.0 by default).
	VALUES(4)	maximum incremental transfer in MW (1000.0 by default).
	VALUES(5)	low voltage threshold in pu in the low voltage check (0.8 by default).
	VALUES(6)	percent of rating in the excessive branch loading check (100.0 by default).
Character LABELS(2)*12	Is an array of two elements specifying subsystem labels (input). The values are as follows:	
	LABELS(1)	label of the study (source) system for which generation is to be increased and/or for which the load is to be decreased (no default allowed).
	LABELS(2)	label of the opposing (sink) system for which generation is to be decreased and/or for which the load is to be increased (no default allowed).
Character DFXFILE*260	Is the name of Distribution Factor Data file (input; no default allowed).	
Character THRFILE*260	Is the name of Load Throwover Data file; blank for none (input; blank by default).	
Character ECDFILE*260	Is the name of Economic Dispatch Data file; blank for none (input; blank by default).	
Character ACCFILE*260	Is the name of PV Results File (input; no default allowed).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR > 0	as for PV_ENGINE_1A.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.230 PV_ENGINE_1A

Use this API routine to run the second release of the first PV analysis calculation engine. The same set of automatic adjustments, and the same treatment of induction motors that fail to solve due to low terminal voltage, is applied in base case and contingency case transfer increment solutions.

Batch command syntax:

```
BAT_PV_ENGINE_1A  OPTIONS(1)...OPTIONS(17)  VALUES(1)...VALUES(6)
LABELS(1)  LABELS(2)  DFXFILE  THRFILE  ECDFILE  ACCFILE
```

Python syntax:

```
ierr = pv_engine_1a(options, values, labels, dfxfile, thrfile,
ecdfile, accfile)
```

Fortran syntax:

```
CALL PV_ENGINE_1A(OPTIONS, VALUES, LABELS, DFXFILE, THRFILE, ECDFILE,
ACCFILE, IERR)
```

where:

Integer OPTIONS(17) Is an array of seventeen elements specifying solution options (input). The values are as follows:

OPTIONS(1)	tap adjustment flag (tap adjustment option setting by default).
OPTIONS(1) = 0	disable.
OPTIONS(1) = 1	enable stepping adjustment.
OPTIONS(1) = 2	enable direct adjustment.
OPTIONS(2)	area interchange adjustment flag (area interchange adjustment option setting by default).
OPTIONS(2) = 0	disable.
OPTIONS(2) = 1	enable using tie line flows only in calculating area interchange.
OPTIONS(2) = 2	enable using tie line flows and loads in calculating area interchange.
OPTIONS(3)	phase shift adjustment flag (phase shift adjustment option setting by default).
OPTIONS(3) = 0	disable.
OPTIONS(3) = 1	enable.
OPTIONS(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
OPTIONS(4) = 0	disable.
OPTIONS(4) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
OPTIONS(5) = 0	disable.
OPTIONS(5) = 1	enable.
OPTIONS(5) = 2	enable continuous mode, disable discrete mode.
OPTIONS(6)	induction motor treatment flag; applied when an induction motor fails to solve due to low terminal voltage (0 by default).
OPTIONS(6) = 0	stall.
OPTIONS(6) = 1	trip.
OPTIONS(7)	non-divergent solution flag (non-divergent solution option setting by default).
OPTIONS(7) = 0	disable.
OPTIONS(7) = 1	enable.
OPTIONS(8)	solution method flag (0 by default).
OPTIONS(8) = 0	FDNS.
OPTIONS(8) = 1	FNSL.
OPTIONS(8) = 2	optimized FDNS.
OPTIONS(9)	var limit code for the contingency case initial power flow solution (0 by default).
OPTIONS(9) = 0	apply immediately.
OPTIONS(9) = 1	initially ignore, then apply automatically.
OPTIONS(10)	var limit code for the subsequent transfer increment cases (0 by default).
OPTIONS(10) = 0	apply immediately.
OPTIONS(10) = 1	initially ignore, then apply automatically.
OPTIONS(11)	rating set (rating set program option setting by default).
OPTIONS(11) = 1	RATEA.
OPTIONS(11) = 2	RATEB.
OPTIONS(11) = 3	RATEC.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(12)	study (source) system transfer dispatch method (1 by default).
OPTIONS(12) = 1	buses and distribution factors from DFAX file for buses with positive MW machines.
OPTIONS(12) = 2	buses and distribution factors from DFAX file for buses with positive MW constant MVA load.
OPTIONS(12) = 3	buses and distribution factors from DFAX file for buses with either positive MW machines or positive MW constant MVA load.
OPTIONS(12) = 4	subsystem buses with positive MW constant MVA load in proportion to their MW load.
OPTIONS(12) = 5	subsystem buses with positive MW machines in proportion to their MW output.
OPTIONS(12) = 6	subsystem buses with positive MW machines in proportion to their MBASEs.
OPTIONS(12) = 7	subsystem buses with positive MW machines in proportion to their reserve (PMAXMC - PGENMC).
OPTIONS(12) = 8	subsystem buses with positive MW machines via ECDI with unit commitment disabled.
OPTIONS(13)	opposing (sink) system transfer dispatch method (1 by default).
OPTIONS(13) = 1	buses and distribution factors from DFAX file for buses with positive MW machines.
OPTIONS(13) = 2	buses and distribution factors from DFAX file for buses with positive MW constant MVA load.
OPTIONS(13) = 3	buses and distribution factors from DFAX file for buses with either positive MW machines or positive MW constant MVA load.
OPTIONS(13) = 4	subsystem buses with positive MW constant MVA load in proportion to their MW load.
OPTIONS(13) = 5	subsystem buses with positive MW machines in proportion to their MW output.
OPTIONS(13) = 6	subsystem buses with positive MW machines in proportion to their MBASEs.
OPTIONS(13) = 7	subsystem buses with positive MW machines in proportion to their reserve (PGENMC - PMINMC).
OPTIONS(13) = 8	subsystem buses with positive MW machines via ECDI with unit commitment disabled.
OPTIONS(14)	generation plant limits flag for transfer methods 1, 3, 5, 6 and 7 (0 by default).
OPTIONS(14) = 0	no limits.
OPTIONS(14) = 1	honor machine active power limits.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTIONS(15)	positive load flag for transfer methods 2, 3 and 4 (0 by default).
	OPTIONS(15) = 0	no limits.
	OPTIONS(15) = 1	enforce non-negative net active power constant MVA load.
	OPTIONS(16)	flag to check for low voltage at any monitored bus (0 by default).
	OPTIONS(16) = 0	disable check.
	OPTIONS(16) = 1	enable check.
	OPTIONS(17)	flag to check for excessive loading on any monitored branch (0 by default).
	OPTIONS(17) = 0	disable check.
	OPTIONS(17) = 1	enable check.
Real VALUES(6)	Is an array of six elements specifying solution values (input). The values are as follows:	
	VALUES(1)	mismatch tolerance in MW and Mvar (Newton solution convergence tolerance, TOLN, by default).
	VALUES(2)	initial transfer increment in MW (100.0 by default).
	VALUES(3)	transfer increment tolerance in MW (10.0 by default).
	VALUES(4)	maximum incremental transfer in MW (1000.0 by default).
	VALUES(5)	low voltage threshold in pu in the low voltage check (0.8 by default).
	VALUES(6)	percent of rating in the excessive branch loading check (100.0 by default).
Character LABELS(2)*12	Is an array of two elements specifying subsystem labels (input). The values are as follows:	
	LABELS(1)	label of the study (source) system for which generation is to be increased and/or for which the load is to be decreased (no default allowed).
	LABELS(2)	label of the opposing (sink) system for which generation is to be decreased and/or for which the load is to be increased (no default allowed).
Character DFXFILE*260	Is the name of Distribution Factor Data file (input; no default allowed).	
Character THRFILE*260	Is the name of Load Throwover Data file; blank for none (input; blank by default).	
Character ECDFILE*260	Is the name of Economic Dispatch Data file; blank for none (input; blank by default).	
Character ACCFILE*260	Is the name of PV Results File (input; no default allowed).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid VALUES value.
IERR = 2	invalid OPTIONS value.
IERR = 3	generators are converted.
IERR = 4	buses in island(s) without a swing bus; use activity TREE.
IERR = 5	largest mismatch exceeds mismatch tolerance.
IERR = 6	initial transfer increment is less than transfer increment tolerance.
IERR = 7	no participating dispatch buses in study system.
IERR = 8	no participating dispatch buses in opposing system.
IERR = 9	subsystem engine error.
IERR = 10	ECDI engine error in subsystem aaa.
IERR = 11	base case ECDI solution did not converge.
IERR = 12	no Distribution Factor Data File specified.
IERR = 13	no PV Results File specified.
IERR = 14	no Economic Dispatch Data File specified.
IERR = 15	in-service induction machines are in the "stalled" or "tripped" state.
IERR = 16	bus type code and series element status inconsistencies.
IERR = 21	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
IERR = 22	monitored elements exceed limit when adding multi-section line members.
IERR = 23	no subsystems defined.
IERR = 24	subsystem not found.
IERR = 25	specified subsystems overlap.
IERR = 26	no in-service buses in subsystem WORLD.
IERR = 50	error opening Economic Dispatch Data File.
IERR = 51	error opening PV Results File.
IERR = 52	error opening Distribution Factor Data File.
IERR = 53	error opening Load Throwover Data File.
IERR = 56	prerequisite requirements for API are not met.
IERR = 99	terminated by user interrupt or file error.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.231 PV_ENGINE_2

This API routine is obsolete. It has been replaced by the API routine PV_ENGINE_6, and is implemented by a call to PV_ENGINE_6.

Use this API routine to run the second release of the PV analysis calculation engine. It provides for a different set of automatic adjustments in contingency case solutions than is used in the base case transfer increment solutions.

Batch command syntax:

```
BAT_PV_ENGINE_2  OPTIONS(1)...OPTIONS(21)  VALUES(1)...VALUES(6)
LABELS(1)  LABELS(2)  DFXFILE  THRFILE  ECDFILE  ACCFILE
```

Python syntax:

```
ierr = pv_engine_2(options, values, labels, dfxfile, thrfile,
ecdfile, accfile)
```

Fortran syntax:

```
CALL PV_ENGINE_2(OPTIONS, VALUES, LABELS, DFXFILE, THRFILE, ECDFILE,
ACCFILE, IERR)
```

where:

Integer OPTIONS(21)	Is an array of twenty-one elements specifying solution options (input). The values are as follows:	
	OPTIONS(1)	base case tap adjustment flag (tap adjustment option setting by default).
	OPTIONS(1) = 0	disable.
	OPTIONS(1) = 1	enable stepping adjustment.
	OPTIONS(1) = 2	enable direct adjustment.
	OPTIONS(2)	base case area interchange adjustment flag (area interchange adjustment option setting by default).
	OPTIONS(2) = 0	disable.
	OPTIONS(2) = 1	enable using tie line flows only in calculating area interchange.
	OPTIONS(2) = 2	enable using tie line flows and loads in calculating area interchange.
	OPTIONS(3)	base case phase shift adjustment flag (phase shift adjustment option setting by default).
	OPTIONS(3) = 0	disable.
	OPTIONS(3) = 1	enable.
	OPTIONS(4)	base case dc tap adjustment flag (dc tap adjustment option setting by default).
	OPTIONS(4) = 0	disable.
	OPTIONS(4) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(5)	base case switched shunt adjustment flag (switched shunt adjustment option setting by default).
OPTIONS(5) = 0	disable.
OPTIONS(5) = 1	enable.
OPTIONS(5) = 2	enable continuous mode, disable discrete mode.
OPTIONS(6)	non-divergent solution flag (non-divergent solution option setting by default).
OPTIONS(6) = 0	disable.
OPTIONS(6) = 1	enable.
OPTIONS(7)	solution method flag (0 by default).
OPTIONS(7) = 0	FDNS.
OPTIONS(7) = 1	FNSL.
OPTIONS(7) = 2	optimized FDNS.
OPTIONS(8)	var limit code for the contingency case power flow solutions (0 by default).
OPTIONS(8) = 0	apply immediately.
OPTIONS(8) = 1	initially ignore, then apply automatically.
OPTIONS(9)	var limit code for the base case transfer increment solutions (0 by default).
OPTIONS(9) = 0	apply immediately.
OPTIONS(9) = 1	initially ignore, then apply automatically.
OPTIONS(10)	rating set (rating set program option setting by default).
OPTIONS(10) = 1	RATEA.
OPTIONS(10) = 2	RATEB.
OPTIONS(10) = 3	RATEC.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(11)	study (source) system transfer dispatch method (1 by default).
OPTIONS(11) = 1	buses and distribution factors from DFAX file for buses with positive MW machines.
OPTIONS(11) = 2	buses and distribution factors from DFAX file for buses with positive MW constant MVA load.
OPTIONS(11) = 3	buses and distribution factors from DFAX file for buses with either positive MW machines or positive MW constant MVA load.
OPTIONS(11) = 4	subsystem buses with positive MW constant MVA load in proportion to their MW load.
OPTIONS(11) = 5	subsystem buses with positive MW machines in proportion to their MW output.
OPTIONS(11) = 6	subsystem buses with positive MW machines in proportion to their MBASEs.
OPTIONS(11) = 7	subsystem buses with positive MW machines in proportion to their reserve (PMAXMC - PGENMC).
OPTIONS(11) = 8	subsystem buses with positive MW machines via ECDI with unit commitment disabled.
OPTIONS(12)	opposing (sink) system transfer dispatch method (1 by default).
OPTIONS(12) = 1	buses and distribution factors from DFAX file for buses with positive MW machines.
OPTIONS(12) = 2	buses and distribution factors from DFAX file for buses with positive MW constant MVA load.
OPTIONS(12) = 3	buses and distribution factors from DFAX file for buses with either positive MW machines or positive MW constant MVA load.
OPTIONS(12) = 4	subsystem buses with positive MW constant MVA load in proportion to their MW load.
OPTIONS(12) = 5	subsystem buses with positive MW machines in proportion to their MW output.
OPTIONS(12) = 6	subsystem buses with positive MW machines in proportion to their MBASEs.
OPTIONS(12) = 7	subsystem buses with positive MW machines in proportion to their reserve (PGENMC - PMINMC).
OPTIONS(12) = 8	subsystem buses with positive MW machines via ECDI with unit commitment disabled.
OPTIONS(13)	generation plant limits flag for transfer methods 1, 3, 5, 6 and 7 (0 by default).
OPTIONS(13) = 0	no limits.
OPTIONS(13) = 1	honor machine active power limits.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(14)	positive load flag for transfer methods 2, 3 and 4 (0 by default).
OPTIONS(14) = 0	no limits.
OPTIONS(14) = 1	enforce non-negative net active power constant MVA load.
OPTIONS(15)	flag to check for low voltage at any monitored bus (0 by default).
OPTIONS(15) = 0	disable check.
OPTIONS(15) = 1	enable check.
OPTIONS(16)	flag to check for excessive loading on any monitored branch (0 by default).
OPTIONS(16) = 0	disable check.
OPTIONS(16) = 1	enable check.
OPTIONS(17)	contingency case tap adjustment flag (base case tap adjustment flag, OPTIONS(1), by default).
OPTIONS(17) = 0	disable.
OPTIONS(17) = 1	enable stepping adjustment.
OPTIONS(17) = 2	enable direct adjustment.
OPTIONS(18)	contingency case area interchange adjustment flag (base case area interchange adjustment flag, OPTIONS(2), by default).
OPTIONS(18) = 0	disable.
OPTIONS(18) = 1	enable using tie line flows only in calculating area interchange.
OPTIONS(18) = 2	enable using tie line flows and loads in calculating area interchange.
OPTIONS(19)	contingency case phase shift adjustment flag (base case phase shift adjustment flag, OPTIONS(3), by default).
OPTIONS(19) = 0	disable.
OPTIONS(19) = 1	enable.
OPTIONS(20)	contingency case dc tap adjustment flag (base case dc tap adjustment flag, OPTIONS(4), by default).
OPTIONS(20) = 0	disable.
OPTIONS(20) = 1	enable.
OPTIONS(21)	contingency case switched shunt adjustment flag (base case switched shunt adjustment flag, OPTIONS(5), by default).
OPTIONS(21) = 0	disable.
OPTIONS(21) = 1	enable.
OPTIONS(21) = 2	enable continuous mode, disable discrete mode.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real VALUES(6)	Is an array of six elements specifying solution values (input). The values are as follows:	
	VALUES(1)	mismatch tolerance in MW and Mvar (Newton solution convergence tolerance, TOLN, by default).
	VALUES(2)	initial transfer increment in MW (100.0 by default).
	VALUES(3)	transfer increment tolerance in MW (10.0 by default).
	VALUES(4)	maximum incremental transfer in MW (1000.0 by default).
	VALUES(5)	low voltage threshold in pu in the low voltage check (0.8 by default).
	VALUES(6)	percent of rating in the excessive branch loading check (100.0 by default).
Character LABELS(2)*12	Is an array of two elements specifying subsystem labels (input). The values are as follows:	
	LABELS(1)	label of the study (source) system for which generation is to be increased and/or for which the load is to be decreased (no default allowed).
	LABELS(2)	label of the opposing (sink) system for which generation is to be decreased and/or for which the load is to be increased (no default allowed).
Character DFXFILE*260	Is the name of Distribution Factor Data file (input; no default allowed).	
Character THRFILE*260	Is the name of Load Throwover Data file; blank for none (input; blank by default).	
Character ECDFILE*260	Is the name of Economic Dispatch Data file; blank for none (input; blank by default).	
Character ACCFILE*260	Is the name of PV Results File (input; no default allowed).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR > 0	as for PV_ENGINE_6.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.232 PV_ENGINE_3

This API routine is obsolete. It has been replaced by the API routine PV_ENGINE_6, and is implemented by a call to PV_ENGINE_6.

Use this API to run the third release of the PV analysis calculation engine. It provides for a different set of automatic adjustments in contingency case solutions than is used in the base case transfer increment solutions, and allows a negative minimum incremental transfer to be specified..

Batch command syntax:

```
BAT_PV_ENGINE_3  OPTIONS(1)...OPTIONS(21)  VALUES(1)...VALUES(7)
LABELS(1)  LABELS(2)  DFXFILE  THRFILE  ECDFILE  ACCFILE
```

Python syntax:

```
ierr = pv_engine_3(options, values, labels, dfxfile, thrfile,
ecdfile, accfile)
```

Fortran syntax:

```
CALL PV_ENGINE_3(OPTIONS, VALUES, LABELS, DFXFILE, THRFILE, ECDFILE,
ACCFILE, IERR)
```

where:

Integer OPTIONS(21)	Is an array of twenty-one elements specifying solution options (input). The values are as follows:	
	OPTIONS(1)	base case tap adjustment flag (tap adjustment option setting by default).
	OPTIONS(1) = 0	disable.
	OPTIONS(1) = 1	enable stepping adjustment.
	OPTIONS(1) = 2	enable direct adjustment.
	OPTIONS(2)	base case area interchange adjustment flag (area interchange adjustment option setting by default).
	OPTIONS(2) = 0	disable.
	OPTIONS(2) = 1	enable using tie line flows only in calculating area interchange.
	OPTIONS(2) = 2	enable using tie line flows and loads in calculating area interchange.
	OPTIONS(3)	base case phase shift adjustment flag (phase shift adjustment option setting by default).
	OPTIONS(3) = 0	disable.
	OPTIONS(3) = 1	enable.
	OPTIONS(4)	base case dc tap adjustment flag (dc tap adjustment option setting by default).
	OPTIONS(4) = 0	disable.
	OPTIONS(4) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(5)	base case switched shunt adjustment flag (switched shunt adjustment option setting by default).
OPTIONS(5) = 0	disable.
OPTIONS(5) = 1	enable.
OPTIONS(5) = 2	enable continuous mode, disable discrete mode.
OPTIONS(6)	non-divergent solution flag (non-divergent solution option setting by default).
OPTIONS(6) = 0	disable.
OPTIONS(6) = 1	enable.
OPTIONS(7)	solution method flag (0 by default).
OPTIONS(7) = 0	FDNS.
OPTIONS(7) = 1	FNSL.
OPTIONS(7) = 2	optimized FDNS.
OPTIONS(8)	var limit code for the contingency case power flow solutions (0 by default).
OPTIONS(8) = 0	apply immediately.
OPTIONS(8) = 1	initially ignore, then apply automatically.
OPTIONS(9)	var limit code for the base case transfer increment solutions (0 by default).
OPTIONS(9) = 0	apply immediately.
OPTIONS(9) = 1	initially ignore, then apply automatically.
OPTIONS(10)	rating set (rating set program option setting by default).
OPTIONS(10) = 1	RATEA.
OPTIONS(10) = 2	RATEB.
OPTIONS(10) = 3	RATEC.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(11)	study (source) system transfer dispatch method (1 by default).
OPTIONS(11) = 1	buses and distribution factors from DFAX file for buses with positive MW machines.
OPTIONS(11) = 2	buses and distribution factors from DFAX file for buses with positive MW constant MVA load.
OPTIONS(11) = 3	buses and distribution factors from DFAX file for buses with either positive MW machines or positive MW constant MVA load.
OPTIONS(11) = 4	subsystem buses with positive MW constant MVA load in proportion to their MW load.
OPTIONS(11) = 5	subsystem buses with positive MW machines in proportion to their MW output.
OPTIONS(11) = 6	subsystem buses with positive MW machines in proportion to their MBASEs.
OPTIONS(11) = 7	subsystem buses with positive MW machines in proportion to their reserve (PMAXMC - PGENMC).
OPTIONS(11) = 8	subsystem buses with positive MW machines via ECDI with unit commitment disabled.
OPTIONS(12)	opposing (sink) system transfer dispatch method (1 by default).
OPTIONS(12) = 1	buses and distribution factors from DFAX file for buses with positive MW machines.
OPTIONS(12) = 2	buses and distribution factors from DFAX file for buses with positive MW constant MVA load.
OPTIONS(12) = 3	buses and distribution factors from DFAX file for buses with either positive MW machines or positive MW constant MVA load.
OPTIONS(12) = 4	subsystem buses with positive MW constant MVA load in proportion to their MW load.
OPTIONS(12) = 5	subsystem buses with positive MW machines in proportion to their MW output.
OPTIONS(12) = 6	subsystem buses with positive MW machines in proportion to their MBASEs.
OPTIONS(12) = 7	subsystem buses with positive MW machines in proportion to their reserve (PGENMC - PMINMC).
OPTIONS(12) = 8	subsystem buses with positive MW machines via ECDI with unit commitment disabled.
OPTIONS(13)	generation plant limits flag for transfer methods 1, 3, 5, 6 and 7 (0 by default).
OPTIONS(13) = 0	no limits.
OPTIONS(13) = 1	honor machine active power limits.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(14)	positive load flag for transfer methods 2, 3 and 4 (0 by default).
OPTIONS(14) = 0	no limits.
OPTIONS(14) = 1	enforce non-negative net active power constant MVA load.
OPTIONS(15)	flag to check for low voltage at any monitored bus (0 by default).
OPTIONS(15) = 0	disable check.
OPTIONS(15) = 1	enable check.
OPTIONS(16)	flag to check for excessive loading on any monitored branch (0 by default).
OPTIONS(16) = 0	disable check.
OPTIONS(16) = 1	enable check.
OPTIONS(17)	contingency case tap adjustment flag (base case tap adjustment flag, OPTIONS(1), by default).
OPTIONS(17) = 0	disable.
OPTIONS(17) = 1	enable stepping adjustment.
OPTIONS(17) = 2	enable direct adjustment.
OPTIONS(18)	contingency case area interchange adjustment flag (base case area interchange adjustment flag, OPTIONS(2), by default).
OPTIONS(18) = 0	disable.
OPTIONS(18) = 1	enable using tie line flows only in calculating area interchange.
OPTIONS(18) = 2	enable using tie line flows and loads in calculating area interchange.
OPTIONS(19)	contingency case phase shift adjustment flag (base case phase shift adjustment flag, OPTIONS(3), by default).
OPTIONS(19) = 0	disable.
OPTIONS(19) = 1	enable.
OPTIONS(20)	contingency case dc tap adjustment flag (base case dc tap adjustment flag, OPTIONS(4), by default).
OPTIONS(20) = 0	disable.
OPTIONS(20) = 1	enable.
OPTIONS(21)	contingency case switched shunt adjustment flag (base case switched shunt adjustment flag, OPTIONS(5), by default).
OPTIONS(21) = 0	disable.
OPTIONS(21) = 1	enable.
OPTIONS(21) = 2	enable continuous mode, disable discrete mode.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real VALUES(7)	Is an array of seven elements specifying solution values (input). The values are as follows:	
	VALUES(1)	mismatch tolerance in MW and Mvar (Newton solution convergence tolerance, TOLN, by default).
	VALUES(2)	initial transfer increment in MW (100.0 by default).
	VALUES(3)	transfer increment tolerance in MW (10.0 by default).
	VALUES(4)	maximum incremental transfer in MW (1000.0 by default).
	VALUES(5)	low voltage threshold in pu in the low voltage check (0.8 by default).
	VALUES(6)	percent of rating in the excessive branch loading check (100.0 by default).
	VALUES(7)	minimum incremental transfer in MW ($\leq$ 0.0) (0.0 by default).
Character LABELS(2)*12	Is an array of two elements specifying subsystem labels (input). The values are as follows:	
	LABELS(1)	label of the study (source) system for which generation is to be increased and/or for which the load is to be decreased (no default allowed).
	LABELS(2)	label of the opposing (sink) system for which generation is to be decreased and/or for which the load is to be increased (no default allowed).
Character DFXFILE*260	Is the name of Distribution Factor Data file (input; no default allowed).	
Character THRFILE*260	Is the name of Load Throwover Data file; blank for none (input; blank by default).	
Character ECDFILE*260	Is the name of Economic Dispatch Data file; blank for none (input; blank by default).	
Character ACCFILE*260	Is the name of PV Results File (input; no default allowed).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR > 0	as for PV_ENGINE_6.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.233 PV_ENGINE_4

This API routine is obsolete. It has been replaced by the API routine PV_ENGINE_6, and is implemented by a call to PV_ENGINE_6.

Use this API routine to run the fourth release of the P-V analysis calculation engine. It provides for a different set of automatic adjustments, and for different treatment of induction motors that fail to solve due to low terminal voltage, in contingency case solutions than is used in the base case transfer increment solutions, and allows a negative minimum incremental transfer to be specified.

Batch command syntax:

```
BAT_PV_ENGINE_4  OPTIONS(1)...OPTIONS(23)  VALUES(1)...VALUES(7)
LABELS(1)  LABELS(2)  DFXFILE  THRFILE  ECDFILE  ACCFILE
```

Python syntax:

```
ierr = pv_engine_4(options, values, labels, dfxfile, thrfile,
ecdfile, accfile)
```

Fortran syntax:

```
CALL PV_ENGINE_4(OPTIONS, VALUES, LABELS, DFXFILE, THRFILE, ECDFILE,
ACCFILE, IERR)
```

where:

Integer OPTIONS(23)	Is an array of twenty-three elements specifying solution options (input). The values are as follows:	
	OPTIONS(1)	base case tap adjustment flag (tap adjustment option setting by default).
	OPTIONS(1) = 0	disable.
	OPTIONS(1) = 1	enable stepping adjustment.
	OPTIONS(1) = 2	enable direct adjustment.
	OPTIONS(2)	base case area interchange adjustment flag (area interchange adjustment option setting by default).
	OPTIONS(2) = 0	disable.
	OPTIONS(2) = 1	enable using tie line flows only in calculating area interchange.
	OPTIONS(2) = 2	enable using tie line flows and loads in calculating area interchange.
	OPTIONS(3)	base case phase shift adjustment flag (phase shift adjustment option setting by default).
	OPTIONS(3) = 0	disable.
	OPTIONS(3) = 1	enable.
	OPTIONS(4)	base case dc tap adjustment flag (dc tap adjustment option setting by default).
	OPTIONS(4) = 0	disable.
	OPTIONS(4) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(5)	base case switched shunt adjustment flag (switched shunt adjustment option setting by default).
OPTIONS(5) = 0	disable.
OPTIONS(5) = 1	enable.
OPTIONS(5) = 2	enable continuous mode, disable discrete mode.
OPTIONS(6)	base case induction motor treatment flag; applied when an induction motor fails to solve due to low terminal voltage (0 by default).
OPTIONS(6) = 0	stall.
OPTIONS(6) = 1	trip.
OPTIONS(7)	non-divergent solution flag (non-divergent solution option setting by default).
OPTIONS(7) = 0	disable.
OPTIONS(7) = 1	enable.
OPTIONS(8)	solution method flag (0 by default).
OPTIONS(8) = 0	FDNS.
OPTIONS(8) = 1	FNSL.
OPTIONS(8) = 2	optimized FDNS.
OPTIONS(9)	var limit code for the contingency case power flow solutions (0 by default).
OPTIONS(9) = 0	apply immediately.
OPTIONS(9) = 1	initially ignore, then apply automatically.
OPTIONS(10)	var limit code for the base case transfer increment solutions (0 by default).
OPTIONS(10) = 0	apply immediately.
OPTIONS(10) = 1	initially ignore, then apply automatically.
OPTIONS(11)	rating set (rating set program option setting by default).
OPTIONS(11) = 1	RATEA.
OPTIONS(11) = 2	RATEB.
OPTIONS(11) = 3	RATEC.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(12)	study (source) system transfer dispatch method (1 by default).
OPTIONS(12) = 1	buses and distribution factors from DFAX file for buses with positive MW machines.
OPTIONS(12) = 2	buses and distribution factors from DFAX file for buses with positive MW constant MVA load.
OPTIONS(12) = 3	buses and distribution factors from DFAX file for buses with either positive MW machines or positive MW constant MVA load.
OPTIONS(12) = 4	subsystem buses with positive MW constant MVA load in proportion to their MW load.
OPTIONS(12) = 5	subsystem buses with positive MW machines in proportion to their MW output.
OPTIONS(12) = 6	subsystem buses with positive MW machines in proportion to their MBASEs.
OPTIONS(12) = 7	subsystem buses with positive MW machines in proportion to their reserve (PMAXMC - PGENMC).
OPTIONS(12) = 8	subsystem buses with positive MW machines via ECDI with unit commitment disabled.
OPTIONS(13)	opposing (sink) system transfer dispatch method (1 by default).
OPTIONS(13) = 1	buses and distribution factors from DFAX file for buses with positive MW machines.
OPTIONS(13) = 2	buses and distribution factors from DFAX file for buses with positive MW constant MVA load.
OPTIONS(13) = 3	buses and distribution factors from DFAX file for buses with either positive MW machines or positive MW constant MVA load.
OPTIONS(13) = 4	subsystem buses with positive MW constant MVA load in proportion to their MW load.
OPTIONS(13) = 5	subsystem buses with positive MW machines in proportion to their MW output.
OPTIONS(13) = 6	subsystem buses with positive MW machines in proportion to their MBASEs.
OPTIONS(13) = 7	subsystem buses with positive MW machines in proportion to their reserve (PGENMC - PMINMC).
OPTIONS(13) = 8	subsystem buses with positive MW machines via ECDI with unit commitment disabled.
OPTIONS(14)	generation plant limits flag for transfer methods 1, 3, 5, 6 and 7 (0 by default).
OPTIONS(14) = 0	no limits.
OPTIONS(14) = 1	honor machine active power limits.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(15)	positive load flag for transfer methods 2, 3 and 4 (0 by default).
OPTIONS(15) = 0	no limits.
OPTIONS(15) = 1	enforce non-negative net active power constant MVA load.
OPTIONS(16)	flag to check for low voltage at any monitored bus (0 by default).
OPTIONS(16) = 0	disable check.
OPTIONS(16) = 1	enable check.
OPTIONS(17)	flag to check for excessive loading on any monitored branch (0 by default).
OPTIONS(17) = 0	disable check.
OPTIONS(17) = 1	enable check.
OPTIONS(18)	contingency case tap adjustment flag (base case tap adjustment flag, OPTIONS(1), by default).
OPTIONS(18) = 0	disable.
OPTIONS(18) = 1	enable stepping adjustment.
OPTIONS(18) = 2	enable direct adjustment.
OPTIONS(19)	contingency case area interchange adjustment flag (base case area interchange adjustment flag, OPTIONS(2), by default).
OPTIONS(19) = 0	disable.
OPTIONS(19) = 1	enable using tie line flows only in calculating area interchange.
OPTIONS(19) = 2	enable using tie line flows and loads in calculating area interchange.
OPTIONS(20)	contingency case phase shift adjustment flag (base case phase shift adjustment flag, OPTIONS(3), by default).
OPTIONS(20) = 0	disable.
OPTIONS(20) = 1	enable.
OPTIONS(21)	contingency case dc tap adjustment flag (base case dc tap adjustment flag, OPTIONS(4), by default).
OPTIONS(21) = 0	disable.
OPTIONS(21) = 1	enable.
OPTIONS(22)	contingency case switched shunt adjustment flag (base case switched shunt adjustment flag, OPTIONS(5), by default).
OPTIONS(22) = 0	disable.
OPTIONS(22) = 1	enable.
OPTIONS(22) = 2	enable continuous mode, disable discrete mode.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTIONS(23)	contingency case induction motor treatment flag; applied when an induction motor fails to solve due to low terminal voltage (0 by default).
	OPTIONS(23) = 0	stall.
	OPTIONS(23) = 1	trip.
Real VALUES(7)	Is an array of seven elements specifying solution values (input). The values are as follows:	
	VALUES(1)	mismatch tolerance in MW and Mvar (Newton solution convergence tolerance, TOLN, by default).
	VALUES(2)	initial transfer increment in MW (100.0 by default).
	VALUES(3)	transfer increment tolerance in MW (10.0 by default).
	VALUES(4)	maximum incremental transfer in MW (1000.0 by default).
	VALUES(5)	low voltage threshold in pu in the low voltage check (0.8 by default).
	VALUES(6)	percent of rating in the excessive branch loading check (100.0 by default).
	VALUES(7)	minimum incremental transfer in MW ($\leq$ 0.0) (0.0 by default).
Character LABELS(2)*12	Is an array of two elements specifying subsystem labels (input). The values are as follows:	
	LABELS(1)	label of the study (source) system for which generation is to be increased and/or for which the load is to be decreased (no default allowed).
	LABELS(2)	label of the opposing (sink) system for which generation is to be decreased and/or for which the load is to be increased (no default allowed).
Character DFXFILE*260	Is the name of Distribution Factor Data file (input; no default allowed).	
Character THRFILE*260	Is the name of Load Throwover Data file; blank for none (input; blank by default).	
Character ECDFILE*260	Is the name of Economic Dispatch Data file; blank for none (input; blank by default).	
Character ACCFILE*260	Is the name of PV Results File (input; no default allowed).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR > 0	as for PV_ENGINE_6.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.234 PV_ENGINE_5

This API routine is obsolete. It has been replaced by the API routine PV_ENGINE_6, and is implemented by a call to PV_ENGINE_6.

Use this API routine to run the fifth release of the P-V analysis calculation engine. It provides for a different set of automatic adjustments, and for different treatment of induction motors that fail to solve due to low terminal voltage, in contingency case solutions than is used in the base case transfer increment solutions; it allows a negative minimum incremental transfer to be specified; for transfer increment methods involving load, it allows a power factor to be specified; and it provides for dispatching power unbalances resulting from the application of contingencies.

Batch command syntax:

```
BAT_PV_ENGINE_5  OPTIONS(1)...OPTIONS(24)  VALUES(1)...VALUES(8)
LABELS(1)...LABELS(3)  DFXFILE  THRFILE  ECDFILE  INLFILE  ACCFILE
```

Python syntax:

```
ierr = pv_engine_5(options, values, labels, dfxfile, thrfile,
ecdfile, inlfile, accfile)
```

Fortran syntax:

```
CALL PV_ENGINE_5(OPTIONS, VALUES, LABELS, DFXFILE, THRFILE, ECDFILE,
INLFILE, ACCFILE, IERR)
```

where:

Integer OPTIONS(24)	Is an array of twenty-four elements specifying solution options (input). The values are as follows:	
	OPTIONS(1)	base case tap adjustment flag (tap adjustment option setting by default).
	OPTIONS(1) = 0	disable.
	OPTIONS(1) = 1	enable stepping adjustment.
	OPTIONS(1) = 2	enable direct adjustment.
	OPTIONS(2)	base case area interchange adjustment flag (area interchange adjustment option setting by default).
	OPTIONS(2) = 0	disable.
	OPTIONS(2) = 1	enable using tie line flows only in calculating area interchange.
	OPTIONS(2) = 2	enable using tie line flows and loads in calculating area interchange.
	OPTIONS(3)	base case phase shift adjustment flag (phase shift adjustment option setting by default).
	OPTIONS(3) = 0	disable.
	OPTIONS(3) = 1	enable.
	OPTIONS(4)	base case dc tap adjustment flag (dc tap adjustment option setting by default).
	OPTIONS(4) = 0	disable.
	OPTIONS(4) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(5)	base case switched shunt adjustment flag (switched shunt adjustment option setting by default).
OPTIONS(5) = 0	disable.
OPTIONS(5) = 1	enable.
OPTIONS(5) = 2	enable continuous mode, disable discrete mode.
OPTIONS(6)	base case induction motor treatment flag; applied when an induction motor fails to solve due to low terminal voltage (0 by default).
OPTIONS(6) = 0	stall.
OPTIONS(6) = 1	trip.
OPTIONS(7)	non-divergent solution flag (non-divergent solution option setting by default).
OPTIONS(7) = 0	disable.
OPTIONS(7) = 1	enable.
OPTIONS(8)	solution method flag (0 by default).
OPTIONS(8) = 0	FDNS.
OPTIONS(8) = 1	FNSL.
OPTIONS(8) = 2	optimized FDNS.
OPTIONS(9)	var limit code for the contingency case power flow solutions (0 by default).
OPTIONS(9) = 0	apply immediately.
OPTIONS(9) = 1	initially ignore, then apply automatically.
OPTIONS(10)	var limit code for the base case transfer increment solutions (0 by default).
OPTIONS(10) = 0	apply immediately.
OPTIONS(10) = 1	initially ignore, then apply automatically.
OPTIONS(11)	rating set (rating set program option setting by default).
OPTIONS(11) = 1	RATEA.
OPTIONS(11) = 2	RATEB.
OPTIONS(11) = 3	RATEC.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(12)	study (source) system transfer dispatch method (1 by default).
OPTIONS(12) = 1	buses and distribution factors from DFAX file for buses with positive MW machines.
OPTIONS(12) = 2	buses and distribution factors from DFAX file for buses with positive MW constant MVA load.
OPTIONS(12) = 3	buses and distribution factors from DFAX file for buses with either positive MW machines or positive MW constant MVA load.
OPTIONS(12) = 4	subsystem buses with positive MW constant MVA load in proportion to their MW load.
OPTIONS(12) = 5	subsystem buses with positive MW machines in proportion to their MW output.
OPTIONS(12) = 6	subsystem buses with positive MW machines in proportion to their MBASEs.
OPTIONS(12) = 7	subsystem buses with positive MW machines in proportion to their reserve (PMAXMC - PGENMC).
OPTIONS(12) = 8	subsystem buses with positive MW machines via ECDI with unit commitment disabled.
OPTIONS(12) = 9	subsystem buses with positive MW machines via ECDI with unit commitment enabled.
OPTIONS(13)	opposing (sink) system transfer dispatch method (1 by default).
OPTIONS(13) = 1	buses and distribution factors from DFAX file for buses with positive MW machines.
OPTIONS(13) = 2	buses and distribution factors from DFAX file for buses with positive MW constant MVA load.
OPTIONS(13) = 3	buses and distribution factors from DFAX file for buses with either positive MW machines or positive MW constant MVA load.
OPTIONS(13) = 4	subsystem buses with positive MW constant MVA load in proportion to their MW load.
OPTIONS(13) = 5	subsystem buses with positive MW machines in proportion to their MW output.
OPTIONS(13) = 6	subsystem buses with positive MW machines in proportion to their MBASEs.
OPTIONS(13) = 7	subsystem buses with positive MW machines in proportion to their reserve (PGENMC - PMINMC).
OPTIONS(13) = 8	subsystem buses with positive MW machines via ECDI with unit commitment disabled.
OPTIONS(13) = 9	subsystem buses with positive MW machines via ECDI with unit commitment enabled.
OPTIONS(14)	generation plant limits flag for transfer methods 1, 3, 5, 6 and 7 (0 by default).
OPTIONS(14) = 0	no limits.
OPTIONS(14) = 1	honor machine active power limits.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(15)	positive load flag for transfer methods 2, 3 and 4 (0 by default).
OPTIONS(15) = 0	no limits.
OPTIONS(15) = 1	enforce non-negative net active power constant MVA load.
OPTIONS(16)	flag to check for low voltage at any monitored bus (0 by default).
OPTIONS(16) = 0	disable check.
OPTIONS(16) = 1	enable check using the threshold specified in VALUES(5).
OPTIONS(16) = 2	enable check using the normal lower voltage limits for base and contingency case solutions.
OPTIONS(16) = 3	enable check using the normal lower voltage limits for base case solutions and the emergency lower voltage limits for contingency case solutions.
OPTIONS(16) = 4	enable check using the emergency lower voltage limits for base and contingency case solutions.
OPTIONS(17)	flag to check for excessive loading on any monitored branch (0 by default).
OPTIONS(17) = 0	disable check.
OPTIONS(17) = 1	enable check.
OPTIONS(18)	dispatch mode for power unbalances resulting from the application of contingencies (0 by default).
OPTIONS(18) = 0	disable.
OPTIONS(18) = 1	in-service subsystem machines using reserve.
OPTIONS(18) = 2	in-service subsystem machines using Pmax.
OPTIONS(18) = 3	in-service subsystem machines using inertia.
OPTIONS(18) = 4	in-service subsystem machines using governor droop.
OPTIONS(19)	contingency case tap adjustment flag (base case tap adjustment flag, OPTIONS(1), by default).
OPTIONS(19) = 0	disable.
OPTIONS(19) = 1	enable stepping adjustment.
OPTIONS(19) = 2	enable direct adjustment.
OPTIONS(20)	contingency case area interchange adjustment flag (base case area interchange adjustment flag, OPTIONS(2), by default).
OPTIONS(20) = 0	disable.
OPTIONS(20) = 1	enable using tie line flows only in calculating area interchange.
OPTIONS(20) = 2	enable using tie line flows and loads in calculating area interchange.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTIONS(21)	contingency case phase shift adjustment flag (base case phase shift adjustment flag, OPTIONS(3), by default).
	OPTIONS(21) = 0	disable.
	OPTIONS(21) = 1	enable.
	OPTIONS(22)	contingency case dc tap adjustment flag (base case dc tap adjustment flag, OPTIONS(4), by default).
	OPTIONS(22) = 0	disable.
	OPTIONS(22) = 1	enable.
	OPTIONS(23)	contingency case switched shunt adjustment flag (base case switched shunt adjustment flag, OPTIONS(5), by default).
	OPTIONS(23) = 0	disable.
	OPTIONS(23) = 1	enable.
	OPTIONS(23) = 2	enable continuous mode, disable discrete mode.
	OPTIONS(24)	contingency case induction motor treatment flag; applied when an induction motor fails to solve due to low terminal voltage (0 by default).
	OPTIONS(24) = 0	stall.
	OPTIONS(24) = 1	trip.
Real VALUES(8)	Is an array of eight elements specifying solution values (input). The values are as follows:	
	VALUES(1)	mismatch tolerance in MW and Mvar (Newton solution convergence tolerance, TOLN, by default).
	VALUES(2)	initial transfer increment in MW (100.0 by default).
	VALUES(3)	transfer increment tolerance in MW (10.0 by default).
	VALUES(4)	maximum incremental transfer in MW (1000.0 by default).
	VALUES(5)	low voltage threshold in pu in the low voltage check (0.8 by default).
	VALUES(6)	percent of rating in the excessive branch loading check (100.0 by default).
	VALUES(7)	minimum incremental transfer in MW ($\leq$ 0.0) (0.0 by default).
	VALUES(8)	power factor for load increases in dispatch methods 2, 3 and 4 (0.0 by default). Specify as $\leq$ 0.0 to retain the original power factor at each load that is changed.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character LABELS(3)*12	Is an array of three elements specifying subsystem labels (input). The values are as follows: LABELS(1) label of the study (source) system for which generation is to be increased and/or for which the load is to be decreased (no default allowed). LABELS(2) label of the opposing (sink) system for which generation is to be decreased and/or for which the load is to be increased (no default allowed). LABELS(3) label of the dispatch subsystem; used if OPTIONS(18) is 1 through 4 (blank by default).
Character DFXFILE*260	Is the name of Distribution Factor Data file (input; no default allowed).
Character THRFILE*260	Is the name of Load Throwover Data file; blank for none (input; blank by default).
Character ECDFILE*260	Is the name of Economic Dispatch Data file; blank for none (input; blank by default).
Character INLFILE*260	Is the name of Inertia and Governor Response Data file; blank for none (input; blank by default).
Character ACCFILE*260	Is the name of PV Results File (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 as for PV_ENGINE_6.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.235 PV_ENGINE_6

Use this API routine to run the sixth release of the P-V analysis calculation engine. It provides for a different set of automatic adjustments, and for different treatment of induction motors that fail to solve due to low terminal voltage, in contingency case solutions than is used in the base case transfer increment solutions; it allows a negative minimum incremental transfer to be specified; for transfer increment methods involving load, it allows a power factor to be specified; it provides for dispatching power unbalances resulting from the application of contingencies; and it allows for the preservation of milestone results in a ZIP Archive Output File.

Batch command syntax:

```
BAT_PV_ENGINE_6  OPTIONS(1)...OPTIONS(25)  VALUES(1)...VALUES(8)
LABELS(1)...LABELS(3)  DFXFILE  THRFILE  ECDFILE  INLFILE  ACCFILE
ZIPFILE
```

Python syntax:

```
ierr = pv_engine_6(options, values, labels, dfxfile, thrfile,
ecdfile, inlfile, accfile, zipfile)
```

Fortran syntax:

```
CALL PV_ENGINE_6(OPTIONS, VALUES, LABELS, DFXFILE, THRFILE, ECDFILE,
INLFILE, ACCFILE, ZIPFILE, IERR)
```

where:

Integer OPTIONS(25)	Is an array of twenty-five elements specifying solution options (input). The values are as follows:	
	OPTIONS(1)	base case tap adjustment flag (tap adjustment option setting by default).
	OPTIONS(1) = 0	disable.
	OPTIONS(1) = 1	enable stepping adjustment.
	OPTIONS(1) = 2	enable direct adjustment.
	OPTIONS(2)	base case area interchange adjustment flag (area interchange adjustment option setting by default).
	OPTIONS(2) = 0	disable.
	OPTIONS(2) = 1	enable using tie line flows only in calculating area interchange.
	OPTIONS(2) = 2	enable using tie line flows and loads in calculating area interchange.
	OPTIONS(3)	base case phase shift adjustment flag (phase shift adjustment option setting by default).
	OPTIONS(3) = 0	disable.
	OPTIONS(3) = 1	enable.
	OPTIONS(4)	base case dc tap adjustment flag (dc tap adjustment option setting by default).
	OPTIONS(4) = 0	disable.
	OPTIONS(4) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(5)	base case switched shunt adjustment flag (switched shunt adjustment option setting by default).
OPTIONS(5) = 0	disable.
OPTIONS(5) = 1	enable.
OPTIONS(5) = 2	enable continuous mode, disable discrete mode.
OPTIONS(6)	base case induction motor treatment flag; applied when an induction motor fails to solve due to low terminal voltage (0 by default).
OPTIONS(6) = 0	stall.
OPTIONS(6) = 1	trip.
OPTIONS(7)	non-divergent solution flag (non-divergent solution option setting by default).
OPTIONS(7) = 0	disable.
OPTIONS(7) = 1	enable.
OPTIONS(8)	solution method flag (0 by default).
OPTIONS(8) = 0	FDNS.
OPTIONS(8) = 1	FNSL.
OPTIONS(8) = 2	optimized FDNS.
OPTIONS(9)	var limit code for the contingency case power flow solutions (0 by default).
OPTIONS(9) = 0	apply immediately.
OPTIONS(9) = 1	initially ignore, then apply automatically.
OPTIONS(10)	var limit code for the base case transfer increment solutions (0 by default).
OPTIONS(10) = 0	apply immediately.
OPTIONS(10) = 1	initially ignore, then apply automatically.
OPTIONS(11)	rating set (rating set program option setting by default).
OPTIONS(11) = 1	RATEA.
OPTIONS(11) = 2	RATEB.
OPTIONS(11) = 3	RATEC.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(12)	study (source) system transfer dispatch method (1 by default).
OPTIONS(12) = 1	buses and distribution factors from DFAX file for buses with positive MW machines.
OPTIONS(12) = 2	buses and distribution factors from DFAX file for buses with positive MW constant MVA load.
OPTIONS(12) = 3	buses and distribution factors from DFAX file for buses with either positive MW machines or positive MW constant MVA load.
OPTIONS(12) = 4	subsystem buses with positive MW constant MVA load in proportion to their MW load.
OPTIONS(12) = 5	subsystem buses with positive MW machines in proportion to their MW output.
OPTIONS(12) = 6	subsystem buses with positive MW machines in proportion to their MBASEs.
OPTIONS(12) = 7	subsystem buses with positive MW machines in proportion to their reserve (PMAXMC - PGENMC).
OPTIONS(12) = 8	subsystem buses with positive MW machines via ECDI with unit commitment disabled.
OPTIONS(12) = 9	subsystem buses with positive MW machines via ECDI with unit commitment enabled.
OPTIONS(13)	opposing (sink) system transfer dispatch method (1 by default).
OPTIONS(13) = 1	buses and distribution factors from DFAX file for buses with positive MW machines.
OPTIONS(13) = 2	buses and distribution factors from DFAX file for buses with positive MW constant MVA load.
OPTIONS(13) = 3	buses and distribution factors from DFAX file for buses with either positive MW machines or positive MW constant MVA load.
OPTIONS(13) = 4	subsystem buses with positive MW constant MVA load in proportion to their MW load.
OPTIONS(13) = 5	subsystem buses with positive MW machines in proportion to their MW output.
OPTIONS(13) = 6	subsystem buses with positive MW machines in proportion to their MBASEs.
OPTIONS(13) = 7	subsystem buses with positive MW machines in proportion to their reserve (PGENMC - PMINMC).
OPTIONS(13) = 8	subsystem buses with positive MW machines via ECDI with unit commitment disabled.
OPTIONS(13) = 9	subsystem buses with positive MW machines via ECDI with unit commitment enabled.
OPTIONS(14)	generation plant limits flag for transfer methods 1, 3, 5, 6 and 7 (0 by default).
OPTIONS(14) = 0	no limits.
OPTIONS(14) = 1	honor machine active power limits.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(15)	positive load flag for transfer methods 2, 3 and 4 (0 by default).
OPTIONS(15) = 0	no limits.
OPTIONS(15) = 1	enforce non-negative net active power constant MVA load.
OPTIONS(16)	flag to check for low voltage at any monitored bus (0 by default).
OPTIONS(16) = 0	disable check.
OPTIONS(16) = 1	enable check using the threshold specified in VALUES(5).
OPTIONS(16) = 2	enable check using the normal lower voltage limits for base and contingency case solutions.
OPTIONS(16) = 3	enable check using the normal lower voltage limits for base case solutions and the emergency lower voltage limits for contingency case solutions.
OPTIONS(16) = 4	enable check using the emergency lower voltage limits for base and contingency case solutions.
OPTIONS(17)	flag to check for excessive loading on any monitored branch (0 by default).
OPTIONS(17) = 0	disable check.
OPTIONS(17) = 1	enable check.
OPTIONS(18)	dispatch mode for power unbalances resulting from the application of contingencies (0 by default).
OPTIONS(18) = 0	disable.
OPTIONS(18) = 1	in-service subsystem machines using reserve.
OPTIONS(18) = 2	in-service subsystem machines using Pmax.
OPTIONS(18) = 3	in-service subsystem machines using inertia.
OPTIONS(18) = 4	in-service subsystem machines using governor droop.
OPTIONS(19)	ZIP Archive flag (0 by default).
OPTIONS(19) = 0	No ZIP Archive file.
OPTIONS(19) = 1	Write ZIP Archive file ZIPFILE; preserve each system condition at its largest solved incremental transfer level.
OPTIONS(19) = 2	Write ZIP Archive file ZIPFILE; preserve each system condition at all of its solved incremental transfer levels.
OPTIONS(20)	contingency case tap adjustment flag (base case tap adjustment flag, OPTIONS(1), by default).
OPTIONS(20) = 0	disable.
OPTIONS(20) = 1	enable stepping adjustment.
OPTIONS(20) = 2	enable direct adjustment.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(21)	contingency case area interchange adjustment flag (base case area interchange adjustment flag, OPTIONS(2), by default).
OPTIONS(21) = 0	disable.
OPTIONS(21) = 1	enable using tie line flows only in calculating area interchange.
OPTIONS(21) = 2	enable using tie line flows and loads in calculating area interchange.
OPTIONS(22)	contingency case phase shift adjustment flag (base case phase shift adjustment flag, OPTIONS(3), by default).
OPTIONS(22) = 0	disable.
OPTIONS(22) = 1	enable.
OPTIONS(23)	contingency case dc tap adjustment flag (base case dc tap adjustment flag, OPTIONS(4), by default).
OPTIONS(23) = 0	disable.
OPTIONS(23) = 1	enable.
OPTIONS(24)	contingency case switched shunt adjustment flag (base case switched shunt adjustment flag, OPTIONS(5), by default).
OPTIONS(24) = 0	disable.
OPTIONS(24) = 1	enable.
OPTIONS(24) = 2	enable continuous mode, disable discrete mode.
OPTIONS(25)	contingency case induction motor treatment flag; applied when an induction motor fails to solve due to low terminal voltage (0 by default).
OPTIONS(25) = 0	stall.
OPTIONS(25) = 1	trip.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real VALUES(8)	Is an array of eight elements specifying solution values (input). The values are as follows:	
	VALUES(1)	mismatch tolerance in MW and Mvar (Newton solution convergence tolerance, TOLN, by default).
	VALUES(2)	initial transfer increment in MW (100.0 by default).
	VALUES(3)	transfer increment tolerance in MW (10.0 by default).
	VALUES(4)	maximum incremental transfer in MW (1000.0 by default).
	VALUES(5)	low voltage threshold in pu in the low voltage check (0.8 by default).
	VALUES(6)	percent of rating in the excessive branch loading check (100.0 by default).
	VALUES(7)	minimum incremental transfer in MW ($\leq$ 0.0) (0.0 by default).
	VALUES(8)	power factor for load increases in dispatch methods 2, 3 and 4 (0.0 by default). Specify as $\leq$ 0.0 to retain the original power factor at each load that is changed.
Character LABELS(3)*12	Is an array of three elements specifying subsystem labels (input). The values are as follows:	
	LABELS(1)	label of the study (source) system for which generation is to be increased and/or for which the load is to be decreased (no default allowed).
	LABELS(2)	label of the opposing (sink) system for which generation is to be decreased and/or for which the load is to be increased (no default allowed).
	LABELS(3)	label of the dispatch subsystem; used if OPTIONS(18) is 1 through 4 (blank by default).
Character DFXFILE*260	Is the name of Distribution Factor Data file (input; no default allowed).	
Character THRFILE*260	Is the name of Load Throwover Data file; blank for none (input; blank by default).	
Character ECDFILE*260	Is the name of Economic Dispatch Data file; blank for none (input; blank by default).	
Character INLFILE*260	Is the name of Inertia and Governor Response Data file; blank for none (input; blank by default).	
Character ACCFILE*260	Is the name of PV Results File (input; no default allowed).	
Character ZIPFILE*260	Is the name of ZIP Archive Output File (input; blank by default).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid VALUES value.
IERR = 2	invalid OPTIONS value.
IERR = 3	generators are converted.
IERR = 4	buses in island(s) without a swing bus; use activity TREE.
IERR = 5	largest mismatch exceeds mismatch tolerance.
IERR = 6	initial transfer increment is less than transfer increment tolerance.
IERR = 7	no participating dispatch buses in study system.
IERR = 8	no participating dispatch buses in opposing system.
IERR = 9	subsystem engine error.
IERR = 10	ECDI engine in subsystem aaa.
IERR = 11	base case ECDI solution did not converge.
IERR = 12	no Distribution Factor Data File specified.
IERR = 13	no P-V Solution Output File specified.
IERR = 14	no Economic Dispatch Data File specified.
IERR = 15	no ZIP Archive Output File specified.
IERR = 16	in-service induction machines are in the "stalled" or "tripped" state.
IERR = 17	bus type code and series element status inconsistencies.
IERR = 18	no dispatch subsystem is specified.
IERR = 19	more than 10 islands in the base case.
IERR = 20	invalid base/contingency case adjustment flag combination.
IERR = 21	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
IERR = 22	monitored elements exceed limit when adding multi-section line members.
IERR = 23	no subsystems defined.
IERR = 24	subsystem not found.
IERR = 25	specified subsystems overlap.
IERR = 26	no in-service buses in subsystem WORLD.
IERR = 31	dispatch subsystem not found.
IERR = 50	error opening Economic Dispatch Data File.
IERR = 51	error opening PV Results File.
IERR = 52	error opening Distribution Factor Data File.
IERR = 53	error opening Load Throwover Data File.
IERR = 54	error opening Inertia and Governor Response Data File.
IERR = 55	error opening ZIP Archive Output File.
IERR = 56	prerequisite requirements for API are not met.
IERR = 99	terminated by user interrupt or file error.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.236 QV_ENGINE

This API routine is obsolete. It has been replaced by the API routine QV_ENGINE_4, and is implemented by a call to the QV_ENGINE_4 API routine.

Use this API to run the QV analysis calculation engine.

Batch command syntax:

```
BAT_QV_ENGINE  OPTIONS(1)...OPTIONS(10)  VALUES(1)...VALUES(4)
DFXFILE  THRFILE  ACCFILE
```

Python syntax:

```
ierr = qv_engine(options, values, dfxfile, thrfile, accfile)
```

Fortran syntax:

```
CALL QV_ENGINE(OPTIONS, VALUES, DFXFILE, THRFILE, ACCFILE, IERR)
```

where:

Integer OPTIONS(10)	Is an array of ten elements specifying solution options (input). The values are as follows:	
	OPTIONS(1)	tap adjustment flag (tap adjustment option setting by default).
	OPTIONS(1) = 0	disable.
	OPTIONS(1) = 1	enable stepping adjustment.
	OPTIONS(1) = 2	enable direct adjustment.
	OPTIONS(2)	area interchange adjustment flag (area interchange adjustment option setting by default).
	OPTIONS(2) = 0	disable.
	OPTIONS(2) = 1	enable using tie line flows only in calculating area interchange.
	OPTIONS(2) = 2	enable using tie line flows and loads in calculating area interchange.
	OPTIONS(3)	phase shift adjustment flag (phase shift adjustment option setting by default).
	OPTIONS(3) = 0	disable.
	OPTIONS(3) = 1	enable.
	OPTIONS(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
	OPTIONS(4) = 0	disable.
	OPTIONS(4) = 1	enable.
	OPTIONS(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
	OPTIONS(5) = 0	disable.
	OPTIONS(5) = 1	enable.
	OPTIONS(5) = 2	enable continuous mode, disable discrete mode.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTIONS(6)	non-divergent solution flag (non-divergent solution option setting by default).
	OPTIONS(6) = 0	disable.
	OPTIONS(6) = 1	enable.
	OPTIONS(7)	solution method flag (0 by default).
	OPTIONS(7) = 0	FDNS.
	OPTIONS(7) = 1	FNSL.
	OPTIONS(7) = 2	optimized FDNS.
	OPTIONS(8)	var limit code for the VHI power flow solution (0 by default).
	OPTIONS(8) = 0	apply immediately.
	OPTIONS(8) = 1	initially ignore, then apply automatically.
	OPTIONS(9)	var limit code for the subsequent voltage decrement cases (0 by default).
	OPTIONS(9) = 0	apply immediately.
	OPTIONS(9) = 1	initially ignore, then apply automatically.
	OPTIONS(10)	study bus number (no default allowed).
Real VALUES(4)	Is an array of four elements specifying solution values (input). The values are as follows:	
	VALUES(1)	the mismatch tolerance in MW and Mvar (Newton solution convergence tolerance, TOLN, by default).
	VALUES(2)	the initial (maximum) pu voltage setpoint at the study bus (VHI) (1.1 by default).
	VALUES(3)	the minimum pu voltage setpoint at the study bus (VLO) (0.9 by default).
	VALUES(4)	the pu voltage setpoint decrement (positive) at the study bus (DLTAV) (0.01 by default).
Character DFXFILE*260	Is the name of Distribution Factor Data file (input; no default allowed).	
Character THRFILE*260	Is the name of Load Throwover Data file; blank for none (input; blank by default).	
Character ACCFILE*260	Is the name of QV Results File (input; no default allowed).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR > 0	as for QV_ENGINE_4.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.237 QV_ENGINE_2

This API routine is obsolete. It has been replaced by the API routine QV_ENGINE_4, and is implemented by a call to the QV_ENGINE_4 API routine.

Use this API routine to run the second release of the Q-V analysis calculation engine.

Batch command syntax:

```
BAT_QV_ENGINE_2  OPTIONS(1)...OPTIONS(11)  VALUES(1)...VALUES(4)
DFXFILE  THRFILE  ACCFILE
```

Python syntax:

```
ierr = qv_engine_2(options, values, dfxfile, thrfile, accfile)
```

Fortran syntax:

```
CALL QV_ENGINE_2(OPTIONS, VALUES, DFXFILE, THRFILE, ACCFILE, IERR)
```

where:

Integer OPTIONS(11) Is an array of eleven elements specifying solution options (input). The values are as follows:

OPTIONS(1)	tap adjustment flag (tap adjustment option setting by default).
OPTIONS(1) = 0	disable.
OPTIONS(1) = 1	enable stepping adjustment.
OPTIONS(1) = 2	enable direct adjustment.
OPTIONS(2)	area interchange adjustment flag (area interchange adjustment option setting by default).
OPTIONS(2) = 0	disable.
OPTIONS(2) = 1	enable using tie line flows only in calculating area interchange.
OPTIONS(2) = 2	enable using tie line flows and loads in calculating area interchange.
OPTIONS(3)	phase shift adjustment flag (phase shift adjustment option setting by default).
OPTIONS(3) = 0	disable.
OPTIONS(3) = 1	enable.
OPTIONS(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
OPTIONS(4) = 0	disable.
OPTIONS(4) = 1	enable.
OPTIONS(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
OPTIONS(5) = 0	disable.
OPTIONS(5) = 1	enable.
OPTIONS(5) = 2	enable continuous mode, disable discrete mode.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTIONS(6)	induction motor treatment flag; applied when an induction motor fails to solve due to low terminal voltage (0 by default).
	OPTIONS(6) = 0	stall.
	OPTIONS(6) = 1	trip.
	OPTIONS(7)	non-divergent solution flag (non-divergent solution option setting by default).
	OPTIONS(7) = 0	disable.
	OPTIONS(7) = 1	enable.
	OPTIONS(8)	solution method flag (0 by default).
	OPTIONS(8) = 0	FDNS.
	OPTIONS(8) = 1	FNSL.
	OPTIONS(8) = 2	optimized FDNS.
	OPTIONS(9)	var limit code for the VHI power flow solution (0 by default).
	OPTIONS(9) = 0	apply immediately.
	OPTIONS(9) = 1	initially ignore, then apply automatically.
	OPTIONS(10)	var limit code for the subsequent voltage decrement cases (0 by default).
	OPTIONS(10) = 0	apply immediately.
	OPTIONS(10) = 1	initially ignore, then apply automatically.
	OPTIONS(11)	study bus number (no default allowed).
Real VALUES(4)	Is an array of four elements specifying solution values (input). The values are as follows:	
	VALUES(1)	the mismatch tolerance in MW and Mvar (Newton solution convergence tolerance, TOLN, by default).
	VALUES(2)	the initial (maximum) pu voltage setpoint at the study bus (VHI) (1.1 by default).
	VALUES(3)	the minimum pu voltage setpoint at the study bus (VLO) (0.9 by default).
	VALUES(4)	the pu voltage setpoint decrement (positive) at the study bus (DLTAV) (0.01 by default).
Character DFXFILE*260	Is the name of Distribution Factor Data file (input; no default allowed).	
Character THRFILE*260	Is the name of Load Throwover Data file; blank for none (input; blank by default).	
Character ACCFILE*260	Is the name of QV Results File (input; no default allowed).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR > 0	as for QV_ENGINE_4.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.238 QV_ENGINE_3

This API routine is obsolete. It has been replaced by the API routine QV_ENGINE_4, and is implemented by a call to the QV_ENGINE_4 API routine.

Use this API routine to run the third release of the Q-V analysis calculation engine.

Batch command syntax:

```
BAT_QV_ENGINE_3  OPTIONS(1)...OPTIONS(12)  VALUES(1)...VALUES(4)
LABEL  DFXFILE  THRFILE  INLFILE  ACCFILE
```

Python syntax:

```
ierr = qv_engine_3(options, values, label, dfxfile, thrfile, inlfile,
accfile)
```

Fortran syntax:

```
CALL QV_ENGINE_3(OPTIONS, VALUES, LABEL, DFXFILE, THRFILE, INLFILE,
ACCFILE, IERR)
```

where:

Integer OPTIONS(12)	Is an array of twelve elements specifying solution options (input). The values are as follows:	
	OPTIONS(1)	tap adjustment flag (tap adjustment option setting by default).
	OPTIONS(1) = 0	disable.
	OPTIONS(1) = 1	enable stepping adjustment.
	OPTIONS(1) = 2	enable direct adjustment.
	OPTIONS(2)	area interchange adjustment flag (area interchange adjustment option setting by default).
	OPTIONS(2) = 0	disable.
	OPTIONS(2) = 1	enable using tie line flows only in calculating area interchange.
	OPTIONS(2) = 2	enable using tie line flows and loads in calculating area interchange.
	OPTIONS(3)	phase shift adjustment flag (phase shift adjustment option setting by default).
	OPTIONS(3) = 0	disable.
	OPTIONS(3) = 1	enable.
	OPTIONS(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
	OPTIONS(4) = 0	disable.
	OPTIONS(4) = 1	enable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
OPTIONS(5) = 0	disable.
OPTIONS(5) = 1	enable.
OPTIONS(5) = 2	enable continuous mode, disable discrete mode.
OPTIONS(6)	induction motor treatment flag; applied when an induction motor fails to solve due to low terminal voltage (0 by default).
OPTIONS(6) = 0	stall.
OPTIONS(6) = 1	trip.
OPTIONS(7)	non-divergent solution flag (non-divergent solution option setting by default).
OPTIONS(7) = 0	disable.
OPTIONS(7) = 1	enable.
OPTIONS(8)	solution method flag (0 by default).
OPTIONS(8) = 0	FDNS.
OPTIONS(8) = 1	FNSL.
OPTIONS(8) = 2	optimized FDNS.
OPTIONS(9)	var limit code for the VHI power flow solution (0 by default).
OPTIONS(9) = 0	apply immediately.
OPTIONS(9) = 1	initially ignore, then apply automatically.
OPTIONS(10)	var limit code for the subsequent voltage decrement cases (0 by default).
OPTIONS(10) = 0	apply immediately.
OPTIONS(10) = 1	initially ignore, then apply automatically.
OPTIONS(11)	study bus number (no default allowed).
OPTIONS(12)	dispatch mode for power unbalances resulting from the application of contingencies (0 by default).
OPTIONS(12) = 0	disable.
OPTIONS(12) = 1	in-service subsystem machines using reserve.
OPTIONS(12) = 2	in-service subsystem machines using Pmax.
OPTIONS(12) = 3	in-service subsystem machines using inertia.
OPTIONS(12) = 4	in-service subsystem machines using governor droop.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real VALUES(4)	Is an array of four elements specifying solution values (input). The values are as follows: VALUES(1) the mismatch tolerance in MW and Mvar (Newton solution convergence tolerance, TOLN, by default). VALUES(2) the initial (maximum) pu voltage setpoint at the study bus (VHI) (1.1 by default). VALUES(3) the minimum pu voltage setpoint at the study bus (VLO) (0.9 by default). VALUES(4) the pu voltage setpoint decrement (positive) at the study bus (DLTAV) (0.01 by default).
Character LABEL*12	Is the label of the dispatch subsystem; used if OPTIONS(12) is 1 through 4 (blank by default).
Character DFXFILE*260	Is the name of Distribution Factor Data file (input; no default allowed).
Character THRFILE*260	Is the name of Load Throwover Data file; blank for none (input; blank by default).
Character INLFILE*260	Is the name of Inertia and Governor Response Data file; blank for none (input; blank by default).
Character ACCFILE*260	Is the name of QV Results File (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 as for QV_ENGINE_4.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.239 QV_ENGINE_4

Use this API routine to run the fourth release of the Q-V analysis calculation engine.

Batch command syntax:

```
BAT_QV_ENGINE_4  OPTIONS(1)...OPTIONS(13)  VALUES(1)...VALUES(4)
LABEL  DFXFILE  THRFILE  INLFILE  ACCFILE  ZIPFILE
```

Python syntax:

```
ierr = qv_engine_4(options, values, label, dfxfile, thrfile, inlfile,
accfile, zipfile)
```

Fortran syntax:

```
CALL QV_ENGINE_4(OPTIONS, VALUES, LABEL, DFXFILE, THRFILE, INLFILE,
ACCFILE, ZIPFILE, IERR)
```

where:

Integer OPTIONS(13)	Is an array of thirteen elements specifying solution options (input). The values are as follows:	
	OPTIONS(1)	tap adjustment flag (tap adjustment option setting by default).
	OPTIONS(1) = 0	disable.
	OPTIONS(1) = 1	enable stepping adjustment.
	OPTIONS(1) = 2	enable direct adjustment.
	OPTIONS(2)	area interchange adjustment flag (area interchange adjustment option setting by default).
	OPTIONS(2) = 0	disable.
	OPTIONS(2) = 1	enable using tie line flows only in calculating area interchange.
	OPTIONS(2) = 2	enable using tie line flows and loads in calculating area interchange.
	OPTIONS(3)	phase shift adjustment flag (phase shift adjustment option setting by default).
	OPTIONS(3) = 0	disable.
	OPTIONS(3) = 1	enable.
	OPTIONS(4)	dc tap adjustment flag (dc tap adjustment option setting by default).
	OPTIONS(4) = 0	disable.
	OPTIONS(4) = 1	enable.
	OPTIONS(5)	switched shunt adjustment flag (switched shunt adjustment option setting by default).
	OPTIONS(5) = 0	disable.
	OPTIONS(5) = 1	enable.
	OPTIONS(5) = 2	enable continuous mode, disable discrete mode.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(6)	induction motor treatment flag; applied when an induction motor fails to solve due to low terminal voltage (0 by default).
OPTIONS(6) = 0	stall.
OPTIONS(6) = 1	trip.
OPTIONS(7)	non-divergent solution flag (non-divergent solution option setting by default).
OPTIONS(7) = 0	disable.
OPTIONS(7) = 1	enable.
OPTIONS(8)	solution method flag (0 by default).
OPTIONS(8) = 0	FDNS.
OPTIONS(8) = 1	FNSL.
OPTIONS(8) = 2	optimized FDNS.
OPTIONS(9)	var limit code for the VHI power flow solution (0 by default).
OPTIONS(9) = 0	apply immediately.
OPTIONS(9) = 1	initially ignore, then apply automatically.
OPTIONS(10)	var limit code for the subsequent voltage decrement cases (0 by default).
OPTIONS(10) = 0	apply immediately.
OPTIONS(10) = 1	initially ignore, then apply automatically.
OPTIONS(11)	study bus number (no default allowed).
OPTIONS(12)	dispatch mode for power unbalances resulting from the application of contingencies (0 by default).
OPTIONS(12) = 0	disable.
OPTIONS(12) = 1	in-service subsystem machines using reserve.
OPTIONS(12) = 2	in-service subsystem machines using Pmax.
OPTIONS(12) = 3	in-service subsystem machines using inertia.
OPTIONS(12) = 4	in-service subsystem machines using governor droop.
OPTIONS(13)	ZIP Archive flag (0 by default).
OPTIONS(13) = 0	No ZIP Archive file.
OPTIONS(13) = 1	Write ZIP Archive file ZIPFILE; preserve each system condition at its lowest solved study bus voltage level.
OPTIONS(13) = 2	Write ZIP Archive file ZIPFILE; preserve each system condition at all solved study bus voltage levels.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real VALUES(4)	Is an array of four elements specifying solution values (input). The values are as follows: VALUES(1) the mismatch tolerance in MW and Mvar (Newton solution convergence tolerance, TOLN, by default). VALUES(2) the initial (maximum) pu voltage setpoint at the study bus (VHI) (1.1 by default). VALUES(3) the minimum pu voltage setpoint at the study bus (VLO) (0.9 by default). VALUES(4) the pu voltage setpoint decrement (positive) at the study bus (DLTAV) (0.01 by default).
Character LABEL*12	Is the label of the dispatch subsystem; used if OPTIONS(12) is 1 through 4 (blank by default).
Character DFXFILE*260	Is the name of Distribution Factor Data file (input; no default allowed).
Character THRFILE*260	Is the name of Load Throwover Data file; blank for none (input; blank by default).
Character INLFILE*260	Is the name of Inertia and Governor Response Data file; blank for none (input; blank by default).
Character ACCFILE*260	Is the name of QV Results File (input; no default allowed).
Character ZIPFILE*260	Is the name of ZIP Archive Output File (input; blank by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid VALUES value.
IERR = 2	invalid OPTIONS value.
IERR = 3	generators are converted.
IERR = 4	buses in island(s) without a swing bus; use activity TREE.
IERR = 5	largest mismatch exceeds mismatch tolerance.
IERR = 6	VHI is not greater than VLO
IERR = 7	study bus not found.
IERR = 8	invalid study bus.
IERR = 9	error when adding a plant to the study bus.
IERR = 10	error when adding a machine to the study bus.
IERR = 11	base case VHI solution did not converge.
IERR = 12	no Distribution Factor Data File specified.
IERR = 13	no Q-V Solution Output File specified.
IERR = 14	no ZIP Archive Output File specified.
IERR = 15	in-service induction machines are in the "stalled" or "tripped" state.
IERR = 16	bus type code and series element status inconsistencies.
IERR = 17	no dispatch subsystem is specified.
IERR = 18	more than 10 islands in the base case.
IERR = 21	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
IERR = 22	monitored elements exceed limit when adding multi-section line members.
IERR = 23	no subsystems defined.
IERR = 24	subsystem not found.
IERR = 51	error opening QV Results File.
IERR = 52	error opening Distribution Factor Data File.
IERR = 53	error opening Load Throwover Data File.
IERR = 54	error opening Inertia and Governor Response Data File.
IERR = 55	error opening ZIP Archive Output File.
IERR = 56	prerequisite requirements for API are not met.
IERR = 99	terminated by user interrupt or file error.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.240 RANK

Use this API to estimate the severity of designated single branch outage contingencies and builds a Contingency Solution Output File with contingencies specified in decreasing order of their estimated severities (activity RANK).

Batch command syntax:

```
BAT_RANK  SID  ALL  IVAL(1)...IVAL(7) RVAL(1)...RVAL(4)  SSFILE
MNFILE  OFILE
```

Python syntax:

```
ierr = rank(sid, all, ival, rval, ssfile, mnfile, ofile)
```

Fortran syntax:

```
CALL RANKNW(SID, ALL, IVAL, RVAL, SSFILE, MNFILE, OFILE, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all buses.
	ALL = 0	process only buses in subsystem SID.
Integer IVAL(7)	Is an array of seven elements specifying ranking options (input).	
	IVAL(1)	allow ties from specified subsystem as contingencies (0 = no, 1 = yes; 0 by default).
	IVAL(2)	number of branch contingencies from overload ranking to include (>= 0) (0 by default).
	IVAL(3)	rating set (1=RATEA, 2=RATEB, 3=RATEC) (rating set program option setting by default).
	IVAL(4)	convert ratings to estimated MW ratings (0 = no, 1 = yes; 0 by default).
	IVAL(5)	number of voltage depression contingencies to include (>=0) (0 by default).
	IVAL(6)	code indicating the set of branches to include in the voltage depression performance index (0 by default).
	IVAL(6) = 0	include all branches
	IVAL(6) = 1	include only those with a nonzero rating.
	IVAL(7)	include contingencies that create swingless islands (0 = no, 1 = yes; 0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real RVAL(4)	Is an array of four elements specifying ranking parameters (input).	
	RVAL(1)	MW mismatch threshold (>= 0.0) (Newton solution convergence tolerance, TOLN, by default).
	RVAL(2)	small reactance threshold (THRSHZ <= RVAL(2) <= 0.005) (0.0005 by default).
	RVAL(3)	percent of rating (0.0 <= RVAL(3) <= 200.0) (100.0 by default).
	RVAL(4)	MW threshold for islanding contingencies (>= 0.0) (100.0 by default).
Character SSFILE*260	Is the name of the Subsystem Description File, or blank for no sub-system input (input; blank by default).	
Character MNFILE*260	Is the name of the Monitored Element Description File (input; no default allowed if IVAL(2) > 0 (i.e., if the branch outage overload ranker is enabled)).	
Character OFILE*260	Is the name of the Contingency Solution Output File, or blank for output to the progress window (input; blank by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	invalid IVAL value.
	IERR = 4	invalid RVAL value.
	IERR = 5	MNFILE is blank.
	IERR = 6	no contingency types enabled.
	IERR = 7	error opening SSFILE or MNFILE.
	IERR = 8	generators are converted.
	IERR = 9	buses in island(s) without a swing bus; use activity TREE.
	IERR = 10	largest mismatch exceeds mismatch tolerance.
	IERR = 11	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.241 RANK_BRN_AND_MAC

Use this API to estimate the severity of designated single element outage contingencies and builds a Contingency Description Data File with contingencies specified in decreasing order of their estimated severities (activity RANK).

Batch command syntax:

```
BAT_RANK_BRN_AND_MAC  SID  ALL  IVAL(1)...IVAL(8) RVAL(1)...RVAL(4)
SSFILE  MNFILE  OFILE
```

Python syntax:

```
ierr = rank_brn_and_mac(sid, all, ival, rval, ssfile, mnfile, ofile)
```

Fortran syntax:

```
CALL RANKAPI(SID, ALL, IVAL, RVAL, SSFILE, MNFILE, OFILE, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all buses.
	ALL = 0	process only buses in subsystem SID.
Integer IVAL(8)	Is an array of eight elements specifying ranking options (input).	
	IVAL(1)	allow ties from specified subsystem as contingencies (0 = no, 1 = yes; 0 by default).
	IVAL(2)	number of branch contingencies from overload ranking to include (>= 0) (0 by default).
	IVAL(3)	rating set (1=RATEA, 2=RATEB, 3=RATEC) (rating set program option setting by default).
	IVAL(4)	convert ratings to estimated MW ratings (0 = no, 1 = yes; 0 by default).
	IVAL(5)	number of voltage depression contingencies to include (>=0) (0 by default).
	IVAL(6)	code indicating the set of branches to include in the voltage depression performance index (0 by default).
	IVAL(6) = 0	include all branches
	IVAL(6) = 1	include only those with a nonzero rating.
	IVAL(7)	include contingencies that create swingless islands (0 = no, 1 = yes; 0 by default).
	IVAL(8)	number of machine contingencies from overload ranking to include (>=0) (0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real RVAL(4)	Is an array of four elements specifying ranking parameters (input).	
	RVAL(1)	MW mismatch threshold (>= 0.0) (Newton solution convergence tolerance, TOLN, by default).
	RVAL(2)	small reactance threshold (THRSHZ <= RVAL(2) <= 0.005) (0.0005 by default).
	RVAL(3)	percent of rating (0.0 <= RVAL(3) <= 200.0) (100.0 by default).
	RVAL(4)	MW threshold for islanding contingencies (>= 0.0) (100.0 by default).
Character SSFILE*260	Is the name of the Subsystem Description File, or blank for no subsystem input (input; blank by default).	
Character MNFILE*260	Is the name of the Monitored Element Description File (input; no default allowed if IVAL(2) > 0 (i.e., if the branch outage overload ranker is enabled) or if IVAL(8) > 0 (i.e., if the machine outage overload ranker is enabled)).	
Character OFILE*260	Is the name of the output Contingency Description Data File, or blank for output to the progress window (input; blank by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	invalid IVAL value.
	IERR = 4	invalid RVAL value.
	IERR = 5	MNFILE is blank.
	IERR = 6	no contingency types enabled.
	IERR = 7	error opening SSFILE or MNFILE.
	IERR = 8	generators are converted.
	IERR = 9	buses in island(s) without a swing bus; use activity TREE.
	IERR = 10	largest mismatch exceeds mismatch tolerance.
	IERR = 11	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.242 RATE

This API routine is obsolete. It has been replaced by the API routine RATE_2, and is implemented by a call to RATE_2.

Use this API to check branch loadings (activities RATE, OLTL, OLTR and RAT3).

Batch command syntax:

```
BAT_RATE  SID  ALL  LINFLG  RATING  FMAX
```

Python syntax:

```
ierr = rate(sid, all, linflg, rating, fmax)
```

Fortran syntax:

```
CALL RATENW(SID, ALL, LINFLG, RATING, FMAX, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all branches.
	ALL = 0	process only branches connected to buses in subsystem SID.
Integer LINFLG	Is the branch check to be applied (input; 1 by default).	
	LINFLG = 1	check all lines.
	LINFLG = 2	check non-transformer branches.
	LINFLG = 3	check transformers.
	LINFLG = 4	check all lines, all ratings.
Integer RATING	Is the rating to be used (input; default rating set program option setting by default).	
	RATING = 1	RATEA.
	RATING = 2	RATEB.
	RATING = 3	RATEC.
Real FMAX	Is the line loading limit, in percent; 0.0 =< FMAX =< 500.0 (input; 100.0 by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	invalid LINFLG value.
	IERR = 4	invalid RATING value.
	IERR = 5	invalid FMAX value.
	IERR = 6	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.243 RATE_2

Use this API to check branch loadings (activities RATE, OLTL, OLTR and RAT3).

Batch command syntax:

```
BAT_RATE_2  SID  ALL  LINFLG  TRNFLG  BRKFLG  RATING  FMAX
```

Python syntax:

```
ierr = rate_2(sid, all, linflg, trnflg, brkflg, rating, fmax)
```

Fortran syntax:

```
CALL RATE_2(SID, ALL, LINFLG, TRNFLG, BRKFLG, RATING, FMAX, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 process all branches. ALL = 0 process only branches connected to buses in subsystem SID.
Integer LINFLG	Include non-transformer branches that are not breakers or switches in the branch check to be applied (input; 1 by default). LINFLG = 0 no. LINFLG = 1 yes
Integer TRNFLG	Include transformer branches in the branch check to be applied (input; 1 by default). TRNFLG = 0 no. TRNFLG = 1 yes
Integer BRKFLG	Include branches that are breakers or switches in the branch check to be applied (input; 1 by default). BRKFLG = 0 no. BRKFLG = 1 yes
Integer RATING	Is the rating to be used (input; default rating set program option setting by default). RATING = 0 all rating sets. RATING = 1 RATEA. RATING = 2 RATEB. RATING = 3 RATEC.
Real FMAX	Is the line loading limit, in percent; 0.0 =< FMAX =< 500.0 (input; 100.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	invalid LINFLG value.
	IERR = 4	invalid TRNFLG value.
	IERR = 5	invalid BRKFLG value.
	IERR = 6	invalid RATING value.
	IERR = 7	invalid FMAX value.
	IERR = 8	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.244 RATING_SET

Use this API to set the default rating set option setting to either RATEA, RATEB or RATEC (activity OPTN).

Batch command syntax:

```
BAT_RATING_SET  IVAL
```

Python syntax:

```
ierr = rating_set(ival)
```

or:

```
ierr,ival = rating_set()
```

Fortran syntax:

```
CALL RATING_SET_API(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Is the value of the option setting (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 1 by default for input).	
	IVAL = 1	RATEA.
	IVAL = 2	RATEB.
	IVAL = 3	RATEC.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE = 0	set the option setting to the value specified in IVAL.
	IOCODE = 1	return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid IVAL value.
	IERR = 2	invalid IOCODE value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.245 RAWD

This API routine is obsolete. It has been replaced by the API routine RAWD_2, and is implemented by a call to RAWD_2.

Use this API to replicate the working case in the form of a Power Flow Raw Data File (activity RAWD).

Batch command syntax:

```
BAT_RAWD  SID  ALL  STATUS(1)...STATUS(6)  OUT  OFILE
```

Python syntax:

```
ierr = rawd(sid, all, status, out, ofile)
```

Fortran syntax:

```
CALL RAWDAPI(SID, ALL, STATUS, OUT, OFILE, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all buses when STATUS(4) = 0 area ties only when STATUS(4) = 1.
	ALL = 0	process only buses in subsystem SID.
Integer STATUS(6)	Is an array of six elements specifying output options (input). The values are as follows:	
	STATUS(1)	include records for Type 4 buses (0 = no, 1 = yes; 1 by default).
	STATUS(2)	include records for out-of-service branches (0 = no, 1 = yes; 1 by default).
	STATUS(3)	include records for equipment in the subsystem (0 = no, 1 = yes; 1 by default).
	STATUS(4)	include records for subsystem ties (0 = no, 1 = yes; 0 by default).
	STATUS(5)	use bus names as bus identifiers (0 = no, 1 = yes; 0 by default).
	STATUS(6)	Raw Data File type (0 by default).
	STATUS(6) = 0	for activity READ with IC=0.
	STATUS(6) = 1	for activity READ with IC=1.
	STATUS(6) = 2	for activity RDCH.
Integer OUT	Is the output destination selection (input; 1 by default).	
	OUT = 0	direct output to a data file specified by OFILE.
	OUT = 1	direct output to the progress window.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character OFILE*260	Is the name of the file to which data records are to be written (input; no default allowed when OUT is 0).
Integer IERR	Is the error code (output).
	IERR = 0 no error.
	IERR = 1 invalid SID value or subsystem SID is not defined.
	IERR = 2 invalid ALL value.
	IERR = 3 invalid STATUS value.
	IERR = 4 invalid OUT value.
	IERR = 5 error opening OFILE.
	IERR = 6 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.246 RAWD_2

Use this API to replicate the working case in the form of a Power Flow Raw Data File (activity RAWD).

Batch command syntax:

```
BAT_RAWD_2  SID  ALL  STATUS(1)...STATUS(7)  OUT  OFILE
```

Python syntax:

```
ierr = rawd_2(sid, all, status, out, ofile)
```

Fortran syntax:

```
CALL RAWDAPI_2(SID, ALL, STATUS, OUT, OFILE, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all buses when STATUS(4) = 0 only area tie branches when STATUS(4) = 1.
	ALL = 0	process only buses in subsystem SID.
Integer STATUS(7)	Is an array of seven elements specifying output options (input). The values are as follows:	
	STATUS(1)	include records for Type 4 buses (0 = no, 1 = yes; 1 by default).
	STATUS(2)	include records for out-of-service branches (0 = no, 1 = yes; 1 by default).
	STATUS(3)	include records for equipment in the subsystem (0 = no, 1 = yes; 1 by default).
	STATUS(4)	include records for subsystem tie branches (0 = no, 1 = yes; 0 by default).
	STATUS(5)	include load record option (honored when ALL=0) (0 for all loads at subsystem buses 1 for subsystem loads at all buses 2 for all loads at subsystem buses and subsystem loads at non-subsystem buses; 0 by default).
	STATUS(6)	use bus names as bus identifiers (0 = no, 1 = yes; 0 by default).
	STATUS(7)	Raw Data File type (0 by default).
	STATUS(7) = 0	for activity READ with IC=0.
	STATUS(7) = 1	for activity READ with IC=1.
	STATUS(7) = 2	for activity RDCH.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer OUT	Is the output destination selection (input; 1 by default). OUT = 0 direct output to a data file specified by OFILE. OUT = 1 direct output to the progress window.
Character OFILE*260	Is the name of the file to which data records are to be written (input; no default allowed when OUT is 0).
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 invalid STATUS value. IERR = 4 invalid OUT value. IERR = 5 error opening OFILE. IERR = 6 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.247 RDCH

Use this API to read power flow change data into the working case (activity RDCH).

Batch command syntax:

```
BAT_RDCH  NUMNAM  IFILE
```

Python syntax:

```
ierr = rdch(numnam, ifile)
```

Fortran syntax:

```
CALL RDCHAPI(NUMNAM, IFILE, IERR)
```

where:

Integer NUMNAM	Is the flag for bus number or name specification on input records (input; 0 by default). NUMNAM = 0 bus numbers. NUMNAM = 1 bus names.
Character IFILE*260	Is the filename of Power Flow Data File, Options (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid NUMNAM. IERR = 10 error opening IFILE. IERR = 11 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.248 RDCHRAWVERSION

Use this API to read a Power Flow Change Raw Data File into the working case. Allowed raw file formats are PSS®E-15 through current format.

Batch command syntax:

```
BAT_RDCHRAWVERSION NUMNAM VERNUM IFILE
```

Python syntax:

```
ierr = rdchrawversion(numnam, vernum, ifile)
```

Fortran syntax:

```
CALL RDCHRAWVERSION(NUMNAM, VERNUM, IFILE, IERR)
```

where:

Integer NUMNAM	Is the flag for bus number or name specification on input records (input; 0 by default). NUMNAM = 0 bus numbers. NUMNAM = 1 bus names.
Character VERNUM*12	Is the version number corresponding to the format of IFILE (input; no default allowed). VERNUM is in the format of a PSS®E release number. **Example:** If IFILE format is of PSS®E-29.5.1: VERNUM = ’29’ = ’29.5’ = ’29.5.1’
Character IFILE*260	Is the name of Power Flow Raw Data file (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid NUMNAM value. IERR = 2 invalid VERNUM value. IERR = 3 unable to convert file. IERR = 4 error opening temporary file. IERR = 5 invalid revision number. IERR = 10 error opening IFILE. IERR = 11 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.249 RDEQ

Use this API to build an electrical equivalent of radial and, optionally, two-point Type 1 buses **outside of** a specified subsystem of the working case (activity RDEQ).

Batch command syntax:

```
BAT_RDEQ  SID  ALL  STATUS(1)...STATUS(8)
```

Python syntax:

```
ierr = rdeq(sid, all, status)
```

Fortran syntax:

```
CALL RDEQNW(SID, ALL, STATUS, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all buses.
	ALL = 0	process only buses outside of subsystem SID.
Integer STATUS(8)	Is an array of eight elements (input). They are used to control equivalencing options. A value of 0 means the option is not enabled. A value of 1 means the option is enabled. They are as follows.	
	STATUS(1)	build an electrical equivalent of radial buses only (0 by default).
	STATUS(2)	suppress the equivalencing option for transformers (0 by default).
	STATUS(3)	suppress the equivalencing option for zero impedance lines (0 by default).
	STATUS(4)	suppress the equivalencing option for controlled buses (0 by default).
	STATUS(5)	suppress the equivalencing option for mutually coupled branches (0 by default).
	STATUS(6)	suppress the equivalencing option for area boundary buses (0 by default).
	STATUS(7)	suppress the equivalencing option for zone boundary buses (0 by default).
	STATUS(8)	apply transformer impedance correction to zero sequence network (1 by default).
Integer IERR	Is the error code (output).	
	IERR = 0	no error
	IERR = 1	invalid SID value or subsystem SID is not defined
	IERR = 2	invalid ALL value
	IERR = 3	invalid STATUS value

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IERR = 4	error writing the working case to a temporary file
IERR = 5	load table is full; restoring original working case
IERR = 6	fixed bus shunt table is full; restoring original working case
IERR = 7	error opening temporary file
IERR = 8	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.250 READ

Use this API to read a Power Flow Raw Data File and add all the data specified in it to the working case (activity READ).

Batch command syntax:

```
BAT_READ  NUMNAM  IFILE
```

Python syntax:

```
ierr = read(numnam, ifile)
```

Fortran syntax:

```
CALL READAPI(NUMNAM, IFILE, IERR)
```

where:

Integer NUMNAM	Is the flag for bus number or name specification on input records (input; 0 by default). NUMNAM = 0 bus numbers. NUMNAM = 1 bus names.
Character IFILE*260	Is the filename of the Power Flow Raw Data file (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid NUMNAM value. IERR = 2 invalid revision number. IERR = 3 unable to convert file. IERR = 4 error opening temporary file. IERR = 10 error opening IFILE. IERR = 11 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.251 READRAWVERSION

Use this API to read a Power Flow Raw Data File and add all the data specified in it to the working case. Allowed raw file formats are PSS®E-15 through current format.

Batch command syntax:

```
BAT_READRAWVERSION NUMNAM VERNUM IFILE
```

Python syntax:

```
ierr = readrawversion(numnam, vernum, ifile)
```

Fortran syntax:

```
CALL READRAWVERSION(NUMNAM, VERNUM, IFILE, IERR)
```

where:

Integer NUMNAM	Is the flag for bus number or name specification on input records (input; 0 by default). NUMNAM = 0 bus numbers. NUMNAM = 1 bus names.
Character VERNUM*12	Is the version number corresponding to the format of IFILE (input; no default allowed). VERNUM is in the format of a PSS®E release number. **Example:** If IFILE format is of PSS®E-29.5.1: VERNUM = ’29’ = ’29.5’ = ’29.5.1’
Character IFILE*260	Is the name of Power Flow Raw Data file (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid NUMNAM value. IERR = 2 invalid VERNUM value. IERR = 3 unable to convert file. IERR = 4 error opening temporary file. IERR = 5 invalid revision number. IERR = 10 error opening IFILE. IERR = 11 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.252 READSUB

Use this API to read a Power Flow Raw Data File and add subsystem data specified in it to the working case (activity READ,OPT).

Batch command syntax:

```
BAT_READSUB  NUMNAM  IFILE  STATUS(1)  STATUS(2)  STATUS(3)  USEKV
BASEKV(1)  BASEKV(2)  SIDAREA  SIDOWNER  SIDZONE
```

Python syntax:

```
ierr = readsub(numnam, ifile, status, usekv, basekv, sidarea,
sidowner, sidzone)
```

Fortran syntax:

```
CALL READSUB(NUMNAM, IFILE, STATUS, USEKV, BASEKV, SIDAREA, SIDOWNER,
SIDZONE, IERR)
```

where:

Integer NUMNAM	Is the flag for bus number or name specification on input records (input; 0 by default).	
	NUMNAM = 0	bus numbers.
	NUMNAM = 1	bus names.
Character IFILE*260	Is the filename of the Power Flow Raw Data file (input; no default allowed).	
Integer STATUS(3)	Is an array of three elements specifying options (input). The values are as follows:	
	STATUS(1)	append flag (0 by default).
	STATUS(1) = 0	initialize working case before reading data.
	STATUS(1) = 1	append data to that already in the working case.
	STATUS(2)	boundary bus identification flag (0 by default).
	STATUS(2) = 0	do not change type codes of boundary buses.
	STATUS(2) = 1	change type codes of boundary buses in preparation for the equivalencing calculation.
	STATUS(3)	data input option (1 by default).
	STATUS(3) = 1	add only data from within subsystem.
	STATUS(3) = 2	add only ties from subsystem.
	STATUS(3) = 3	add subsystem data plus ties.
Integer USEKV	Is the subsystem base voltage flag (input; 0 by default).	
	USEKV = 0	do not use base voltage as a subsystem selection criterion.
	USEKV = 1	use base voltage as a subsystem selection criterion.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real BASEKV(2)	Is an array of two elements specifying the base voltage range (input). BASEKV(1) is the minimum base voltage (0.0 by default). BASEKV(2) is the maximum base voltage (BASEKV(1) by default).
Integer SIDAREA	Is the area subsystem identifier (0 through 11), or –1 to bypass subsystem specification by area (input; -1 by default).
Integer SIDOWNER	Is the owner subsystem identifier (0 through 11), or –1 to bypass subsystem specification by owner (input; -1 by default).
Integer SIDZONE	Is the zone subsystem identifier (0 through 11), or –1 to bypass subsystem specification by zone (input; -1 by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid NUMNAM value. IERR = 2 invalid STATUS value. IERR = 3 invalid USEKV value. IERR = 4 invalid BASEKV value(s). IERR = 5 BASEKV(1) is greater than BASEKV(2). IERR = 6 invalid SIDAREA value or area subsystem SIDAREA is not defined. IERR = 7 invalid SIDOWNER value or owner subsystem SIDOWNER is not defined. IERR = 8 invalid SIDZONE value or zone subsystem SIDZONE is not defined. IERR = 9 no subsystem specified. IERR = 10 error opening IFILE. IERR = 11 invalid revision number. IERR = 12 prerequisite requirements for API are not met. IERR = 14 unable to convert file. IERR = 15 error opening temporary file.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.253 READSUBRAWVERSION

Use this API to read a Power Flow Raw Data File and add subsystem data specified in it to the working case. Allowed raw file formats are PSS®E-15 through current format.

Batch command syntax:

```
BAT_READSUBRAWVERSION  NUMNAM  VERNUM  IFILE  STATUS(1)  STATUS(2)
STATUS(3)  USEKV  BASEKV(1)  BASEKV(2)  SIDAREA  SIDOWNER  SIDZONE
```

Python syntax:

```
ierr = readsubrawversion(numnam, vernum, ifile, status, usekv,
basekv, sidarea, sidowner, sidzone)
```

Fortran syntax:

```
CALL READSUBRAWVERSION(NUMNAM, VERNUM, IFILE, STATUS, USEKV, BASEKV,
SIDAREA, SIDOWNER, SIDZONE, IERR)
```

where:

Integer NUMNAM	Is the flag for bus number or name specification on input records (input; 0 by default). NUMNAM = 0 bus numbers. NUMNAM = 1 bus names.
Character VERNUM*12	Is the version number corresponding to the format of IFILE (input; no default allowed). VERNUM is in the format of a PSS®E release number. **Example:** If IFILE format is of PSS®E-29.5.1: VERNUM = ’29’ = ’29.5’ = ’29.5.1’
Character IFILE*260	Is the filename of the Power Flow Raw Data file (input; no default allowed).
Integer STATUS(3)	Is an array of three elements specifying options (input). The values are as follows: STATUS(1) append flag (0 by default). STATUS(1) = 0 initialize working case before reading data. STATUS(1) = 1 append data to that already in the working case. STATUS(2) boundary bus identification flag (0 by default). STATUS(2) = 0 do not change type codes of boundary buses. STATUS(2) = 1 change type codes of boundary buses in preparation for the equivalencing calculation. STATUS(3) data input option (1 by default). STATUS(3) = 1 add only data from within subsystem. STATUS(3) = 2 add only ties from subsystem. STATUS(3) = 3 add subsystem data plus ties.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer USEKV	Is the subsystem base voltage flag (input; 0 by default).	
	USEKV = 0	do not use base voltage as a subsystem selection criterion.
	USEKV = 1	use base voltage as a subsystem selection criterion.
Real BASEKV(2)	Is an array of two elements specifying the base voltage range (input).	
	BASEKV(1)	is the minimum base voltage (0.0 by default).
	BASEKV(2)	is the maximum base voltage (BASEKV(1) by default).
Integer SIDAREA	Is the area subsystem identifier (0 through 11), or –1 to bypass subsystem specification by area (input; -1 by default).	
Integer SIDOWNER	Is the owner subsystem identifier (0 through 11), or –1 to bypass subsystem specification by owner (input; -1 by default).	
Integer SIDZONE	Is the zone subsystem identifier (0 through 11), or –1 to bypass subsystem specification by zone (input; -1 by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid NUMNAM value.
	IERR = 2	invalid STATUS value.
	IERR = 3	invalid USEKV value.
	IERR = 4	invalid BASEKV value(s).
	IERR = 5	BASEKV(1) is greater than BASEKV(2).
	IERR = 6	invalid SIDAREA value or area subsystem SIDAREA is not defined.
	IERR = 7	invalid SIDOWNER value or owner subsystem SIDOWNER is not defined.
	IERR = 8	invalid SIDZONE value or zone subsystem SIDZONE is not defined.
	IERR = 9	no subsystem specified.
	IERR = 10	error opening IFILE.
	IERR = 11	invalid revision number.
	IERR = 12	prerequisite requirements for API are not met.
	IERR = 13	invalid VERNUM value.
	IERR = 14	unable to convert file.
	IERR = 15	error opening temporary file.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.254 RECN

Use this API to electrically reconnect a bus (activity RECN).

Batch command syntax:

```
BAT_RECN  BUS
```

Python syntax:

```
ierr = recn(bus)
```

Fortran syntax:

```
CALL RECNAPI(BUS, IERR)
```

where:

Integer BUS	Is the bus number of bus to be reconnected (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 bus not found.
	IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.255 RECORDING_OPTIONS

Use this API to specify options that control the API recording process.

Batch command syntax:

```
BAT_RECORDING_OPTIONS RECOPTS(1), RECOPTS(2)
```

Python syntax:

```
ierr = recording_options(recopts)
```

Fortran syntax:

```
CALL RECORDING_OPTIONS(RECOPTS, IERR)
```

where:

Integer RECOPTS(2)	Recording options. For future use (input).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid RECOPTS value.
	IERR = 2	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.256 REGB

Use this API to tabulate those buses where voltages are controlled by generation, switched shunts, and/or other voltage controlling equipment (activity REGB).

Batch command syntax:

```
BAT_REGB  SID  ALL
```

Python syntax:

```
ierr = regb(sid, all)
```

Fortran syntax:

```
CALL REGBNW(SID, ALL, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.257 RELIND

This API routine is obsolete. It has been replaced by the API routine RELIND_2, and is implemented by a call to RELIND_2.

Use this API to run probabilistic reliability assessment for transmission systems. Probabilistic indices on system problems are calculated in entire system or a given subsystem, these problems include flow overloading, voltage violation and loss of load, etc.

Batch command syntax:

```
BAT_RELIND  OPTIONS(1)...OPTIONS(7)  VALUES(1)...VALUES(3)  LABEL
ACCFILE  DFXFILE  PRBFILE  STSFILE
```

Python syntax:

```
ierr = relind(options, values, label, accfile, dfxfile, prbfile,
stsfile)
```

Fortran syntax:

```
CALL RELINDAPI(OPTIONS, VALUES, LABEL, ACCFILE, DFXFILE, PRBFILE,
STSFILE, IERR)
```

where:

Integer OPTIONS(7) Is an array of seven elements specifying output options (input). The values are as follows:

OPTIONS(1)	report format (1 by default).
OPTIONS(1) = 1	system problem summary report.
OPTIONS(1) = 2	system loss of load report.
OPTIONS(1) = 3	bus loss of load report.
OPTIONS(1) = 4	branch flow overloading report.
OPTIONS(1) = 5	bus voltage violation report.
OPTIONS(1) = 6	contingency summary.
OPTIONS(1) = 7	system problem indices.
OPTIONS(1) = 8	system loss of load indices.
OPTIONS(1) = 9	bus loss of load indices.
OPTIONS(1) = 10	branch overload problem indices.
OPTIONS(1) = 11	bus voltage problem indices.
OPTIONS(1) = 12	contingency summary with outage statistics.
OPTIONS(2)	output mode (1 by default).
OPTIONS(2) = 1	post contingency mode.
OPTIONS(2) = 2	post tripping mode.
OPTIONS(2) = 3	post corrective action mode.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTIONS(3)	rating set (rating set program option setting by default).
	OPTIONS(3) = 1	rating A
	OPTIONS(3) = 2	rating B
	OPTIONS(3) = 3	rating C
	OPTIONS(4)	include ties from study subsystem (0 by default).
	OPTIONS(4) = 0	disable
	OPTIONS(4) = 1	enable
	OPTIONS(5)	include interfaces (0 by default)
	OPTIONS(5) = 0	disable
	OPTIONS(5) = 1	enable
	OPTIONS(6)	output Outage Statistics Data File (0 by default)
	OPTIONS(6) = 0	disable
	OPTIONS(6) = 1	enable
	OPTIONS(7)	normalize probability (0 by default)
	OPTIONS(7) = 0	disable
	OPTIONS(7) = 1	enable
Real VALUES(3)	Is an array of three elements specifying reliability assessment parameters:	
	VALUES(1)	percent of rating set, (>=0 and <=200.0, 100.0 by default)
	VALUES(2)	probability cutoff (>=0; 0.0001 by default).
	VALUES(3)	frequency cutoff (>=0; 0.0001 by default).
Character LABEL*12	Is the name of the study subsystem (blank by default, probabilistic reliability assessment is carried out in entire system).	
Character ACCFILE*260	Is the name of the Contingency Solution Output File (no default allowed).	
Character DFXFILE*260	Is the name of the Distribution Factor Data File (no default allowed).	
Character PRBFILE*260	Is the name of the Outage Statistics Data File (blank for none if OPTIONS(1) is between 1 and 6, inclusive; no default allowed if OPTIONS(1) is greater).	
Character STSFILE*260	Is the name of Output Outage Statistics Data File (blank for none if OPTIONS(6) is zero or OPTIONS(1) is from 1 to 6; otherwise no default allowed).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR

Is the error code (output).

IERR = 0	no error
IERR = 1	invalid OPTIONS value
IERR = 2	no dispatch subsystem specified
IERR = 3	no Contingency Solution Output File specified
IERR = 4	no Distribution Factor Data File specified
IERR = 5	no Outage Statistic Data File specified
IERR = 6	no Output Outage Statistic Data File specified
IERR = 7	generators are converted
IERR = 8	buses in island(s) without a swing bus--use activity TREE.
IERR = 9	subsystem not found in Distribution Factor Data File
IERR = 10	Contingency Solution Output File is not created by Multiple Level Contingency Analysis of PSS®E 31 or later
IERR = 11	case data is not the one used in contingency analysis
IERR = 12	Distribution Factor Data File is not the one used in contingency analysis
IERR = 13	error reading Contingency Solution Output file
IERR = 51	error opening Contingency Solution Output File
IERR = 52	error opening Distribution Factor Data File
IERR = 53	error opening Outage Statistic Data File
IERR = 54	error opening Output Outage Statistic Data File
IERR = 55	prerequisite requirements for function not met

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.258 RELIND_2

The API is the second release of probabilistic reliability assessment function. Use this API to run probabilistic reliability assessment for transmission systems. Probabilistic indices on system problems are calculated in entire system or a given subsystem, these problems include flow overloading, voltage violation and loss of load, etc.

Batch command syntax:

```
BAT_RELIND_2  OPTIONS(1)...OPTIONS(8)  VALUES(1)...VALUES(3)  LABEL
ACCFILE  DFXFILE  PRBFILE  STSFILE
```

Python syntax:

```
ierr = relind_2(options, values, label, accfile, dfxfile, prbfile,
stsfile)
```

Fortran syntax:

```
CALL RELINDAPI_2(OPTIONS, VALUES, LABEL, ACCFILE, DFXFILE, PRBFILE,
STSFILE, IERR)
```

where:

Integer OPTIONS(8)	Is an array of eight elements specifying output options (input). The values are as follows:	
	OPTIONS(1)	report format (1 by default).
	OPTIONS(1) = 1	system problem summary report.
	OPTIONS(1) = 2	system loss of load report.
	OPTIONS(1) = 3	bus loss of load report.
	OPTIONS(1) = 4	branch flow overloading report.
	OPTIONS(1) = 5	bus voltage violation report.
	OPTIONS(1) = 6	contingency summary.
	OPTIONS(1) = 7	system problem indices.
	OPTIONS(1) = 8	system loss of load indices.
	OPTIONS(1) = 9	bus loss of load indices.
	OPTIONS(1) = 10	branch overload problem indices.
	OPTIONS(1) = 11	bus voltage problem indices.
	OPTIONS(1) = 12	contingency summary with outage statistics.
	OPTIONS(2)	output mode (1 by default).
	OPTIONS(2) = 1	post contingency mode.
	OPTIONS(2) = 2	post tripping mode.
	OPTIONS(2) = 3	post corrective action mode.
	OPTIONS(3)	rating set (rating set program option setting by default).
	OPTIONS(3) = 1	rating A
	OPTIONS(3) = 2	rating B
	OPTIONS(3) = 3	rating C

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTIONS(4)	voltage limit.
	OPTIONS(4) = 1	normal
	OPTIONS(4) = 2	emergency
	OPTIONS(5)	include ties from study subsystem (0 by default).
	OPTIONS(5) = 0	disable
	OPTIONS(5) = 1	enable
	OPTIONS(6)	include interfaces (0 by default)
	OPTIONS(6) = 0	disable
	OPTIONS(6) = 1	enable
	OPTIONS(7)	output Outage Statistics Data File (0 by default)
	OPTIONS(7) = 0	disable
	OPTIONS(7) = 1	enable
	OPTIONS(8)	normalize probability (0 by default)
	OPTIONS(8) = 0	disable
	OPTIONS(8) = 1	enable
Real VALUES(3)	Is an array of three elements specifying reliability assessment parameters:	
	VALUES(1)	percent of rating set, (>=0 and <=200.0, 100.0 by default)
	VALUES(2)	probability cutoff (>=0; 0.0001 by default).
	VALUES(3)	frequency cutoff (>=0; 0.0001 by default).
Character LABEL*32	Is the name of the study subsystem (blank by default, probabilistic reliability assessment is carried out in entire system).	
Character ACCFILE*260	Is the name of the Contingency Solution Output File (no default allowed).	
Character DFXFILE*260	Is the name of the Distribution Factor Data File (no default allowed).	
Character PRBFILE*260	Is the name of the Outage Statistics Data File (blank for none if OPTIONS(1) is between 1 and 6, inclusive; no default allowed if OPTIONS(1) is greater).	
Character STSFILE*260	Is the name of Output Outage Statistics Data File (blank for none if OPTIONS(6) is zero or OPTIONS(1) is from 1 to 6; otherwise no default allowed).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error
IERR = 1	invalid OPTIONS value
IERR = 2	no dispatch subsystem specified
IERR = 3	no Contingency Solution Output File specified
IERR = 4	no Distribution Factor Data File specified
IERR = 5	no Outage Statistic Data File specified
IERR = 6	no Output Outage Statistic Data File specified
IERR = 7	generators are converted
IERR = 8	buses in island(s) without a swing bus--use activity TREE.
IERR = 9	subsystem not found in Distribution Factor Data File
IERR = 10	Contingency Solution Output File is not created by Multiple Level Contingency Analysis of PSS¬ÆE 31 or later
IERR = 11	case data is not the one used in contingency analysis
IERR = 12	Distribution Factor Data File is not the one used in contingency analysis
IERR = 13	error reading Contingency Solution Output file
IERR = 51	error opening Contingency Solution Output File
IERR = 52	error opening Distribution Factor Data File
IERR = 53	error opening Outage Statistic Data File
IERR = 54	error opening Output Outage Statistic Data File
IERR = 55	prerequisite requirements for function not met

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.259 REMM

Use this API to read transaction event data from a Transactions Raw Data File and add it to PSS®E working memory (activity REMM).

Batch command syntax:

```
BAT_REMM  OPT  IFILE
```

Python syntax:

```
ierr = remm(opt, ifile)
```

Fortran syntax:

```
CALL REMMAPI(OPT, IFILE, IERR)
```

where:

Integer OPT	Is the flag for new or change data input records (input; 0 by default). OPT = 0 — new data; replaces existing transactions data. OPT = 1 — change data; modifies existing transactions data.
Character IFILE*260	Is the filename of Transactions Raw Data file (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 — no error occurred. IERR = 1 — invalid OPT value. IERR = 2 — no transactions defined; change mode invalid. IERR = 3 — error opening IFILE. IERR = 4 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.260 REPORT_OUTPUT

Use this API to specify the report output device.

Batch command syntax:

```
BAT_REPORT_OUTPUT  ISLCT  FILARG  OPTIONS(1)  OPTIONS(2)
```

Python syntax:

```
ierr = report_output(islct, filarg, options)
```

Fortran syntax:

```
CALL OPEN_REPORT_DEVICE(ISLCT, FILARG, OPTIONS, IERR)
```

where:

Integer ISLCT	Is the virtual device selector (input).	
	ISLCT = 1	standard destination. In the PSS®E GUI standard output for alerts is the Report View.
	ISLCT = 2	direct output to a file.
	ISLCT = 3	direct output to a printer.
	ISLCT = 4	direct output to progress device.
	ISLCT = 5	direct output to report device.
	ISLCT = 6	no output.
Character FILARG*260	Is the printer name or filename; ignored if ISLCT is not 2 or 3 (input).	
Integer OPTIONS(2)	Is an array of two elements specifying open and printing options; ignored if ISLCT is not 2 or 3 (input). The values are as follows:	
	OPTIONS(1)	file and printer open options:
	OPTIONS(1) = 0	open with carriage control format and, for files, for overwrite of existing files
	OPTIONS(1) = 1	open with list format.
	OPTIONS(1) = 2	open file for append (file output only).
	OPTIONS(1) = 2	set printer orientation to portrait mode (printer output only).
	OPTIONS(1) = 4	open printer in immediate print mode (printer output only).
	The OPTIONS(1) values may be summed to select multiple options.	
	OPTIONS(2)	number of copies to print (printer output only).
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	prerequisite requirements for API are not met.
	IERR = 2	failed to open device.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.261 RESQ

Use this API to read sequence data from a Sequence Data File and add it to the working case (activity RESQ).

Batch command syntax:

```
BAT_RESQ  IFILE
```

Python syntax:

```
ierr = resq(ifile)
```

Fortran syntax:

```
CALL RESQAPI(IFILE, IERR)
```

where:

Character IFILE*260	Is the filename of Sequence Data file (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 error opening IFILE.
	IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.262 RESQVERSION

Use this API to read a network Sequence Data File and add all the data specified in it to the working case. Allowed seq file formats are PSS®E-15 through the current format.

Batch command syntax:

```
BAT_RESQVERSION  VERNUM  IFILE
```

Python syntax:

```
ierr = resqversion(vernum, ifile)
```

Fortran syntax:

```
CALL RESQVERSION(VERNUM, IFILE, IERR)
```

where:

Character VERNUM*12	Is the version number corresponding to the format of the IFILE (input; no default allowed). VERNUM is in the format of a PSS®E release number. **Example**: If IFILE format is of PSS®E-29.5.1: VERNUM = ’29’ = ’29.5’ = ’29.5.1’
Character IFILE*260	Is the name of the power flow Sequence Data File (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 2 invalid VERNUM value. IERR = 3 unable to convert file. IERR = 4 error opening temporary file. IERR = 5 invalid revision number. IERR = 10 error opening IFILE. IERR = 11 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.263 RETI

Use this API to read the long title from an input file into the working case (activity RETI).

Batch command syntax:

```
BAT_RETI  IFILE
```

Python syntax:

```
ierr = reti(ifile)
```

Fortran syntax:

```
CALL RETIAPI(IFILE, IERR)
```

where:

Character IFILE*260	Is the filename of long title data input file (input; no default allowed).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	error opening IFILE.
	IERR = 2	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.264 REV29_NAMES

Use this API to set the extended bus name input format to either the PSS®E-29 or PSS®E-30 format.

Batch command syntax:

```
BAT_REV29_NAMES  IVAL
```

Python syntax:

```
ierr = rev29_names(ival)
```

Fortran syntax:

```
CALL REV29_NAMES_API(IVAL, IERR)
```

where:

Integer IVAL	Is the extended bus name input option flag (input; 0 by default).	
	IVAL = 0	PSS®E-30 format (12 characters plus base voltage).
	IVAL = 1	PSS®E-29 or earlier format (8 characters plus base voltage).
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid IVAL value.
	IERR = 2	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.265 RNFI

Use this API to reproduce the results of a working case bus renumbering operation in auxiliary data input files (activity RNFI).

Batch command syntax:

```
BAT_RNFI  FTYPE  TRNFILE  INPFILE  OUTFILE
```

Python syntax:

```
ierr = rnfi(ftype, trnfile, inpfile, outfile)
```

Fortran syntax:

```
CALL RNFIAPI(FTYPE, TRNFILE, INPFILE, OUTFILE, IERR)
```

where:

Integer FTYPE	Is the type of data file to be processed (input; 1 by default). FTYPE = 1 Sequence Data File. FTYPE = 2 Drawing Coordinate Data File. FTYPE = 3 Dynamics Data File. FTYPE = 4 Machine Impedance Data File. FTYPE = 5 Economic Dispatch Data File. FTYPE = 6 Inertia and Governor Response Data File. FTYPE = 7 Breaker Duty Data File. FTYPE = 8 Fault Specification Data File. FTYPE = 9 Fault Control Data File. FTYPE = 10 Machine Capability Curve Data File. FTYPE = 11 Load Throwover Data File. FTYPE = 12 ANSI Fault Specification Data File. FTYPE = 13 Subsystem Description Data File. FTYPE = 14 Monitored Element Data File. FTYPE = 15 Contingency Description Data File. FTYPE = 16 Tripping Element Data File.
Character TRNFILE*260	Is the name of Bus Renumbering Translation File (input; no default allowed).
Character INPFILE*260	Is the name of input data file to be renumbered (input; no default allowed).
Character OUTFILE*260	Is the name of output data file (blank for output to the report window) (input; blank by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid FTYPE value.
	IERR = 2	error opening translation file.
	IERR = 3	error opening input data file.
	IERR = 4	error opening output data file.
	IERR = 5	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.266 RWCM

Use this API to replicate the working case in IEEE Common tape format (activity RWCM).

Batch command syntax:

```
BAT_RWCM  OPT  OUT  OFILE
```

Python syntax:

```
ierr = rwcm(opt, out, ofile)
```

Fortran syntax:

```
CALL RWCMAPI(OPT, OUT, OFILE, IERR)
```

where:

Integer OPT	Is the dc line/FACTS/GNE device option (input; 0 by default).	
	OPT = 0	ignore dc lines, FACTS devices, and GNE devices.
	OPT = 1	add powers to loads.
Integer OUT	Is the output selection option (input; 1 by default).	
	OUT = 0	direct output to the data file specified by OFILE.
	OUT = 1	direct output to the progress window.
Character OFILE*260	Is the filename of the output file (input; no default allowed when OUT is 0).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid OPT value.
	IERR = 2	invalid OUT value.
	IERR = 3	buses in case with 5 or 6 digit bus numbers.
	IERR = 4	n buses in case but IEEE common format can handle only 9999.
	IERR = 5	error opening OFILE.
	IERR = 6	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.267 RWMA

Use this API to replicate machine parametric data from the working case in the form of a Machine Impedance Data File (activity RWMA).

Batch command syntax:

```
BAT_RWMA  SID  ALL  STATUS(1)  STATUS(2)  STATUS(3)  OUT  OFILE
```

Python syntax:

```
ierr = rwma(sid, all, status, out, ofile)
```

Fortran syntax:

```
CALL RWMANW(SID, ALL, STATUS, OUT, OFILE, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all buses.
	ALL = 0	process only buses in subsystem SID.
Integer STATUS(3)	Is an array of three elements (input). They are used to control output options. They are as follows.	
	STATUS(1)	include records for out-of-service machines (0 by default).
	STATUS(1) = 0	exclude them.
	STATUS(1) = 1	include them.
	STATUS(2)	used when calculating P fractions (0 by default).
	STATUS(2) = 0	use PGEN.
	STATUS(2) = 1	use MBASE.
	STATUS(2) = 2	use PMAX.
	STATUS(3)	used when calculating Q fractions (0 by default).
	STATUS(3) = 0	use QGEN.
	STATUS(3) = 1	use MBASE.
	STATUS(3) = 2	use QMAX.
Integer OUT	Is the output option selection (input; 1 by default).	
	OUT = 0	direct output to a data file specified by OFILE.
	OUT = 1	direct output to the progress window.
Character OFILE*260	Is the output filename (input; no default allowed when OUT is 0).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	invalid STATUS value.
	IERR = 4	invalid OUT value.
	IERR = 5	error opening OFILE.
	IERR = 6	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.268 RWMM

Use this API to replicate transaction event data in the form of a Transactions Data File (activity RWMM).

Batch command syntax:

```
BAT_RWMM  OUT  OFILE
```

Python syntax:

```
ierr = rwmm(out, ofile)
```

Fortran syntax:

```
CALL RWMMAPI(OUT, OFILE, IERR)
```

where:

Integer OUT	Is the output selection option (input; 1 by default).	
	OUT = 0	direct output to the data file specified by OFILE.
	OUT = 1	direct output to the progress window.
Character OFILE*260	Is the name of the file to which data records are to be written (input; no default allowed when OUT is 0).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid OUT value.
	IERR = 2	error opening OFILE.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.269 RWSQ

This API is obsolete. It has been replaced by the API RWSQ_2 and is implemented by a call to the RWSQ_2 API routine.

Use this API to replicate the sequence data contained in the working case in the form of a Sequence Data File (activity RWSQ).

Batch command syntax:

```
BAT_RWSQ  SID  ALL  STATUS(1)...STATUS(4)  OUT  OFILE
```

Python syntax:

```
ierr = rwsq(sid, all, status, out, ofile)
```

Fortran syntax:

```
CALL RWSQNW(SID, ALL, STATUS, OUT, OFILE, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all buses when STATUS(4) is 0; area ties only when STATUS(4) is 1.
	ALL = 0	process only buses in subsystem SID.
Integer STATUS(4)	Is an array of four elements specifying output options (input). The values are as follows.	
	STATUS(1)	include records for Type 4 buses (0 = no, 1 = yes; 1 by default).
	STATUS(2)	include records for out-of-service branches (0 = no, 1 = yes; 1 by default).
	STATUS(3)	include records for equipment in the subsystem (0 = no, 1 = yes; 1 by default).
	STATUS(4)	include records for subsystem ties (0 = no, 1 = yes; 0 by default).
Integer OUT	Is the output option selection (input; 1 by default).	
	OUT = 0	direct output to a data file specified by OFILE.
	OUT = 1	direct output to the progress window.
Character OFILE*260	Is the name of the file to which data records are to be written (input; no default allowed when OUT is 0).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	invalid STATUS value.
	IERR = 4	invalid OUT value.
	IERR = 5	error opening OFILE.
	IERR = 6	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.270 RWSQ_2

Use this API to replicate the sequence data contained in the working case in the form of a Sequence Data File (activity RWSQ).

Batch command syntax:

```
BAT_RWSQ_2  SID  ALL  STATUS(1)...STATUS(5)  OUT  OFILE
```

Python syntax:

```
ierr = rwsq_2(sid, all, status, out, ofile)
```

Fortran syntax:

```
CALL RWSQNW_2(SID, ALL, STATUS, OUT, OFILE, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all buses when STATUS(4) is 0; area ties only when STATUS(4) is 1.
	ALL = 0	process only buses in subsystem SID.
Integer STATUS(5)	Is an array of five elements specifying output options (input). The values are as follows.	
	STATUS(1)	include records for Type 4 buses (0 = no, 1 = yes; 1 by default).
	STATUS(2)	include records for out-of-service branches (0 = no, 1 = yes; 1 by default).
	STATUS(3)	include records for equipment in the subsystem (0 = no, 1 = yes; 1 by default).
	STATUS(4)	include records for subsystem ties (0 = no, 1 = yes; 0 by default).
	STATUS(5)	Seq data file type (0 by default)
	STATUS(5)=0	For activity RESQ with IC=0
	STATUS(5)=1	For activity RESQ with IC=1
Integer OUT	Is the output option selection (input; 1 by default).	
	OUT = 0	direct output to a data file specified by OFILE.
	OUT = 1	direct output to the progress window.
Character OFILE*260	Is the name of the file to which data records are to be written (input; no default allowed when OUT is 0).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	invalid STATUS value.
	IERR = 4	invalid OUT value.
	IERR = 5	error opening OFILE.
	IERR = 6	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.271 SAVE

Use this API to save the PSS®E working case in a Saved Case File (activity SAVE).

Batch command syntax:

```
BAT_SAVE  SFILE
```

Python syntax:

```
ierr = save(sfile)
```

Fortran syntax:

```
CALL SAVENW(SFILE, IERR)
```

where:

Character SFILE*260	Is the name of the PSS®E Saved Case File (input; no default allowed). If SFILE is "*", SAVENW attempts to use the most recently accessed Saved Case File. If no Saved Case File was accessed in the current execution of PSS®E, SAVENW returns an error code.
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 SFILE is blank.
	IERR = 2 error writing to SFILE.
	IERR = 3 error opening SFILE.
	IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.272 SAVEUCTEFILE

Use this API to translate a PSSE RAW file to a UCTE data file.

Batch command syntax:

```
BAT_SAVEUCTEFILE IFILE RFILE UFILE
```

Python syntax:

```
saveuctefile(ifile, rfile, ufile)
```

where:

Character IFILE*260	Is the name of an existing UCTE translation options input file (input).
Character RFILE*260	Is the name of the existing PSSE RAW input file (input).
Character UFILE*260	Is the name of the target UCTE data output file (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.273 SCAL

Use this API to uniformly increase or decrease any or all specified bus quantities for a specified group of buses (activity SCAL). It **must** be referenced using one of the following procedures:

- One reference with APIOPT=1, followed by one reference with APIOPT=2.
- One reference with APIOPT=0. This automatically combines the processing of one APIOPT 1 reference, followed by one APIOPT 2 reference.

Batch command syntax:

```
BAT_SCAL  SID  ALL  APIOPT  STATUS(1)...STATUS(4)
SCALVAL(1)...SCALVAL(7)
```

Python syntax:

```
ierr,totals,moto = scal(sid, all, apiopt, status, scalval)
```

Fortran syntax:

```
CALL SCALNW(SID, ALL, APIOPT, TOTALS, MOTO, STATUS, SCALVAL, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 — process all buses. ALL = 0 — process only buses in subsystem SID.
Integer APIOPT	Is the mode of operation in the API (input; 0 by default). APIOPT = 0 — initialize for scaling, then run the scaling and post-processing housekeeping. APIOPT = 1 — initialize for scaling. APIOPT = 2 — run the scaling and post-processing housekeeping.
Real TOTALS(11)	Is an array of eleven elements **returned** when APIOPT = 0 or 1. They are as follows. TOTALS(1) — load Mvar total. TOTALS(2) — load MW total. TOTALS(3) — generation MW total. TOTALS(4) — shunt MW total. TOTALS(5) — reactor Mvar total. TOTALS(6) — capacitor Mvar total. TOTALS(7) — motor load MW total. TOTALS(8) — generator PMAX. TOTALS(9) — generator PMIN. TOTALS(10) — motor load PMIN. TOTALS(11) — motor load PMAX.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer MOTO

Is **returned** when APIOPT = 0 or 1.

MOTO = 0	no motors in the specified subsystem.
MOTO = 1	motors present in the specified subsystem.

Integer STATUS(4)

Is an array of four elements that are used to control scaling (input). The first three elements are checked and used when APIOPT = 0 or 2; the fourth element is checked and saved when APIOPT = 0 or 1. They are as follows.

STATUS(1)	method used to scale active power load, generation and motor data, and bus shunt data (0 by default).
STATUS(1) = 0	no scaling.
STATUS(1) = 1	specify new total powers.
STATUS(1) = 2	specify percent changes.
STATUS(1) = 3	specify incremental powers.
STATUS(2)	flag to enforce machine power limits (0 by default).
STATUS(2) = 0	ignore machine power limits.
STATUS(2) = 1	enforce machine power limits.
STATUS(3)	flag to specify the scaling rule to be enforced when changing the reactive power load (0 by default).
STATUS(3) = 0	no change.
STATUS(3) = 1	constant P/Q ratio.
STATUS(3) = 2	new total Q load.
STATUS(3) = 3	percent change.
STATUS(3) = 4	new power factor.
STATUS(3) = 5	incremental Q load.
STATUS(4)	is the bus type code flag for load scaling; normally defaulted except when used by OPF (0 by default).
STATUS(4) = 0	all buses in subsystem.
STATUS(4) = 1	only Type 1 buses in subsystem.
STATUS(4) = 2	only Type 2 and 3 buses in subsystem.

Real SCALVAL(7)

Is an array of seven elements used as scaling targets (input). Based on the value of STATUS(1), entries (1) through (6) are either total powers (present total power by default), percent changes (0.0 by default), or incremental changes (0.0 by default).

SCALVAL(1)	load MW total/percent/increment.
SCALVAL(2)	generation MW total/percent/increment.
SCALVAL(3)	shunt MW total/percent/increment.
SCALVAL(4)	reactor Mvar total/percent/increment.
SCALVAL(5)	capacitor Mvar total/percent/increment.
SCALVAL(6)	motor load MW total/percent/increment.
SCALVAL(7)	reactive load scaling parameter.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

If STATUS(3) = 2, SCALVAL(7) = new total Mvar load (present total Mvar load by default).

If STATUS(3) = 3, SCALVAL(7) = percent change (0.0 by default).

If STATUS(3) = 4, SCALVAL(7) = new power factor (1.0 by default).

If STATUS(3) = 5, SCALVAL(7) = incremental Mvar load change (0.0 by default).

Integer IERR

Is the error code (output).

IERR = 0	no error.
IERR = 1	invalid SID value or subsystem SID is not defined.
IERR = 2	invalid ALL value.
IERR = 3	invalid APIOPT value.
IERR = 4	invalid STATUS value.
IERR = 5	unexpected APIOPT value.
IERR = 6	invalid power factor in SCALVAL(7).
IERR = 7	generator or motor totals beyond limits.
IERR = 8	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.274 SCAL_2

Use this API routine to uniformly increase or decrease any or all specified bus quantities for a specified group of buses (activity SCAL). It **must** be referenced using one of the following procedures:

- One reference with APIOPT=1, followed by one reference with APIOPT=2.
- One reference with APIOPT=0. This automatically combines the processing of one APIOPT 1 reference, followed by one APIOPT 2 reference.

Batch command syntax:

```
BAT_SCAL_2  SID  ALL  APIOPT  STATUS(1)...STATUS(5)
SCALVAL(1)...SCALVAL(7)
```

Python syntax:

```
ierr,totals,moto = scal_2(sid, all, apiopt, status, scalval)
```

Fortran syntax:

```
CALL SCAL_2(SID, ALL, APIOPT, TOTALS, MOTO, STATUS, SCALVAL, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all buses.
	ALL = 0	process only buses in subsystem SID.
Integer APIOPT	Is the mode of operation in the API (input; 0 by default).	
	APIOPT = 0	initialize for scaling, then run the scaling and post-processing housekeeping.
	APIOPT = 1	initialize for scaling.
	APIOPT = 2	run the scaling and post-processing housekeeping.
Real TOTALS(11)	Is an array of eleven elements **returned** when APIOPT = 0 or 1. They are as follows.	
	TOTALS(1)	load Mvar total.
	TOTALS(2)	load MW total.
	TOTALS(3)	generation MW total.
	TOTALS(4)	shunt MW total.
	TOTALS(5)	reactor Mvar total.
	TOTALS(6)	capacitor Mvar total.
	TOTALS(7)	motor load MW total.
	TOTALS(8)	generator PMAX.
	TOTALS(9)	generator PMIN.
	TOTALS(10)	motor load PMIN.
	TOTALS(11)	motor load PMAX.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer MOTO

Is **returned** when APIOPT = 0 or 1. It indicates the presence of motors that are modeled as conventional generators with negative active power settings.

MOTO = 0	no motors in the specified subsystem.
MOTO = 1	motors present in the specified subsystem.

Integer STATUS(5)

Is an array of five elements that are used to control scaling (input). The first and fifth elements are checked and saved when APIOPT = 0 or 1; the second, third and fourth elements are checked and used when APIOPT = 0 or 2. They are as follows.

STATUS(1)	interruptible load scaling option (0 by default).
STATUS(1) = 0	scale both interruptible and non-interruptible scalable loads.
STATUS(1) = 1	scale only non-interruptible scalable loads.
STATUS(1) = 2	scale only interruptible scalable loads.
STATUS(2)	method used to scale active power load, generation and motor data, and bus shunt data (0 by default).
STATUS(2) = 0	no scaling.
STATUS(2) = 1	specify new total powers.
STATUS(2) = 2	specify percent changes.
STATUS(2) = 3	specify incremental powers.
STATUS(3)	flag to enforce machine power limits (0 by default).
STATUS(3) = 0	ignore machine power limits.
STATUS(3) = 1	enforce machine power limits.
STATUS(4)	flag to specify the scaling rule to be enforced when changing the reactive power load (0 by default).
STATUS(4) = 0	no change.
STATUS(4) = 1	constant P/Q ratio.
STATUS(4) = 2	new total Q load.
STATUS(4) = 3	percent change.
STATUS(4) = 4	new power factor.
STATUS(4) = 5	incremental Q load.
STATUS(5)	is the bus type code flag for load scaling; normally defaulted except when used by OPF (0 by default).
STATUS(5) = 0	all buses in subsystem.
STATUS(5) = 1	only Type 1 buses in subsystem.
STATUS(5) = 2	only Type 2 and 3 buses in subsystem.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real SCALVAL(7)

Is an array of seven elements used as scaling targets (input). Based on the value of STATUS(1), entries (1) through (6) are either total powers (present total power by default), percent changes (0.0 by default), or incremental changes (0.0 by default).

SCALVAL(1)	load MW total/percent/increment.
SCALVAL(2)	generation MW total/percent/increment.
SCALVAL(3)	shunt MW total/percent/increment.
SCALVAL(4)	reactor Mvar total/percent/increment.
SCALVAL(5)	capacitor Mvar total/percent/increment.
SCALVAL(6)	motor load MW total/percent/increment.
SCALVAL(7)	reactive load scaling parameter.

If STATUS(3) = 2, SCALVAL(7) = new total Mvar load (present total Mvar load by default).

If STATUS(3) = 3, SCALVAL(7) = percent change (0.0 by default).

If STATUS(3) = 4, SCALVAL(7) = new power factor (1.0 by default).

If STATUS(3) = 5, SCALVAL(7) = incremental Mvar load change (0.0 by default).

Integer IERR

Is the error code (output).

IERR = 0	no error.
IERR = 1	invalid SID value or subsystem SID is not defined.
IERR = 2	invalid ALL value.
IERR = 3	invalid APIOPT value.
IERR = 4	invalid STATUS value.
IERR = 5	unexpected APIOPT value.
IERR = 6	invalid power factor in SCALVAL(7).
IERR = 7	generator or motor totals beyond limits.
IERR = 8	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.275 SCEQ

Use this API to construct network equivalents of the positive and zero sequence networks and calculates source impedances at equivalent source nodes for all three sequences in preparation for the unbalanced fault analysis activities of PSS®E (activity SCEQ).

Batch command syntax:

```
BAT_SCEQ  SID  ALL  STATUS  THRESH  RFILE  SFILE
```

Python syntax:

```
ierr = sceq(sid, all, status, thresh, rfile, sfile)
```

Fortran syntax:

```
CALL SCEQNW(SID, ALL, STATUS, THRESH, RFILE, SFILE, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Integer STATUS	Is the transformer impedance correction option (input; 1 by default). STATUS = 0 do not apply to zero sequence. STATUS = 1 apply to zero sequence.
Real THRESH	Is the equivalent branch threshold tolerance (input; 10.0 by default).
Character RFILE*260	Is the output Power Flow Raw Data File; blank for none (input; blank by default).
Character SFILE*260	Is the output Sequence Data File; blank for none. (input; blank by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 invalid STATUS value. IERR = 4 case is not subject to classical assumptions. IERR = 5 generators are not converted. IERR = 6 error writing the working case to a temporary file. IERR = 7 no buses being retained. IERR = 8 no buses being processed for equivalence. IERR = 9 too many branches in the equivalent. IERR = 10 bus type codes other than 1, 2, 4, 5 and 6 present. IERR = 11 error opening RFILE or SFILE. IERR = 12 error opening temporary file. IERR = 13 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.276 SCGR

This API routine is obsolete. It has been replaced by the API routine ASCC_2. To display fault analysis results on the active Slider Diagram, use the API routine ASCC_2. From the GUI, use the **Fault>Automatic sequencing fault calculation (ASCC)…** menu entry.

Use this API routine to calculate three phase and line-to-ground fault solutions (activity SCGR). Results may be programmatically accessed via the internally accessible fetch routines SCGTXT, SCGBUS, SCGBRN, and SCG3WN.

Usage is:

- invoke SCGR API with APIOPT = 0;
- multiple calls to fetch routines;
- optionally, repeat of above steps for different fault and/or different pre-fault conditions;
- final invocation of SCGR API with APIOPT = 1.

Batch command syntax:

```
BAT_SCGR  SID  ALL  APIOPT  STATUS(1)...STATUS(8)
```

Python syntax:

```
ierr = scgr(sid, all, apiopt, status)
```

Fortran syntax:

```
CALL SCGRAPI(SID, ALL, APIOPT, STATUS, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all buses.
	ALL = 0	process only buses in subsystem SID.
Integer APIOPT	Is the mode of operation of SCGRAPI (input; 0 by default).	
	APIOPT = 0	apply fault calculations and fill arrays used by fetch routines.
	APIOPT = 1	deallocate arrays used by fetch routines.
Integer STATUS(8)	Is an array of eight elements specifying fault and output options (input). The values are as follows:	
	STATUS(1)	fault type (0 by default):
	STATUS(1) = 0	three-phase.
	STATUS(1) = 1	line-to-ground.
	STATUS(2)	L-G fault output (ignored if STATUS(1) is 0) (0 by default).
	STATUS(2) = 0	A phase.
	STATUS(2) = 1	$3 * I_0$.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

STATUS(3)	fault location (0 by default).
STATUS(3) = 0	all buses in subsystem.
STATUS(3) = >0	bus number of the subsystem bus to be faulted.
STATUS(4)	flat conditions option (0 by default).
STATUS(4) = 0	working case conditions.
STATUS(4) = 1	flat conditions.
STATUS(5)	double-precision fault calculation option (ignored, but must be specified as 0 or 1; 0 by default; double precision always used).
STATUS(6)	dc line and FACTS device option (ignored if STATUS(4) is 1) (0 by default).
STATUS(6) = 0	block.
STATUS(6) = 1	represent as load.
STATUS(7)	transformer impedance correction option (ignored if STATUS(1) is 0 or STATUS(4) is 1 (1 by default).
STATUS(7) = 0	do not apply to zero sequence.
STATUS(7) = 1	apply to zero sequence.
STATUS(8)	branch results option (0 by default).
STATUS(8) = 0	currents.
STATUS(8) = 1	apparent impedances.

Integer IERR

Is the error code (output).

IERR = 1	SCGR is obsolete; use "Fault>Automatic sequencing fault calculation (ASCC)..." (API routine ASCC_2).
IERR = 8	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.277 SCMU

Use this API to calculate simultaneous unbalances (activity SCMU). It **must** be called once with APIOPT set to 1, once for each of the simultaneous faults to be applied with APIOPT set to 2, and once with APIOPT set to 3.

Batch command syntax:

```
BAT_SCMU  APIOPT  OPTIONS(1)...OPTIONS(7)  RVAL(1)...RVAL(5)  CKTARG
```

Python syntax:

```
ierr = scmu(apiopt, options, rval, cktarg)
```

Fortran syntax:

```
CALL SCMUAPI(APIOPT, OPTIONS, RVAL, CKTARG, IERR)
```

where:

Integer APIOPT — Is the mode of operation (input; no default allowed).

APIOPT = 1	initialize; automatically runs activity SEQD if the sequence network setup option setting is enabled.
APIOPT = 2	specify an unbalance to be applied; must be called once for each simultaneous unbalance to be imposed.
APIOPT = 3	solve in the presence of the specified unbalances and terminate.
APIOPT = 4	remove specified unbalances (cancels any unbalances specified by any preceding APIOPT 2 references, and leaves memory as it is immediately following an APIOPT 1 reference).

Integer OPTIONS(7) — Is an array of seven elements specifying options and input values to be used in the current calculation phase; ignored when APIOPT is 3 or 4 (input).

When APIOPT is 1, the first three elements of OPTIONS define options to be used in the construction of the sequence admittance matrices. These values are used only when the sequence network setup option is enabled, and activity SEQD is to be automatically executed.

OPTIONS(1)	dc line and FACTS device option (0 by default).
OPTIONS(1) = 0	block and ignore.
OPTIONS(1) = 1	convert to constant admittance load.
OPTIONS(2)	transformer impedance correction option (1 by default).
OPTIONS(2) = 0	do not apply to zero sequence.
OPTIONS(2) = 1	apply to zero sequence.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(3)	fault analysis generator reactance option; ignored if generators are already "converted" (0 by default).
OPTIONS(3) = 0	use subtransient reactance.
OPTIONS(3) = 1	use transient reactance.
OPTIONS(3) = 2	use synchronous reactance.

When APIOPT is 2, the first element of OPTIONS defines the type of unbalance to be applied.

OPTIONS(1)	unbalance type (no default allowed).
OPTIONS(1) = 1	first line-to-ground fault.
OPTIONS(1) = 2	second line-to-ground fault.
OPTIONS(1) = 3	first line-to-line or line-to-line-to-ground fault.
OPTIONS(1) = 4	second line-to-line or line-to-line-to-ground fault.
OPTIONS(1) = 5	one phase closed.
OPTIONS(1) = 6	two phases closed.
OPTIONS(1) = 7	three-phase fault (uses fault types 2 and 4).
OPTIONS(1) = 8	branch unbalance with one end opened.
OPTIONS(1) = 9	branch unbalance with both ends closed.

The remaining elements of OPTIONS are dependent on the value specified for OPTIONS(1) and are described at the end of this section.

Real RVAL(5) — Is an array of five elements specifying input values used to define the current fault; ignored except when APIOPT is 2. The elements of RVAL are dependent on the value specified for OPTIONS(1) and are described at the end of this section (input).

Character CKTARG*2 — Is the circuit identifier of branch to be faulted; ignored except when APIOPT is 2 and OPTIONS(1) is 8 or 9 (input; ’1’ by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	unexpected APIOPT value.
	IERR = 2	invalid APIOPT value.
	IERR = 3	invalid unbalance type.
	IERR = 4	bus not found.
	IERR = 5	bus is out-of-service.
	IERR = 6	invalid phase value.
	IERR = 7	branch not found.
	IERR = 8	three-winding transformer not found.
	IERR = 9	invalid open end bus.
	IERR = 10	invalid distance factor.
	IERR = 11	terminated due to invalid fault specification or application.
	IERR = 12	SEQD results are not compatible with the working case.
	IERR = 13	fault analysis is invalid due to the presence of in-service induction machines that are in the "stalled" or "tripped" state
	IERR = 14	bus type code and series element status inconsistencies.
	IERR = 15	invalid OPTIONS value (from SEQD).
	IERR = 16	prerequisite requirements for API are not met.

When APIOPT is 2 and OPTIONS(1) is 1 or 2, the following additional elements of OPTIONS and RVAL are used.

Integer OPTIONS(7)	OPTIONS(2)	bus to be faulted (no default allowed).
	OPTIONS(3)	faulted phase (1 by default).
	OPTIONS(3) = 1	phase A.
	OPTIONS(3) = 2	phase B.
	OPTIONS(3) = 3	phase C.
Real RVAL(5)	RVAL(1)	resistance of the complex L-G fault impedance in pu (0.0 by default).
	RVAL(2)	reactance of the complex L-G fault impedance in pu (0.0 by default).

When APIOPT is 2 and OPTIONS(1) is 3 or 4, the following additional elements of OPTIONS and RVAL are used.

Integer OPTIONS(7)	OPTIONS(2)	bus to be faulted (no default allowed).
	OPTIONS(3)	excluded phase (1 by default).
	OPTIONS(3) = 1	phase A.
	OPTIONS(3) = 2	phase B.
	OPTIONS(3) = 3	phase C.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real RVAL(5)	RVAL(1)	resistance of the complex L-L fault impedance in pu (0.0 by default).
	RVAL(2)	reactance of the complex L-L fault impedance in pu (0.0 by default).
	RVAL(3)	resistance of the complex L-G fault impedance in pu (0.0 by default).
	RVAL(4)	reactance of the complex L-G fault impedance in pu (0.0 by default).

When APIOPT is 2 and OPTIONS(1) is 5, the following additional elements of OPTIONS and RVAL are used.

Integer OPTIONS(7)	OPTIONS(2)	one of the buses to be connected by the phase closed unbalance (no default allowed).
	OPTIONS(3)	the other bus to be connected by the phase closed unbalance (no default allowed).
	OPTIONS(4)	phase to be closed (1 by default).
	OPTIONS(4) = 1	phase A.
	OPTIONS(4) = 2	phase B.
	OPTIONS(4) = 3	phase C.
Real RVAL(5)	RVAL(1)	resistance of the complex phase impedance in pu (0.0 by default).
	RVAL(2)	reactance of the complex phase impedance in pu (0.0 by default).

When APIOPT is 2 and OPTIONS(1) is 6, the following additional elements of OPTIONS and RVAL are used.

Integer OPTIONS(7)	OPTIONS(2)	one of the buses to be connected by the phase closed unbalance (no default allowed).
	OPTIONS(3)	the other bus to be connected by the phase closed unbalance (no default allowed).
	OPTIONS(4)	excluded phase (1 by default).
	OPTIONS(4) = 1	phase A.
	OPTIONS(4) = 2	phase B.
	OPTIONS(4) = 3	phase C.
Real RVAL(5)	RVAL(1)	resistance of the complex phase impedance in pu (0.0 by default).
	RVAL(2)	reactance of the complex phase impedance in pu (0.0 by default).

When APIOPT is 2 and OPTIONS(1) is 7, the following additional elements of OPTIONS and RVAL are used.

Integer OPTIONS(7)	OPTIONS(2)	bus to be faulted (no default allowed).
Real RVAL(5)	None.	

When APIOPT is 2 and OPTIONS(1) is 8, the following additional elements of OPTIONS and RVAL are used.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer OPTIONS(7)	OPTIONS(2)	one of the buses connected by the branch to be faulted (no default allowed).
	OPTIONS(3)	another bus connected by the branch to be faulted (no default allowed).
	OPTIONS(4)	third bus connected by the three-winding transformer to be faulted, or zero for a two-terminal branch (0 by default).
	OPTIONS(5)	type of in-line fault to be applied (1 by default).
	OPTIONS(5) = 1	line-to-ground.
	OPTIONS(5) = 2	line-to-line or line-to-line-to-ground fault.
	OPTIONS(5) = 3	three-phase.
	OPTIONS(6)	faulted (if OPTIONS(5) is 1) or excluded (if OPTIONS(5) is 2) phase (1 by default).
	OPTIONS(6) = 1	phase A.
	OPTIONS(6) = 2	phase B.
	OPTIONS(6) = 3	phase C.
	OPTIONS(7)	opened end bus (must be identical to one of OPTIONS(2), OPTIONS(3), or, if it is non-zero, OPTIONS(4)) (OPTIONS(2) by default).
Real RVAL(5)	RVAL(1)	resistance of the complex L-G (if OPTIONS(5) is 1) or L-L (if OPTIONS(5) is 2) fault impedance in pu (0.0 by default).
	RVAL(2)	reactance of the complex L-G (if OPTIONS(5) is 1) or L-L (if OPTIONS(5) is 2) fault impedance in pu (0.0 by default).
	RVAL(3)	resistance of the complex L-G fault impedance in pu (if OPTIONS(5) is 2) (0.0 by default).
	RVAL(4)	reactance of the complex L-G fault impedance in pu (if OPTIONS(5) is 2) (0.0 by default).
	RVAL(5)	fault location from closed end bus in pu of total line length; must be 1.0 if the branch is a transformer (1.0 by default).

When APIOPT is 2 and OPTIONS(1) is 9, the following additional elements of OPTIONS and RVAL are used.

Integer OPTIONS(7)	OPTIONS(2)	one of the buses connected by the branch to be faulted (no default allowed).
	OPTIONS(3)	the other bus connected by the branch to be faulted (no default allowed).
	OPTIONS(4)	type of in-line fault to be applied (1 by default).
	OPTIONS(4) = 1	line-to-ground.
	OPTIONS(4) = 2	line-to-line or line-to-line-to-ground fault.
	OPTIONS(4) = 3	three-phase.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTIONS(5)	faulted (if OPTIONS(4) is 1) or excluded (if OPTIONS(4) is 2) phase (1 by default).
	OPTIONS(5) = 1	phase A.
	OPTIONS(5) = 2	phase B.
	OPTIONS(5) = 3	phase C.
Real RVAL(5)	RVAL(1)	resistance of the complex L-G (if OPTIONS(4) is 1) or L-L (if OPTIONS(4) is 2) fault impedance in pu (0.0 by default).
	RVAL(2)	reactance of the complex L-G (if OPTIONS(4) is 1) or L-L (if OPTIONS(4) is 2) fault impedance in pu (0.0 by default).
	RVAL(3)	resistance of the complex L-G fault impedance in pu (if OPTIONS(4) is 2) (0.0 by default).
	RVAL(4)	reactance of the complex L-G fault impedance in pu (if OPTIONS(4) is 2) (0.0 by default).
	RVAL(5)	fault location from bus OPTIONS(2) in pu of total line length (0.5 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.278 SCOP

Use this API to tabulate unbalanced fault solution output (activity SCOP). It **must** be preceded by the execution of activity SCMU, either interactively, from a Response File, or via Python. It **must** then be referenced using one of the following procedures:

- One reference with APIOPT=1, followed by any number of references with APIOPT=2, followed by one reference with APIOPT=3.
- One reference with APIOPT=0. This automatically combines the processing of one APIOPT 1 reference, followed by one APIOPT 2 reference, followed by one APIOPT 3 reference.

Batch command syntax:

```
BAT_SCOP  SID  ALL  UNITS  APIOPT
```

Python syntax:

```
ierr = scop(sid, all, units, apiopt)
```

Fortran syntax:

```
CALL SCOPAPI(SID, ALL, UNITS, APIOPT, IERR)
```

where:

Integer SID	Is a valid subsystem identifier; used only when APIOPT is 0 or 2 and ALL is 0. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag; used only when APIOPT is 0 or 2 (input; 1 by default).	
	ALL = 1	process all buses.
	ALL = 0	process only buses in subsystem SID.
Integer UNITS	Is the branch quantity output flag; used only when APIOPT is 0 or 2 (input; 1 by default).	
	UNITS = 1	currents.
	UNITS = 2	apparent impedances.
	UNITS = 3	apparent admittances.
Integer APIOPT	Is the mode of operation (input; 0 by default).	
	APIOPT = 0	initialize for short circuit output, produce the output, and run post-processing clean up.
	APIOPT = 1	initialize for short circuit output.
	APIOPT = 2	produce output.
	APIOPT = 3	post-processing clean up.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid SID value or subsystem SID is not defined.
IERR = 2	invalid ALL value.
IERR = 3	invalid UNITS value.
IERR = 4	invalid APIOPT value.
IERR = 5	unexpected APIOPT value.
IERR = 6	SCMU results error; one of the following occurred: • No SCMU results file is open. • SCMU results are not compatible with the working case. • Error reading the SCMU results file.
IERR = 7	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.279 SENSITIVITY_FLOW

This API is used to calculate sensitivity factors of a branch flow to MW power at buses, MW power at generator buses, MW power at load buses, phase angle of phase shifters, tap position of tap changing transformers as well as admittance of switched shunts.

Batch command syntax:

```
BAT_SENSITIVITY_FLOW FLWOPT(1) … FLWOPT(5) OPTIONS(1)… OPTIONS(9)
VALUES(1) VALUES(2) BRNCKT LABELS(1) LABELS(2) DFXFILE
```

Python syntax:

```
ierr = sensitivity_flow (flowopt,options,values,brnckt,labels,
dfxfile)
```

Fortran syntax:

```
CALL
SENSITIVITY_FLOW(FLWOPT,OPTIONS,VALUES,BRNCKT,LABELS,DFXFILE,IERR)
```

where:

Integer FLWOPT — is an array of 5 elements specifying options for sensitivity analysis to a branch flow (input).

(1) Is the from bus of the branch for which sensitivity factors are calculated (no default is allowed)

(2) Is the to bus of the branch (no default is allowed)

(3) Is the third bus connected by a three-winding transformer, or zero for a non-transformer branch or a two-winding transformer (0 by default)

(4) Is network model on which sensitivity analysis is based (0 by default)

0 => DC network model

1 => AC network model

(5) Is the branch flow type to which the sensitivity factors are calculated (If DC network is used, only the branch MW flow is valid for sensitivity analysis. 0 by default)

0 => MW flow

1 => MVar flow

2 => MVA flow

3 => Amps flow

Integer OPTIONS — is an array of 9 elements (input), the value of each element is as follows:

(1) Is the flag to calculate sensitivity factors at buses (1 by default)

0 => Disable, do not calculate sensitivity factors at buses

1 => Enable, calculate sensitivity factors at buses

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

(2) Is the flag to calculate sensitivity factors at generator buses (0 by default)

0 => Disable

1 => Enable

(3) Is the flag to calculate sensitivity factors at load buses (0 by default)

0 => Disable

1 => Enable

(4) Is the flag to calculate sensitivity factors of the branch flow to phase shifters (0 by default)

0 => Disable

1 => Enable

(5) Is the flag to calculate sensitivity factors of the branch flow to tap changing transformers (ignore if DC network model is used, 0 by default)

0 => Disable

1 => Enable

(6) Is the flag to calculate sensitivity factors of the branch flow to switched shunts (ignore if DC network model is used, 0 by default)

0 => Disable

1 => Enable

(7) Is the flag to specify the transfer from the components to which sensitivity factors of the branch flow are calculated to the opposing subsystem (the flag is ignored if none of bus, generator and load sensitivity calculation flags is selected, 0 by default)

0 => Import, the positive transfer is from the components to the opposing subsystem.

1 => Export, the positive transfer is from the opposing subsystem to the components.

(8) Is the opposing subsystem type (0 by default)

0 =>Slack bus

1 =>Single bus

2 =>Subsystem

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

(9) Is ignored if opposing subsystem type is slack bus. It is a bus number when opposing subsystem type is a single bus; or one of the following seven dispatch modes if the opposing subsystem type is a subsystem (1 by default):

1 => Buses & participating factors from DFAX file for buses with positive MW machines

2 => Buses & participating factors from DFAX file for buses with positive MW constant MVA load

3 => Buses & participating factors from DFAX file for buses with either positive MW machines or positive MW constant MVA load

4 => Subsystem buses with positive MW constant MVA load in proportion to their MW load

5 => Subsystem buses with positive MW machines in proportion to their MW output

6 => Subsystem buses with positive MW machines in proportion to their MBASE

7 => Subsystem buses with positive MW machines in proportion to their reserves (Pmaxmc-Pgenmc)

Real VALUES — Is an array of two elements specifying sensitivity analysis parameters (input).

VALUES (1) Is mismatch tolerance for continuing sensitivity analysis in AC mode, ignored in DC mode (TOLN by default)

VALUES (2) Is sensitivity cutoff. Sensitivity factors less than the cutoff are not presented (0.005 by default)

Character BRNCKT*2 — Is the branch circuit ID (input; blank by default)

Character LABELS*32 — Is an array of two elements specifying subsystem labels (input)

LABELS (1) Is the label of the subsystem containing components to which the sensitivity factors of the branch flow are calculated (no default allowed)

LABELS (2) Is the opposing subsystem label when the opposing subsystem type is a subsystem (blank by default)

Character DFXFILE*260 — Is the DFAX file name (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR Is the error return code (output)

0 => No error

1 => Invalid OPTIONS value

2 => Invalid VALUES value

3 => The branch can not be found

4 => Generators are converted

5 => Buses in island(s) without a swing bus--use activity TREE

6 => Dispatch bus does not exist

7 => No Distribution Factor Data input file specified

8 => Subsystem containing components to which sensitivity factors are calculated is not specified

9=> Opposing subsystem is not specified

10 => Largest mismatch exceeds mismatch tolerance

11 => The branch bus does not exist

12 => The branch is out-of-service

13 => The branch is a zero impedance line

14 => The branch is a multi-section line but the Report Multi-Section line option is disabled

21=> file DFXFILE is not in the form of a PSS™E-25 or later DFAX file, run DFAX

22=> Error in linearizing Jacobian matrix with AC network model

51=> error opening Distribution Factor Data File

52=> prerequisite requirements for function not met

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.280 SENSITIVITY_FLOWS

This API is used to calculate sensitivity factors of flows on the branches in a subsystem to MW power at buses, MW power at generator buses, MW power at load buses, phase angle of phase shifters, tap position of tap changing transformers as well as admittance of switched shunts.

Batch command syntax:

```
BAT_SENSITIVITY_FLOWS FLWOPT(1) FLWOPT(2) OPTIONS(1)… OPTIONS(9)
VALUES(1) VALUES(2) LABELS(1)… LABELS(3) DFXFILE
```

Python syntax:

```
ierr = sensitivity_flows
(flowpt,options,values,brnckt,lables,dfxfile)
```

Fortran syntax:

```
Call  SENSITIVITY_FLOWS(FLWOPT,OPTIONS,VALUES,LABELS,DFXFILE,IERR)
```

where:

Integer FLWOPT — is an array of 2 elements specifying options for sensitivity analysis for branches in a subsystem (input).

(1) Is network model on which sensitivity analysis is based (0 by default)

0 => DC network model

1 => AC network model

(2) Is the branch flow type to which the sensitivity factors are calculated (when DC network is used, only the branch MW flow is valid for sensitivity analysis, 0 by default)

0 => MW flow

1 => MVar flow

2 => MVA flow

3 => Amps flow

Integer OPTIONS — is an array of 9 integer elements (input), the value of each element is as follows:

OPTIONS (1) — Is the flag to calculate sensitivity factors at buses (1 by default)

0 => Disable, do not calculate sensitivity factors at buses

1 => Enable, calculate sensitivity factors at buses

OPTIONS (2) — Is the flag to calculate sensitivity factors at generator buses (0 by default)

0 => Disable

1 => Enable

OPTIONS (3) — Is the flag to calculate sensitivity factors at load buses (0 by default)

0 => Disable

1 => Enable

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS (4)	is the flag to calculate sensitivity factors of the branch flow to phase shifters (0 by default) 0 => Disable 1 => Enable
OPTIONS (5)	Is the flag to calculate sensitivity factors of the branch flow to tap changing transformers (ignore if DC network model is used, 0 by default) 0 => Disable 1 => Enable
OPTIONS (6)	Is the flag to calculate sensitivity factors of the branch flow to switched shunts (ignore if DC network model is used, 0 by default) 0 => Disable 1 => Enable
OPTIONS (7)	Is the flag to specify the transfer from the components to which sensitivity factors of the branch flow are calculated to the opposing subsystem (the flag is ignored if none of bus, generator and load sensitivity calculation flags is selected, 0 by default) 0 => Import, the positive transfer is from the components to the opposing subsystem. 1 => Export, the positive transfer is from the opposing subsystem to the components.
OPTIONS (8)	Is the opposing subsystem type (0 by default) 0 =>Slack bus 1 =>Single bus 2 =>Subsystem

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTIONS (9)	Is ignored if opposing subsystem type is slack bus. It is a bus number when opposing subsystem type is a single bus; or one of the following seven dispatch modes if the opposing subsystem type is a subsystem (1 by default): 1 => Buses & participating factors from DFAX file for buses with positive MW machines 2 => Buses & participating factors from DFAX file for buses with positive MW constant MVA load 3 => Buses & participating factors from DFAX file for buses with either positive MW machines or positive MW constant MVA load 4 => Subsystem buses with positive MW constant MVA load in proportion to their MW load 5 => Subsystem buses with positive MW machines in proportion to their MW output 6 => Subsystem buses with positive MW machines in proportion to their MBASE 7 => Subsystem buses with positive MW machines in proportion to their reserves (Pmaxmc-Pgenmc)
Real VALUES	Is an array of two elements specifying sensitivity analysis parameters (input).	
	VALUES (1)	Is mismatch tolerance for continuing sensitivity analysis in AC mode, ignored in DC mode (TOLN by default)
	VALUES (2)	Is sensitivity cutoff. Sensitivity factors less than the cutoff are not presented (0.005 by default)
Character LABELS(3)*12	Is an array of three elements specifying subsystem labels (input)	
	LABELS(1) =	Is the label of the subsystem containing components to which the sensitivity factors of the branch flow are calculated (no default allowed)
	LABELS(2) =	Is the label of opposing subsystem when the opposing subsystem type is a subsystem (blank by default)
	LABELS(3) =	Is the label of the subsystem including branches for which sensitivity factors are calculated (no default allowed)
Character DFXFILE*260	Is the DFAX file name (input; no default allowed).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error return code (output).	
	0 =>	No error
	1 =>	Invalid OPTIONS value
	2 =>	Invalid VALUES value
	3 =>	Subsystem containing branches for which sensitivity factors are calculated is not specified
	4 =>	Generators are converted
	5 =>	Buses in island(s) without a swing bus--use activity TREE
	6 =>	Dispatch bus does not exist
	7 =>	No Distribution Factor Data input file specified
	8 =>	Subsystem containing components to which sensitivity factors are calculated is not specified
	9 =>	Opposing subsystem is not specified
	10 =>	Largest mismatch exceeds mismatch tolerance
	21 =>	file DFXFILE is not in the form of a PSS™E-25 or later DFAX file, run DFAX
	22 =>	Error linearizing Jacobian matrix in AC network model
	51 =>	error opening Distribution Factor Data File
	52 =>	prerequisite requirements for function not met

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.281 SENSITIVITY_VOLTAGE

This API is used to calculate sensitivity factors of a bus voltage to MW and MVar power at buses, MW power at generator buses, MW and MVar at load buses, phase angle of phase shifters, tap position of tap changing transformers as well as admittance of switched shunts. An AC network model must be used for sensitivity analysis on voltages.

Batch command syntax:

```
BAT_SENSITIVITY_VOLTAGE IBUS OPTIONS(1)… OPTIONS(9) VALUES(1)
VALUES(2) LABELS(1) LABELS(2) DFXFILE
```

Python syntax:

```
ierr = sensitivity_voltage (ibus, options,values,labels,dfxfile)
```

Fortran syntax:

```
Call  SENSITIVITY_VOLTAGE(IBUS,OPTIONS,VALUES,LABELS,DFXFILE,IERR)
```

where:

Integer IBUS	Is the bus number for which sensitivity factors are calculated (input, no default allowed)	
Integer OPTIONS	is an array of 9 integer elements (input), the value of each element is as follows:	
	OPTIONS (1)	Is the flag to calculate sensitivity factors at buses (1 by default) 0 => Disable, do not calculate sensitivity factors at buses 1 => Enable, calculate sensitivity factors at buses
	OPTIONS (2)	Is the flag to calculate sensitivity factors at generator buses (0 by default) 0 => Disable 1 => Enable
	OPTIONS (3)	Is the flag to calculate sensitivity factors at load buses (0 by default) 0 => Disable 1 => Enable
	OPTIONS (4)	Is the flag to calculate sensitivity factors of the branch flow to phase shifters (0 by default) 0 => Disable 1 => Enable
	OPTIONS (5)	Is the flag to calculate sensitivity factors of the branch flow to tap changing transformers 0 => Disable 1 => Enable

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTIONS (6)	Is the flag to calculate sensitivity factors of the branch flow to switched shunts 0 => Disable 1 => Enable
	OPTIONS (7)	Is the flag to specify the transfer from the components to which sensitivity factors of the branch flow are calculated to the opposing subsystem (the flag is ignored if none of bus, generator and load sensitivity calculation flags is selected, 0 by default) 0 => Import. The positive transfer is from the components to the opposing subsystem. 1 => Export. The positive transfer is from the opposing subsystem to the components.
	OPTIONS (8)	Is the opposing subsystem type (0 by default) 0 =>Slack bus 1 =>Single bus 2 =>Subsystem
	OPTIONS (9)	Is ignored if opposing subsystem type is slack bus. It is a bus number when opposing subsystem type is a single bus; or one of the following seven dispatch modes if the opposing subsystem type is a subsystem (1 by default): 1 => Buses & participating factors from DFAX file for buses with positive MW machines 2 => Buses & participating factors from DFAX file for buses with positive MW constant MVA load 3 => Buses & participating factors from DFAX file for buses with either positive MW machines or positive MW constant MVA load 4 => Subsystem buses with positive MW constant MVA load in proportion to their MW load 5 => Subsystem buses with positive MW machines in proportion to their MW output 6 => Subsystem buses with positive MW machines in proportion to their MBASE 7 => Subsystem buses with positive MW machines in proportion to their reserves (Pmaxmc-Pgenmc)
Real VALUES	Is an array of two elements specifying sensitivity analysis parameters (input).	
	VALUES (1)	Is mismatch tolerance for continuing sensitivity analysis in AC mode, ignored in DC mode (TOLN by default)
	VALUES (2)	Is sensitivity cutoff. Sensitivity factors less than the cutoff are not presented (0.005 by default)

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character LABELS*32	Is an array of two elements specifying subsystem labels (input)	
	LABELS(1) =	Is the label of the subsystem containing components to which the sensitivity factors of the bus voltage are calculated (no default allowed)
	LABELS(2) =	Is the opposing subsystem label when the opposing subsystem type is a subsystem
Character DFXFILE*260	Is the DFAX file name (input; no default allowed).	
Integer ERRARG	Is error return code (output)	
	ERRARG = 0	No error
	ERRARG = 1	Invalid OPTIONS value
	ERRARG = 2	Invalid VALUES value
	ERRARG = 3	The bus does not exist
	ERRARG = 4	Generators are converted
	ERRARG = 5	Buses in island(s) without a swing bus--use activity TREE
	ERRARG = 6	Dispatch bus does not exist
	ERRARG = 7	No Distribution Factor Data input file specified
	ERRARG = 8	Subsystem containing components to which sensitivity factors are calculated is not specified
	ERRARG = 9	Opposing subsystem is not specified
	ERRARG = 10	Largest mismatch exceeds mismatch tolerance
	ERRARG = 11	The bus is a type 2 or type 3 bus
	ERRARG = 21	file DFXFILE is not in the form of a PSS™E-25 or later DFAX file, run DFAX
	ERRARG = 22	Error linearizing Jacobian matrix in AC network model
	ERRARG = 51	error opening Distribution Factor Data File
	ERRARG = 52	prerequisite requirements for function not met

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.282 SENSITIVITY_VOLTAGES

This API is used to calculate sensitivity factors of bus voltages in a subsystem to MW and MVar power at buses, MW power at generator buses, MW and MVar at load buses, phase angle of phase shifters, tap position of tap changing transformers as well as admittance of switched shunts. An AC network model must be used for sensitivity analysis on voltages.

Batch command syntax:

```
BAT_SENSITIVITY_VOLTAGES OPTIONS(1)… OPTIONS(9) VALUES(1) VALUES(2)
LABELS(1) LABELS(3) DFXFILE
```

Python syntax:

```
ierr = sensitivity_voltages (options,values,labels,dfxfile)
```

Fortran syntax:

```
Call  SENSITIVITY_VOLTAGES(OPTIONS,VALUES,LABELS,DFXFILE,IERR)
```

where:

Integer OPTIONS(9)	is an array of 9 integer elements (input), the value of each element is as follows:	
	OPTIONS (1)	Is the flag to calculate sensitivity factors at buses (1 by default)
	OPTIONS (1) = 0	Disable, do not calculate sensitivity factors at buses
	OPTIONS (1) = 1	Enable, calculate sensitivity factors at buses
	OPTIONS (2)	Is the flag to calculate sensitivity factors at generator buses (0 by default)
	OPTIONS (2) = 0	Disable
	OPTIONS (2) = 1	Enable
	OPTIONS (3)	Is the flag to calculate sensitivity factors at load buses (0 by default)
	OPTIONS(3) = 0	Disable
	OPTIONS(3) = 1	Enable
	OPTIONS (4)	is the flag to calculate sensitivity factors of the branch flow to phase shifters (0 by default)
	OPTIONS(4) = 0	Disable
	OPTIONS(4) = 1	Enable
	OPTIONS (5)	Is the flag to calculate sensitivity factors of the branch flow to tap changing transformers
	OPTIONS(5) = 0	Disable
	OPTIONS(5) = 1	Enable
	OPTIONS (6)	Is the flag to calculate sensitivity factors of the branch flow to switched shunts
	OPTIONS(6) = 0	Disable
	OPTIONS(6) = 1	Enable

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTIONS (7)	Is the flag to specify the transfer from the components to which sensitivity factors of the branch flow are calculated to the opposing subsystem (the flag is ignored if none of bus, generator and load sensitivity calculation flags is selected, 0 by default)
	OPTIONS(7) = 0	Import, the positive transfer is from the components to the opposing subsystem.
	OPTIONS(7) = 1	Export, the positive transfer is from the opposing subsystem to the components.
	OPTIONS (8)	Is the opposing subsystem type (0 by default)
	OPTIONS(8) = 0	Slack bus
	OPTIONS(8) = 1	Single bus
	OPTIONS(8) = 2	Subsystem
	OPTIONS (9)	Is ignored if opposing subsystem type is slack bus. It is a bus number when opposing subsystem type is a single bus; or one of the following seven dispatch modes if the opposing subsystem type is a subsystem (1 by default):
	OPTIONS(9) = 1	Buses & participating factors from DFAX file for buses with positive MW machines
	OPTIONS(9) = 2	Buses & participating factors from DFAX file for buses with positive MW constant MVA load
	OPTIONS(9) = 3	Buses & participating factors from DFAX file for buses with either positive MW machines or positive MW constant MVA load
	OPTIONS(9) = 4	Subsystem buses with positive MW constant MVA load in proportion to their MW load
	OPTIONS(9) = 5	Subsystem buses with positive MW machines in proportion to their MW output
	OPTIONS(9) = 6	Subsystem buses with positive MW machines in proportion to their MBASE
	OPTIONS(9) = 7	Subsystem buses with positive MW machines in proportion to their reserves (Pmaxmc-Pgenmc)
Real VALUES	Is an array of two elements specifying sensitivity analysis parameters (input).	
	VALUES (1)	Is mismatch tolerance for continuing sensitivity analysis in AC mode, ignored in DC mode (TOLN by default)
	VALUES (2)	Is sensitivity cutoff. Sensitivity factors less than the cutoff are not presented (0.005 by default)

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character LABELS*32	Is an array of three elements specifying subsystem labels (input).	
	LABELS(1) =	Is the label of the subsystem containing components to which the sensitivity factors of the bus voltage are calculated (no default allowed)
	LABELS(2) =	Is the label of opposing subsystem when the opposing subsystem type is a subsystem (blank by default)
	LABELS(3) =	Is the label of the subsystem including buses for which sensitivity factors are calculated (no default allowed)
Character DFXFILE*260	Is the DFAX file name (input; no default allowed).	
Integer IERR	Is the error code (output).	
	IERR = 0	No error
	IERR = 1	Invalid OPTIONS(%%) value:%%
	IERR = 2	Invalid VALUES(%%) value:%%
	IERR = 3	The bus %% does not exist
	IERR = 4	Generators are converted
	IERR = 5	Buses in island(s) without a swing bus--use activity TREE
	IERR = 6	Dispatch bus does not exist
	IERR = 7	No Distribution Factor Data input file specified
	IERR = 8	Subsystem containing components to which sensitivity factors are calculated is not specified
	IERR = 9	Opposing subsystem is not specified
	IERR = 10	Largest mismatch exceeds mismatch tolerance
	IERR = 21	file DFXFILE is not in the form of a PSS™E-25 or later DFAX file, run DFAX
	IERR = 22	Error in linearizing Jacobian matrix with AC network model
	IERR = 51	error opening Distribution Factor Data File
	IERR = 52	prerequisite requirements for function not met

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.283 SENSITIVITY_INTERFACE

This API is used to calculate sensitivity factors of an interface flow to MW power at buses, MW power at generator buses, MW power at load buses, phase angle of phase shifters, tap position of tap changing transformers as well as admittance of switched shunts.

Batch command syntax:

```
BAT_ SENSITIVITY_INTERFACE MODE OPTIONS(1)… OPTIONS(9) VALUES(1)
VALUES(2) LABELS(1)… LABELS(3) DFXFILE
```

Python syntax:

```
ierr = sensitivity_interface (mode,options,values,labels,dfxfile)
```

Fortran syntax:

```
Call  SENSITIVITY_INTERFACE(MODE,OPTIONS,VALUES,LABELS,DFXFILE,IERR)
```

where:

Integer MODE	Is network model on which sensitivity analysis is based (input; 0 by default) 0 => DC network model 1 => AC network model

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer OPTIONS(9) is an array of 9 integer elements (input), the value of each element is as follows:

(1) Is the flag to calculate sensitivity factors at buses (1 by default)

0 => Disable, do not calculate sensitivity factors at buses

1 => Enable, calculate sensitivity factors at buses

(2) Is the flag to calculate sensitivity factors at generator buses (0 by default)

0 => Disable

1 => Enable

(3) Is the flag to calculate sensitivity factors at load buses (0 by default)

0 => Disable

1 => Enable

(4) is the flag to calculate sensitivity factors of the branch flow to phase shifters (0 by default)

0 => Disable

1 => Enable

(5) is the flag to calculate sensitivity factors of the branch flow to phase shifters (0 by default)

0 => Disable

1 => Enable

(6) Is the flag to calculate sensitivity factors of the branch flow to switched shunts (ignore if DC network model is used, 0 by default)

0 => Disable

1 => Enable

(7) Is the flag to specify the transfer from the components to which sensitivity factors of the branch flow are calculated to the opposing subsystem (the flag is ignored if none of bus, generator and load sensitivity calculation flags is selected, 0 by default)

0 => Import, the positive transfer is from the components to the opposing subsystem.

1 => Export, the positive transfer is from the opposing subsystem to the components.

(8) Is the opposing subsystem type (0 by default)

0 =>Slack bus

1 =>Single bus

2 =>Subsystem

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

(9) Is ignored if opposing subsystem type is slack bus. It is a bus number when opposing subsystem type is a single bus; or one of the following seven dispatch modes if the opposing subsystem type is a subsystem (1 by default):

1 => Buses & participating factors from DFAX file for buses with positive MW machines

2 => Buses & participating factors from DFAX file for buses with positive MW constant MVA load

3 => Buses & participating factors from DFAX file for buses with either positive MW machines or positive MW constant MVA load

4 => Subsystem buses with positive MW constant MVA load in proportion to their MW load

5 => Subsystem buses with positive MW machines in proportion to their MW output

6 => Subsystem buses with positive MW machines in proportion to their MBASE

7 => Subsystem buses with positive MW machines in proportion to their reserves (Pmaxmc-Pgenmc)

Real (VALUES) Is an array of two elements specifying sensitivity analysis parameters (input).

VALUES (1) Is mismatch tolerance for continuing sensitivity analysis in AC mode, ignored in DC mode (TOLN by default)

VALUES (2) Is sensitivity cutoff. Sensitivity factors less than the cutoff are not presented (0.005 by default)

Character LABELS*32 Is an array of three elements specifying subsystem/interface labels (input).

(1) Is the label of the subsystem containing components to which the sensitivity factors of the interface flow are calculated (no default allowed)

(2) Is the label of opposing subsystem when the opposing subsystem type is a subsystem (blank by default)

(3) Is the label of the interface for which sensitivity factors are calculated (no default allowed)

Character DFXFILE*260 Is the DFAX file name (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR Is error return code (output).

0 => No error
1 => Invalid OPTIONS value
2 => Invalid VALUES value
3 => Interface is not specified
4 => Generators are converted
5 => Buses in island(s) without a swing bus--use activity TREE
6 => Dispatch bus does not exist
7 => No Distribution Factor Data input file specified
8 => Subsystem containing components to which sensitivity factors are calculated is not specified
9=> Opposing subsystem is not specified
10 => Largest mismatch exceeds mismatch tolerance
21=> file DFXFILE is not in the form of a PSS™E-25 or later DFAX file, run DFAX
22=> Error in linearizing Jacobian matrix with AC network model
51=> error opening Distribution Factor Data File
52=> prerequisite requirements for function not met

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.284 SCOP

Use this API to tabulate unbalanced fault solution output (activity SCOP). It **must** be preceded by the execution of activity SCMU, either interactively, from a Response File, or via Python. It **must** then be referenced using one of the following procedures:

- One reference with APIOPT=1, followed by any number of references with APIOPT=2, followed by one reference with APIOPT=3.
- One reference with APIOPT=0. This automatically combines the processing of one APIOPT 1 reference, followed by one APIOPT 2 reference, followed by one APIOPT 3 reference.

Batch command syntax:

```
BAT_SCOP  SID  ALL  UNITS  APIOPT
```

Python syntax:

```
ierr = scop(sid, all, units, apiopt)
```

Fortran syntax:

```
CALL SCOPAPI(SID, ALL, UNITS, APIOPT, IERR)
```

where:

Integer SID	Is a valid subsystem identifier; used only when APIOPT is 0 or 2 and ALL is 0. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag; used only when APIOPT is 0 or 2 (input; 1 by default).	
	ALL = 1	process all buses.
	ALL = 0	process only buses in subsystem SID.
Integer UNITS	Is the branch quantity output flag; used only when APIOPT is 0 or 2 (input; 1 by default).	
	UNITS = 1	currents.
	UNITS = 2	apparent impedances.
	UNITS = 3	apparent admittances.
Integer APIOPT	Is the mode of operation (input; 0 by default).	
	APIOPT = 0	initialize for short circuit output, produce the output, and run post-processing clean up.
	APIOPT = 1	initialize for short circuit output.
	APIOPT = 2	produce output.
	APIOPT = 3	post-processing clean up.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid SID value or subsystem SID is not defined.
IERR = 2	invalid ALL value.
IERR = 3	invalid UNITS value.
IERR = 4	invalid APIOPT value.
IERR = 5	unexpected APIOPT value.
IERR = 6	SCMU results error; one of the following occurred: • No SCMU results file is open. • SCMU results are not compatible with the working case. • Error reading the SCMU results file.
IERR = 7	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.285 SEQD

This API routine is obsolete. It has been replaced by the API routine SEQD_2, and is implemented by a call to the SEQD_2 API routine.

Use this API to factorize the sequence matrices in preparation for unbalanced fault calculations (activity SEQD).

Batch command syntax:

```
BAT_SEQD  OPTIONS(1)  OPTIONS(2)
```

Python syntax:

```
ierr = seqd(options)
```

Fortran syntax:

```
CALL SEQDAPI(OPTIONS, IERR)
```

where:

Integer OPTIONS(2)	Is an array of two elements specifying options to be used in constructing the sequence matrices (input). The values are as follows:	
	OPTIONS(1)	dc line and FACTS device option (0 by default).
	OPTIONS(1) = 0	block and ignore.
	OPTIONS(1) = 1	convert to constant admittance load.
	OPTIONS(2)	transformer impedance correction option (1 by default).
	OPTIONS(2) = 0	do not apply to zero sequence.
	OPTIONS(2) = 1	apply to zero sequence.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid OPTIONS value.
	IERR = 2	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.286 SEQD_2

Use this API routine to factorize the sequence matrices in preparation for unbalanced fault calculations SCMU and SPCB (activity SEQD).

Batch command syntax:

```
BAT_SEQD_2  OPTIONS(1)  OPTIONS(2)  OPTIONS(3)
```

Python syntax:

```
ierr = seqd_2(options)
```

Fortran syntax:

```
CALL SEQD_2(OPTIONS, IERR)
```

where:

Integer OPTIONS(3) Is an array of three elements specifying options to be used in constructing the sequence matrices (input). The values are as follows:

OPTIONS(1)	dc line and FACTS device option (0 by default).
OPTIONS(1) = 0	block and ignore.
OPTIONS(1) = 1	convert to constant admittance load.
OPTIONS(2)	transformer impedance correction option (1 by default).
OPTIONS(2) = 0	do not apply to zero sequence.
OPTIONS(2) = 1	apply to zero sequence.
OPTIONS(3)	fault analysis generator reactance option; ignored if generators are already "converted" (0 by default).
OPTIONS(3) = 0	use subtransient reactance.
OPTIONS(3) = 1	use transient reactance.
OPTIONS(3) = 2	use synchronous reactance.

Integer IERR Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid OPTIONS value.
IERR = 2	fault analysis is invalid due to the presence of in-service induction machines that are in the "stalled" or "tripped" state
IERR = 3	bus type code and series element status inconsistencies.
IERR = 4	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.287 SEQUENCE_NETWORK_SETUP

Use this API to enable or disable the fault analysis sequence network setup option (activity OPTN).

Batch command syntax:

```
BAT_SEQUENCE_NETWORK_SETUP  IVAL
```

Python syntax:

```
ierr = sequence_network_setup(ival)
```

or:

```
ierr,ival = sequence_network_setup()
```

Fortran syntax:

```
CALL SEQUENCE_NETWORK_SETUP_API(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Is the value of the option setting (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 1 by default for input).	
	IVAL = 0	SCMU assumes SEQD has already been executed.
	IVAL = 1	SCMU always executes activity SEQD.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE = 0	set the option setting to the value specified in IVAL.
	IOCODE = 1	return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid IVAL value.
	IERR = 2	invalid IOCODE value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.288 SET_INPUT_DEV

Use this API to set the terminal input device to a file.

Batch command syntax:

```
BAT_SET_INPUT_DEV  IDVFIL
```

Python syntax:

```
ierr = set_input_dev(idvfil)
```

Fortran syntax:

```
CALL IDEVAPI(IDVFIL, IERR)
```

where:

Character IDVFIL*260	Is the name of Response File to be used for terminal input.
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 error opening IDVFIL.
	IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.289 SHNT

Use this API to tabulate fixed and/or switched bus shunts contained in the working case (activity SHNT).

Batch command syntax:

```
BAT_SHNT  SID  ALL  OPT
```

Python syntax:

```
ierr = shnt(sid, all, opt)
```

Fortran syntax:

```
CALL SHNTNW(SID, ALL, OPT, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Integer OPT	Is the option to select the type of bus shunts tabulated (input; 1 by default). OPT = 1 all shunts. OPT = 2 fixed bus shunts. OPT = 3 switched shunts.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 invalid OPT value. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.290 SHORT_CIRCUIT_COORDINATES

Use this API to set the fault analysis output coordinates option setting to either rectangular or polar coordinates (activity OPTN).

Batch command syntax:

```
BAT_SHORT_CIRCUIT_COORDINATES  IVAL
```

Python syntax:

```
ierr = short_circuit_coordinates(ival)
```

or:

```
ierr,ival = short_circuit_coordinates()
```

Fortran syntax:

```
CALL SHORT_CIRCUIT_COORDINATES_API(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Is the value of the option setting (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 1 by default for input).	
	IVAL = 0	rectangular coordinates.
	IVAL = 1	polar coordinates.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE = 0	set the option setting to the value specified in IVAL.
	IOCODE = 1	return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid IVAL value.
	IERR = 2	invalid IOCODE value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.291 SHORT_CIRCUIT_MODELING

Use this API to set the fault analysis modeling option setting to either *normal three-phase* or *center tapped two-phase* (activity OPTN).

Batch command syntax:

```
BAT_SHORT_CIRCUIT_MODELING  IVAL
```

Python syntax:

```
ierr = short_circuit_modeling(ival)
```

or:

```
ierr,ival = short_circuit_modeling()
```

Fortran syntax:

```
CALL SHORT_CIRCUIT_MODELING_API(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Is the value of the option setting (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 1 by default for input).	
	IVAL = 0	center tapped two-phase.
	IVAL = 1	normal three-phase.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE = 0	set the option setting to the value specified in IVAL.
	IOCODE = 1	return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid IVAL value.
	IERR = 2	invalid IOCODE value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.292 SHORT_CIRCUIT_UNITS

Use this API to set the fault analysis output option setting to either per unit or physical units (activity OPTN).

Batch command syntax:

```
BAT_SHORT_CIRCUIT_UNITS  IVAL
```

Python syntax:

```
ierr = short_circuit_units(ival)
```

or:

```
ierr,ival = short_circuit_units()
```

Fortran syntax:

```
CALL SHORT_CIRCUIT_UNITS_API(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Is the value of the option setting (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 1 by default for input).	
	IVAL = 0	per unit.
	IVAL = 1	physical units.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE = 0	set the option setting to the value specified in IVAL.
	IOCODE = 1	return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid IVAL value.
	IERR = 2	invalid IOCODE value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.293 SHORT_CIRCUIT_WARNING

Use this API to enable or disable the fault analysis warning option setting (activity OPTN).

Batch command syntax:

```
BAT_SHORT_CIRCUIT_WARNING  IVAL
```

Python syntax:

```
ierr = short_circuit_warning(ival)
```

or:

```
ierr,ival = short_circuit_warning()
```

Fortran syntax:

```
CALL SHORT_CIRCUIT_WARNING_API(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Is the value of the option setting (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 1 by default for input).	
	IVAL = 0	disable printing of RESQ/TRSQ/solution warnings.
	IVAL = 1	print RESQ/TRSQ/solution warnings.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE = 0	set the option setting to the value specified in IVAL.
	IOCODE = 1	return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid IVAL value.
	IERR = 2	invalid IOCODE value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.294 SHOW

Use this API to tabulate summaries of Saved Case and/or Snapshot Files (activity SHOW).

Batch command syntax:

```
BAT_SHOW  OPTION  SFILE
```

Python syntax:

```
ierr = show(option, sfile)
```

Fortran syntax:

```
CALL SHOWAPI(OPTION, SFILE, IERR)
```

where:

Integer OPTION	Is the code for the type(s) of files to report (input; 1 by default). OPTION = 0 file specified in SFILE. OPTION = 1 all .sav and .snp files. OPTION = 2 all .sav files. OPTION = 3 all .snp files.
Character SFILE*260	Is the name of Saved Case or Snapshot File (input; no default allowed when OPTION is 0).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid OPTION value. IERR = 2 SFILE is blank when OPTION is 0. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.295 SIZE

Use this API to obtain a summary of the number of components in the working case (activity SIZE).

Batch command syntax:

```
BAT_SIZE
```

Python syntax:

```
ierr = size()
```

Fortran syntax:

```
CALL SIZEAPI(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.296 SOLV

Use this API to apply the Gauss-Seidel power flow calculation (activity SOLV).

Batch command syntax:

```
BAT_SOLV  OPTIONS(1)...OPTIONS(6)
```

Python syntax:

```
ierr = solv(options)
```

Fortran syntax:

```
CALL SOLVAPI(OPTIONS, IERR)
```

where:

Integer OPTIONS(6)	Is an array of six element specifying solution options (input). The values are as follows:	
	OPTIONS(1)	tap adjustment flag (use tap adjustment option setting by default).
	OPTIONS(1) = 0	disable.
	OPTIONS(1) = 1	enable.
	OPTIONS(2)	area interchange adjustment flag (use area interchange adjustment option setting by default).
	OPTIONS(2) = 0	disable.
	OPTIONS(2) = 1	enable using tie line flows only in calculating area interchange.
	OPTIONS(2) = 2	enable using tie line flows and loads in calculating area interchange.
	OPTIONS(3)	flat start flag (0 by default).
	OPTIONS(3) = 0	do not flat start.
	OPTIONS(3) = 1	flat start.
	OPTIONS(4)	dc tap adjustment flag (use dc tap adjustment option setting by default).
	OPTIONS(4) = 0	disable.
	OPTIONS(4) = 1	enable.
	OPTIONS(5)	switched shunt adjustment flag (use switched shunt adjustment option setting by default).
	OPTIONS(5) = 0	disable.
	OPTIONS(5) = 1	enable.
	OPTIONS(5) = 2	enable continuous mode, disable discrete mode.
	OPTIONS(6)	var limit flag (0 by default).
	OPTIONS(6) = 0	apply var limits.
	OPTIONS(6) = 1	ignore var limits.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid OPTIONS value.
	IERR = 2	generators are converted.
	IERR = 3	buses in island(s) without a swing bus; use activity TREE.
	IERR = 4	bus type code and series element status inconsistencies.
	IERR = 5	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.297 SPCB

Use this API to calculate positive sequence equivalents of branch unbalances (activity SPCB).

Batch command syntax:

```
BAT_SPCB  IBUS  JBUS  CKT  OPTIONS(1)...OPTIONS(4)
VALUES(1)...VALUES(5)
```

Python syntax:

```
ierr = spcb(ibus, jbus, ckt, options, values)
```

Fortran syntax:

```
CALL SPCBAPI(IBUS, JBUS, CKT, OPTIONS, VALUES, IERR)
```

where:

Integer IBUS	Is the from bus of branch subjected to unbalance (input; no default allowed).
Integer JBUS	Is the to bus of branch subjected to unbalance (input; no default allowed).
Character CKT*2	Is the circuit identifier of branch subjected to unbalance (input; ’1’ by default).
Integer OPTIONS(4)	Is an array of four elements array specifying unbalance options (input). The values are as follows:

OPTIONS(1)	unbalance code (0 by default).
OPTIONS(1) = 0	no unbalance; Thevenin impedances only.
OPTIONS(1) = 1	one phase open.
OPTIONS(1) = 2	two phases open.
OPTIONS(1) = 3	in-line fault.
OPTIONS(1) = 4	one breaker open.
OPTIONS(1) = 5	no unbalance.
OPTIONS(2)	path to ground code (used when OPTIONS(1) is 1 or 4) (0 by default).
OPTIONS(2) = 0	no path to ground.
OPTIONS(2) = 1	include a path to ground.
OPTIONS(3)	type of in-line fault code (used when OPTIONS(1) is 3) (1 by default).
OPTIONS(3) = 1	line-to-ground.
OPTIONS(3) = 2	line-to-line-to-ground.
OPTIONS(3) = 3	three-phase.
OPTIONS(4)	open end code (used when OPTIONS(1) is 4) (1 by default).
OPTIONS(4) = 0	breaker at to bus is open.
OPTIONS(4) = 1	breaker at from bus is open.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real VALUES(5) Is an array of five elements specifying unbalance parameters (input).

VALUES(1)	fault location as fraction of line from from bus; 0.0 < VALUES(1) < 1.0 (used when OPTIONS(1) is 1 or 4 and OPTIONS(2) is 1; or when OPTIONS(1) is 3) (0.5 by default).
VALUES(2) & (3)	Rl-g and Xl-g (used when OPTIONS(1) is 1 or 4 and OPTIONS(2) is 1; or when OPTIONS(1) is 3 and OPTIONS(3) is 1 or 2) (0.0 by default).
VALUES(4) & (5)	Rl-l and Xl-l (used when OPTIONS(1) is 3 and OPTIONS(3) is 2) (0.0 by default).

Integer IERR Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid with two phase modeling option enabled.
IERR = 2	bus not found.
IERR = 3	invalid OPTIONS value.
IERR = 4	invalid VALUES value.
IERR = 5	SEQD temporary file is not open.
IERR = 6	SEQD results are not compatible with the working case.
IERR = 7	terminated due to invalid fault specification or application.
IERR = 8	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.298 SPIL

Use this API to calculate transmission interchange limits using a linear network (dc) model (activity SPIL).

Batch command syntax:

```
BAT_SPIL  OPTIONS(1)...OPTIONS(16)  VALUES(1)...VALUES(7)
LABELS(1)...LABELS(8)  DFXFILE  PRTFILE
```

Python syntax:

```
ierr = spil(options, values, labels, dfxfile, prtfile)
```

Fortran syntax:

```
CALL SPILAPI(OPTIONS, VALUES, LABELS, DFXFILE, PRTFILE, IERR)
```

where:

Integer OPTIONS(16)	Is an array of sixteen elements specifying calculation and reporting options (input). The values are as follows:	
	OPTIONS(1)	base case rating set (rating set program option setting by default).
	OPTIONS(1) = 1	RATEA.
	OPTIONS(1) = 2	RATEB.
	OPTIONS(1) = 3	RATEC.
	OPTIONS(2)	contingency case rating set (rating set program option setting by default).
	OPTIONS(2) = 1	RATEA.
	OPTIONS(2) = 2	RATEB.
	OPTIONS(2) = 3	RATEC.
	OPTIONS(3)	base case line flow code (0 by default).
	OPTIONS(3) = 0	dc base case.
	OPTIONS(3) = 1	ac base case.
	OPTIONS(4)	phase shifter code (0 by default).
	OPTIONS(4) = 0	locked in base case shift solutions.
	OPTIONS(4) = 1	regulating in base case shift solutions.
	OPTIONS(5)	list study system buses (0 by default).
	OPTIONS(5) = 0	no.
	OPTIONS(5) = 1	yes.
	OPTIONS(6)	list opposing system buses (0 by default).
	OPTIONS(6) = 0	no.
	OPTIONS(6) = 1	yes.
	OPTIONS(7)	list study system tie lines (0 by default).
	OPTIONS(7) = 0	no.
	OPTIONS(7) = 1	yes.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTIONS(8)	add study system tie lines to monitored line list (0 by default).
	OPTIONS(8) = 0	no.
	OPTIONS(8) = 1	yes.
	OPTIONS(9)	output format code (0 by default).
	OPTIONS(9) = 0	summary output.
	OPTIONS(9) = 1	full output.
	OPTIONS(10)	interchange limit output code (0 by default).
	OPTIONS(10) = 0	incremental transfer capability.
	OPTIONS(10) = 1	total transfer capability.
	OPTIONS(11)	apply summary minimum distribution factor, VALUES(4), to solution reports (0 by default).
	OPTIONS(11) = 0	no.
	OPTIONS(11) = 1	yes.
	OPTIONS(12)	convert MVA ratings to estimated MW ratings (0 by default).
	OPTIONS(12) = 0	no.
	OPTIONS(12) = 1	yes.
	OPTIONS(13)	code to discontinue activity if MW mismatch exceeds VALUES(1) (0 by default).
	OPTIONS(13) = 0	discontinue.
	OPTIONS(13) = 1	continue.
	OPTIONS(14)	summary table contingency descriptions (1 by default).
	OPTIONS(14) = 0	contingency labels.
	OPTIONS(14) = 1	contingency events.
	OPTIONS(14) = 2	both contingency labels and events.
	OPTIONS(15)	number of elements to include in flow tables (60000 by default).
	OPTIONS(16)	summary table maximum times for reporting the same element (5 by default).
Real VALUES(7)	Is an array of seven elements specifying tolerances and thresholds (input).	
	VALUES(1)	required MW mismatch tolerance for continuing (Newton solution convergence tolerance, TOLN, by default).
	VALUES(2)	percent of rating for reporting an element as overloaded (100.0 by default).
	VALUES(3)	summary table maximum import or export (>0.) (99999.0 by default).
	VALUES(4)	summary table minimum distribution factor magnitude (0.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	VALUES(5)	minimum contingency case pre-shift flow change (0.0 by default).
	VALUES(6)	minimum contingency case distribution factor change (0.0 by default).
	VALUES(7)	study system generation shift (100.0 by default).
Character LABELS(8)*12	Is an array of eight elements specifying subsystem and interface labels (input).	
	LABELS(1)	study system label (no default allowed).
	LABELS(2)	opposing system label (no default allowed).
	LABELS(3)	interface label for optional interface limits summary table (blank for none) (blank by default).
	LABELS(4)	interface label for optional interface limits summary table (blank for none) (blank by default).
	LABELS(5)	interface label for optional interface limits summary table (blank for none) (blank by default).
	LABELS(6)	interface label for optional interface limits summary table (blank for none) (blank by default).
	LABELS(7)	interface label for optional interface limits summary table (blank for none) (blank by default).
	LABELS(8)	interface label for optional interface limits summary table (blank for none) (blank by default).
Character DFXFILE*260	Is the name of Distribution Factor Data File (input; no default allowed).	
Character PRTFILE*260	Is the name of Subsystem Participation Data file; blank for none (input; blank by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid OPTIONS value.
	IERR = 2	invalid VALUES value.
	IERR = 3	no Distribution Factor Data File specified.
	IERR = 4	same label specified for study and opposing systems.
	IERR = 5	generators are converted.
	IERR = 6	buses in island(s) without a swing bus; use activity TREE.
	IERR = 7	largest mismatch exceeds mismatch tolerance.
	IERR = 8	error opening file DFXFILE.
	IERR = 9	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IERR = 10	file DFXFILE does not contain distribution factors; run DFAX with distribution factor option enabled.
IERR = 11	no monitored branches or interfaces defined.
IERR = 12	no subsystems defined.
IERR = 13	subsystem not found.
IERR = 14	specified subsystems overlap.
IERR = 15	no in-service buses in subsystem WORLD.
IERR = 16	no tie lines from study system.
IERR = 17	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.299 SPLT

Use this API to add a bus to the working case and place a zero impedance line between bus BUS and the new bus (activity SPLT). No other equipment is connected to the new bus by this API.

Batch command syntax:

```
BAT_SPLT  BUS  NEWNUM  'NEWNAM'  NEWKV
```

Python syntax:

```
ierr = splt(bus, newnum, newnam, newkv)
```

Fortran syntax:

```
CALL SPLTAPI(BUS, NEWNUM, NEWNAM, NEWKV, IERR)
```

where:

Integer BUS	Is the number of the bus to be split (input; no default allowed).
Integer NEWNUM	Is the number to be assigned to the new bus (input; no default allowed).
Character NEWNAM*12	Is the name to be assigned to the new bus (input; blank by default).
Real NEWKV	Is the base voltage to be assigned to the new bus (input; base voltage of bus BUS by default).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 bus not found.
	IERR = 2 invalid NEWNUM value.
	IERR = 3 bus number is already assigned to another bus.
	IERR = 4 invalid NEWKV value.
	IERR = 5 bus, branch, or branch ownership tables are full.
	IERR = 6 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.300 SQEX

Use this API to tabulate all fault analysis data pertaining to a specified bus (activity SQEX).

Batch command syntax:

```
BAT_SQEX  SID  ALL
```

Python syntax:

```
ierr = sqex(sid, all)
```

Fortran syntax:

```
CALL SQEXNW(SID, ALL, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.301 SRAIND

Use this API to run the substation reliability analysis. It applies failure modes and affects analysis of arbitrary bus-breaker configurations and measures the impact of substation equipment failures upon the power transfer capability of the substation.

Batch command syntax:

```
BAT_SRAIND OPTIONS(1)…OPTIONS(3)  VALUES(1)…VALUES(5), LABELS(1) …
LABELS(3),  DFXFILE, PRBFILE
```

Python syntax:

```
ierr = sraind(options, values, labels, dfxfile, prbfile)
```

Fortran syntax:

```
CALL SRAINDAPI(OPTIONS, VALUES, LABELS, DFXFILE, PRBFILE, IERR)
```

where:

Integer OPTIONS(3)	Is an array of three elements specifying output options. The values are as follows:	
	OPTIONS(1)	report format (1 by default).
	OPTIONS(1) =1	contingency report.
	OPTIONS(1) =2	bus load curtailment report.
	OPTIONS(1) =3	substation load curtailment report.
	OPTIONS(2)	rating set (rating set program option setting by default).
	OPTIONS(2) =1	rating A.
	OPTIONS(2) =2	rating B.
	OPTIONS(2) =3	rating C.
	OPTIONS(3)	include system ties (0 by default).
	OPTIONS(3) =0	disable.
	OPTIONS(3) =1	enable.
Real VALUES(5)	Is an array of five elements specifying substation reliability assessment parameters. The values are as follows:	
	VALUES(1)	percent of flow rating (>=0, 100.0 by default).
	VALUES(2)	frequency cutoff (>=0, 0.0001 by default).
	VALUES(3)	percent of peak load (>=0, 100.0 by default).
	VALUES(4)	switching time in hours (>=0, 0.0 by default).
	VALUES(5)	breaker failure rate in hours per year (>=0, 0.0 by default).
Character LABELS(3)*12	Is an array of three elements specifying subsystem labels (input). The values are as follows:	
	LABELS(1)	label of substation subsystem (no default is allowed).
	LABELS(2)	label of source subsystem (no default is allowed).
	LABELS(3)	label of sink subsystem (no default is allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character*260 DFXFILE	Is the name of Distribution Factor Data File (no default allowed).	
Character*260 PRBFILE	Is the name of Outage Statistic Data File (no default allowed).	
Interger IERR	Is an error code (output)	
	IERR=0	no error
	IERR=1	invalid OPTIONS value
	IERR=2	invalid VALUES value
	IERR=3	Substation/Source/Sink subsytem not specified
	IERR=4	no Distribution Factor Data File specified
	IERR=5	no Outage Statistic Data File specified
	IERR=6	File is not in the form of a PSS®E-25 or later DFAX file; run DFAX
	IERR=7	subsystem not found in Distribution Factor Data file
	IERR=8	Source/Sink buses not in substation subsystem
	IERR=9	Source and Sink systems overlap
	IERR=51	error opening Distribution Factor Data File
	IERR=52	error opening Outage Statistic Data File
	IERR=53	File system errors

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.302 SQLI

Use this API to tabulate the sequence data in a form suitable for problem data documentation (activity SQLI).

Batch command syntax:

```
BAT_SQLI  SID  ALL  OPT
```

Python syntax:

```
ierr = sqli(sid, all, opt)
```

Fortran syntax:

```
CALL SQLINW(SID, ALL, OPT, IERR)
```

where:

Integer SID — Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).

Integer ALL — Is the *all buses* or specified subsystem flag (input; 1 by default).

ALL = 1	process all buses.
ALL = 0	process only buses in subsystem SID.

Integer OPT — Is category of sequence data to be tabulated (input; 1 by default).

= 1, full listing	= 2, bus load data
= 3, generator data	= 4, branch data
= 5, two-winding positive sequence data	= 6, mutual coupling data
= 7, switched shunt data	= 8, three-winding general data
= 9, line shunt data	= 10, fixed shunt data
=11, induction machine data	=12, two-winding zero sequence data
=13, three-winding positive sequence data	=14, three-winding zero sequence data

Integer IERR — Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid SID value or subsystem SID is not defined.
IERR = 2	invalid ALL value.
IERR = 3	invalid OPT value.
IERR = 4	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.303 STOP

This API routine is obsolete. It has been replaced by the API routine STOP_2, and is implemented by a call to STOP_2.

Use this API to end the operation of PSS®E (activity STOP).

Batch command syntax:

```
BAT_STOP  DELETE
```

Python syntax:

```
ierr = stop(delete)
```

Fortran syntax:

```
CALL STOPAPI(DELETE, IERR)
```

where:

Integer DELETE	Is the delete working files flag (input; 0 by default).	
	DELETE = 1	delete working files.
	DELETE =	anything except 1, keep working files
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.304 STOP_2

Use this API to end the operation of PSS®E (activity STOP).

Batch command syntax:

```
BAT_STOP_2
```

Python syntax:

```
ierr = stop_2()
```

Fortran syntax:

```
CALL STOPAPI_2(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.305 SUBS

Use this API to summarize conditions in the working case by tabulating the conditions at each swing system bus, conditions at each area slack area bus, number of components, generation/ load/shunt totals, and loss/line shunt/charging totals by voltage levels (activity SUBS).

Batch command syntax:

```
BAT_SUBS  SID  ALL
```

Python syntax:

```
ierr = subs(sid, all)
```

Fortran syntax:

```
CALL SUBSNW(SID, ALL, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.306 SWITCHED_SHUNT_ADJUSTMENT

Use this API to enable or disable the switched shunt adjustment option setting (activity OPTN).

Batch command syntax:

```
BAT_SWITCHED_SHUNT_ADJUSTMENT  IVAL
```

Python syntax:

```
ierr = switched_shunt_adjustment(ival)
```

or:

```
ierr,ival = switched_shunt_adjustment()
```

Fortran syntax:

```
CALL SWITCHED_SHUNT_ADJUSTMENT_API(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Is the value of the option setting (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 1 by default for input).	
	IVAL = 0	disabled.
	IVAL = 1	enabled.
	IVAL = 2	continuous mode switched shunts enabled, discrete mode switched shunts locked.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE = 0	set the option setting to the value specified in IVAL.
	IOCODE = 1	return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid IVAL value.
	IERR = 2	invalid IOCODE value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.307 TAP_ADJUSTMENT

Use this API to set the default tap adjustment option setting to either *disabled*, *stepping* or *direct* (activity OPTN).

Batch command syntax:

```
BAT_TAP_ADJUSTMENT  IVAL
```

Python syntax:

```
ierr = tap_adjustment(ival)
```

or:

```
ierr,ival = tap_adjustment()
```

Fortran syntax:

```
CALL TAP_ADJUSTMENT_API(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Is the value of the option setting (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 1 by default for input).	
	IVAL = 0	locked (i.e., tap adjustment is disabled).
	IVAL = 1	tap adjustment using the stepping method enabled.
	IVAL = 2	tap adjustment using the direct method enabled.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE = 0	set the option setting to the value specified in IVAL.
	IOCODE = 1	return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid IVAL value.
	IERR = 2	invalid IOCODE value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.308 TEXT

This API does nothing. Its purpose is to provide recordable documentation (activity TEXT).

Batch command syntax:

```
BAT_TEXT  'CARG'
```

Python syntax:

```
ierr = text(carg)
```

Fortran syntax:

```
CALL TEXT_API(CARG, IERR)
```

where:

Character CARG*72	Is the string of text (input; blank by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.309 TFLG

Use this API to set or reset the adjustment control mode flags for all automatically adjustable transformers contained in the specified subsystem (activity TFLG).

Batch command syntax:

```
BAT_TFLG  SID  ALL  OPT
```

Python syntax:

```
ierr = tflg(sid, all, opt)
```

Fortran syntax:

```
CALL TFLGNW(SID, ALL, OPT, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all buses.
	ALL = 0	process only buses in subsystem SID.
Integer OPT	Is the option to select the setting given to the adjustment control mode flags of those transformers being processed (input; 1 by default).	
	OPT = 0	disable.
	OPT = 1	enable.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	invalid OPT value.
	IERR = 4	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.310 TIES

Use this API to tabulate the flows on all area tie lines, with tie flows grouped by area (activity TIES).

Batch command syntax:

```
BAT_TIES  SID  ALL
```

Python syntax:

```
ierr = ties(sid, all)
```

Fortran syntax:

```
CALL TIESNW(SID, ALL, IERR)
```

where:

Integer SID	Is a valid area subsystem identifier. Valid subsystem identifiers range from 0 to 11. Area subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all areas* or specified subsystem flag (input; 1 by default). ALL = 1 process all areas. ALL = 0 process only areas in area subsystem SID.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.311 TIEZ

Use this API to tabulate the flows on all zone tie lines, with tie flows grouped by zone (activity TIEZ).

Batch command syntax:

```
BAT_TIEZ  SID  ALL
```

Python syntax:

```
ierr = tiez(sid, all)
```

Fortran syntax:

```
CALL TIEZNW(SID, ALL, IERR)
```

where:

Integer SID	Is a valid zone subsystem identifier. Valid subsystem identifiers range from 0 to 11. Zone subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all zones* or specified subsystem flag (input; 1 by default). ALL = 1 process all zones. ALL = 0 process only zones in zone subsystem SID.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.312 TIME

Use this API to tabulate timing statistics (activity TIME).

Batch command syntax:

```
BAT_TIME  INIT
```

Python syntax:

```
ierr = time(init)
```

Fortran syntax:

```
CALL TIMEAPI(INIT, IERR)
```

where:

Integer INIT	Is the initialize timers flag (input; positive by default).	
	INIT = ≤0	do not initialize.
	INIT = >0	initialize.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.313 TLST

Use this API to tabulate those transformers in the working case where off-nominal turns ratio or phase shift angle may be adjusted by the power flow solution activities (activity TLST).

Batch command syntax:

```
BAT_TLST  SID  ALL  OPT  VREV
```

Python syntax:

```
ierr = tlst(sid, all, opt, vrev)
```

Fortran syntax:

```
CALL TLSTNW(SID, ALL, OPT, VREV, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Integer OPT	Is the option to select the transformers tabulated (input; 0 by default). OPT = 0 all. OPT = 1 violations only.
Integer VREV	Is the option to change voltage output units from the default units to the alternate units (input; 0 by default). VREV = 0 use default voltage units. VREV = 1 use alternate voltage units.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 invalid OPT value. IERR = 4 invalid VREV value. IERR = 5 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.314 TLTG

Use this API to calculate transmission interchange limits using a linear network (dc) model (activity TLTG).

Batch command syntax:

```
BAT_TLTG  OPTIONS(1)...OPTIONS(17)  VALUES(1)...VALUES(7)
LABELS(1)...LABELS(8)  DFXFILE
```

Python syntax:

```
ierr = tltg(options, values, labels, dfxfile)
```

Fortran syntax:

```
CALL TLTGAPI(OPTIONS, VALUES, LABELS, DFXFILE, IERR)
```

where:

Integer OPTIONS(17)	Is an array of seventeen elements specifying calculation and reporting options (input). The values are as follows:	
	OPTIONS(1)	base case rating set (rating set program option setting by default).
	OPTIONS(1) = 1	RATEA.
	OPTIONS(1) = 2	RATEB.
	OPTIONS(1) = 3	RATEC.
	OPTIONS(2)	contingency case rating set (rating set program option setting by default).
	OPTIONS(2) = 1	RATEA.
	OPTIONS(2) = 2	RATEB.
	OPTIONS(2) = 3	RATEC.
	OPTIONS(3)	base case line flow code (0 by default).
	OPTIONS(3) = 0	dc base case.
	OPTIONS(3) = 1	ac base case.
	OPTIONS(4)	phase shifter code (0 by default).
	OPTIONS(4) = 0	locked in base case shift solutions.
	OPTIONS(4) = 1	regulating in base case shift solutions.
	OPTIONS(5)	code for the treatment of base case line flow constraints in contingency cases (0 by default).
	OPTIONS(5) = 0	ignore.
	OPTIONS(5) = 1	include.
	OPTIONS(6)	list study system buses (0 by default).
	OPTIONS(6) = 0	no.
	OPTIONS(6) = 1	yes.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

OPTIONS(7)	list opposing system buses (0 by default).
OPTIONS(7) = 0	no.
OPTIONS(7) = 1	yes.
OPTIONS(8)	list study system tie lines (0 by default).
OPTIONS(8) = 0	no.
OPTIONS(8) = 1	yes.
OPTIONS(9)	add study system tie lines to monitored line list (0 by default).
OPTIONS(9) = 0	no.
OPTIONS(9) = 1	yes.
OPTIONS(10)	output format code (0 by default).
OPTIONS(10) = 0	summary output.
OPTIONS(10) = 1	full output.
OPTIONS(11)	interchange limit output code (0 by default).
OPTIONS(11) = 0	incremental transfer capability.
OPTIONS(11) = 1	total transfer capability.
OPTIONS(12)	apply summary minimum distribution factor, VALUES(4), to solution reports (0 by default).
OPTIONS(12) = 0	no.
OPTIONS(12) = 1	yes.
OPTIONS(13)	convert MVA ratings to estimated MW ratings (0 by default).
OPTIONS(13) = 0	no.
OPTIONS(13) = 1	yes.
OPTIONS(14)	code to discontinue activity if MW mismatch exceeds VALUES(1) (0 by default).
OPTIONS(14) = 0	discontinue.
OPTIONS(14) = 1	continue.
OPTIONS(15)	summary table contingency descriptions (1 by default).
OPTIONS(15) = 0	contingency labels.
OPTIONS(15) = 1	contingency events.
OPTIONS(15) = 2	both contingency labels and events.
OPTIONS(16)	number of elements to include in flow tables (60000 by default).
OPTIONS(17)	summary table maximum times for reporting the same element (5 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real VALUES(7)	Is an array of seven elements specifying tolerances and thresholds (input).	
	VALUES(1)	required MW mismatch tolerance for continuing (Newton solution convergence tolerance, TOLN, by default).
	VALUES(2)	percent of rating for reporting an element as overloaded (100.0 by default).
	VALUES(3)	summary table maximum import or export (>0.) (99999.0 by default).
	VALUES(4)	summary table minimum distribution factor magnitude (0.0 by default).
	VALUES(5)	minimum contingency case pre-shift flow change (0.0 by default).
	VALUES(6)	minimum contingency case distribution factor change (0.0 by default).
	VALUES(7)	study system generation shift (100.0 by default).
Character LABELS(8)*12	Is an array of eight elements specifying subsystem and interface labels (input).	
	LABELS(1)	study system label (no default allowed).
	LABELS(2)	opposing system label (no default allowed).
	LABELS(3)	interface label for optional interface limits summary table (blank for none) (blank by default).
	LABELS(4)	interface label for optional interface limits summary table (blank for none) (blank by default).
	LABELS(5)	interface label for optional interface limits summary table (blank for none) (blank by default).
	LABELS(6)	interface label for optional interface limits summary table (blank for none) (blank by default).
	LABELS(7)	interface label for optional interface limits summary table (blank for none) (blank by default).
	LABELS(8)	interface label for optional interface limits summary table (blank for none) (blank by default).
Character DFXFILE*260	Is the name of Distribution Factor Data File (input; no default allowed).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid OPTIONS value.
IERR = 2	invalid VALUES value.
IERR = 3	no Distribution Factor Data File specified.
IERR = 4	same label specified for study and opposing systems.
IERR = 5	generators are converted.
IERR = 6	buses in island(s) without a swing bus; use activity TREE.
IERR = 7	largest mismatch exceeds mismatch tolerance.
IERR = 8	error opening file DFXFILE.
IERR = 9	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX.
IERR = 10	file DFXFILE does not contain distribution factors; run DFAX with distribution factor option enabled.
IERR = 11	no monitored branches or interfaces defined.
IERR = 12	no subsystems defined.
IERR = 13	subsystem not found.
IERR = 14	specified subsystems overlap.
IERR = 15	no in-service buses in subsystem WORLD.
IERR = 16	no tie lines from study system.
IERR = 17	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.315 TPCH

Use this API to check the adjustment data associated with voltage or flow controlling transformers (activity TPCH). It **must** be referenced using one of the following procedures:

- One reference with APIOPT=1, followed by any number of either APIOPT=2 references and/or pairs of references with APIOPT=2 and then 3, followed by one reference with APIOPT=4.
- One reference with APIOPT=0. This automatically combines the processing of one APIOPT 1 reference, followed by one APIOPT 2 reference, followed by one APIOPT 3 reference if NTRN from the APIOPT 2 reference is greater than zero, followed by one APIOPT 4 reference.

Batch command syntax:

```
BAT_TPCH  SID  ALL  APIOPT  OPT  STATUS(1)  STATUS(2)  THRSH(1)
THRSH(2)
```

Python syntax:

```
ierr,ttyp,ntrn = tpch(sid, all, apiopt, opt, status, thrsh)
```

Fortran syntax:

```
CALL TPCHNW(SID, ALL, APIOPT, OPT, STATUS, THRSH, TTYP, NTRN, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; used only when APIOPT is 0 or 1; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; used only when APIOPT is 0 or 1; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Integer APIOPT	Is the mode of operation (input; 0 by default). APIOPT = 0 initialize, check, modify adjustment data, and terminate. APIOPT = 1 initialize for check and data modification. APIOPT = 2 check transformers. APIOPT = 3 modify adjustment data. APIOPT = 4 terminate transformer checks.
Integer OPT	Is the option to select the type of adjustment (input; used only when APIOPT is 0 or 2; 1 by default). = 1, tap step = 0. (V/Mvar) = 2, 0. < step < minimum (V/ Mvar) = 3, tap step > maximum (V/ Mvar) = 4, voltage band < 2*step (V) = 5, voltage band < minimum (V) = 6, voltage band > maximum (V) = 8, flow band < minimum (MW/Mvar) = 9, flow band > maximum (MW/Mvar)

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer STATUS(2) Is a two-element array specifying data modification options (input; used only when APIOPT is 0 or 3).

STATUS(1)	modify steps or voltage bands, when OPT = 4 (1 by default).
STATUS(1) = 1	modify steps.
STATUS(1) = 2	modify voltage bands.
STATUS(2)	used to indicate which transformers to change where OPTs 1, 2, 3, 8 or 9 (0 by default).
STATUS(2) = 0	change both types.
STATUS(2) = 1	change voltage controlling transformers when OPT = 1, 2 or 3; change MW controlling transformers when OPT = 8 or 9.
STATUS(2) = 2	change Mvar controlling transformers.

Real THRSH(2) Is a two-element array specifying data checking and modification parameters (input). The values are as follows.

THRSH(1)	threshold to use in initial test (used only when APIOPT is 0 or 2; defaults: 0.00625 when OPT is 2 or 3; 0.02 when OPT is 5 or 6; 5.0 when OPT is 8 or 9).
THRSH(2)	new tap step, voltage band, or flow band (used only when APIOPT is 0 or 3).

If OPT 1, THRSH(2), is the new tap step (0.00625 by default).

If OPT 2->3, THRSH(2), is the new tap step (THRSH(1) by default).

If OPT 5->6, THRSH(2), is the new voltage band (THRSH(1) by default).

If OPT 8->9, THRSH(2), is the new flow band (THRSH(1) by default).

Logical TTYP Is **returned** when APIOPT = 0, 2 or 3 and is TRUE. if both voltage (or MW if OPT = 8 or 9) and Mvar controlling transformers were found.

Integer NTRN Is **returned** when APIOPT = 0, 2 or 3 and is the number of transformers that need adjustment when applying the adjustment type indicated by OPT.

Integer IERR Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid SID value or subsystem SID is not defined.
IERR = 2	invalid ALL value.
IERR = 3	invalid APIOPT value.
IERR = 4	invalid OPT value.
IERR = 5	invalid STATUS value.
IERR = 6	invalid THRSH value.
IERR = 7	unexpected APIOPT value.
IERR = 8	no controlling transformers in case.
IERR = 9	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.316 TRANSACTION_DELETE

Use this API to delete a transaction event from PSS®E working memory.

Batch command syntax:

```
BAT_TRANSACTION_DELETE  EVENT
```

Python syntax:

```
ierr = transaction_delete(event)
```

Fortran syntax:

```
CALL TRANSACTION_DELETE(EVENT, IERR, NERR)
```

where:

Integer EVENT	Is the transaction event number.	
Integer IERR	Is the error code.	
	IERR = 0	no error occurred.
	IERR = 1	invalid transaction event number.
	IERR = 2	prerequisite requirements for API are not met.
Integer NERR	Is the number of error entries in PBUF02 (output).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.317 TRANSFORMER_PERCENT_UNITS

Use this API to set the transformer percent units option setting to either *MVA* or *current expressed as MVA* (activity OPTN).

Batch command syntax:

```
BAT_TRANSFORMER_PERCENT_UNITS  IVAL
```

Python syntax:

```
ierr = transformer_percent_units(ival)
```

or:

```
ierr,ival = transformer_percent_units()
```

Fortran syntax:

```
CALL TRANSFORMER_PERCENT_UNITS_API(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Is the value of the option setting (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 1 by default for input).	
	IVAL = 0	MVA.
	IVAL = 1	current expressed as MVA.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE = 0	set the option setting to the value specified in IVAL.
	IOCODE = 1	return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid IVAL value.
	IERR = 2	invalid IOCODE value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.318 TRANSMISSION_LINE_UNITS

Use this API to set the transmission line units option setting to either per unit or ohms (activity OPTN).

Batch command syntax:

```
BAT_TRANSMISSION_LINE_UNITS  IVAL
```

Python syntax:

```
ierr = transmission_line_units(ival)
```

or:

```
ierr,ival = transmission_line_units()
```

Fortran syntax:

```
CALL TRANSMISSION_LINE_UNITS_API(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Is the value of the option setting (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 1 by default for input).	
	IVAL = 0	per unit.
	IVAL = 1	ohms and microfarads.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE = 0	set the option setting to the value specified in IVAL.
	IOCODE = 1	return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid IVAL value.
	IERR = 2	invalid IOCODE value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.319 TREE

Use this API to check for the existence of in-service ac islands that do not contain a Type 3 (swing) bus (activity TREE). Following each successful call, it returns BUSES as the number of buses in a swingless island (0 for no more swingless islands). When a BUSES value of 0 is returned, no further calls are needed.

The API must be called once with APIOPT set to 1. If BUSES is returned as 0 (i.e., there are no swingless islands), no further calls are needed. Otherwise, if BUSES is greater than zero, it **must** be called one or more times with APIOPT set to 2 and OPTION set to indicate the disposition of the current swingless island. APIOPT 2 calls **are required** until either BUSES is returned as zero or an APIOPT 2 call is made with OPTION set to a negative value.

Batch command syntax:

```
BAT_TREE  APIOPT  OPTION
```

Python syntax:

```
ierr,buses = tree(apiopt, option)
```

Fortran syntax:

```
CALL TREEAPI(APIOPT, OPTION, BUSES, IERR)
```

where:

Integer APIOPT	Is the mode of operation (input; no default allowed).	
	APIOPT = 1	initialize and check for the presence of a swingless island.
	APIOPT = 2	process previously detected island as dictated by OPTION; then check for the presence of another swingless island.
Integer OPTION	Is the option for the handling of previously detected swingless island (input; used only when APIOPT is 2; -1 by default).	
	OPTION = <0	leave this island alone and terminate activity TREE.
	OPTION = 0	leave this island alone and check for another swingless island.
	OPTION = >0	disconnect this island, then check for another swingless island.
Integer BUSES	Is **returned** as the number of buses in this swingless island; 0 if no more swingless islands (output).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid APIOPT value.
	IERR = 2	unexpected APIOPT value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.320 TYSL

Use this API to run switching study network solutions (activity TYSL).

Batch command syntax:

```
BAT_TYSL  OPT
```

Python syntax:

```
ierr = tysl(opt)
```

Fortran syntax:

```
CALL TYSLAPI(OPT, IERR)
```

where:

Integer OPT	Is the flat start flag (input; 0 by default).	
	OPT = 0	use present voltage vector as starting point.
	OPT = 1	flat start.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid OPT value.
	IERR = 2	generators are not converted.
	IERR = 3	network is not ordered.
	IERR = 4	error reading the admittance matrix temporary file; one of the following occurred: • Admittance matrix temporary file is not open. • Reading matrix pointers: different sizes specified than were used in writing them. • Reading matrix terms: different sizes specified than were used in writing them.
	IERR = 5	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.321 USER

Use this API to run the user-written activity, subroutine USERAC (activity USER).

Batch command syntax:

```
BAT_USER
```

Python syntax:

```
ierr = user()
```

Fortran syntax:

```
CALL USERAPI(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.322 VAMM

Use this API to tabulate the vector absolute MW-mile report (activity ALOC).

Batch command syntax:

```
BAT_VAMM
```

Python syntax:

```
ierr = vamm()
```

Fortran syntax:

```
CALL VAMMAPI(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	no transactions defined.
	IERR = 4	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.323 VAMO

Use this API to tabulate the vector absolute MW-ohm report (activity ALOC).

Batch command syntax:

```
BAT_VAMO
```

Python syntax:

```
ierr = vamo()
```

Fortran syntax:

```
CALL VAMOAPI(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	no transactions defined.
	IERR = 4	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.324 VCHK

Use this API to tabulate those buses where voltage magnitude is outside a specified range (activity VCHK).

Batch command syntax:

```
BAT_VCHK  SID  ALL  VLO  VHI
```

Python syntax:

```
ierr = vchk(sid, all, vlo, vhi)
```

Fortran syntax:

```
CALL VCHKNW(SID, ALL, VLO, VHI, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Real VLO	Is the lower limit of the voltage band (input; 0.95 by default).
Real VHI	Is the upper limit of the voltage band (input; 1.05 by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.325 VOLTAGE_INPUT

Use this API to set the voltage input option setting to either per unit or kV (activity OPTN).

Batch command syntax:

```
BAT_VOLTAGE_INPUT  IVAL
```

Python syntax:

```
ierr = voltage_input(ival)
```

or:

```
ierr,ival = voltage_input()
```

Fortran syntax:

```
CALL VOLTAGE_INPUT_API(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Is the value of the option setting (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 1 by default for input).	
	IVAL = 0	per unit.
	IVAL = 1	kV.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE = 0	set the option setting to the value specified in IVAL.
	IOCODE = 1	return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid IVAL value.
	IERR = 2	invalid IOCODE value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.326 VOLTAGE_OUTPUT

Use this API to set the voltage output option setting to either per unit or kV (activity OPTN).

Batch command syntax:

```
BAT_VOLTAGE_OUTPUT  IVAL
```

Python syntax:

```
ierr = voltage_output(ival)
```

or:

```
ierr,ival = voltage_output()
```

Fortran syntax:

```
CALL VOLTAGE_OUTPUT_API(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Is the value of the option setting (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 1 by default for input).	
	IVAL = 0	per unit.
	IVAL = 1	kV.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE = 0	set the option setting to the value specified in IVAL.
	IOCODE = 1	return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid IVAL value.
	IERR = 2	invalid IOCODE value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.327 VSMO

Use this API to tabulate the vector MW-ohm report (activity ALOC).

Batch command syntax:

```
BAT_VSMO
```

Python syntax:

```
ierr = vsmo()
```

Fortran syntax:

```
CALL VSMOAPI(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	no transactions defined.
	IERR = 4	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.328 WRITERAWVERSION

Use this API to replicate the working case in the form of Power Flow Raw Data File compatible with PSS®E 15 or later.

Batch command syntax:

```
BAT_WRITERAWVERSION  'VERNUM'  OUT  OFILE
```

Python syntax:

```
ierr = writerawversion(vernum, out, ofile)
```

Fortran syntax:

```
CALL WRITERAWVERSION(VERNUM, OUT, OFILE, IERR)
```

where:

Character VERNUM*12	Is the value of OFILE version number (input; no default allowed). VERNUM = '15' or later. Example: If OFILE format is to be PSS®E 29.5.1: VERNUM = '29 '= '29.5' = '29.5.1'
Integer OUT	Is the flag for OUTPUT DEVICE CODE (input; 1 by default). OUT = 0 to OFILE. OUT = 1 to REPORT WINDOW.
Character OFILE*260	Is the filename of the Power Flow Raw Data file (Output; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid VERNUM value. IERR = 2 invalid OUT value. IERR = 3 Memory Allocation Error. IERR = 4 Error Reading Data. IERR = 5 Error saving Temporary 'sav' File. IERR = 6 Error opening Temporary 'sav' File. IERR = 7 Error deleting Temporary 'sav' File. IERR = 8 Error opening OFILE. IERR = 9 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.329 WRITESEQVERSION

This API is obsolete. It has been replaced by the API WRITESEQVERSION_2 and is implemented by a call to the WRITESEQVERSION_2 API routine.

Use this API to replicate the working case in the form of a Network Sequence Data File compatible with PSS®E 27 or later.

Batch command syntax:

```
BAT_WRITESEQVERSION  'VERNUM'  OUT  OFILE
```

Python syntax:

```
ierr = writeseqversion(vernum, out, ofile)
```

Fortran syntax:

```
CALL WRITESEQVERSION(VERNUM, OUT, OFILE, IERR)
```

where:

Character VERNUM*12	Is the value of OFILE version number (input; no default allowed). VERNUM = '27' or later. **Example:** If OFILE format is to be PSS®E 29.5.1: VERNUM = '29' = '29.5' = '29.5.1'
Integer OUT	Is the flag for OUTPUT DEVICE CODE (input; 1 by default). OUT = 0 to OFILE. OUT = 1 to REPORT WINDOW.
Character OFILE*260	Is the filename of the Network Sequence Data file (Output; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid VERNUM value. IERR = 2 invalid OUT value. IERR = 3 Error opening OFILE. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.330 WRITESEQVERSION_2

Use this API to replicate the working case in the form of a Network Sequence Data File compatible with PSS®E 27 or later.

Batch command syntax:

```
BAT_WRITESEQVERSION_2 'VERNUM'  OUT  OFILE GZOPTN
```

Python syntax:

```
ierr = writeseqversion_2 (vernum, out, ofile, gzoptn)
```

Fortran syntax:

```
CALL WRITESEQVERSION_2(VERNUM, OUT, OFILE, GZOPTN, IERR)
```

where:

Character VERNUM*12	Is the value of OFILE version number (input; no default allowed). VERNUM = '27' or later. **Example:** If OFILE format is to be PSS®E 29.5.1: VERNUM = '29' = '29.5' = '29.5.1'
Integer OUT	Is the flag for OUTPUT DEVICE CODE (input; 1 by default). OUT = 0 to OFILE. OUT = 1 to REPORT WINDOW.
Character OFILE*260	Is the filename of the Network Sequence Data file (Output; no default allowed).
Integer GZOPTN	Is generator reactance option (input; 0 by default) GZOPTN = 0, save subtransient reactance GZOPTN = 1, save transient reactance GZOPTN = 2, save synchronous reactance
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid VERNUM value. IERR = 2 invalid OUT value. IERR = 3 Error opening OFILE. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.331 WRITE_OPTIONS_FILE

Use this API to save the present values of the PSS®E-25 program option settings to the file psse.opt (activity OPTN).

Batch command syntax:

```
BAT_WRITE_OPTIONS_FILE
```

Python syntax:

```
ierr = write_options_file()
```

Fortran syntax:

```
CALL WRITE_OPTIONS_FILE_API(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.332 XEQV

Use this API to build an electrical equivalent of the portion of the working case **outside of** a specified subsystem of the working case.

Batch command syntax:

```
BAT_XEQV  SID  ALL  STATUS(1)...STATUS(6)  DVAL(1)  DVAL(2)
```

Python syntax:

```
ierr = xeqv(sid, all, status, dval)
```

Fortran syntax:

```
CALL XEQV(SID, ALL, STATUS, DVAL, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all buses.
	ALL = 0	process only buses outside of subsystem SID.
Integer STATUS(6)	Is an array of six elements (input). They are used to control equivalencing options. A value of 0 means the option is not enabled. A value of 1 means the option is enabled. They are as follows.	
	STATUS(1)	retain area boundary buses (0 by default).
	STATUS(2)	retain zone boundary buses (0 by default).
	STATUS(3)	suppress the equivalencing option for phase shifters (0 by default).
	STATUS(4)	retain controlled buses (0 by default).
	STATUS(5)	retain existing branches between retained buses (0 by default).
	STATUS(6)	net load and shunt at retained buses (0 by default).
Real DVAL(2)	Is an array of two elements (input). They are as follows.	
	DVAL(1)	minimum active or reactive power generation for retaining generator buses (0.0 by default).
	DVAL(2)	branch threshold tolerance (10.0 by default).
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	invalid STATUS value.
	IERR = 4	invalid DVAL value.
	IERR = 5	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.333 ZONE

This API is obsolete. It has been replaced by the API ZONE_2, and is implemented by a call to ZONE_2. Use this API to tabulate zone totals by zone (activity ZONE).

Batch command syntax:

```
BAT_ZONE  SID  ALL
```

Python syntax:

```
ierr = zone(sid, all)
```

Fortran syntax:

```
CALL ZONENW(SID, ALL, IERR)
```

where:

Integer SID	Is a valid zone subsystem identifier. Valid subsystem identifiers range from 0 to 11. Zone subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all zones* or specified subsystem flag (input; 1 by default). ALL = 1 process all zones. ALL = 0 process only zones in zone subsystem SID.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.334 ZONE_2

Use this API to tabulate zone totals by zone (activity ZONE).

Batch command syntax:

```
BAT_ZONE_2 SID  ALL  OPT
```

Python syntax:

```
ierr = zone_2(sid, all, opt)
```

Fortran syntax:

```
CALL ZONE_2(SID, ALL, OPT, IERR)
```

where:

Integer SID	Is a valid zone subsystem identifier. Valid subsystem identifiers range from 0 to 11. Zone subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all zones* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all zones.
	ALL = 0	process only zones in zone subsystem SID.
Integer OPT	Is the load and induction machine reporting option (input; 2 by default).	
	OPT = 1	tabulate loads, induction generators and induction motors connected to zone buses
	OPT = 2	tabulate loads, induction generators and induction motors assigned to the zone
	OPT = 3	tabulate three load and induction machine subtotals for each zone: • zone load and induction machines connected to zone buses • zone load and induction machines connected to non-zone buses • non-zone load and induction machines connected to zone buses
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	invalid OPT value.
	IERR = 4	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.335 ZONE_AREA

Use this API to tabulate zone totals by zone, along with subtotals by area.

Batch command syntax:

```
BAT_ZONE_AREA  SID  ALL
```

Python syntax:

```
ierr = zone_area(sid, all)
```

Fortran syntax:

```
CALL ZONE_AREA(SID, ALL, IERR)
```

where:

Integer SID	Is a valid zone subsystem identifier. Valid subsystem identifiers range from 0 to 11. Zone subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all zones* or specified subsystem flag (input; 1 by default). ALL = 1 process all zones. ALL = 0 process only zones in zone subsystem SID.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.336 ZONM

This API routine is obsolete. It has been replaced by the API routine ZONM_2, and is implemented by a call to the ZONM_2 API routine.

Use this API to reassign the buses and loads in a specified subsystem of the working case from their original zone to a designated zone (activity ZONM).

Batch command syntax:

```
BAT_ZONM  SID  ALL  STATUS(1)  STATUS(2)  IZNEW
```

Python syntax:

```
ierr = zonm(sid, all, status, iznew)
```

Fortran syntax:

```
CALL ZONMNW(SID, ALL, STATUS, IZNEW, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Integer STATUS(2)	Is an array of two elements (input). The values are as follows: STATUS(1) = 1 change zone assignments of buses to IZNEW; otherwise = 0 (0 by default). STATUS(2) = 1 change zone assignments of loads to IZNEW; otherwise = 0 (0 by default).
Integer IZNEW	Is the new zone to be assigned to specified buses and/or loads. IZNEW must be between 1 and 9999 (input; 1 by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 invalid STATUS value. IERR = 4 invalid new zone number. IERR = 5 zone table is full; cannot add zone to the working case. IERR = 6 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

1.337 ZONM_2

This API routine is the second release of the zone renumbering function. Use this API routine to reassign the buses, loads and/or induction machines in a specified subsystem of the working case from their original zone to a designated zone (activity ZONM).

Batch command syntax:

```
BAT_ZONM_2  SID  ALL  STATUS(1)  STATUS(2)  STATUS(3)  IZNEW
```

Python syntax:

```
ierr = zonm_2(sid, all, status, iznew)
```

Fortran syntax:

```
CALL ZONM_2(SID, ALL, STATUS, IZNEW, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Integer STATUS(3)	Is an array of three elements (input). The values are as follows: STATUS(1) = 1 change zone assignments of buses to IZNEW; 0 to leave zone assignments of buses unchanged (0 by default). STATUS(2) = 1 change zone assignments of loads to IZNEW; 0 to leave zone assignments of loads unchanged (0 by default). STATUS(3) = 1 change zone assignments of induction machines to IZNEW; 0 to leave zone assignments of induction machines unchanged (0 by default).
Integer IZNEW	Is the new zone to be assigned to specified buses, load, and/or induction machines. IZNEW must be between 1 and 9999 (input; 1 by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 invalid STATUS value. IERR = 4 invalid new zone number. IERR = 5 zone table is full; cannot add zone to the working case. IERR = 6 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

SIEMENS

Chapter 2

Power Flow Data Changing

This chapter describes the API routines used to add or modify power flow data.

For most categories of power flow data, there are a pair of API routines for specifying data; one of these contains "_DATA" as part of the API name, and the other contains "_CHNG" in its API name. The "_DATA" API routines may be used to add a new network element to the working case, or to modify the data of an existing network element. The "_CHNG" API routines may be used only to modify the data of an existing network element; specifying a network element that does not already exist in the working case to one of the "_CHNG" API routines is treated as a fatal error by the API routine.

Generally, when an API is adding a new element to the working case, data items that are defaulted are set to the value listed in the data item's description. When an API is modifying the data of an existing element, defaulted data items are unchanged. Note that, except for the SEQ_MUTUAL_DATA API, the SEQ_ APIs always modify data of an existing element.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.1 AREA_DATA

Use this API to modify area data in the working case (see *PSS®E Program Operation Manual*, Areas, Zones and Owners).

Batch command syntax:

```
BAT_AREA_DATA  I   INTGAR(1)   REALAR(1)   REALAR(2)   'ARNAME'
```

Python syntax:

```
ierr = area_data(i, intgar, realar, arname)
```

Fortran syntax:

```
CALL AREA_API_2(I, INTGAR, REALAR, ARNAME, IERR)
```

where:

Integer I	Is the area number between 1 and 9999 (input; no default allowed).	
Integer INTGAR(1)	Is an array of one element (input).	
	INTGAR(1)	ISW, area slack bus number (0 by default).
Real REALAR(2)	Is an array of two elements (input).	
	REALAR(1)	PDES, desired MW area interchange (0.0 by default).
	REALAR(2)	PTOL, desired MW interchange tolerance (0.1 * SBASE by default; for example, when system base MVA is 100.0, the default value of PTOL is 10 MW).
Character ARNAME*12	Is the area name (input; blank by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR > 0	fatal error occurred.
	IERR = 1	invalid area number.
	IERR = 2	area table is full.
	IERR = 3	prerequisite requirements for API are not met.
	IERR = -1	data error, warning and information messages; one or more of: • Area slack bus not found. • Slack bus is not in this area. • No plant at slack bus. • Type code of slack bus is not 2 or 3. • A Type 3 bus is in the area but it is not the area slack bus. • Interchange tolerance is not positive. • No equipment is assigned to area.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.2 BRANCH_CHNG

Use this API routine to modify the data of an existing non-transformer branch in the working case (see *PSS®E Program Operation Manual*, Non-Transformer Branch Data).

Batch command syntax:

```
BAT_BRANCH_CHNG  I  J  'CKT'  INTGAR(1)...INTGAR(6)
REALAR(1)...REALAR(15)
```

Python syntax:

```
ierr = branch_chng(i, j, ckt, intgar, realar)
```

Fortran syntax:

```
CALL BRANCH_CHNG(I, J, CKT, INTGAR, REALAR, IERR)
```

where:

Integer I	Is the bus number of from bus (input; no default allowed).
Integer J	Is the bus number of to bus (input; no default allowed).
Character CKT*2	Is the circuit identifier (input; ’1’ by default).
Integer INTGAR(6)	Is an array of six elements (input). See the BRANCH_DATA API routine for details.
Real REALAR(15)	Is an array of fifteen elements (input). See the BRANCH_DATA API routine for details.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no errors occurred
IERR > 0	fatal error occurred
IERR = 1	bus not found
IERR = 2	circuit identifier is more than two characters
IERR = 3	branch exists but it is a two-winding transformer
IERR = 9	branch not found
IERR = 10	prerequisite requirements for API are not met
IERR = -1	data error, warning and information messages; one or more of:

- branch reactance is 0.0
- line length is negative
- zero impedance line table is full; branch is treated as a normal line
- invalid metered end bus
- invalid branch status
- invalid owner number
- invalid ownership fraction
- no ownership data is specified
- multi-section line grouping deleted
- branch ownership table is full--only first n owners retained
- branch is no longer treated as a zero impedance line
- owner table is full.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.3 BRANCH_DATA

Use this API to modify the data of an existing non-transformer branch in the working case or to add a new non-transformer branch to the working case (see *PSS®E Program Operation Manual*, Non-Transformer Branch Data).

Batch command syntax:

```
BAT_BRANCH_DATA  I  J  'CKT'  INTGAR(1)...INTGAR(6)
REALAR(1)...REALAR(15)
```

Python syntax:

```
ierr = branch_data(i, j, ckt, intgar, realar)
```

Fortran syntax:

```
CALL BRANCH_API_2(I, J, CKT, INTGAR, REALAR, IERR)
```

where:

Integer I	Is the bus number of from bus (input; no default allowed).
Integer J	Is the bus number of to bus (input; no default allowed).
Character CKT*2	Is the circuit identifier (input; ’1’ by default).
Integer INTGAR(6)	Is an array of six elements (input).
	INTGAR(1) — ST, branch status (1 by default).
	INTGAR(2) — METBUS, metered end bus number (I or J) (I by default).
	INTGAR(3) — O1, first owner number (owner of bus I by default).
	INTGAR(4) — O2, second owner number (0 by default).
	INTGAR(5) — O3, third owner number (0 by default).
	INTGAR(6) — O4, fourth owner number (0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real REALAR(15) Is an array of fifteen elements (input).

REALAR(1)	R, nominal branch resistance (0.0 by default).
REALAR(2)	X, nominal branch reactance (THRSHZ by default; 0.0001 if THRSHZ = 0.0).
REALAR(3)	B, total line charging (0.0 by default).
REALAR(4)	RATEA, rating set A line rating (0.0 by default).
REALAR(5)	RATEB, rating set B line rating (0.0 by default).
REALAR(6)	RATEC, rating set C line rating (0.0 by default).
REALAR(7)	GI, real line shunt at bus I end. (0.0 by default)
REALAR(8)	BI, reactive line shunt at bus I end (0.0 by default).
REALAR(9)	GJ, real line shunt at bus J end (0.0 by default).
REALAR(10)	BJ, reactive line shunt at bus J end (0.0 by default).
REALAR(11)	LEN, line length (0.0 by default).
REALAR(12)	F1, first owner fraction (1.0 by default).
REALAR(13)	F2, second owner fraction (1.0 by default).
REALAR(14)	F3, third owner fraction (1.0 by default).
REALAR(15)	F4, fourth owner fraction (1.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no errors occurred
IERR > 0	fatal error occurred
IERR = 1	bus not found
IERR = 2	circuit identifier is more than two characters
IERR = 3	branch exists but it is a two-winding transformer
IERR = 4	branch from a bus to itself
IERR = 5	blank circuit identifier is invalid
IERR = 6	circuit identifier may not start with &
IERR = 7	branch table is full
IERR = 8	branch ownership table is full
IERR = 9	prerequisite requirements for API are not met
IERR = 10	prerequisite requirements for API are not met
IERR = -1	data error, warning and information messages; one or more of:

- branch reactance is 0.0
- line length is negative
- zero impedance line table is full; branch is treated as a normal line
- invalid metered end bus
- invalid branch status
- invalid owner number
- invalid ownership fraction
- no ownership data is specified
- multi-section line grouping deleted
- branch ownership table is full--only first n owners retained
- branch is no longer treated as a zero impedance line
- owner table is full.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.4 BUS_CHNG_3

This API routine is the third release of the bus data specification function. Use this API routine to modify the data of an existing bus in the working case (see *PSS®E Program Operation Manual*, Bus Data).

Batch command syntax:

```
BAT_BUS_CHNG_3  I  INTGAR(1)...INTGAR(4)  REALAR(1)...REALAR(7)
'NAME'
```

Python syntax:

```
ierr = bus_chng_3(i, intgar, realar, name)
```

Fortran syntax:

```
CALL BUS_CHNG_3(I, INTGAR, REALAR, NAME, IERR)
```

where:

Integer I	Is the bus number (input; no default allowed).
Integer INTGAR(4)	Is an array of four elements (input). See the BUS_DATA_3 API routine for details.
Real REALAR(7)	Is an array of seven elements (input). See the BUS_DATA_3 API routine for details.
Character NAME*12	Is the bus name (input; blank by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no errors occurred.
	IERR > 0 — fatal error occurred.
	IERR = 1 — invalid bus number.
	IERR = 3 — working case is not initialized.
	IERR = 4 — bus not found.
	IERR = 5 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IERR = -1 data error, warning and information messages; one or more of:

- Invalid base voltage.
- Invalid bus type code.
- Invalid area, zone or owner number.
- Bus name is more than 12 characters.
- New area number but bus is the area swing for its old area.
- Bus type code is 3 but is not the swing bus for its area.
- Area table is full.
- Zone table is full.
- Owner table is full.
- High voltage limit is not greater than the low voltage limit.
- Normal voltage limit is less restrictive than the emergency limit.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.5 BUS_DATA

This API routine is obsolete. It has been replaced by the API routine BUS_DATA_3, and is implemented by a call to the BUS_DATA_3 API routine followed by a call to the SHUNT_DATA API using a shunt identifier of '1'.

Use this API to modify the data of an existing bus in the working case or to add a new bus to the working case.

Batch command syntax:

```
BAT_BUS_DATA  I  INTGAR(1)...INTGAR(4)  REALAR(1)...REALAR(5)  'NAME'
```

Python syntax:

```
ierr = bus_data(i, intgar, realar, name)
```

Fortran syntax:

```
CALL BUS_API(I, INTGAR, REALAR, NAME, IERR, NERR)
```

where:

Integer I	Is the bus number.	
Integer INTGAR(4)	Is an array of four elements.	
	INTGAR(1)	IDE, bus type code.
	INTGAR(2)	AREA, area number.
	INTGAR(3)	ZONE, zone number.
	INTGAR(4)	OWNER, owner number.
Real REALAR(5)	Is an array of five elements.	
	REALAR(1)	GL, fixed shunt admittance (conductance).
	REALAR(2)	BL, fixed shunt admittance (susceptance).
	REALAR(3)	BASKV, bus base voltage in kV.
	REALAR(4)	VM, bus voltage magnitude in pu.
	REALAR(5)	VA, bus voltage phase angle.
Character NAME*12	Is the bus name.	
Integer IERR	Is the error code.	
	IERR = 0	no errors occurred.
	IERR > 0	fatal error occurred.
	IERR = 1	invalid bus number.
	IERR = 2	bus table is full.
	IERR = 3	working case is not initialized.
	IERR = 5	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	IERR = -1	data error, warning and information messages; one or more of: • Invalid base voltage. • Invalid bus type code. • Invalid area, zone or owner number. • Bus name is more than 12 characters. • New area number but bus is the area swing for its old area. • Bus type code is 3 but is not the swing bus for its area. • Area table is full. • Zone table is full. • Owner table is full.
Integer NERR	The number of entries in PBUF02 containing error messages.	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.6 BUS_DATA_2

This API routine is obsolete. It has been replaced by the API routine BUS_DATA_3, and is implemented by a call to the BUS_DATA_3 API routine.

This API is the second release of the bus data specification engine. Use this API to modify the data of an existing bus in the working case or to add a new bus to the working case (see *PSS®E Program Operation Manual*, Bus Data).

Batch command syntax:

```
BAT_BUS_DATA_2  I  INTGAR(1)...INTGAR(4)  REALAR(1)...REALAR(3)
'NAME'
```

Python syntax:

```
ierr = bus_data_2(i, intgar, realar, name)
```

Fortran syntax:

```
CALL BUS_API_2(I, INTGAR, REALAR, NAME, IERR)
```

where:

Integer I	Is the bus number (input; no default allowed).	
Integer INTGAR(4)	Is an array of four elements (input).	
	INTGAR(1)	IDE, bus type code (1 by default).
	INTGAR(2)	AREA, area number (1 by default).
	INTGAR(3)	ZONE, zone number (1 by default).
	INTGAR(4)	OWNER, owner number (1 by default).
Real REALAR(3)	Is an array of three elements (input).	
	REALAR(1)	BASKV, bus base voltage in kV (0.0 by default).
	REALAR(2)	VM, bus voltage magnitude in pu. (1.0 by default)
	REALAR(3)	VA, bus voltage phase angle (0.0 by default).
Character NAME*12	Is the bus name (input; blank by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no errors occurred.
	IERR > 0	fatal error occurred.
	IERR = 1	invalid bus number.
	IERR = 2	bus table is full.
	IERR = 3	working case is not initialized.
	IERR = 5	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IERR = -1 data error, warning and information messages; one or more of:

- Invalid base voltage.
- Invalid bus type code.
- Invalid area, zone or owner number.
- Bus name is more than 12 characters.
- New area number but bus is the area swing for its old area.
- Bus type code is 3 but is not the swing bus for its area.
- Area table is full.
- Zone table is full.
- Owner table is full.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.7 BUS_DATA_3

This API routine is the third release of the bus data specification function. Use this API routine to modify the data of an existing bus in the working case, or to add a new bus to the working case (see *PSS®E Program Operation Manual*, Bus Data).

Batch command syntax:

```
BAT_BUS_DATA_3  I  INTGAR(1)...INTGAR(4)  REALAR(1)...REALAR(7)
'NAME'
```

Python syntax:

```
ierr = bus_data_3(i, intgar, realar, name)
```

Fortran syntax:

```
CALL BUS_API_3(I, INTGAR, REALAR, NAME, IERR)
```

where:

Integer I	Is the bus number (input; no default allowed).	
Integer INTGAR(4)	Is an array of four elements (input).	
	INTGAR(1)	IDE, bus type code (1 by default).
	INTGAR(2)	AREA, area number (1 by default).
	INTGAR(3)	ZONE, zone number (1 by default).
	INTGAR(4)	OWNER, owner number (1 by default).
Real REALAR(7)	Is an array of seven elements (input).	
	REALAR(1)	BASKV, bus base voltage in kV (0.0 by default).
	REALAR(2)	VM, bus voltage magnitude in pu. (1.0 by default)
	REALAR(3)	VA, bus voltage phase angle (0.0 by default).
	REALAR(4)	NMAXV, normal bus voltage magnitude high limit in pu. (1.1 by default)
	REALAR(5)	NMINV, normal bus voltage magnitude low limit in pu. (0.9 by default)
	REALAR(6)	EMAXV, emergency bus voltage magnitude high limit in pu. (1.1 by default)
	REALAR(7)	EMINV, emergency bus voltage magnitude low limit in pu. (0.9 by default)
Character NAME*12	Is the bus name (input; blank by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no errors occurred.
	IERR > 0	fatal error occurred.
	IERR = 1	invalid bus number.
	IERR = 2	bus table is full.
	IERR = 3	working case is not initialized.
	IERR = 5	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IERR = -1 data error, warning and information messages; one or more of:

- Invalid base voltage.
- Invalid bus type code.
- Invalid area, zone or owner number.
- Bus name is more than 12 characters.
- New area number but bus is the area swing for its old area.
- Bus type code is 3 but is not the swing bus for its area.
- Area table is full.
- Zone table is full.
- Owner table is full.
- High voltage limit is not greater than the low voltage limit.
- Normal voltage limit is less restrictive than the emergency limit.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.8 BUS_NUMBER

Use this API to change the bus number of an existing bus in the working case.

Batch command syntax:

```
BAT_BUS_NUMBER  IBUS  NEWBUS
```

Python syntax:

```
ierr = bus_number(ibus, newbus)
```

Fortran syntax:

```
CALL BUS_NUMBER_API_2(IBUS, NEWBUS, IERR)
```

where:

Integer IBUS	Is the current bus number of the bus for which the number is to be changed (input; no default allowed).
Integer NEWBUS	Is the new bus number (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 — no errors occurred. IERR > 0 — fatal error occurred. IERR = 1 — bus not found. IERR = 2 — invalid new bus number. IERR = 3 — bus number NEWBUS is already assigned to another bus. IERR = 4 — prerequisite requirements for API are not met. IERR = -1 — data error, warning and information messages; one or more of: • None.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.9 CASE_TITLE_DATA

Use this API to change the two line case title in the working case (see *PSS®E Program Operation Manual*, Case Identification Data).

Batch command syntax:

```
BAT_CASE_TITLE_DATA  'LINE1'  'LINE2'
```

Python syntax:

```
ierr = case_title_data(line1, line2)
```

Fortran syntax:

```
CALL CASE_TITLE_API_2(LINE1, LINE2, IERR)
```

where:

Character LINE1*60	Is the first title line (input; unchanged by default).
Character LINE2*60	Is the second title line (input; unchanged by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no errors occurred.
	IERR > 0 — fatal error occurred.
	IERR = 1 — prerequisite requirements for API are not met.
	IERR = -1 — data error, warning and information messages; one or more of: • Title line exceeds 60 characters.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.10 FACTS_CHNG_2

This API routine is the second release of the FACTS device data specification function. Use this API routine to modify the data of an existing FACTS device in the working case (see *PSS®E Program Operation Manual*, FACTS Device Data).

Batch command syntax:

```
BAT_FACTS_CHNG_2  'NAME'  INTGAR(1)...INTGAR(6)
REALAR(1)...REALAR(13)  'MNAME'
```

Python syntax:

```
ierr,realaro = facts_chng_2(name, intgar, realari, mname)
```

Fortran syntax:

```
CALL FACTS_CHNG_2(NAME, INTGAR, REALAR, MNAME, IERR)
```

where:

Character*12 NAME	Is the non-blank name of the FACTS device for which data is to be specified (input; no default allowed).
Integer INTGAR(6)	Is an array of six elements (input). See the FACTS_DATA_2 API routine for details.
Real REALAR(19)	Is an array of nineteen elements (first thirteen are input values; last six are output values). See the FACTS_DATA_2 API routine for details.
Real REALARI(13)	Is an array of thirteen elements corresponding to input values REALAR(1) through REALAR(13) above.
Real REALARO(6)	Is an array of six elements corresponding to output values REALAR(14) through REALAR(19) above.
Character*12 MNAME	Is the name of the FACTS device that is the master device for this IPFC slave device (input; blank by default).
Integer IERR	Is the error code (output).

IERR = 0	no errors occurred
IERR > 0	error occurred
IERR = 2	no sending end bus is specified
IERR = 3	bus not found
IERR = 4	the same bus is specified for the sending and terminal end buses
IERR = 5	blank FACTS device name is invalid
IERR = 6	FACTS device not found
IERR = 7	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IERR = -1 warnings – one or more of:

- Invalid FACTS device control mode.
- Invalid owner number.
- Invalid series voltage reference code.
- Regulated bus not found.
- Shunt element specified as remotely regulating itself.
- Type code of regulated bus is not 1 or 2.
- Shunt element voltage setpoint must be positive.
- Terminal bus voltage limit must be positive.
- VTMX is less than VTMN.
- Series voltage limit must be positive.
- Shunt current limit must not be negative.
- Bridge real power transfer limit must not be negative.
- Series current limit must not be negative.
- Series reactance must be positive.
- Shunt current limit is not 0.0 for an IPFC device.
- Bridge real power transfer limit is not 0.0 for an IPFC device.
- Impedance setpoint has reactance of 0.0.
- RMPCT is not positive.
- Master IPFC FACTS device not found.
- Bus is dummy bus of a multi-section line.
- Multi-section line grouping deleted.
- Owner table is full.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.11 FACTS_DATA

This API is obsolete. It has been replaced by the API FACTS_DATA_2, and is implemented by a call to the FACTS_DATA_2 API.

Use this API to modify the data of an existing FACTS device in the working case or to add a new FACTS device to the working case.

Batch command syntax:

```
BAT_FACTS_DATA  FD  INTGAR(1)...INTGAR(5)  REALAR(1)...REALAR(13)
```

Python syntax:

```
ierr,realaro = facts_data(fd, intgar, realari)
```

Fortran syntax:

```
CALL FACTS_API(FD, INTGAR, REALAR, IERR, NERR)
```

where:

Integer FD	Is the number (which the API converts to a left-justified name) of the FACTS device for which data is to be specified (input; no default allowed).	
Integer INTGAR(5)	Is an array of five elements.	
	INTGAR(1)	I, sending end bus number.
	INTGAR(2)	J, terminal end bus number (0 for STATCON).
	INTGAR(3)	MODE, control mode.
	INTGAR(4)	OWNER, owner number.
	INTGAR(5)	VSREF, series voltage reference code for modes 4 and 7 (0 for sending end voltage; 1 for series current).
Real REALAR(19)	Is an array of nineteen elements.	
	REALAR(1)	PDES, desired real power at terminal bus.
	REALAR(2)	QDES, desired reactive power at terminal bus.
	REALAR(3)	VSET, sending bus voltage setpoint.
	REALAR(4)	SHMX, sending end maximum shunt current (MVA at rated voltage).
	REALAR(5)	TRMX, maximum bridge real power transfer.
	REALAR(6)	VTMN, minimum terminal bus voltage magnitude.
	REALAR(7)	VTMX, maximum terminal bus voltage magnitude.
	REALAR(8)	VSMX, maximum series voltage magnitude.
	REALAR(9)	IMX, maximum series current (MVA at rated voltage).
	REALAR(10)	LINX, pu reactance of temporary series element.
	REALAR(11)	RMPCT, percent of contributed reactive power.
	REALAR(12)	SET1, real component of constant Z (mode = 3), magnitude of constant V (mode = 4), or VD (mode = 7 or 8).
	REALAR(13)	SET2, imaginary component of constant Z (mode = 3), angle of constant V (mode = 4), or VQ (mode = 7 or 8).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

REALAR(14)	real component of present effective series Z (**returned**).
REALAR(15)	imaginary component of present effective series Z (**returned**).
REALAR(16)	magnitude of present series voltage, FDVREF reference (**returned**).
REALAR(17)	angle of present series voltage, FDVREF reference (**returned**).
REALAR(18)	real component of present series voltage, FDVREF reference (**returned**).
REALAR(19)	imaginary component of present series voltage, FDVREF reference (**returned**).

Real REALARI(13) Is an array of thirteen elements corresponding to input values REALAR(1) through REALAR(13) as described above.

Real REALARO(6) Is an array of six elements corresponding to output values REALAR(14) through REALAR(19) as described above.

Integer IERR Is the error code.

IERR = 0	no errors occurred
IERR > 0	error occurred
IERR = 1	FACTS device table is full
IERR = 2	no sending end bus is specified
IERR = 3	bus not found
IERR = 4	the same bus is specified for the sending and terminal end buses
IERR = 6	prerequisite requirements for API are not met.
IERR = 7	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IERR = -1 warnings – one or more of:

- Invalid FACTS device control model.
- Invalid owner number.
- Invalid series voltage reference code.
- Shunt element voltage setpoint must be positive.
- Terminal bus voltage limit must be positive.
- VTMX is less than VTMN.
- Series voltage limit must be positive.
- Shunt current limit must not be negative.
- Bridge real power transfer limit must not be negative.
- Series current limit must not be negative.
- Series reactance must be positive.
- Shunt current limit is not 0.0 for an IPFC device.
- Bridge real power transfer limit is not 0.0 for an IPFC device.
- Impedance setpoint has reactance of 0.0.
- RMPCT is not positive.
- Master IPFC FACTS device not found.
- Bus is dummy bus of a multi-section line.
- Multi-section line grouping deleted.
- Owner table is full.

Integer NERR The number of entries in PBUF02 containing error messages.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.12 FACTS_DATA_2

Use this API to run the second release of the FACTS device data specification engine. It modifies the data of an existing FACTS device in the working case, or to add a new FACTS device to the working case (see *PSS®E Program Operation Manual*, FACTS Device Data).

Batch command syntax:

```
BAT_FACTS_DATA_2  'NAME'  INTGAR(1)...INTGAR(6)
REALAR(1)...REALAR(13)  'MNAME'
```

Python syntax:

```
ierr,realaro = facts_data_2(name, intgar, realari, mname)
```

Fortran syntax:

```
CALL FACTS_API_2(NAME, INTGAR, REALAR, MNAME, IERR)
```

where:

Character*12 NAME	Is the non-blank name of the FACTS device for which data is to be specified (input; no default allowed).	
Integer INTGAR(6)	Is an array of six elements (input).	
	INTGAR(1)	I, sending end bus number (no default allowed).
	INTGAR(2)	J, terminal end bus number (0 for STATCON) (0 by default).
	INTGAR(3)	MODE, control mode (1 by default).
	INTGAR(4)	OWNER, owner number (1 by default).
	INTGAR(5)	VSREF, series voltage reference code for modes 4 and 7 (0 for sending end voltage; 1 for series current) (0 by default).
	INTGAR(6)	REMOT, remote bus number (0 for sending end bus) (0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real REALAR(19) Is an array of nineteen elements (first thirteen are input values; last six are output values).

REALAR(1)	PDES, desired real power at terminal bus (0.0 by default).
REALAR(2)	QDES, desired reactive power at terminal bus (0.0 by default).
REALAR(3)	VSET, sending bus voltage setpoint (1.0 by default).
REALAR(4)	SHMX, sending end maximum shunt current (MVA at rated voltage) (99.99 * SBASE by default).
REALAR(5)	TRMX, maximum bridge real power transfer (99.99 * SBASE by default).
REALAR(6)	VTMN, minimum terminal bus voltage magnitude (0.9 by default).
REALAR(7)	VTMX, maximum terminal bus voltage magnitude (1.1 by default).
REALAR(8)	VSMX, maximum series voltage magnitude (1.0 by default).
REALAR(9)	IMX, maximum series current (MVA at rated voltage) (0.0 by default).
REALAR(10)	LINX, pu reactance of temporary series element (0.05 by default).
REALAR(11)	RMPCT, percent of contributed reactive power (100.0 by default).
REALAR(12)	SET1, real component of constant Z (MODE = 3), magnitude of constant V (MODE = 4), or VD (MODE = 7 or 8) (0.0 by default).
REALAR(13)	SET2, imaginary component of constant Z (MODE = 3), angle of constant V (MODE = 4), or VQ (MODE = 7 or 8) (0.005 by default when MODE = 3; 0.0 otherwise).
REALAR(14)	real component of present effective series Z (**returned**).
REALAR(15)	imaginary component of present effective series Z (**returned**).
REALAR(16)	magnitude of present series voltage, FDVREF reference (**returned**).
REALAR(17)	angle of present series voltage, FDVREF reference (**returned**).
REALAR(18)	real component of present series voltage, FDVREF reference (**returned**).
REALAR(19)	imaginary component of present series voltage, FDVREF reference (**returned**).

Real REALARI(13) Is an array of thirteen elements corresponding to input values REALAR(1) through REALAR(13) as described above.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real REALARO(6)	Is an array of six elements corresponding to output values REALAR(14) through REALAR(19) as described above.
Character*12 MNAME	Is the name of the FACTS device that is the master device for this IPFC slave device (input; blank by default).
Integer IERR	Is the error code (output).

IERR = 0	no errors occurred
IERR > 0	error occurred
IERR = 1	FACTS device table is full
IERR = 2	no sending end bus is specified
IERR = 3	bus not found
IERR = 4	the same bus is specified for the sending and terminal end buses
IERR = 5	blank FACTS device name is invalid
IERR = 6	prerequisite requirements for API are not met.
IERR = 7	prerequisite requirements for API are not met.
IERR = -1	warnings – one or more of:

- Invalid FACTS device control mode.
- Invalid owner number.
- Invalid series voltage reference code.
- Regulated bus not found.
- Shunt element specified as remotely regulating itself.
- Type code of regulated bus is not 1 or 2.
- Shunt element voltage setpoint must be positive.
- Terminal bus voltage limit must be positive.
- VTMX is less than VTMN.
- Series voltage limit must be positive.
- Shunt current limit must not be negative.
- Bridge real power transfer limit must not be negative.
- Series current limit must not be negative.
- Series reactance must be positive.
- Shunt current limit is not 0.0 for an IPFC device.
- Bridge real power transfer limit is not 0.0 for an IPFC device.
- Impedance setpoint has reactance of 0.0.
- RMPCT is not positive.
- Master IPFC FACTS device not found.
- Bus is dummy bus of a multi-section line.
- Multi-section line grouping deleted.
- Owner table is full.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.13 GNE_CHNG

Use this API routine to modify the data of an existing GNE device in the working case (see *PSS®E Program Operation Manual*, GNE Device Data).

Batch command syntax:

```
BAT_GNE_CHNG  'NAME'  'MODEL'  NTERM  TRMNLS(1)...TRMNLS(NTERM)
INTGAR(1)...INTGAR(3)  NRELCN  RELCON(1)...RELCON(NRELCN)  NINTCN
INTCON(1)...INTCON(NINTCN)  NCHRCN  CHRCON(1)...CHRCON(NCHRCN)
```

Python syntax:

```
ierr = gne_chng(name, model, nterm, trmnls, intgar, nrelcn, relcon,
nintcn, intcon, nchrcn, chrcon)
```

Fortran syntax:

```
CALL GNE_CHNG(NAME, MODEL, NTERM, TRMNLS, INTGAR, NRELCN, RELCON,
NINTCN, INTCON, NCHRCN, CHRCON, IERR)
```

where:

Character*12 NAME	Is the non-blank name of the GNE device for which data is to be specified (input; no default allowed).
Character*16 MODEL	Is the non-blank name of the BOSL model to be used by this GNE device. The BOSL model file "MODEL.mac" must be available (input; no default allowed).
Integer NTERM	The number of terminals (i.e., buses) to which the model is connected. NTERM must be 1 for GNE-PQ and GNE-I models, and it must be either 1 or 2 for a GNE-Y model (input; no default allowed).
Integer TRMNLS(NTERM)	Is an array of NTERM elements containing the bus numbers of the buses to which this model is connected (input; no default allowed).
Integer INTGAR(3)	Is an array of three elements (input). See the GNE_DATA API routine for details.
Integer NRELCN	Is the number of real (floating point) constants required by the model (input; no default allowed).
Real RELCON(NRELCN)	Is an array of NRELCN elements containing real model constants (input; 0.0 by default).
Integer NINTCN	Is the number of integer constants required in calculating the inputs required by the model (input; no default allowed).
Integer INTCON(NINTCN)	Is an array of NINTCN elements containing integer model constants (usually bus numbers) (input; TRMNLS(1) by default).
Integer NCHRCN	Is the number of two-character identifiers required in calculating the inputs required by the model (input; no default allowed).
Character*2 CHRCON(NCHRCN)	Is an array of NCHRCN elements containing character model constants (e.g., circuit identifiers) (input; ’1’ by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no errors occurred
IERR > 0	error occurred
IERR = 1	blank GNE device name is invalid
IERR = 2	invalid number of terminal buses for this BOSL model
IERR = 3	bus not found
IERR = 4	model is not a GNE model
IERR = 5	number of constants specified differs from those required in the model's .mac file
IERR = 7	blank model name is invalid for a new GNE device instance
IERR = 8	error getting model options
IERR = 9	model has OutOpt not equal to 0 or 1
IERR = 10	model has OutFac not equal to 0
IERR = 11	GNE .mac file is not found or an error occurred while reading it
IERR = 12	GNE device not found
IERR = 13	prerequisite requirements for API are not met.
IERR = -1	warnings – one or more of: • Invalid GNE device status. • Invalid owner number. • Invalid non-metered end bus. • Bus specified in an integer constant is not found. • Owner table is full.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.14 GNE_DATA

Use this API routine to run the GNE device data specification function. It modifies the data of an existing GNE device in the working case, or adds a new GNE device to the working case (see *PSS®E Program Operation Manual*, GNE Device Data).

Batch command syntax:

```
BAT_GNE_DATA  'NAME'  'MODEL'  NTERM  TRMNLS(1)...TRMNLS(NTERM)
INTGAR(1)...INTGAR(3)  NRELCN  RELCON(1)...RELCON(NRELCN)  NINTCN
INTCON(1)...INTCON(NINTCN)  NCHRCN  CHRCON(1)...CHRCON(NCHRCN)
```

Python syntax:

```
ierr = gne_data(name, model, nterm, trmnls, intgar, nrelcn, relcon,
nintcn, intcon, nchrcn, chrcon)
```

Fortran syntax:

```
CALL GNE_API(NAME, MODEL, NTERM, TRMNLS, INTGAR, NRELCN, RELCON,
NINTCN, INTCON, NCHRCN, CHRCON, IERR)
```

where:

Character*12 NAME	Is the non-blank name of the GNE device for which data is to be specified (input; no default allowed).
Character*16 MODEL	Is the non-blank name of the BOSL model to be used by this GNE device. The BOSL model file "MODEL.mac" must be available (input; no default allowed).
Integer NTERM	The number of terminals (i.e., buses) to which the model is connected. NTERM must be 1 for GNE-PQ and GNE-I models, and it must be either 1 or 2 for a GNE-Y model (input; no default allowed).
Integer TRMNLS(NTERM)	Is an array of NTERM elements containing the bus numbers of the buses to which this model is connected (input; no default allowed).
Integer INTGAR(3)	Is an array of three elements (input). INTGAR(1) STATUS, model status (0 for out-of-service, 1 for in-service) (1 by default). INTGAR(2) OWNER, owner number (the owner to which bus TRMNLS(1) is assigned by default). INTGAR(3) NMETR, non-metered end bus number (TRMNLS(NTERM) by default).
Integer NRELCN	Is the number of real (floating point) constants required by the model (input; no default allowed).
Real RELCON(NRELCN)	Is an array of NRELCN elements containing real model constants (input; 0.0 by default).
Integer NINTCN	Is the number of integer constants required in calculating the inputs required by the model (input; no default allowed).
Integer INTCON(NINTCN)	Is an array of NINTCN elements containing integer model constants (usually bus numbers) (input; TRMNLS(1) by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer NCHRCN	Is the number of two-character identifiers required in calculating the inputs required by the model (input; no default allowed).
Character*2 CHRCON(NCHRCN)	Is an array of NCHRCN elements containing character model constants (e.g., circuit identifiers) (input; ’1’ by default).
Integer IERR	Is the error code (output).

IERR = 0	no errors occurred
IERR > 0	error occurred
IERR = 1	blank GNE device name is invalid
IERR = 2	invalid number of terminal buses for this BOSL model
IERR = 3	bus not found
IERR = 4	model is not a GNE model
IERR = 5	number of constants specified differs from those required in the model’s .mac file
IERR = 6	GNE device instance table is full
IERR = 7	blank model name is invalid for a new GNE device instance
IERR = 8	error getting model options
IERR = 9	model has OutOpt not equal to 0 or 1
IERR = 10	model has OutFac not equal to 0
IERR = 11	GNE .mac file is not found or an error occurred while reading it
IERR = 13	prerequisite requirements for API are not met.
IERR = -1	warnings – one or more of: • Invalid GNE device status. • Invalid owner number. • Invalid non-metered end bus. • Bus specified in an integer constant is not found. • Owner table is full.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.15 IMPEDANCE_CORRECTION_DATA

Use this API to modify the data of an existing transformer impedance correction table in the working case or to add a new transformer impedance correction table to the working case (see *PSS®E Program Operation Manual*, Transformer Impedance Correction Table Data).

Batch command syntax:

```
BAT_IMPEDANCE_CORRECTION_DATA  I   REALAR(1,1)  REALAR(2,1)
REALAR(1,2)  REALAR(2,2)...REALAR(1,11)  REALAR(2,11)
```

Python syntax:

```
ierr,intgar = impedance_correction_data(i, realar)
```

Fortran syntax:

```
CALL IMPEDANCE_CORRECTION_TAB_API_2(I, INTGAR, REALAR, IERR)
```

where:

Integer I	Is the table number (must be between 1 and the number of tables allowed at the present size level of PSS®E) (input; no default allowed).
Integer INTGAR(2)	Is an array of two elements (output). INTGAR(1) number of entries in table (**returned**). INTGAR(2) table type flag (0=ratio, 1=angle) (**returned**).
Real REALAR(2,11)	Is an array of 22 values dimensioned (2,11), where REALAR(1,i) specifies T_i, and REALAR(2,i) specifies F_i. The values are specified such that the left subscript varies more rapidly: (1,1), (2,1), (1,2), (2,2), … (1,11), (2,11). (Input; REALAR(1,1) = REALAR(2,1) =1.0 by default; all other values are 0.0 by default)
Integer IERR	Is the error code (output). IERR = 0 no errors occurred IERR > 0 fatal error occurred IERR = 1 invalid transformer impedance correction table number IERR = 3 abscissa values are not in ascending order IERR = 4 abscissa value is outside of the range of -180 to +180 IERR = 5 table has fewer than 2 points IERR = 6 prerequisite requirements for API are not met IERR = -1 data error, warning and information messages; one or more of: • point is beyond previous end of table • invalid scaling factor.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.16 INDUCTION_MACHINE_CHNG

Use this API routine to modify the data of an existing induction machine in the working case (see *PSS®E Program Operation Manual*, Induction Machine Data).

Batch command syntax:

```
BAT_INDUCTION_MACHINE_CHNG  I  'ID'  INTGAR(1)...INTGAR(9)
REALAR(1)...REALAR(23)
```

Python syntax:

```
ierr = induction_machine_chng(i, id, intgar, realar)
```

Fortran syntax:

```
CALL INDUCTION_MACHINE_CHNG(I, ID, INTGAR, REALAR, IERR, NERR)
```

where:

Integer I	Is the bus number (input; no default allowed).
Character ID*2	Is the machine identifier (input; ’1’ by default).
Integer INTGAR(9)	Is an array of nine elements (input). See the INDUCTION_MACHINE_DATA API routine for details.
Real REALAR(23)	Is an array of twenty-three elements (input). See the INDUCTION_MACHINE_DATA API routine for details.
Integer IERR	Is the error code.
	IERR = 0 — no errors occurred.
	IERR > 0 — fatal error occurred.
	IERR = 1 — bus not found.
	IERR = 2 — machine identifier is more than two characters.
	IERR = 5 — induction machine not found.
	IERR = 6 — prerequisite requirements for API are not met.
	IERR = -1 — data error, warning and information messages; one or more of:
	• invalid machine status.
	• invalid machine standard code.
	• invalid machine design code.
	• invalid area, zone or owner number.
	• invalid mechanical torque variation method code.
	• invalid base power code.
	• invalid power setpoint code.
	• machine base power is not positive.
	• machine rated voltage is negative.
	• machine rated voltage is ignored because the bus has no base voltage specified for it.
	• bus base voltage and machine rated voltage differ by more than 20%.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

- power setpoint exceeds the threshold determined from the standard code and the power setpoint code.
- invalid machine inertia.
- magnitude of load torque variation constant D is not between 0.2 and 5.0.
- the sign of load torque variation constant D is not consistent with that of the machine power setpoint.
- machine parameter is negative.
- XM is not positive.
- Rn or Xn exceeds 999.0.
- Rn and Xn are both 0.0.
- one of R1 and Xn is 999.0 and the other is not.
- E1 is greater than E2; interchanging E1 and E2.
- E1 and E2 are equal.
- SE1 is greater than SE2; interchanging SE1 and SE2.
- IA1 is greater than IA2; interchanging IA1 and IA2.
- IA1 and IA2 are equal.
- IA1 is less than 1.0.
- XAMULT must be between 0.0 and 1.0.
- area, zone or owner table is full.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.17 INDUCTION_MACHINE_DATA

Use this API routine to modify the data of an existing induction machine in the working case, or to add a new induction machine to the working case (see *PSS®E Program Operation Manual,* Induction Machine Data).

Batch command syntax:

```
BAT_INDUCTION_MACHINE_DATA  I  'ID'  INTGAR(1)...INTGAR(9)
REALAR(1)...REALAR(23)
```

Python syntax:

```
ierr = induction_machine_data(i, id, intgar, realar)
```

Fortran syntax:

```
CALL INDUCTION_MACHINE_API(I, ID, INTGAR, REALAR, IERR, NERR)
```

where:

Integer I	Is the bus number (input; no default allowed).	
Character ID*2	Is the machine identifier (input; ’1’ by default).	
Integer INTGAR(9)	Is an array of nine elements (input).	
	INTGAR(1)	STAT, induction machine status (1 by default).
	INTGAR(2)	SCODE, machine standard type (1 for NEMA, 2 for IEC) (1 by default).
	INTGAR(3)	DCODE, machine design code (0 for custom, 1 through 5 for MEMA designs A through E) (2 by default).
	INTGAR(4)	AREA, area number (area of bus I by default).
	INTGAR(5)	ZONE, zone number (zone of bus I by default).
	INTGAR(6)	OWNER, owner number (owner of bus I by default).
	INTGAR(7)	TCODE, type of mechanical torque variation (1 for simple power law, 2 for WECC model) (1 by default).
	INTGAR(8)	BCODE, base machine power code (1 for mechanical power output in MW, 2 for machine input apparent power in MVA) (1 by default).
	INTGAR(9)	PCODE, power setpoint code (1 for mechanical power output in MW, 2 for electrical power drawn in MW) (1 by default).
Real REALAR(23)	Is an array of twenty-three elements (input).	
	REALAR(1)	MBASE, machine base power (SBASE by default).
	REALAR(2)	RATEKV, machine rated voltage (kV line to line); 0.0 for bus base voltage (0.0 by default).
	REALAR(3)	PSET, machine power setpoint in MW (positive for a motor) (0.0 by default).
	REALAR(4)	H, machine inertia in pu on machine base power (1.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	REALAR(5)	A, load torque variation constant (1.0 by default).
	REALAR(6)	B, load torque variation constant (1.0 by default).
	REALAR(7)	D, load torque variation constant (1.0 by default).
	REALAR(8)	E, load torque variation constant (1.0 by default).
	REALAR(9)	RA, armature resistance (0.02494 by default).
	REALAR(10)	XA, armature leakage reactance (0.07482 by default).
	REALAR(11)	XM, magnetizing reactance (2.32786 by default).
	REALAR(12)	R1, first cage resistance (0.02078 by default).
	REALAR(13)	X1, first cage reactance (0.09145 by default).
	REALAR(14)	R2, second cage resistance (0.12471 by default).
	REALAR(15)	X2, second cage reactance (0.03326 by default).
	REALAR(16)	X3, third reactance (0.0 by default).
	REALAR(17)	E1, first voltage point on saturation curve (1.0 by default).
	REALAR(18)	SE1, saturation factor at E1 (0.04 by default).
	REALAR(19)	E2, second voltage point on saturation curve (1.2 by default).
	REALAR(20)	SE2, saturation factor at E2 (0.2 by default).
	REALAR(21)	IA1, starting stator current for applying stator leakage reactance saturation (1.0 by default).
	REALAR(22)	IA2, stator current at wiich the stator leakage reactance is fully saturated (5.0 by default).
	REALAR(23)	XAMULT, Xa multiplier at full leakage reactance saturation (1.0 by default).
Integer IERR	Is the error code.	
	IERR = 0	no errors occurred.
	IERR > 0	fatal error occurred.
	IERR = 1	bus not found.
	IERR = 2	machine identifier is more than two characters.
	IERR = 3	blank machine identifier is invalid.
	IERR = 4	induction machine table is full.
	IERR = 6	prerequisite requirements for API are not met.
	IERR = -1	data error, warning and information messages; one or more of: • invalid machine status. • invalid machine standard code. • invalid machine design code. • invalid area, zone or owner number. • invalid mechanical torque variation method code. • invalid base power code. • invalid power setpoint code.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

- machine base power is not positive.
- machine rated voltage is negative.
- machine rated voltage is ignored because the bus has no base voltage specified for it.
- bus base voltage and machine rated voltage differ by more than 20%.
- power setpoint exceeds the threshold determined from the standard code and the power setpoint code.
- invalid machine inertia.
- magnitude of load torque variation constant D is not between 0.2 and 5.0.
- the sign of load torque variation constant D is not consistent with that of the machine power setpoint.
- machine parameter is negative.
- XM is not positive.
- Rn or Xn exceeds 999.0.
- Rn and Xn are both 0.0.
- one of R1 and Xn is 999.0 and the other is not.
- E1 is greater than E2; interchanging E1 and E2.
- E1 and E2 are equal.
- SE1 is greater than SE2; interchanging SE1 and SE2.
- IA1 is greater than IA2; interchanging IA1 and IA2.
- IA1 and IA2 are equal.
- IA1 is less than 1.0.
- XAMULT must be between 0.0 and 1.0.
- area, zone or owner table is full.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.18 LOAD_CHNG_4

This API routine is the fourth release of the load data specification function. Use this API routine to modify the data of an existing load in the working case (see *PSS®E Program Operation Manual*, Load Data).

Batch command syntax:

```
BAT_LOAD_CHNG_4 I 'ID' INTGAR(1)...INTGAR(6) REALAR(1)...REALAR(6)
```

Python syntax:

```
ierr = load_chng_4(i, id, intgar, realar)
```

Fortran syntax:

```
CALL LOAD_CHNG_4(I, ID, INTGAR, REALAR, IERR)
```

where:

Integer I	Is the bus number (input; no default allowed).
Character ID*2	Is the load identifier (input; ’1’ by default).
Integer INTGAR(6)	Is an array of six elements (input). See the LOAD_DATA_4 API routine for details.
Real REALAR(6)	Is an array of six elements (input). See the LOAD_DATA_4 API routine for details.
Integer IERR	Is the error code (output). IERR = 0 — no errors occurred. IERR > 0 — fatal error occurred. IERR = 1 — bus not found. IERR = 2 — load identifier is more than two characters. IERR = 5 — load not found. IERR = 6 — prerequisite requirements for API are not met. IERR = -1 — data error, warning and information messages; one or more of: • invalid load status. • invalid area, zone or owner number. • area table is full. • zone table is full. • owner table is full. • invalid load scaling flag. • invalid interruptible load flag.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.19 LOAD_DATA

This API routine is obsolete. It has been replaced by the API routine LOAD_DATA_4, and is implemented by a call to the LOAD_DATA_4 API.

Use this API to modify the data of an existing load in the working case or to add a new load to the working case.

Batch command syntax:

```
BAT_LOAD_DATA  I  'ID'  INTGAR(1)...INTGAR(4)  REALAR(1)...REALAR(6)
```

Python syntax:

```
ierr = load_data(i, id, intgar, realar)
```

Fortran syntax:

```
CALL LOAD_API_2(I, ID, INTGAR, REALAR, IERR)
```

where:

Integer I	Is the bus number (input; no default allowed).	
Character ID*2	Is the load identifier (input; ’1’ by default).	
Integer INTGAR(4)	Is an array of four elements (input).	
	INTGAR(1)	STATUS, load status (1 by default).
	INTGAR(2)	AREA, area number (area of bus I by default).
	INTGAR(3)	ZONE, zone number (zone of bus I by default).
	INTGAR(4)	OWNER, owner number (owner of bus I by default).
Real REALAR(6)	Is an array of six elements (input).	
	REALAR(1)	PL, constant power active load (0.0 by default).
	REALAR(2)	QL, constant power reactive load (0.0 by default).
	REALAR(3)	IP, constant current active load (0.0 by default).
	REALAR(4)	IQ, constant current reactive load (0.0 by default).
	REALAR(5)	YP, constant admittance active load (0.0 by default).
	REALAR(6)	YQ, constant admittance reactive load (0.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no errors occurred
IERR > 0	fatal error occurred
IERR = 1	bus not found
IERR = 2	load identifier is more than two characters
IERR = 3	blank load identifier is invalid
IERR = 4	load table is full
IERR = 5	prerequisite requirements for API are not me
IERR = 6	prerequisite requirements for API are not met
IERR = -1	data error, warning and information messages; one or more of: • invalid load status. • invalid area, zone or owner number. • Area table is full. • Zone table is full. • Owner table is full.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.20 LOAD_DATA_3

This API routine is obsolete. It has been replaced by the API routine LOAD_DATA_4, and is implemented by a call to the LOAD_DATA_4 API.

Batch command syntax:

```
BAT_LOAD_DATA_3 I 'ID' INTGAR(1)...INTGAR(5) REALAR(1)...REALAR(6)
```

Python syntax:

```
ierr = load_data_3(i, id, intgar, realar)
```

Fortran syntax:

```
CALL LOAD_API_3(I, ID, INTGAR, REALAR, IERR)
```

where:

Integer I	Is the bus number (input; no default allowed).	
Character ID*2	Is the load identifier (input; ’1’ by default).	
Integer INTGAR(5)	Is an array of five elements (input).	
	INTGAR(1)	STATUS, load status (1 by default).
	INTGAR(2)	AREA, area number (area of bus I by default).
	INTGAR(3)	ZONE, zone number (zone of bus I by default).
	INTGAR(4)	OWNER, owner number (owner of bus I by default).
	INTGAR(5)	SCALE, load scaling flag (0 = fixed, non-conforming; 1 = scalable, conforming) (1 by default).
Real REALAR(6)	Is an array of six elements (input).	
	REALAR(1)	PL, constant power active load (0.0 by default).
	REALAR(2)	QL, constant power reactive load (0.0 by default).
	REALAR(3)	IP, constant current active load (0.0 by default).
	REALAR(4)	IQ, constant current reactive load (0.0 by default).
	REALAR(5)	YP, constant admittance active load (0.0 by default).
	REALAR(6)	YQ, constant admittance reactive load (0.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no errors occurred
	IERR > 0	fatal error occurred
	IERR = 1	bus not found
	IERR = 2	load identifier is more than two characters
	IERR = 3	blank load identifier is invalid
	IERR = 4	load table is full
	IERR = 5	prerequisite requirements for API are not met
	IERR = 6	prerequisite requirements for API are not met
	IERR = -1	data error, warning and information messages; one or more of: • invalid load status. • invalid area, zone or owner number. • area table is full. • zone table is full. • owner table is full. • invalid load scaling flag.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.21 LOAD_DATA_4

This API routine is the fourth release of the load data specification function. Use this API routine to modify the data of an existing load in the working case, or to add a new load to the working case (see *PSS®E Program Operation Manual*, Load Data).

Batch command syntax:

```
BAT_LOAD_DATA_4 I 'ID' INTGAR(1)...INTGAR(6) REALAR(1)...REALAR(6)
```

Python syntax:

```
ierr = load_data_4(i, id, intgar, realar)
```

Fortran syntax:

```
CALL LOAD_API_4(I, ID, INTGAR, REALAR, IERR)
```

where:

Integer I	Is the bus number (input; no default allowed).	
Character ID*2	Is the load identifier (input; ’1’ by default).	
Integer INTGAR(6)	Is an array of six elements (input).	
	INTGAR(1)	STATUS, load status (1 by default).
	INTGAR(2)	AREA, area number (area of bus I by default).
	INTGAR(3)	ZONE, zone number (zone of bus I by default).
	INTGAR(4)	OWNER, owner number (owner of bus I by default).
	INTGAR(5)	SCALE, load scaling flag (0 = fixed, non-conforming; 1 = scalable, conforming) (1 by default).
	INTGAR(6)	INTRPT, interruptible load flag (0 = non-interruptible; 1 = interruptible) (0 by default).
Real REALAR(6)	Is an array of six elements (input).	
	REALAR(1)	PL, constant power active load (0.0 by default).
	REALAR(2)	QL, constant power reactive load (0.0 by default).
	REALAR(3)	IP, constant current active load (0.0 by default).
	REALAR(4)	IQ, constant current reactive load (0.0 by default).
	REALAR(5)	YP, constant admittance active load (0.0 by default).
	REALAR(6)	YQ, constant admittance reactive load (0.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no errors occurred.
IERR > 0	fatal error occurred.
IERR = 1	bus not found.
IERR = 2	load identifier is more than two characters.
IERR = 3	blank load identifier is invalid.
IERR = 4	load table is full.
IERR = 6	prerequisite requirements for API are not met.
IERR = -1	data error, warning and information messages; one or more of: • invalid load status. • invalid area, zone or owner number. • area table is full. • zone table is full. • owner table is full. • invalid load scaling flag. • invalid interruptible load flag.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.22 LONG_TITLE_DATA

Use this API to change the sixteen line long title in the working case (see *PSS®E Program Operation Manual*, activity CHTI).

Batch command syntax:

```
BAT_LONG_TITLE_DATA  'LNGLNS(1)'...'LNGLNS(16)'
```

Python syntax:

```
ierr = long_title_data(lnglns)
```

Fortran syntax:

```
CALL LONG_TITLE_API_2(LNGLNS, IERR)
```

where:

Character LNGLNS(16)*72	Is an array of sixteen title lines (input; defaulted lines are unchanged).
Integer IERR	Is the error code (output).
	IERR = 0 — no errors occurred.
	IERR > 0 — fatal error occurred.
	IERR = 1 — prerequisite requirements for API are not met.
	IERR = -1 — data error, warning and information messages; one or more of: • Title line exceeds 72 characters.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.23 MACHINE_CHNG_2

This API routine is the second release of the machine data specification function. Use this API routine to modify the data of an existing machine in the working case (see *PSS®E Program Operation Manual*, Fixed Bus Shunt Data).

Batch command syntax:

```
BAT_MACHINE_CHNG_2  I  'ID'  INTGAR(1)...INTGAR(6)
REALAR(1)...REALAR(17)
```

Python syntax:

```
ierr = machine_chng_2(i, id, intgar, realar)
```

Fortran syntax:

```
CALL MACHINE_CHNG_2(I, ID, INTGAR, REALAR, IERR)
```

where:

Integer I	Is the bus number (input; no default allowed).
Character ID*2	Is the machine identifier (input; ’1’ by default).
Integer INTGAR(6)	Is an array of six elements (input). See the MACHINE_DATA_2 API routine for details.
Real REALAR(17)	Is an array of seventeen elements (input). See the MACHINE_DATA_2 API routine for details.
Integer IERR	Is the error code (output).
	IERR = 0 — no errors occurred.
	IERR > 0 — fatal error occurred.
	IERR = 1 — bus not found.
	IERR = 2 — machine identifier is more than two characters.
	IERR = 3 — no plant data at this bus.
	IERR = 7 — machine not found.
	IERR = 8 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IERR = -1 data error, warning and information messages; one or more of:

- invalid machine status
- invalid wind machine control mode
- invalid wind machine power factor
- invalid ZSORCE reactance
- invalid owner number
- invalid ownership fraction
- no ownership data is specified
- QMAX is less than QMIN
- PMAX is less than PMIN
- MBASE is not positive
- wind machine has negative active power
- machine is no longer treated as a wind machine
- wind machine table is full--retained as a non-wind machine
- machine ownership table is full--only first n owners retained
- owner table is full.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.24 MACHINE_DATA

This API is obsolete. It has been replaced by the API MACHINE_DATA_2, and is implemented by a call to the MACHINE_DATA_2 API.

Use this API to modify the data of an existing machine in the working case or to add a new machine to a plant bus in the working case.

Batch command syntax:

```
BAT_MACHINE_DATA  I  'ID'  INTGAR(1)...INTGAR(5)
REALAR(1)...REALAR(16)
```

Python syntax:

```
ierr = machine_data(i, id, intgar, realar)
```

Fortran syntax:

```
CALL MACHINE_API(I, ID, INTGAR, REALAR, IERR, NERR)
```

where:

Integer I	Is the bus number.	
Character ID*2	Is the machine identifier.	
Integer INTGAR(5)	Is an array of five elements.	
	INTGAR(1)	STAT, machine status.
	INTGAR(2)	O1, first owner number.
	INTGAR(3)	O2, second owner number.
	INTGAR(4)	O3, third owner number.
	INTGAR(5)	O4, fourth owner number.
Real REALAR(16)	Is an array of sixteen elements.	
	REALAR(1)	PG, machine active power output.
	REALAR(2)	QG, machine reactive power output.
	REALAR(3)	QT, machine reactive power upper limit.
	REALAR(4)	QB, machine reactive power lower limit.
	REALAR(5)	PT, machine active power upper limit.
	REALAR(6)	PB, machine active power lower limit.
	REALAR(7)	MBASE, machine MVA base.
	REALAR(8)	ZR, machine resistance.
	REALAR(9)	ZX, machine reactance.
	REALAR(10)	RT, step-up transformer resistance.
	REALAR(11)	XT, step-up transformer reactance.
	REALAR(12)	GTAP, step-up transformer tap ratio.
	REALAR(13)	F1, first owner fraction.
	REALAR(14)	F2, second owner fraction.
	REALAR(15)	F3, third owner fraction.
	REALAR(16)	F4, fourth owner fraction.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code.

IERR = 0	no errors occurred
IERR > 0	fatal error occurred
IERR = 1	bus not found
IERR = 2	machine identifier is more than two characters
IERR = 3	no plant data at bus
IERR = 4	blank machine identifier is invalid
IERR = 5	machine table full
IERR = 6	machine ownership table full
IERR = 7	prerequisite requirements for API are not met
IERR = 8	prerequisite requirements for API are not met
IERR = -1	data error, warning and information messages; one or more of: • invalid machine status • invalid ZSORCE reactance • invalid ZSORCE: (0.0,0.0) • invalid owner number • invalid ownership fraction • no ownership data is specified • QMAX is less than QMIN • PMAX is less than PMIN • MBASE is not positive • machine ownership table is full--only first n owners retained • owner table is full.

Integer NERR — The number of entries in PBUF02 containing error messages.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.25 MACHINE_DATA_2

Use this API to run the second release of the machine data specification engine. It modifies the data of an existing machine in the working case, or to add a new machine to a plant bus in the working case (see *PSS®E Program Operation Manual*, Fixed Bus Shunt Data).

Batch command syntax:

```
BAT_MACHINE_DATA_2  I  'ID'  INTGAR(1)...INTGAR(6)
REALAR(1)...REALAR(17)
```

Python syntax:

```
ierr = machine_data_2(i, id, intgar, realar)
```

Fortran syntax:

```
CALL MACHINE_API_2(I, ID, INTGAR, REALAR, IERR)
```

where:

Integer I	Is the bus number (input; no default allowed).	
Character ID*2	Is the machine identifier (input; ’1’ by default).	
Integer INTGAR(6)	Is an array of six elements (input).	
	INTGAR(1)	STAT, machine status (1 by default).
	INTGAR(2)	O1, first owner number (owner of bus I by default).
	INTGAR(3)	O2, second owner number (0 by default).
	INTGAR(4)	O3, third owner number (0 by default).
	INTGAR(5)	O4, fourth owner number (0 by default).
	INTGAR(6)	WMOD, wind machine reactive power limits mode (0 if this machine is not a wind machine) (0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real REALAR(17) Is an array of seventeen elements (input).

REALAR(1)	PG, machine active power output (0.0 by default).
REALAR(2)	QG, machine reactive power output (0.0 by default).
REALAR(3)	QT, machine reactive power upper limit (9999.0 by default).
REALAR(4)	QB, machine reactive power lower limit (-9999.0 by default).
REALAR(5)	PT, machine active power upper limit (9999.0 by default).
REALAR(6)	PB, machine active power lower limit (-9999.0 by default).
REALAR(7)	MBASE, machine MVA base (SBASE by default).
REALAR(8)	ZR, machine resistance (0.0 by default).
REALAR(9)	ZX, machine reactance (1.0 by default).
REALAR(10)	RT, step-up transformer resistance (0.0 by default).
REALAR(11)	XT, step-up transformer reactance (0.0 by default).
REALAR(12)	GTAP, step-up transformer tap ratio (1.0 by default).
REALAR(13)	F1, first owner fraction (1.0 by default).
REALAR(14)	F2, second owner fraction (1.0 by default).
REALAR(15)	F3, third owner fraction (1.0 by default).
REALAR(16)	F4, fourth owner fraction (1.0 by default).
REALAR(17)	WPF, wind machine power factor (1.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no errors occurred
IERR > 0	fatal error occurred
IERR = 1	bus not found
IERR = 2	machine identifier is more than two characters
IERR = 3	no plant data at this bus
IERR = 4	blank machine identifier is invalid
IERR = 5	machine table is full
IERR = 6	machine ownership table is full
IERR = 8	prerequisite requirements for API are not met
IERR = -1	data error, warning and information messages; one or more of:

- invalid machine status
- invalid wind machine control mode
- invalid wind machine power factor
- invalid ZSORCE: reactance
- invalid owner number
- invalid ownership fraction
- no ownership data is specified
- QMAX is less than QMIN
- PMAX is less than PMIN
- MBASE is not positive
- wind machine has negative active power
- machine is no longer treated as a wind machine
- wind machine table is full--retained as a non-wind machine
- machine ownership table is full--only first n owners retained
- owner table is full.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.26 MULTI_SECTION_LINE_DATA

Use this API to modify the definition of an existing multi-section line grouping in the working case or to add a new multi-section line grouping to the working case (see *PSS®E Program Operation Manual*, Multi-Section Line Grouping Data).

Batch command syntax:

```
BAT_MULTI_SECTION_LINE_DATA  I  J  'ID'  INTGAR(1)...INTGAR(10)
```

Python syntax:

```
ierr = multi_section_line_data(i, j, id, intgar)
```

Fortran syntax:

```
CALL MULTI_SECTION_LINE_API_2(I, J, ID, INTGAR, IERR)
```

where:

Integer I	Is the bus number of from bus (input; no default allowed).	
Integer J	Is the bus number of to bus (input; no default allowed).	
Character ID*2	Is a multi-section line identifier; ID(1:1) must be '&' (input; '&1' by default).	
Integer INTGAR(10)	Is an array of ten elements (input).	
	INTGAR(1)	METBUS, metered end bus number (must be I or J) (I by default).
	INTGAR(2-10)	DUM1 through DUM9 entries 2 through 10 are the bus numbers of the dummy buses encountered along this multi-section line, starting with the bus to which bus 'I' is connected and ending with the bus to which bus 'J' is connected; entries used are 2 through n+1, where n is the number of dummy buses; if n+1 is less than 10, entries n+2 through 10 must be set to 0. (No default is allowed for entry 2; entries 3 through 10 are 0 by default)
Integer IERR	Is the error code (output).	
	IERR = 0	no errors occurred
	IERR > 0	fatal error occurred
	IERR = 1	bus not found
	IERR = 2	multi-section line identifier is more than two characters
	IERR = 3	multi-section line identifier does not start with &
	IERR = 4	no dummy buses specified
	IERR = 5	line section not found
	IERR = 6	circuit is already a member of another multi-section line
	IERR = 7	bus appears more than once in the multi-section line path
	IERR = 8	dummy bus does not have exactly two branches connected to it
	IERR = 9	dummy bus is a converter bus of a dc line

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IERR = 10	a FACTS device is connected to a dummy bus
IERR = 11	line section table is full
IERR = 12	multi-section line table is full
IERR = 13	prerequisite requirements for API are not met
IERR = -1	data error, warning and information messages; one or more of: • invalid metered end bus • branch status and bus type code changes.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.27 MULTI_SECTION_LINE_EDIT

Use this API to modify the status and/or metered end designation of an existing multi-section line grouping in the working case (see *PSS®E Program Operation Manual*, Multi-Section Line Grouping Data).

Batch command syntax:

```
BAT_MULTI_SECTION_LINE_EDIT  I  J  'ID'  INTGAR(1)  INTGAR(2)
```

Python syntax:

```
ierr = multi_section_line_edit(i, j, id, intgar)
```

Fortran syntax:

```
CALL MULTI_SECTION_LINE_EDIT_API_2(I, J, ID, INTGAR, IERR)
```

where:

Integer I	Is the bus number of from bus (input; no default allowed).	
Integer J	Is the bus number of to bus (input; no default allowed).	
Character ID*2	Is the multi-section line identifier; ID(1:1) must be ’&’ (input; ’&1’ by default).	
Integer INTGAR(2)	Is an array of two elements (input).	
	INTGAR(1)	STAT, branch status (1 for in-service, 0 for out-of-service) (unchanged by default).
	INTGAR(2)	METBUS, metered end bus number (must be I or J) (unchanged by default).
Integer IERR	Is the error code (output).	
	IERR = 0	no errors occurred.
	IERR > 0	fatal error occurred.
	IERR = 1	bus not found.
	IERR = 2	multi-section line identifier is more than two characters.
	IERR = 3	multi-section line identifier does not start with &.
	IERR = 4	multi-section line not found.
	IERR = 5	prerequisite requirements for API are not met.
	IERR = -1	data error, warning and information messages; one or more of: • invalid metered end bus. • invalid branch status.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.28 MULTI_TERM_DC_BUS_DATA

Use this API to modify the data of a dc bus of an existing multi-terminal dc line in the working case or to add a new dc bus to a multi-terminal dc line in the working case (see *PSS®E Program Operation Manual*, Multi-Terminal DC Transmission Line Data). The dc line must have previously been specified via the MULTI_TERM_DC_LINE_DATA API.

Batch command syntax:

```
BAT_MULTI_TERM_DC_BUS_DATA  DC  IDC  INTGAR(1)...INTGAR(5)  REALAR(1)
'DCNAME'
```

Python syntax:

```
ierr,realaro = multi_term_dc_bus_data(dc, idc, intgar, realari,
dcname)
```

Fortran syntax:

```
CALL MULTI_TERMINAL_DC_BUS_API  (NUM, IDC, INTGAR, REALAR, DCNAME,
IERR, NERR)

CALL MULTI_TERMINAL_DC_BUS_API_I(NUM, IDC, INTGAR, REALAR, DCNAME,
IERR)

CALL MULTI_TERMINAL_DC_BUS_API_C(NAME, IDC, INTGAR, REALAR, DCNAME,
IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the multi-terminal dc line for which data for a dc bus is to be specified; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the multi-terminal dc line for which data for a dc bus is to be specified (input; no default allowed).
Character*12 NAME	Is the name of the multi-terminal dc line for which data for a dc bus is to be specified (input; no default allowed).
Integer IDC	Is the dc bus number (must be between 1 and the number of dc buses in multi-terminal dc line DC) (input; no default allowed).
Integer INTGAR(5)	Is an array of five elements (input). INTGAR(1) IB, converter bus number or zero (0 by default). INTGAR(2) AREA, area number (1 by default). INTGAR(3) ZONE, zone number (1 by default). INTGAR(4) OWNER, owner number (1 by default). INTGAR(5) IDC2, second dc bus number (0 by default).
Real REALAR(2)	Is an array of two elements (first is input value; second is output value). REALAR(1) RGRND, resistance to ground (0.0 by default). REALAR(2) dc voltage (kV) (**returned**).
Real REALARI(1)	Is an array of one element corresponding to input values REALAR(1) as described above.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real REALARO(1)	Is an array of one element corresponding to output value REALAR(2) as described above.
Character DCNAME*12	Is the dc bus name (input; blank by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no errors occurred.
	IERR > 0 — fatal error occurred.
	IERR = 1 — dc line not found.
	IERR = 2 — invalid dc bus number.
	IERR = 3 — prerequisite requirements for API are not met.
	IERR = -1 — data error, warning and information messages; one or more of:
	• data for lower numbered dc bus has not yet been entered.
	• converter bus not found.
	• bus is not a converter of this multi-terminal dc line.
	• this dc bus is not the first dc bus of any converter.
	• the first dc bus of some converter is set to this dc bus.
	• invalid second dc bus number.
	• this dc bus is being connected to a converter bus.
	• dc bus no longer connected to some other converter.
	• dc bus is being specified with no converter bus.
	• dc bus no longer connected to converter as its first dc bus.
	• dc bus no longer connected to converter as its second dc bus.
	• bus invalid as second dc bus because it is the first dc bus of another converter.
	• invalid area/zone/owner number.
	• area table is full.
	• zone table is full.
	• owner table is full.
Integer NERR	Is the number of entries in PBUF02 containing error messages.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.29 MULTI_TERM_DC_CONVR_DATA

Use this API to modify the data of a dc converter of an existing multi-terminal dc line in the working case or to add a new converter to a multi-terminal dc line in the working case (see *PSS®E Program Operation Manual*, Multi-Terminal DC Transmission Line Data). The dc line must have previously been specified via the MULTI_TERM_DC_LINE_DATA API.

Batch command syntax:

```
BAT_MULTI_TERM_DC_CONVR_DATA  DC  INTGAR(1)...INTGAR(4)
REALAR(1)...REALAR(13)
```

Python syntax:

```
ierr,realaro = multi_term_dc_convr_data(dc, intgar, realari)
```

Fortran syntax:

```
CALL MULTI_TERMINAL_DC_CONVERTER_API(NUM,INTGAR,REALAR,IERR,NERR)

CALL MULTI_TERM_DC_CONVERTER_API_I(NUM, INTGAR, REALAR, IERR)

CALL MULTI_TERM_DC_CONVERTER_API_C(NAME, INTGAR, REALAR, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the multi-terminal dc line for which data for a converter is to be specified; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the multi-terminal dc line for which data for a converter is to be specified (input; no default allowed).
Character*12 NAME	Is the name of the multi-terminal dc line for which data for a converter is to be specified (input; no default allowed).
Integer INTGAR(4)	Is the array of four elements (input).
	INTGAR(1) — IB, converter bus number (no default allowed).
	INTGAR(2) — N, number of bridges in series (1 by default).
	INTGAR(3) — CNVCOD, pole code (> or = 0 for a positive pole converter, < 0 for a negative pole converter) (0 by default).
	INTGAR(4) — VCCFLG, pole's voltage controlling converter flag (< or = 0 => no, >0 => yes) (0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real REALAR(17) Is an array of seventeen elements (first thirteen are input values; last four are output values).

REALAR(1)	SETVL, scheduled current (amps), power (MW) or voltage (kv) (0.0 by default).
REALAR(2)	ANGMN, minimum firing angle (0.0 by default).
REALAR(3)	ANGMX, maximum firing angle (90.0 by default).
REALAR(4)	TR, transformer ratio (1.0 by default).
REALAR(5)	RC, commutating resistance (0.0 by default).
REALAR(6)	XC, commutating reactance (0.0 by default).
REALAR(7)	EBAS, primary base ac voltage (1.0 by default).
REALAR(8)	TAP, tap setting (1.0 by default).
REALAR(9)	TPMX, maximum tap setting (1.5 by default).
REALAR(10)	TPMN, minimum tap setting (0.51 by default).
REALAR(11)	TSTP, tap step (0.00625 by default).
REALAR(12)	MARG, margin (pu) (0.0 by default).
REALAR(13)	DCPF, converter participation factor (1.0 by default).
REALAR(14)	firing angle (**returned**).
REALAR(15)	converter current (amps) (**returned**).
REALAR(16)	active power into converter (MW or kW) (**returned**).
REALAR(17)	reactive power into converter (MW or kW) (**returned**).

Real REALARI(13) Is an array of thirteen elements corresponding to input values REALAR(1) through REALAR(13) as described above.

Real REALARO(4) Is an array of four elements corresponding to output values REALAR(14) through REALAR(17) as described above.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code(output).

IERR = 0	no errors occurred.
IERR > 0	fatal error occurred.
IERR = 1	dc line not found.
IERR = 2	no converter bus is specified.
IERR = 3	converter bus not found.
IERR = 4	too many converters specified.
IERR = 5	prerequisite requirements for API are not met.
IERR = -1	data error, warning and information messages; one or more of: • converter bus is a dummy bus of a multi-section line. • multi-section line grouping deleted. • invalid number of bridges. • invalid base ac voltage. • converter maximum angle is less than its minimum angle. • converter maximum tap is less than its minimum tap. • invalid tap step. • dc voltage setpoint is not positive. • scheduled voltage is less than mode switch voltage. • this converter bus is now on the other pole; it is no longer the voltage controlling converter of its previous pole. • there is no voltage controlling converter on a pole. • this converter replaces another converter as its pole’s voltage controlling converter. • this converter must remain its pole's voltage controlling converter because it is the only converter on the pole.

Integer NERR — Is the number of entries in PBUF02 containing error messages.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.30 MULTI_TERM_DC_LINE_CHNG

Use this API routine to modify the control mode and mode switch voltage of an existing multi-terminal dc line in the working case (see *PSS®E Program Operation Manual*, Multi-Terminal DC Transmission Line Data). Converter data, dc bus data, and dc link data are specified via the MULTI_TERM_DC_CONVR_DATA, MULTI_TERM_DC_BUS_DATA, and MULTI_TERM_DC_LINK_DATA API routines respectively.

Batch command syntax:

```
BAT_MULTI_TERM_DC_LINE_CHNG  DC  INTGAR(1)  REALAR(1)
```

Python syntax:

```
ierr,intgaro = multi_term_dc_line_chng(dc, intgari, realar)
```

Fortran syntax:

```
CALL MULTI_TERMINAL_DC_LINE_CHNG_I(NUM, INTGAR, REALAR, IERR)

CALL MULTI_TERMINAL_DC_LINE_CHNG_C(NAME,INTGAR, REALAR, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the multi-terminal dc line for which data is to be specified; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the multi-terminal dc line for which data is to be specified (input; no default allowed).
Character*12 NAME	Is the name of the multi-terminal dc line for which data is to be specified (input; no default allowed).
Integer INTGAR(8)	Is an array of eight elements (first is input value; last seven are output values). See the MULTI_TERM_DC_LINE_DATA API routine for details.
Integer INTGARI(1)	Is an array of one element corresponding to input value INTGAR(1) as described above.
Integer INTGARO(7)	Is an array of seven elements corresponding to output values INTGAR(2) through INTGAR(8) as described above.
Real REALAR(1)	Is an array of one element (input). See the MULTI_TERM_DC_LINE_DATA API routine for details.
Integer IERR	Is the error code (output). IERR = 0 — no errors occurred IERR > 0 — fatal error occurred IERR = 2 — blank dc line name is invalid IERR = 3 — dc line not found IERR = 4 — prerequisite requirements for API are not met IERR = -1 — data error, warning and information messages; one or more of: • invalid dc line control mode.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.31 MULTI_TERM_DC_LINE_DATA

Use this API to modify the control mode and mode switch voltage of an existing multi-terminal dc line in the working case or to add a new multi-terminal dc line to the working case (see *PSS®E Program Operation Manual*, Multi-Terminal DC Transmission Line Data). Converter data, dc bus data, and dc link data are specified via the MULTI_TERM_DC_CONVR_DATA, MULTI_TERM_DC_BUS_DATA, and MULTI_TERM_DC_LINK_DATA APIs respectively.

Batch command syntax:

```
BAT_MULTI_TERM_DC_LINE_DATA  DC  INTGAR(1)  REALAR(1)
```

Python syntax:

```
ierr,intgaro = multi_term_dc_line_data(dc, intgari, realar)
```

Fortran syntax:

```
CALL MULTI_TERMINAL_DC_LINE_API  (NUM, INTGAR, REALAR, IERR, NERR)

CALL MULTI_TERMINAL_DC_LINE_API_I(NUM, INTGAR, REALAR, IERR)

CALL MULTI_TERMINAL_DC_LINE_API_C(NAME,INTGAR, REALAR, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the multi-terminal dc line for which data is to be specified; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the multi-terminal dc line for which data is to be specified (input; no default allowed).
Character*12 NAME	Is the name of the multi-terminal dc line for which data is to be specified (input; no default allowed).
Integer INTGAR(8)	Is an array of eight elements (first is input value; last seven are output values).
	INTGAR(1) — MDC, control mode (0 by default).
	INTGAR(2) — positive pole voltage controlling converter index (**returned**).
	INTGAR(3) — negative pole voltage controlling converter index, or zero if the negative pole is not being modeled (**returned**).
	INTGAR(4) — positive pole voltage controlling converter bus number (**returned**).
	INTGAR(5) — positive pole voltage controlling converter bus number (**returned**).
	INTGAR(6) — number of converters (**returned**).
	INTGAR(7) — number of dc buses (**returned**).
	INTGAR(8) — number of dc links (**returned**).
Integer INTGARI(1)	Is an array of one element corresponding to input value INTGAR(1) as described above.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer INTGARO(7)	Is an array of seven elements corresponding to output values INTGAR(2) through INTGAR(8) as described above.
Real REALAR(1)	Is an array of one element (input). REALAR(1) VCMOD, mode switch dc voltage (kV) (0.0 by default).
Integer IERR	Is the error code (output). IERR = 0 no errors occurred IERR > 0 fatal error occurred IERR = 1 multi-terminal dc line table is full IERR = 2 blank dc line name is invalid IERR = 4 prerequisite requirements for API are not met IERR = -1 data error, warning and information messages; one or more of: • invalid dc line control mode.
Integer NERR	Is the number of entries in PBUF02 containing error messages.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.32 MULTI_TERM_DC_LINK_DATA

Use this API to modify the data of a dc link of an existing multi-terminal dc line in the working case or to add a new dc link to a multi-terminal dc line in the working case (see *PSS®E Program Operation Manual*, Multi-Terminal DC Transmission Line Data). The dc line must have previously been specified via the MULTI_TERM_DC_LINE_DATA API.

Batch command syntax:

```
BAT_MULTI_TERM_DC_LINK_DATA  DC  IDC  JDC  'DCCKT'  INTGAR(1)
REALAR(1)  REALAR(2)
```

Python syntax:

```
ierr = multi_term_dc_link_data(dc,idc,jdc,dcckt,intgar,realar)
```

Fortran syntax:

```
CALL MULTI_TERMINAL_DC_LINK_API  (NUM, IDC, JDC, DCCKT, INTGAR,
REALAR, IERR, NERR)

CALL MULTI_TERMINAL_DC_LINK_API_I(NUM, IDC, JDC, DCCKT, INTGAR,
REALAR, IERR)

CALL MULTI_TERMINAL_DC_LINK_API_C(NAME, IDC, JDC, DCCKT, INTGAR,
REALAR, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the multi-terminal dc line for which data for a dc link is to be specified; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the multi-terminal dc line for which data for a dc link is to be specified (input; no default allowed).
Character*12 NAME	Is the name of the multi-terminal dc line for which data for a dc link is to be specified (input; no default allowed).
Integer IDC	Is the dc bus number of from bus (must be between 1 and the number of dc buses in multi-terminal dc line DC) (input; no default allowed).
Integer JDC	Is the dc bus number of to bus (must be between 1 and the number of dc buses in multi-terminal dc line DC) (input; no default allowed).
Character DCCKT*1	Is the circuit identifier (input; '1' by default).
Integer INTGAR(1)	Is an array of one element (input). INTGAR(1) METBUS, metered end dc bus number (IDC by default).
Real REALAR(2)	Is an array of two elements (input). REALAR(1) RDC, dc link resistance (0.0 by default). REALAR(2) LDC, dc link inductance (0.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no errors occurred.
	IERR > 0	fatal error occurred.
	IERR = 1	dc line not found.
	IERR = 2	invalid dc bus number.
	IERR = 3	circuit identifier is more than one character.
	IERR = 4	dc bus not present in this multi-terminal dc line.
	IERR = 5	branch from a bus to itself.
	IERR = 6	too many dc links in this multi-terminal dc line.
	IERR = 7	prerequisite requirements for API are not met.
	IERR = -1	data error, warning and information messages; one or more of: • invalid metered end bus. • invalid dc link resistance. • invalid dc link inductance.
Integer NERR	Is the number of entries in PBUF02 containing error messages.	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.33 OWNER_DATA

Use this API to modify owner data in the working case (see *PSS®E Program Operation Manual*, Interarea Transfer Data).

Batch command syntax:

```
BAT_OWNER_DATA  I  'OWNAME'
```

Python syntax:

```
ierr = owner_data(i, owname)
```

Fortran syntax:

```
CALL OWNER_API_2(I, OWNAME, IERR)
```

where:

Integer I	Is the owner number between 1 and 9999 (input; no default allowed).
Character OWNAME*12	Is the owner name (input; blank by default).
Integer IERR	Is the error code (output). IERR = 0 — no errors occurred. IERR > 0 — fatal error occurred. IERR = 1 — invalid owner number. IERR = 2 — owner table is full. IERR = 3 — prerequisite requirements for API are not met. IERR = -1 — data error, warning and information messages; one or more of: • no equipment is assigned to owner.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.34 PBUS_ADD_MOD

Use this API to add or modify transaction event participating bus data.

Batch command syntax:

```
BAT_PBUS_ADD_MOD  EVENT  IBUS  REALAR(1)  REALAR(2)
```

Python syntax:

```
ierr = pbus_add_mod(event, ibus, realar)
```

Fortran syntax:

```
CALL PBUS_ADD_MOD(EVENT, IBUS, REALAR, IERR, NERR)
```

where:

Integer EVENT	Is the transaction event number.
Integer IBUS	Is the number of the participating bus.
Real REALAR(2)	Is an array of two elements. REALAR(1) participating bus load value. REALAR(2) participating bus generation value.
Integer IERR	Is the error code. IERR = 0 no error occurred. IERR > 0 error occurred. IERR = 2 invalid transaction event number. IERR = 3 transaction event not found. IERR > 3 unable to add or modify participating bus data; error condition detected in data access routine. IERR = 10 prerequisite requirements for API are not met. IERR = -1 warnings: • None.
Integer NERR	Is the number of error entries in PBUF02 (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.35 PLANT_CHNG

Use this API routine to modify the data of an existing plant in the working case (see *PSS®E Program Operation Manual*, Fixed Bus Shunt Data).

Batch command syntax:

```
BAT_PLANT_CHNG  I  INTGAR(1)  REALAR(1)  REALAR(2)
```

Python syntax:

```
ierr = plant_chng(i, intgar, realar)
```

Fortran syntax:

```
CALL PLANT_CHNG(I, INTGAR, REALAR, IERR)
```

where:

Integer I	Is the bus number (input; no default allowed).
Integer INTGAR(1)	Is an array of one element (input). See the PLANT_DATA API routine for details.
Real REALAR(2)	Is an array of two elements (input). See the PLANT_DATA API routine for details.
Integer IERR	Is the error code (output). IERR = 0 — no errors occurred IEER > 0 — fatal error occurred IERR = 1 — bus not found IERR = 3 — no plant at bus IERR = 4 — prerequisite requirements for API are not met IERR = -1 — data error, warning and information messages; one or more of: • invalid scheduled voltage • regulated bus not found • plant specified as remotely regulating itself • type code of regulated bus is not 1 or 2 • RMPCT is not positive.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.36 PLANT_DATA

Use this API to modify the data of an existing plant in the working case, or to add a new plant to the working case (see *PSS®E Program Operation Manual*, Fixed Bus Shunt Data).

Batch command syntax:

```
BAT_PLANT_DATA  I  INTGAR(1)  REALAR(1)  REALAR(2)
```

Python syntax:

```
ierr = plant_data(i, intgar, realar)
```

Fortran syntax:

```
CALL PLANT_API_2(I, INTGAR, REALAR, IERR)
```

where:

Integer I	Is the bus number (input; no default allowed).	
Integer INTGAR(1)	Is an array of one element (input).	
	INTGAR(1)	IREG, remote bus number (0 to control voltage at bus I) (0 by default).
Real REALAR(2)	Is an array of two elements (input).	
	REALAR(1)	VS, scheduled voltage magnitude (1.0 by default).
	REALAR(2)	RMPCT, percent of contributed reactive power (100.0 by default).
Integer IERR	Is the error code (output).	
	IERR = 0	no errors occurred
	IEER > 0	fatal error occurred
	IERR = 1	bus not found
	IERR = 2	plant table is full
	IERR = 4	prerequisite requirements for API are not met
	IERR = -1	data error, warning and information messages; one or more of: • invalid scheduled voltage • regulated bus not found • plant specified as remotely regulating itself • type code of regulated bus is not 1 or 2 • RMPCT is not positive.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.37 SEQ_3_WIND_GROUNDING_DATA

This API routine is obsolete. It has been replaced by the API routine SEQ_THREE_WINDING_DATA_3, and is implemented by a call to the SEQ_THREE_WINDING_DATA_3 API routine.

Use this API to modify the zero sequence grounding data of an existing three-winding transformer in the working case (see *PSS®E Program Operation Manual*, Zero Sequence Transformer Data).

Batch command syntax:

```
BAT_SEQ_3_WIND_GROUNDING_DATA  I  J  K  'ICKT'  INTGAR(1)  REALAR(1)
REALAR(2)
```

Python syntax:

```
ierr = seq_3_wind_grounding_data(i, j, k, ickt, intgar, realar)
```

Fortran syntax:

```
CALL SEQ_3_WINDING_GROUNDING_API_2(I,J,K,ICKT,INTGAR,REALAR,IERR)
```

where:

Integer I	Is the bus number of one of the buses (input; no default allowed).	
Integer J	Is the bus number one of the other buses (input; no default allowed).	
Integer K	Is the bus number of the third bus (input; no default allowed).	
Character ICKT*2	Is the transformer circuit identifier (input; ’1’ by default).	
Integer INTGAR(1)	Is an array of one element (input; defaulted data items are unchanged).	
	INTGAR(1)	CC, connection code.
Real REALAR(2)	Is an array of at least two elements (input; defaulted data items are unchanged).	
	REALAR(1)	RG, zero sequence grounding resistance.
	REALAR(2)	XG, zero sequence grounding reactance.
Integer IERR	Is the error code.	
	IERR = 0	no errors occurred.
	IERR > 0	fatal error occurred.
	IERR = 1	bus not found.
	IERR = 2	circuit identifier is more than two characters.
	IERR = 3	three-winding transformer not found.
	IERR = 4	prerequisite requirements for API are not met.
	IERR = -1	data error, warning and information messages; one or more of: • Invalid winding connection code. • Connection code is such that more than one winding requires a grounding impedance.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.38 SEQ_3_WIND_WINDING_DATA

This API routine is obsolete. It has been replaced by the API routine SEQ_THREE_WINDING_DATA_3, and is implemented by a call to the SEQ_THREE_WINDING_DATA_3 API routine.

Use this API to modify the zero sequence impedance data of one winding of an existing three-winding transformer in the working case (see *PSS®E Program Operation Manual*, Zero Sequence Transformer Data).

Batch command syntax:

```
BAT_SEQ_3_WIND_WINDING_DATA  I  J  K  'ICKT'  WARG  REALAR(1)
REALAR(2)
```

Python syntax:

```
ierr = seq_3_wind_winding_data(i, j, k, ickt, warg, realar)
```

Fortran syntax:

```
CALL SEQ_THREE_WINDING_WINDING_API_2(I,J,K,ICKT,WARG,REALAR,IERR)
```

where:

Integer I	Is the bus number of one of the buses (input; no default allowed).
Integer J	Is the bus number one of the other buses (input; no default allowed).
Integer K	Is the bus number of the third bus (input; no default allowed).
Character ICKT*2	Is the transformer circuit identifier (input; ’1’ by default).
Integer WARG	Is the winding number (1, 2 or 3) (input; no default allowed).
Real REALAR(2)	Is an array of at least two elements (input; defaulted data items are unchanged). REALAR(1) Ri, zero sequence winding resistance. REALAR(2) Xi, zero sequence winding reactance.
Integer IERR	Is the error code (output). IERR = 0 no errors occurred. IERR > 0 fatal error occurred. IERR = 1 bus not found. IERR = 2 circuit identifier is more than two characters. IERR = 3 three-winding transformer not found. IERR = 4 invalid winding number. IERR = 5 prerequisite requirements for API are not met. IERR = -1 data error, warning and information messages; one or more of: • None.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.39 SEQ_BRANCH_DATA

This API is obsolete. It has been replaced by the API routine SEQ_BRANCH_DATA_3, and is implemented by a call to the SEQ_BRANCH_DATA_3 API routine.

Use this API to modify the zero sequence data of an existing non-transformer branch in the working case (see *PSS®E Program Operation Manual*, Zero Sequence Non-Transformer Branch Data).

Batch command syntax:

```
BAT_SEQ_BRANCH_DATA  I  J  'ICKT'  REALAR(1)...REALAR(7)
```

Python syntax:

```
ierr = seq_branch_data(i, j, ickt, realar)
```

Fortran syntax:

```
CALL SEQ_BRANCH_API_2(I, J, ICKT, REALAR, IERR)
```

where:

Integer I	Is the bus number of from bus (input; no default allowed).	
Integer J	Is the bus number of to bus (input; no default allowed).	
Character ICKT*2	Is the circuit identifier (input; ’1’ by default).	
Real REALAR(7)	Is an array of at least seven elements (input; defaulted data items are unchanged).	
	REALAR(1)	RLINZ, zero sequence branch resistance.
	REALAR(2)	XLINZ, zero sequence branch reactance.
	REALAR(3)	BCHZ, zero sequence total line charging.
	REALAR(4)	GI, zero sequence active line shunt at bus I end.
	REALAR(5)	BI, zero sequence reactive line shunt at bus I end.
	REALAR(6)	GJ, zero sequence active line shunt at bus J end.
	REALAR(7)	BJ, zero sequence reactive line shunt at bus J end.
Integer IERR	Is the error code (output).	
	IERR = 0	no errors occurred
	IERR > 0	fatal error occurred
	IERR = 1	bus not found
	IERR = 2	circuit identifier is more than two characters
	IERR = 3	circuit identifier may not start with &
	IERR = 4	branch not found
	IERR = 5	branch exists but it is a two-winding transformer
	IERR = 6	prerequisite requirements for API are not met
	IERR = -1	data error, warning and information messages; one or more of: • branch is treated as a zero impedance line but its zero sequence impedance is non-zero.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.40 SEQ_BRANCH_DATA_3

Use this routine API to modify the zero sequence data of an existing non-transformer branch in the working case (see *PSS®E Program Operation Manual*, Zero Sequence Non-Transformer Branch Data).

Batch command syntax:

```
BAT_SEQ_BRANCH_DATA_3 I  J  'ICKT' INTGAR(1) REALAR(1)...REALAR(8)
```

Python syntax:

```
ierr = seq_branch_data_3(i, j, ickt, intgar, realar)
```

Fortran syntax:

```
CALL SEQ_BRANCH_API_3(I, J, ICKT, INTGAR, REALAR, IERR)
```

where:

Integer I	Is the bus number of from bus (input; no default allowed).	
Integer J	Is the bus number of to bus (input; no default allowed).	
Character ICKT*2	Is the circuit identifier (input; ’1’ by default).	
Integer INTGAR(1)	Is an array of at least one element (input; defaulted data itemsare unchanged).	
	INTGAR(1)	MOV-protected series capacitor flag (0 for not an MOV-protected branch, 1 for an MOV-protected series capacitor with MOV protection enabled, 2 for an MOV-protected series capacitor with MOV protection disabled, 3 for spark gap protection enabled).
Real REALAR(8)	Is an array of at least eight elements (input; defaulted data items are unchanged).	
	REALAR(1)	RLINZ, zero sequence branch resistance.
	REALAR(2)	XLINZ, zero sequence branch reactance.
	REALAR(3)	BCHZ, zero sequence total line charging.
	REALAR(4)	GI, zero sequence active line shunt at bus I end.
	REALAR(5)	BI, zero sequence reactive line shunt at bus I end.
	REALAR(6)	GJ, zero sequence active line shunt at bus J end.
	REALAR(7)	BJ, zero sequence reactive line shunt at bus J end.
	REALAR(8)	IMOV, MOV-protected current level in kA.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no errors occurred
IERR > 0	fatal error occurred
IERR = 1	bus not found
IERR = 2	circuit identifier is more than two characters
IERR = 3	circuit identifier may not start with &
IERR = 4	branch not found
IERR = 5	branch exists but it is a two-winding transformer
IERR = 6	prerequisite requirements for API are not met
IERR = -1	data error, warning and information messages; one or more of: • branch is treated as a zero impedance line but its zero sequence impedance is non-zero. • invalid MOV-protected series capacitor flag. • inappropriate zero or positive sequence data is specified for a branch that is indicated as an MOV-protected series capacitor. • MOV flag is zero but the MOV-protected current level is non-zero

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.41 SEQ_BUS_DATA

This API routine is obsolete. Bus load sequence data is now accessible as load sequence data viat the API routine SEQ_LOAD_DATA.

Use this API routine to modify the sequence data of an existing bus in the working case (see *PSS®E Program Operation Manual*, Negative Sequence Shunt Load Data and Zero Sequence Shunt Load Data).

Batch command syntax:

```
BAT_SEQ_BUS_DATA  I  REALAR(1)...REALAR(4)
```

Python syntax:

```
ierr = seq_bus_data(i, realar)
```

Fortran syntax:

```
CALL SEQ_BUS_API_2(I, REALAR, IERR)
```

where:

Integer I	Is the bus number (input; no default allowed).	
Real REALAR(4)	Is an array of four elements (input; defaulted data items are unchanged).	
	REALAR(1)	GNEG, negative sequence shunt admittance active component.
	REALAR(2)	BNEG, negative sequence shunt admittance reactive component.
	REALAR(3)	GZERO, zero sequence shunt admittance active component.
	REALAR(4)	BZERO, zero sequence shunt admittance reactive component.
Integer IERR	Is the error code (output).	
	IERR = 0	no errors occurred.
	IERR > 0	fatal error occurred.
	IERR = 1	the SEQ_BUS_DATA API routine is obsolete.
	IERR = 2	prerequisite requirements for API are not met.
	IERR = -1	data error, warning and information messages; one or more of: • None.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.42 SEQ_FIXED_SHUNT_DATA

Use this API to modify the zero sequence data of an existing fixed bus shunt in the working case (see *PSS®E Program Operation Manual*, Zero Sequence Fixed Shunt Data).

Batch command syntax:

```
BAT_SEQ_FIXED_SHUNT_DATA  I  'ID'  REALAR(1)  REALAR(2)
```

Python syntax:

```
ierr = seq_fixed_shunt_data(i, id, realar)
```

Fortran syntax:

```
CALL SEQ_FIXED_SHUNT_API(I, ID, REALAR, IERR)
```

where:

Integer I	Is the bus number (input; no default allowed).	
Character ID*2	Is the shunt identifier (input; ’1’ by default).	
Real REALAR(2)	Is an array of two elements (input; defaulted data items are unchanged).	
	REALAR(1)	GSZERO, active power component of fixed shunt zero sequence admittance (conductance).
	REALAR(2)	BSZERO, reactive power component of fixed shunt zero sequence admittance (susceptance).
Integer IERR	Is the error code (output).	
	IERR = 0	no errors occurred.
	IERR > 0	fatal error occurred.
	IERR = 1	bus not found.
	IERR = 2	shunt identifier is more than two characters.
	IERR = 3	fixed shunt not found.
	IERR = 4	prerequisite requirements for API are not met.
	IERR = -1	data error, warning and information messages; one or more of: • Admittance components with different signs in the positive and zero sequences.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.43 SEQ_INDUCTION_MACH_DATA

Use this API routine to modify the sequence data of an existing induction machine in the working case (see *PSS®E Program Operation Manual*, Induction Machine Impedance Data).

Batch command syntax:

```
BAT_SEQ_INDUCTION_MACH_DATA  I  'ID'  INTGAR(1)  INTGAR(2)
REALAR(1)...REALAR(6)
```

Python syntax:

```
ierr = seq_induction_mach_data(i, id, intgar, realar)
```

Fortran syntax:

```
CALL SEQ_INDUCTION_MACH_API(I, ID, INTGAR, REALAR, IERR)
```

where:

Integer I	Is the bus number (input; no default allowed).	
Character ID*2	Is the machine identifier (input; ’1’ by default).	
Integer INTGAR(2)	Is an array of two elements (input; defaulted data items are unchanged).	
	INTGAR(1)	CZG, code indicating the units in which the zero sequence grounding impedance is specified: 1 for per unit 2 for ohms
	INTGAR(2)	CGR, grounding code (0 = ungrounded; 1 = grounded) (0 by default).
Real REALAR(6)	Is an array of six elements (input; defaulted data items are unchanged).	
	REALAR(1)	ILK2IR, ratio of positive sequence locked rotor current to rated current.
	REALAR(2)	ROVERX, ratio of positive sequence resistance to reactance.
	REALAR(3)	R0, machine zero sequence fault analysis resistance.
	REALAR(4)	X0, machine zero sequence fault analysis reactance.
	REALAR(5)	RG, machine zero sequence grounding resistance.
	REALAR(6)	XG, machine zero sequence grounding reactance.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no errors occurred.
IERR > 0	fatal error occurred.
IERR = 1	bus not found.
IERR = 2	machine identifier is more than two characters.
IERR = 3	induction machine not found.
IERR = 4	prerequisite requirements for API are not met.
IERR = -1	data error, warning and information messages; one or more of: • Invalid CZG data value. • invalid grounding code. • CZG is 2 but the bus has no base voltage specified.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.44 SEQ_LOAD_DATA

Use this API routine to modify the sequence data of an existing load in the working case (see *PSS®E Program Operation Manual*, Load Data).

Batch command syntax:

```
BAT_SEQ_LOAD_DATA I  'ID'  INTGAR(1)  REALAR(1)...REALAR(4)
```

Python syntax:

```
ierr = seq_load_data(i, id, intgar, realar)
```

Fortran syntax:

```
CALL SEQ_LOAD_API(I, ID, INTGAR, REALAR, IERR)
```

where:

Integer I	Is the bus number (input; no default allowed).	
Character ID*2	Is the load identifier (input; '1' by default).	
Integer INTGAR(1)	Is an array of one element (input).	
	INTGAR(1)	CGR, load grounding code (0 = ungrounded; 1 = grounded) (0 by default).
Real REALAR(4)	Is an array of four elements (input).	
	REALAR(1)	GNEG, negative sequence active power load component; entered as admittance in MW at 1.0 per unit voltage (0.0 by default).
	REALAR(2)	BNEG, negative sequence reactive power load component; BNEG is negative for an inductive load and positive for a capacitive load; entered as admittance in MVar at 1.0 per unit voltage (0.0 by default).
	REALAR(3)	GZERO, zero sequence active power load component; entered as admittance in MW at 1.0 per unit voltage (0.0 by default).
	REALAR(4)	BZERO, zero sequence reactive power load component; BZERO is negative for an inductive load and positive for a capacitive load; entered as admittance in MVar at 1.0 per unit voltage (0.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no errors occurred
	IERR > 0	fatal error occurred
	IERR = 1	bus not found
	IERR = 2	load identifier is more than two characters
	IERR = 3	load not found
	IERR = 4	prerequisite requirements for API are not met
	IERR = -1	data error, warning and information messages; one or more of: • invalid grounding code.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.45 SEQ_MACHINE_DATA

This API routine is obsolete. It has been replaced by the API routine SEQ_MACHINE_DATA_3, and is implemented by a call to the SEQ_MACHINE_DATA_3 API routine.

Use this API routine to modify the sequence data of an existing machine in the working case (see *PSS®E Program Operation Manual*, Positive Sequence Generator Impedance Data, Negative Sequence Generator Impedance Data and Zero Sequence Generator Impedance Data).

Batch command syntax:

```
BAT_SEQ_MACHINE_DATA  I  'ID'  REALAR(1)...REALAR(6)
```

Python syntax:

```
ierr = seq_machine_data(i, id, realar)
```

Fortran syntax:

```
CALL SEQ_MACHINE_API_2(I, ID, REALAR, IERR)
```

where:

Integer I	Is the bus number (input; no default allowed).	
Character ID*2	Is the machine identifier (input; ’1’ by default).	
Real REALAR(6)	Is an array of six elements (input; defaulted data items are unchanged).	
	REALAR(1)	ZRPOS, machine positive sequence fault analysis resistance.
	REALAR(2)	ZXPOS, machine positive sequence fault analysis reactance.
	REALAR(3)	ZRNEG, machine negative sequence fault analysis resistance.
	REALAR(4)	ZXNEG, machine negative sequence fault analysis reactance.
	REALAR(5)	RZERO, machine zero sequence fault analysis resistance.
	REALAR(6)	XZERO, machine zero sequence fault analysis reactance.
Integer IERR	Is the error code (output).	
	IERR = 0	no errors occurred.
	IERR > 0	fatal error occurred.
	IERR = 1	bus not found.
	IERR = 2	machine identifier is more than two characters.
	IERR = 3	machine not found.
	IERR = 4	prerequisite requirements for API are not met.
	IERR = -1	data error, warning and information messages; one or more of: • Machine impedance is (0.0,0.0).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.46 SEQ_MACHINE_DATA_3

Use this API routine to run the third release of the machine sequence data specification function. Use this API routine to modify the sequence data of an existing machine in the working case (see *PSS®E Program Operation Manual*, Positive Sequence Generator Impedance Data, Negative Sequence Generator Impedance Data and Zero Sequence Generator Impedance Data).

Batch command syntax:

```
BAT_SEQ_MACHINE_DATA_3  I  'ID'  INTGAR(1)  REALAR(1)...REALAR(10)
```

Python syntax:

```
ierr = seq_machine_data_3(i, id, intgar, realar)
```

Fortran syntax:

```
CALL SEQ_MACHINE_API_3(I, ID, INTGAR, REALAR, IERR)
```

where:

Integer I	Is the bus number (input; no default allowed).	
Character ID*2	Is the machine identifier (input; ’1’ by default).	
Integer INTGAR(1)	Is an array of one element (input; defaulted data items are unchanged).	
	INTGAR(1)	CZG, code indicating the units in which the zero sequence grounding impedance is specified: 1 for per unit 2 for ohms:
Real REALAR(10)	Is an array of ten elements (input; defaulted data items are unchanged).	
	REALAR(1)	RPOS, machine positive sequence fault analysis resistance.
	REALAR(2)	X", machine positive sequence fault analysis subtransient reactance.
	REALAR(3)	RNEG, machine negative sequence fault analysis resistance.
	REALAR(4)	XNEG, machine negative sequence fault analysis reactance.
	REALAR(5)	RZERO, machine zero sequence fault analysis resistance.
	REALAR(6)	XZERO, machine zero sequence fault analysis reactance.
	REALAR(7)	X’, machine positive sequence fault analysis transient reactance.
	REALAR(8)	Xs, machine positive sequence fault analysis synchronous reactance.
	REALAR(9)	RG, machine zero sequence grounding resistance.
	REALAR(10)	XG, machine zero sequence grounding reactance.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no errors occurred.
IERR > 0	fatal error occurred.
IERR = 1	bus not found.
IERR = 2	machine identifier is more than two characters.
IERR = 3	machine not found.
IERR = 4	prerequisite requirements for API are not met.
IERR = -1	data error, warning and information messages; one or more of:

- Grounding impedance units code indicates input in ohms but no base voltage has been specified for the bus.
- Invalid grounding impedance units code.
- Invalid grounding impedance units code.
- Invalid positive sequence reactance value.
- Negative sequence impedance is (0.0,0.0).
- Zero sequence impedance is (0.0,0.0).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.47 SEQ_MUTUAL_DATA

Use this API to modify existing zero sequence mutual impedance data in the working case or to add a new zero sequence mutual coupling to the working case (see *PSS®E Program Operation Manual*, Zero Sequence Mutual Impedance Data).

Batch command syntax:

```
BAT_SEQ_MUTUAL_DATA  I  J  'ICKT1'  K  L  'ICKT2'
REALAR(1)...REALAR(6)
```

Python syntax:

```
ierr = seq_mutual_data(i, j, ickt1, k, l, ickt2, realar)
```

Fortran syntax:

```
CALL SEQ_MUTUAL_API_2(I, J, ICKT1, K, L, ICKT2, REALAR, IERR)
```

where:

Integer I	Is the bus number of the from bus of the first branch (input; no default allowed).
Integer J	Is the bus number of the to bus of the first branch (input; no default allowed).
Character ICKT1*2	Is the first branch circuit identifier (input; '1' by default).
Integer K	Is the bus number of the from bus of the second branch (input; no default allowed).
Integer L	Is the bus number of the to bus of the second branch to bus (input; no default allowed).
Character ICKT2*2	Is the second branch circuit identifier (input; '1' by default).
Real REALAR(6)	Is an array of six elements (input).
	REALAR(1) — RM, branch-to-branch mutual resistance (0.0 by default).
	REALAR(2) — XM, branch-to-branch mutual reactance (0.0 by default).
	REALAR(3) — BIJ1, B factor at I end of the first branch (0.0 by default).
	REALAR(4) — BIJ2, B factor at J end of the first branch (1.0 by default).
	REALAR(5) — BKL1, B factor at K end of the second branch (0.0 by default).
	REALAR(6) — BKL2, B factor at L end of the second branch (1.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no errors occurred
IERR > 0	fatal error occurred
IERR = 1	bus not found
IERR = 2	circuit identifier is more than two characters
IERR = 3	circuit identifier may not start with &
IERR = 4	branch not found
IERR = 5	mutual couples a branch to itself
IERR = 6	branch is a transformer
IERR = 7	branch is a zero impedance line
IERR = 8	mutual coupling table is full
IERR = 9	prerequisite requirements for API are not met.
IERR = -1	data error, warning and information messages; one or more of: • Invalid B factor (<0.0 or >1.0). • B1 is not less than B2.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.48 SEQ_SWITCHED_SHUNT_DATA

Use this API to modify the zero sequence data of an existing switched shunt in the working case (see *PSS®E Program Operation Manual*, Zero Sequence Switched Shunt Data).

Batch command syntax:

```
BAT_SEQ_SWITCHED_SHUNT_DATA  I  REALAR(1)...REALAR(8)
```

Python syntax:

```
ierr,realaro = seq_switched_shunt_data(i, realari)
```

Fortran syntax:

```
CALL SEQ_SWITCHED_SHUNT_API_2(I, REALAR, IERR)
```

where:

Integer I	Is the bus number (input; no default allowed).	
Real REALAR(9)	Is an array of nine elements (first eight are input values; last one is output value) (input; defaulted data items are unchanged).	
	REALAR(1)	BZ1, zero sequence increment in MVAR per step for block 1.
	REALAR(2)	BZ2, zero sequence increment in MVAR per step for block 2.
	REALAR(3)	BZ3, zero sequence increment in MVAR per step for block 3.
	REALAR(4)	BZ4, zero sequence increment in MVAR per step for block 4.
	REALAR(5)	BZ5, zero sequence increment in MVAR per step for block 5.
	REALAR(6)	BZ6, zero sequence increment in MVAR per step for block 6.
	REALAR(7)	BZ7, zero sequence increment in MVAR per step for block 7.
	REALAR(8)	BZ8, zero sequence increment in MVAR per step for block 8.
	REALAR(9)	present zero sequence switched shunt admittance in MVAR (**returned**).
Real REALARI(8)	Is an array of eight elements corresponding to input values REALAR(1) through REALAR(8) as described above.	
Real REALARO(1)	Is an array of one element corresponding to output value REALAR(9) as described above.	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no errors occurred.
IERR > 0	fatal error occurred.
IERR = 1	bus not found.
IERR = 2	no switched shunt at bus.
IERR = 3	prerequisite requirements for API are not met.
IERR = -1	data error, warning and information messages; one or more of: • Block has N zero and B non-zero. • Block has admittances with different signs in positive and zero sequences.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.49 SEQ_THREE_WINDING_DATA

This API routine is obsolete. It has been replaced by the API routine SEQ_THREE_WINDING_DATA_3, and is implemented by a call to the SEQ_THREE_WINDING_DATA_3 API routine.

Use this API to modify the zero sequence data of an existing three-winding transformer in the working case.

Batch command syntax:

```
BAT_SEQ_THREE_WINDING_DATA  I  J  K  'ICKT'  INTGAR(1)
REALAR(1)...REALAR(8)
```

Python syntax:

```
ierr = seq_three_winding_data(i, j, ickt, intgar, realar)
```

Fortran syntax:

```
CALL SEQ_THREE_WINDING_API(I, J, K, ICKT, INTGAR, REALAR, IERR, NERR)
```

where:

Integer I	Is the bus number of one of the buses.	
Integer J	Is the bus number one of the other buses.	
Integer K	Is the bus number of the third bus.	
Character ICKT*2	Is the transformer circuit identifier.	
Integer INTGAR(1)	Is an array of one element.	
	INTGAR(1)	CC, connection code.
Real REALAR(8)	Is an array of at least eight elements.	
	REALAR(1)	RG, zero sequence grounding resistance.
	REALAR(2)	XG, zero sequence grounding reactance.
	REALAR(3)	R1, zero sequence winding one resistance.
	REALAR(4)	X1, zero sequence winding one reactance.
	REALAR(5)	R2, zero sequence winding two resistance.
	REALAR(6)	X2, zero sequence winding two reactance.
	REALAR(7)	R3, zero sequence winding three resistance.
	REALAR(8)	X3, zero sequence winding three reactance.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code.	
	IERR = 0	no errors occurred.
	IERR > 0	error occurred.
	IERR = 1	bus not found.
	IERR = 2	circuit identifier is more than two characters.
	IERR = 3	three-winding transformer not found.
	IERR = 4	prerequisite requirements for API are not met.
	IERR = -1	warnings – one or more of: • Invalid winding connection code. • Connection code is such that more than one winding requires a grounding impedance.
Integer NERR	The number of entries in PBUF02 containing error messages.	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.50 SEQ_THREE_WINDING_DATA_3

Use this API routine to modify the zero sequence data of an existing three-winding transformer in the working case.

Batch command syntax:

```
BAT_SEQ_THREE_WINDING_DATA_3  I  J  K  'ICKT'  INTGAR(1)...INTGAR(3)
REALAR(1)...REALAR(14
```

Python syntax:

```
ierr = seq_three_winding_data_3(i, j, ickt, intgar, realar)
```

Fortran syntax:

```
CALL SEQ_THREE_WINDING_API_3(I, J, K, ICKT, INTGAR, REALAR, IERR,
NERR)
```

where:

Integer I	Is the bus number of one of the buses.	
Integer J	Is the bus number one of the other buses.	
Integer K	Is the bus number of the third bus.	
Character ICKT*2	Is the transformer circuit identifier.	
Integer INTGAR(3)	Is an array of three elements (input; defaulted data items are unchanged).	
	INTGAR(1)	CZ0, non-grounding impedance data I/O code.
	INTGAR(2)	CZG, grounding impedance data I/O code.
	INTGAR(3)	CC, connection code.
Real REALAR(14)	Is an array of at least fourteen elements (input; defaulted data items are unchanged).	
	REALAR(1)	RG1, grounding resistance on winding 1 for an impedance grounded transformer.
	REALAR(2)	XG1, grounding reactance on winding 1 for an impedance grounded transformer.
	REALAR(3)	R01, winding 1, or winding 1 bus to winding 2 bus, leakage resistance..
	REALAR(4)	X01, winding 1, or winding 1 bus to winding 2 bus, leakage reactance.
	REALAR(5)	RG2, grounding resistance on winding 2 for an impedance grounded transformer.
	REALAR(6)	XG2, grounding reactance on winding 2 for an impedance grounded transformer.
	REALAR(7)	R02, winding 2, or winding 2 bus to winding 3 bus, leakage resistance..
	REALAR(8)	X02, winding 2, or winding 2 bus to winding 3 bus, leakage reactance.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real REALAR(14) Is an array of at least fourteen elements (input; defaulted data items are unchanged).

REALAR(9)	RG3, grounding resistance on winding 3 for an impedance grounded transformer.
REALAR(10)	XG3, grounding reactance on winding 3 for an impedance grounded transformer.
REALAR(11)	R03, winding 3, or winding 3 bus to winding 1 bus, leakage resistance..
REALAR(12)	X03, winding 3, or winding 3 bus to winding 1 bus, leakage reactance.
REALAR(13)	RN, common neutral grounding resistance.
REALAR(14)	XN, common neutral grounding reactance.

Integer IERR Is the error code.

IERR = 0	no errors occurred.
IERR > 0	error occurred.
IERR = 1	bus not found.
IERR = 2	circuit identifier is more than two characters.
IERR = 3	three-winding transformer not found.
IERR = 4	CZG is 3 but at least one endpoint bus has no base voltage specified.
IERR = 5	invalid CZ0 or CZG value.
IERR = 6	prerequisite requirements for API are not met.
IERR = -1	warnings – one or more of:

- Invalid winding connection code.
- Invalid CZ0 or CZG data value.
- CZ0 or CZG is not 1 with connection code 1 through 9 or a three-digit connection code.
- CZG data value is 3 but at least one endpoint bus has no base voltage specified.
- Non-zero neutral impedance specified but at least one endpoint bus has no base or nominal voltage specified.
- A non-zero value is specified for an impedance that is not used with the specified connection code.
- A value of zero is specified for an impedance that is used with the specified connection code.
- Both ZGn and Z0n are used with the specified connection code and both are (0.0,0.0).
- Connection code is 17 or 18 but at least one endpoint bus has no base or nominal voltage specified.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.51 SEQ_TWO_WINDING_DATA

This API routine is obsolete. It has been replaced by the API routine SEQ_TWO_WINDING_DATA_3, and is implemented by a call to the SEQ_TWO_WINDING_DATA_3 API routine.

Use this API routine to modify the zero sequence data of an existing two-winding transformer in the working case (see *PSS®E Program Operation Manual*, Zero Sequence Transformer Data).

Batch command syntax:

```
BAT_SEQ_TWO_WINDING_DATA  I  J  'ICKT'  INTGAR(1)
REALAR(1)...REALAR(6)
```

Python syntax:

```
ierr = seq_two_winding_data(i, j, ickt, intgar, realar)
```

Fortran syntax:

```
CALL SEQ_TWO_WINDING_API_2(I, J, ICKT, INTGAR, REALAR, IERR)
```

where:

Integer I	Is the bus number of from bus (input; no default allowed).	
Integer J	Is the bus number of to bus (input; no default allowed).	
Character ICKT*2	Is the circuit identifier (input; ’1’ by default).	
Integer INTGAR(1)	Is an array of one element (input; defaulted data items are unchanged).	
	INTGAR(1)	CC, connection code (1 through 8).
Real REALAR(6)	Is an array of six elements (input; defaulted data items are unchanged).	
	REALAR(1)	RG, grounding resistance if connection code < 8; winding 1 side resistance to ground if CC = 8.
	REALAR(2)	XG, grounding reactance if connection code < 8; winding 1 side reactance to ground if CC = 8.
	REALAR(3)	R1, zero sequence transformer resistance.
	REALAR(4)	X1, zero sequence transformer reactance.
	REALAR(5)	RG2, winding 2 side resistance to ground if CC = 8; 0.0 otherwise.
	REALAR(6)	XG2, winding 2 side reactance to ground if CC = 8; 0.0 otherwise.
Integer IERR	Is the error code (output).	
	IERR = 0	no errors occurred.
	IERR > 0	fatal error occurred.
	IERR = 1	bus not found.
	IERR = 2	circuit identifier is more than two characters.
	IERR = 3	circuit identifier may not start with &, * or @.
	IERR = 4	branch not found.
	IERR = 5	branch exists but it is a non-transformer branch.
	IERR = 6	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IERR = -1 data error, warning and information messages; one or more of:

- Invalid winding connection code.
- Zero sequence impedance is (0.0,0.0).
- Non-zero winding 2 impedance to ground not used with the specified connection code.
- Winding n impedance to ground is (0.0,0.0) with connection code 8.
- Grounding impedance is (0.0,0.0) with connection code 9.
- A specified series impedance is (0.0,0.0) with connection code 9.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.52 SEQ_TWO_WINDING_DATA_3

Use this API routine to modify the zero sequence data of an existing two-winding transformer in the working case (see *PSS®E Program Operation Manual*, Zero Sequence Transformer Data).

Batch command syntax:

```
BAT_SEQ_TWO_WINDING_DATA_3  I  J  'ICKT'  INTGAR(1)...INTGAR(3)
REALAR(1)...REALAR(10)
```

Python syntax:

```
ierr = seq_two_winding_data_3(i, j, ickt, intgar, realar)
```

Fortran syntax:

```
CALL SEQ_TWO_WINDING_API_3(I, J, ICKT, INTGAR, REALAR, IERR)
```

where:

Integer I	Is the bus number of from bus (input; no default allowed).	
Integer J	Is the bus number of to bus (input; no default allowed).	
Character ICKT*2	Is the circuit identifier (input; ’1’ by default).	
Integer INTGAR(3)	Is an array of three elements (input; defaulted data items are unchanged).	
	INTGAR(1)	CC, connection code.
	INTGAR(2)	CZ0, leakage impedance data I/O code.
	INTGAR(3)	CZG, grounding impedance data I/O code.
Real REALAR(10)	Is an array of ten elements (input; defaulted data items are unchanged).	
	REALAR(1)	RG1, grounding resistance at the winding 1 bus for an impedance grounded transformer.
	REALAR(2)	XG1, grounding reactance at the winding 1 bus for an impedance grounded transformer.
	REALAR(3)	R01, zero sequence leakage resistance connected to the winding 1 bus..
	REALAR(4)	X01, zero sequence leakage reactance connected to the winding 1 bus.
	REALAR(5)	RG2, grounding resistance at the winding 2 bus for an impedance grounded transformer.
	REALAR(6)	XG2, grounding reactance at the winding 2 bus for an impedance grounded transformer.
	REALAR(7)	R02, zero sequence leakage resistance connected to the winding 2 bus..
	REALAR(8)	X02, zero sequence leakage reactance connected to the winding 2 bus.
	REALAR(9)	RN, common neutral grounding resistance.
	REALAR(10)	XN, common neutral grounding reactance.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no errors occurred.
IERR > 0	fatal error occurred.
IERR = 1	bus not found.
IERR = 2	circuit identifier is more than two characters.
IERR = 3	circuit identifier may not start with &, * or @.
IERR = 4	branch not found.
IERR = 5	branch exists but it is a non-transformer branch.
IERR = 6	CZG is 3 but at least one endpoint bus has no base voltage specified.
IERR = 7	invalid CZ0 or CZG value.
IERR = 8	prerequisite requirements for API are not met.
IERR = -1	data error, warning and information messages; one or more of: • Invalid winding connection code. • Z01 is set to its positive sequence value. • A non-zero value is specified for an impedance that is not used with the specified connection code.. • A value of zero is specified for an impedance that is used with the specified connection code. • Invalid CZ0 or CZG data value. • CZ0 or CZG is not 1 with connection code 1 through 9. • CZG data value is 3 but at least one endpoint bus has no base voltage specified. • Non-zero neutral impedance specified but at least one endpoint bus has no base or nominal voltage specified. • Connection code is 18 through 22 but at least one endpoint bus has no base or nominal voltage specified.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.53 SHUNT_CHNG

Use this API routine to modify the data of an existing fixed bus shunt in the working case (see *PSS®E Program Operation Manual*, Fixed Bus Shunt Data).

Batch command syntax:

```
BAT_SHUNT_CHNG  I  'ID'  INTGAR(1)  REALAR(1)  REALAR(2)
```

Python syntax:

```
ierr = shunt_chng(i, id, intgar, realar)
```

Fortran syntax:

```
CALL SHUNT_CHNG(I, ID, INTGAR, REALAR, IERR)
```

where:

Integer I	Is the bus number (input; no default allowed).
Character ID*2	Is the shunt identifier (input; ’1’ by default).
Integer INTGAR(1)	Is an array of one elements (input). See the SHUNT_DATA API routine for details.
Real REALAR(2)	Is an array of two elements (input). See the SHUNT_DATA API routine for details.
Integer IERR	Is the error code (output).
	IERR = 0 — no errors occurred
	IERR > 0 — fatal error occurred
	IERR = 1 — bus not found
	IERR = 2 — shunt identifier is more than two characters
	IERR = 5 — fixed bus shunt not found
	IERR = 6 — prerequisite requirements for API are not met
	IERR = -1 — data error, warning and information messages; one or more of: • invalid shunt status.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.54 SHUNT_DATA

Use this API to modify the data of an existing fixed bus shunt in the working case or to add a new fixed bus shunt to the working case (see *PSS®E Program Operation Manual*, Fixed Bus Shunt Data).

Batch command syntax:

```
BAT_SHUNT_DATA  I  'ID'  INTGAR(1)  REALAR(1)  REALAR(2)
```

Python syntax:

```
ierr = shunt_data(i, id, intgar, realar)
```

Fortran syntax:

```
CALL SHUNT_API(I, ID, INTGAR, REALAR, IERR)
```

where:

Integer I	Is the bus number (input; no default allowed).	
Character ID*2	Is the shunt identifier (input; ’1’ by default).	
Integer INTGAR(1)	Is an array of one elements (input)	
	INTGAR(1)	STATUS, shunt status (1 by default).
Real REALAR(2)	Is an array of two elements (input).	
	REALAR(1)	GL, fixed shunt admittance (conductance) (0.0 by default).
	REALAR(2)	BL, fixed shunt admittance (susceptance) (0.0 by default).
Integer IERR	Is the error code (output).	
	IERR = 0	no errors occurred
	IERR > 0	fatal error occurred
	IERR = 1	bus not found
	IERR = 2	shunt identifier is more than two characters
	IERR = 3	blank shunt identifier is invalid
	IERR = 4	fixed bus shunt table is full
	IERR = 6	prerequisite requirements for API are not met
	IERR = -1	data error, warning and information messages; one or more of: • invalid shunt status.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.55 SOLUTION_PARAMETERS

This API routine is obsolete. It has been replaced by the API routine SOLUTION_PARAMETERS_4, and is implemented by a call to the SOLUTION_PARAMETERS_4 API routine.

Use this API to modify the power flow solution parameters in the working case.

Batch command syntax:

```
BAT_SOLUTION_PARAMETERS  INTGAR(1)...INTGAR(3)
REALAR(1)...REALAR(16)
```

Python syntax:

```
ierr = solution_parameters(intgar, realar)
```

Fortran syntax:

```
CALL SOLUTION_PARAMETER_API(INTGAR, REALAR, IERR, NERR)
```

where:

Integer INTGAR(3)	Is an array of three elements.	
	INTGAR(1)	ITMX, Gauss-Seidel maximum number of iterations.
	INTGAR(2)	ITMXN, Newton-Raphson maximum number of iterations.
	INTGAR(3)	ITMXTY, TYSL maximum number of iterations.
Real REALAR(16)	Is an array of sixteen elements.	
	REALAR(1)	ACCP, Gauss-Seidel real component acceleration factor.
	REALAR(2)	ACCQ, Gauss-Seidel imaginary component acceleration factor.
	REALAR(3)	ACCM, special MSLV acceleration factor.
	REALAR(4)	TOL, Gauss-Seidel convergence tolerance.
	REALAR(5)	ACCN, Newton-Raphson acceleration factor.
	REALAR(6)	TOLN, Newton-Raphson mismatch convergence tolerance.
	REALAR(7)	ACCTY, TYSL acceleration factor.
	REALAR(8)	TOLTY, TYSL convergence tolerance.
	REALAR(9)	BLOWUP, blow-up threshold.
	REALAR(10)	PQBRAK, constant power load characteristic breakpoint.
	REALAR(11)	THRSHZ, zero impedance line threshold.
	REALAR(12)	ADJTHR, automatic adjustment threshold tolerance.
	REALAR(13)	ACCTAP, tap movement deceleration factor.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	REALAR(14)	TAPLIM, maximum tap change per adjustment.
	REALAR(15)	DVLIM, largest | (delta v)/|v| | for Newton solutions.
	REALAR(16)	NDVFCT, non-divergent required improvement factor.
Integer IERR	Is the error code.	
	IERR = 0	no errors occurred.
	IERR > 0	error occurred.
	IERR = 1	prerequisite requirements for API are not met.
	IERR = -1	warnings: • None.
Integer NERR	The number of entries in PBUF02 containing error messages.	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.56 SOLUTION_PARAMETERS_2

This API routine is obsolete. It has been replaced by the API routine SOLUTION_PARAMETERS_4, and is implemented by a call to the SOLUTION_PARAMETERS_4 API routine.

Use this API to run the second release of the solution parameters data specification engine. It modifies the power flow solution parameters in the working case.

Batch command syntax:

```
BAT_SOLUTION_PARAMETERS_2  INTGAR(1)...INTGAR(3)
REALAR(1)...REALAR(18)
```

Python syntax:

```
ierr = solution_parameters_2(intgar, realar)
```

Fortran syntax:

```
CALL SOLUTION_PARAMETER_API_2(INTGAR, REALAR, IERR, NERR)
```

where:

Integer INTGAR(3)	Is an array of three elements.	
	INTGAR(1)	ITMX, Gauss-Seidel maximum number of iterations.
	INTGAR(2)	ITMXN, Newton-Raphson maximum number of iterations.
	INTGAR(3)	ITMXTY, TYSL maximum number of iterations.
Real REALAR(18)	Is an array of eighteen elements.	
	REALAR(1)	ACCP, Gauss-Seidel real component acceleration factor.
	REALAR(2)	ACCQ, Gauss-Seidel imaginary component acceleration factor.
	REALAR(3)	ACCM, special MSLV acceleration factor.
	REALAR(4)	TOL, Gauss-Seidel convergence tolerance.
	REALAR(5)	ACCN, Newton-Raphson acceleration factor.
	REALAR(6)	TOLN, Newton-Raphson mismatch convergence tolerance.
	REALAR(7)	ACCTY, TYSL acceleration factor.
	REALAR(8)	TOLTY, TYSL convergence tolerance.
	REALAR(9)	BLOWUP, blow-up threshold.
	REALAR(10)	PQBRAK, constant power load characteristic breakpoint.
	REALAR(11)	THRSHZ, zero impedance line threshold.
	REALAR(12)	ADJTHR, automatic adjustment threshold tolerance.
	REALAR(13)	ACCTAP, tap movement deceleration factor.
	REALAR(14)	TAPLIM, maximum tap change per adjustment.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	REALAR(15)	DVLIM, largest | (delta v)/|v| | for Newton solutions.
	REALAR(16)	NDVFCT, non-divergent required improvement factor.
	REALAR(17)	VCTOLQ, Newton-Raphson voltage controlled bus reactive power mismatch convergence tolerance.
	REALAR(18)	VCTOLV, Newton-Raphson voltage controlled bus voltage error convergence tolerance.
Integer IERR	Is the error code.	
	IERR = 0	no errors occurred.
	IERR > 0	error occurred.
	IERR = 1	prerequisite requirements for API are not met.
	IERR = -1	warnings: • None.
Integer NERR	The number of entries in PBUF02 containing error messages.	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.57 SOLUTION_PARAMETERS_3

This API routine is obsolete. It has been replaced by the API routine SOLUTION_PARAMETERS_4, and is implemented by a call to the SOLUTION_PARAMETERS_4 API routine.

This API routine is the third release of the solution parameters data specification function. Use this API to run the third release of the solution parameters data specification engine. It modifies the power flow solution parameters in the working case (see *PSS®E Program Operation Manual*, Sections 6.3.13, 6.3.18, 6.5.1 and 12.7).

Batch command syntax:

```
BAT_SOLUTION_PARAMETERS_3  INTGAR(1)...INTGAR(3)
REALAR(1)...REALAR(19)
```

Python syntax:

```
ierr = solution_parameters_3(intgar, realar)
```

Fortran syntax:

```
CALL SOLUTION_PARAMETER_API_3(INTGAR, REALAR, IERR)
```

where:

Integer INTGAR(3)	Is an array of three elements (input).	
	INTGAR(1)	ITMX, Gauss-Seidel maximum number of iterations (100 by default).
	INTGAR(2)	ITMXN, Newton-Raphson maximum number of iterations (20 by default).
	INTGAR(3)	ITMXTY, TYSL maximum number of iterations (20 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real REALAR(19) Is an array of nineteen elements (input).

REALAR(1)	ACCP, Gauss-Seidel real component acceleration factor (1.6 by default).
REALAR(2)	ACCQ, Gauss-Seidel imaginary component acceleration factor (1.6 by default).
REALAR(3)	ACCM, special MSLV acceleration factor (1.0 by default).
REALAR(4)	TOL, Gauss-Seidel convergence tolerance (0.0001 by default).
REALAR(5)	ACCN, Newton-Raphson acceleration factor (1.0 by default).
REALAR(6)	TOLN, Newton-Raphson mismatch convergence tolerance (default Newton power flow solution tolerance option setting by default).
REALAR(7)	ACCTY, TYSL acceleration factor (1.0 by default).
REALAR(8)	TOLTY, TYSL convergence tolerance (0.00001 by default).
REALAR(9)	BLOWUP, blow-up threshold (5.0 by default).
REALAR(10)	PQBRAK, constant power load characteristic breakpoint (0.7 by default).
REALAR(11)	THRSHZ, zero impedance line threshold (0.0001 by default).
REALAR(12)	ADJTHR, automatic adjustment threshold tolerance (0.005 by default).
REALAR(13)	ACCTAP, tap movement deceleration factor (1.0 by default).
REALAR(14)	TAPLIM, maximum tap change per adjustment (0.05 by default).
REALAR(15)	DVLIM, largest | (delta v)/|v| | for Newton solutions (0.99 by default).
REALAR(16)	NDVFCT, non-divergent required improvement factor (0.99 by default).
REALAR(17)	VCTOLQ, Newton-Raphson voltage controlled bus reactive power mismatch convergence tolerance (default Newton power flow solution tolerance option setting by default).
REALAR(18)	VCTOLV, Newton-Raphson voltage controlled bus voltage error convergence tolerance (0.00001 by default).
REALAR(19)	SWVBND, percent of voltage controlling band mode switched shunts to be adjusted per power flow iteration (100.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no errors occurred.
	IERR > 0	fatal error occurred.
	IERR = 1	prerequisite requirements for API are not met.
	IERR = -1	data error, warning and information messages; one or more of: • Invalid percent value for switched shunt adjustment.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.58 SOLUTION_PARAMETERS_4

This API routine is the fourth release of the solution parameters data specification function. Use this API routine to modify the power flow solution parameters in the working case (see *PSS®E Program Operation Manual*, Sections 6.3.13, 6.3.18, 6.5.1 and 12.7).

Batch command syntax:

```
BAT_SOLUTION_PARAMETERS_4  INTGAR(1)...INTGAR(5)
REALAR(1)...REALAR(19)
```

Python syntax:

```
ierr = solution_parameters_4(intgar, realar)
```

Fortran syntax:

```
CALL SOLUTION_PARAMETER_API_4(INTGAR, REALAR, IERR)
```

where:

Integer INTGAR(5) Is an array of five elements (input).

INTGAR(1)	ITMX, Gauss-Seidel maximum number of iterations (100 by default).
INTGAR(2)	ITMXN, Newton-Raphson maximum number of iterations (20 by default).
INTGAR(3)	ITMXTY, TYSL maximum number of iterations (20 by default).
INTGAR(4)	MXTPSS, maximum number of times taps and/or switched shunts are adjusted during power flow solutions (100 by default).
INTGAR(5)	MXSWIM, maximum number of times an induction machine can change state (e.g., from "running" to "stalled") during power flow solutions (10 by default).

Real REALAR(19) Is an array of nineteen elements (input).

REALAR(1)	ACCP, Gauss-Seidel real component acceleration factor (1.6 by default).
REALAR(2)	ACCQ, Gauss-Seidel imaginary component acceleration factor (1.6 by default).
REALAR(3)	ACCM, special MSLV acceleration factor (1.0 by default).
REALAR(4)	TOL, Gauss-Seidel convergence tolerance (0.0001 by default).
REALAR(5)	ACCN, Newton-Raphson acceleration factor (1.0 by default).
REALAR(6)	TOLN, Newton-Raphson mismatch convergence tolerance (default Newton power flow solution tolerance option setting by default).
REALAR(7)	ACCTY, TYSL acceleration factor (1.0 by default).
REALAR(8)	TOLTY, TYSL convergence tolerance (0.00001 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

REALAR(9)	BLOWUP, blow-up threshold (5.0 by default).REALAR(10)PQBRAK, constant power load characteristic breakpoint (0.7 by default).
REALAR(11)	THRSHZ, zero impedance line threshold (0.0001 by default).
REALAR(12)	ADJTHR, automatic adjustment threshold tolerance (0.005 by default).
REALAR(13)	ACCTAP, tap movement deceleration factor (1.0 by default).
REALAR(14)	TAPLIM, maximum tap change per adjustment (0.05 by default).
REALAR(15)	DVLIM, largest | (delta v)/|v| | for Newton solutions (0.99 by default).
REALAR(16)	NDVFCT, non-divergent required improvement factor (0.99 by default).
REALAR(17)	VCTOLQ, Newton-Raphson voltage controlled bus reactive power mismatch convergence tolerance (default Newton power flow solution tolerance option setting by default).
REALAR(18)	VCTOLV, Newton-Raphson voltage controlled bus voltage error convergence tolerance (0.00001 by default).
REALAR(19)	SWVBND, percent of voltage controlling band mode switched shunts to be adjusted per power flow iteration (100.0 by default).

Integer IERR — Is the error code (output).

IERR = 0	no errors occurred.
IERR > 0	fatal error occurred.
IERR = 1	prerequisite requirements for API are not met.
IERR = -1	data error, warning and information messages; one or more of: • Invalid percent value for switched shunt adjustment. • Invalid induction motor treatment code.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.59 SWITCHED_SHUNT_CHNG_3

Use this API routine to modify the data of an existing switched shunt in the working case (see *PSS®E Program Operation Manual*, Switched Shunt Data).

Batch command syntax:

```
BAT_SWITCHED_SHUNT_CHNG_3  I  INTGAR(1)...INTGAR(12)
REALAR(1)...REALAR(12)  'RMIDNT'
```

Python syntax:

```
ierr = switched_shunt_chng_3(i, intgar, realar, rmidnt)
```

Fortran syntax:

```
CALL SWITCHED_SHUNT_CHNG_3(I, INTGAR, REALAR, RMIDNT, IERR)
```

where:

Integer I	Is the bus number (input; no default allowed).
Integer INTGAR(12)	Is an array of at least twelve elements (input). See the SWITCHED_SHUNT_DATA_3 API routine for details.
Real REALAR(12)	Is an array of at least twelve elements. See the SWITCHED_SHUNT_DATA_3 API routine for details.
Character RMIDNT*12	Is the VSC dc line name (MODSW =4) or FACTS device name (MODSW = 6) (input; blank by default).
Integer IERR	Is the error code (output).

IERR = 0	no errors occurred
IERR > 0	error occurred
IERR = 1	bus not found
IERR = 3	switched shunt not found
IERR = 4	prerequisite requirements for API are not met
IERR = -1	warnings – one or more of: • invalid switched shunt control mode • control mode is 4 and a blank VSC dc line name is specified • VSC dc line not found • control mode is 6 and a blank FACTS device name is specified • FACTS device not found • non-blank device name specified with control mode 0, 1, 2, 3 or 5 • regulated bus not found • voltage controlling switched shunt specified as remotely regulating itself • type code of regulated bus is not 1 or 2 • VHIGH is less than VLOW for voltage controlling switched shunt

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

- no regulated bus specified for plant, VSC converter, or switched shunt control
- control mode is 3 and the regulated bus has no generation
- control mode is 4 and the regulated bus is not a converter of the specified VSC dc line
- control mode is 5 and the regulated bus has no switched shunt
- regulated bus is not the sending end bus of the specified FACTS device
- VLOW is not less than VHIGH for plant, VSC converter, switched shunt, or FACTS device control
- VHIGH is greater than 1.0 for plant, VSC converter, switched shunt, or FACTS device control
- VLOW is less than 0.0 for plant, VSC converter, switched shunt, or FACTS device control
- RMPCT is not positive
- block has N or B zero and the other non-zero
- invalid number of steps in block
- block with non-zero N is beyond a previous N=0 block
- block 1 has no steps.
- Reactor block is out of sequence.
- Initial admittance is below minimum.
- Initial admittance is above maximum.
- Initial admittance is not on a step.
- Block has admittances with different signs in positive and zero sequences.
- Invalid switched shunt status.
- Invalid switched shunt adjustment method.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.60 SWITCHED_SHUNT_DATA

This API is obsolete. It has been replaced by the API SWITCHED_SHUNT_DATA_3, and is implemented by a call to the SWITCHED_SHUNT_DATA_3 API.

Use this API to modify the data of an existing switched shunt in the working case or to add a new switched shunt to the working case.

Batch command syntax:

```
BAT_SWITCHED_SHUNT_DATA  I  INTGAR(1)...INTGAR(10)
REALAR(1)...REALAR(12)  'RMIDNT'
```

Python syntax:

```
ierr = switched_shunt_data(i, intgar, realar, rmidnt)
```

Fortran syntax:

```
CALL SWITCHED_SHUNT_API_2(I, INTGAR, REALAR, RMIDNT, IERR)
```

where:

Integer I	Is the bus number (input; no default allowed).	
Integer INTGAR(10)	Is an array of at least ten elements (input).	
	INTGAR(1)	N1, number of steps for block 1 (0 by default).
	INTGAR(2)	N2, number of steps for block 2 (0 by default).
	INTGAR(3)	N3, number of steps for block 3 (0 by default).
	INTGAR(4)	N4, number of steps for block 4 (0 by default).
	INTGAR(5)	N5, number of steps for block 5 (0 by default).
	INTGAR(6)	N6, number of steps for block 6 (0 by default).
	INTGAR(7)	N7, number of steps for block 7 (0 by default).
	INTGAR(8)	N8, number of steps for block 8 (0 by default).
	INTGAR(9)	MODSW, control mode (1 by default).
	INTGAR(10)	SWREM, remote bus number (0 by default).
Real REALAR(12)	Is an array of at least twelve elements.	
	REALAR(1)	B1, admittance increment per step for block 1 (0.0 by default).
	REALAR(2)	B2, admittance increment per step for block 2 (0.0 by default).
	REALAR(3)	B3, admittance increment per step for block 3 (0.0 by default).
	REALAR(4)	B4, admittance increment per step for block 4 (0.0 by default).
	REALAR(5)	B5, admittance increment per step for block 5 (0.0 by default).
	REALAR(6)	B6, admittance increment per step for block 6 (0.0 by default).
	REALAR(7)	B7, admittance increment per step for block 7 (0.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	REALAR(8)	B8, admittance increment per step for block 8 (0.0 by default).
	REALAR(9)	VSWHI, desired voltage upper limit (1.0 by default).
	REALAR(10)	VSWLO, desired voltage lower limit (1.0 by default).
	REALAR(11)	BINIT, present switched shunt admittance (0.0 by default).
	REALAR(12)	RMPCT, percent of contributed reactive power (100.0 by default).
Character RMIDNT*12	Is the VSC dc line name (MODSW =4) or FACTS device name (MODSW = 6) (input; blank by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no errors occurred
	IERR > 0	error occurred
	IERR = 1	bus not found
	IERR = 2	switched shunt table is full
	IERR = 4	prerequisite requirements for API are not met
	IERR = -1	warnings – one or more of: • invalid switched shunt control mode • control mode is 4 and a blank VSC dc line name is specified • VSC dc line not found • control mode is 6 and a blank FACTS device name is specified • FACTS device not found • non-blank device name specified with control mode 0, 1, 2, 3 or 5 • regulated bus not found • voltage controlling switched shunt specified as remotely regulating itself • type code of regulated bus is not 1 or 2 • VHIGH is less than VLOW for voltage controlling switched shunt • no regulated bus specified for plant, VSC converter, or switched shunt control • control mode is 3 and the regulated bus has no generation • control mode is 4 and the regulated bus is not a converter of the specified VSC dc line • control mode is 5 and the regulated bus has no switched shunt • regulated bus is not the sending end bus of the specified FACTS device

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

- VLOW is not less than VHIGH for plant, VSC converter, switched shunt, or FACTS device control
- VHIGH is greater than 1.0 for plant, VSC converter, switched shunt, or FACTS device control
- VLOW is less than 0.0 for plant, VSC converter, switched shunt, or FACTS device control
-
- RMPCT is not positive
- block has N or B zero and the other non-zero
- invalid number of steps in block
- block with non-zero N is beyond a previous N=0 block
- block 1 has no steps.
- Reactor block is out of sequence.
- Initial admittance is below minimum.
- Initial admittance is above maximum.
- Initial admittance is not on a step.
- Block has admittances with different signs in positive and zero sequences.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.61 SWITCHED_SHUNT_DATA_3

Use this API to modify the data of an existing switched shunt in the working case or to add a new switched shunt to the working case (see *PSS®E Program Operation Manual*, Switched Shunt Data).

Batch command syntax:

```
BAT_SWITCHED_SHUNT_DATA_3  I  INTGAR(1)...INTGAR(12)
REALAR(1)...REALAR(12)  'RMIDNT'
```

Python syntax:

```
ierr = switched_shunt_data_3(i, intgar, realar, rmidnt)
```

Fortran syntax:

```
CALL SWITCHED_SHUNT_API_3(I, INTGAR, REALAR, RMIDNT, IERR)
```

where:

Integer I	Is the bus number (input; no default allowed).	
Integer INTGAR(12)	Is an array of at least twelve elements (input).	
	INTGAR(1)	N1, number of steps for block 1 (0 by default).
	INTGAR(2)	N2, number of steps for block 2 (0 by default).
	INTGAR(3)	N3, number of steps for block 3 (0 by default).
	INTGAR(4)	N4, number of steps for block 4 (0 by default).
	INTGAR(5)	N5, number of steps for block 5 (0 by default).
	INTGAR(6)	N6, number of steps for block 6 (0 by default).
	INTGAR(7)	N7, number of steps for block 7 (0 by default).
	INTGAR(8)	N8, number of steps for block 8 (0 by default).
	INTGAR(9)	MODSW, control mode (1 by default).
	INTGAR(10)	SWREM, remote bus number (0 by default).
	INTGAR(11)	STAT, status (1 by default).
	INTGAR(12)	ADJM, adjustment method (0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real REALAR(12)	Is an array of at least twelve elements.	
	REALAR(1)	B1, admittance increment per step for block 1 (0.0 by default).
	REALAR(2)	B2, admittance increment per step for block 2 (0.0 by default).
	REALAR(3)	B3, admittance increment per step for block 3 (0.0 by default).
	REALAR(4)	B4, admittance increment per step for block 4 (0.0 by default).
	REALAR(5)	B5, admittance increment per step for block 5 (0.0 by default).
	REALAR(6)	B6, admittance increment per step for block 6 (0.0 by default).
	REALAR(7)	B7, admittance increment per step for block 7 (0.0 by default).
	REALAR(8)	B8, admittance increment per step for block 8 (0.0 by default).
	REALAR(9)	VSWHI, desired voltage upper limit (1.0 by default).
	REALAR(10)	VSWLO, desired voltage lower limit (1.0 by default).
	REALAR(11)	BINIT, present switched shunt admittance (0.0 by default).
	REALAR(12)	RMPCT, percent of contributed reactive power (100.0 by default).
Character RMIDNT*12	Is the VSC dc line name (MODSW =4) or FACTS device name (MODSW = 6) (input; blank by default).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR

Is the error code (output).

IERR = 0	no errors occurred
IERR > 0	error occurred
IERR = 1	bus not found
IERR = 2	switched shunt table is full
IERR = 4	prerequisite requirements for API are not met
IERR = -1	warnings – one or more of:

- invalid switched shunt control mode
- control mode is 4 and a blank VSC dc line name is specified
- VSC dc line not found
- control mode is 6 and a blank FACTS device name is specified
- FACTS device not found
- non-blank device name specified with control mode 0, 1, 2, 3 or 5
- regulated bus not found
- voltage controlling switched shunt specified as remotely regulating itself
- type code of regulated bus is not 1 or 2
- VHIGH is less than VLOW for voltage controlling switched shunt
- no regulated bus specified for plant, VSC converter, or switched shunt control
- control mode is 3 and the regulated bus has no generation
- control mode is 4 and the regulated bus is not a converter of the specified VSC dc line
- control mode is 5 and the regulated bus has no switched shunt
- regulated bus is not the sending end bus of the specified FACTS device
- VLOW is not less than VHIGH for plant, VSC converter, switched shunt, or FACTS device control
- VHIGH is greater than 1.0 for plant, VSC converter, switched shunt, or FACTS device control
- VLOW is less than 0.0 for plant, VSC converter, switched shunt, or FACTS device control
- RMPCT is not positive
- block has N or B zero and the other non-zero
- invalid number of steps in block
- block with non-zero N is beyond a previous N=0 block
- block 1 has no steps.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

- Reactor block is out of sequence.
- Initial admittance is below minimum.
- Initial admittance is above maximum.
- Initial admittance is not on a step.
- Block has admittances with different signs in positive and zero sequences.
- Invalid switched shunt status.
- Invalid switched shunt adjustment method.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.62 THREE_WINDING_DATA

This API is obsolete. It has been replaced by the APIs THREE_WND_IMPEDANCE_DATA and the THREE_WND_WINDING_DATA, and is implemented by calls to these APIs.

Use this API to modify the data of an existing three-winding transformer in the working case or to add a new three-winding transformer to the working case.

Batch command syntax:

```
BAT_THREE_WINDING_DATA  I  J  K  'CKT'  INTGAR(1)...INTGAR(17)
REALAR(1)...REALAR(41)  'NAME'
```

Fortran syntax:

```
CALL THREE_WINDING_API(I, J, K, CKT, INTGAR, REALAR, NAME, IERR,
NERR)
```

where:

Integer I	Is the bus number of one of the buses.
Integer J	Is the bus number one of the other buses.
Integer K	Is the bus number of the third bus.
Character CKT*2	Is the transformer circuit identifier.
Integer INTGAR(17)	Is an array of seventeen elements.

INTGAR(1)	O1, first owner number.
INTGAR(2)	O2, second owner number.
INTGAR(3)	O3, third owner number.
INTGAR(4)	O4, fourth owner number.
INTGAR(5)	NTP1, number of tap positions.
INTGAR(6)	TAB1, impedance correction table number.
INTGAR(7)	CONT1, controlled bus number.
INTGAR(8)	SICOD1, negative for controlled bus on winding one side.
INTGAR(9)	COD1, adjustment control mode flag (-3 through +3).
INTGAR(10)	CW, winding data I/O code.
INTGAR(11)	CZ, impedance data I/O code.
INTGAR(12)	CM, magnetizing admittance data I/O code.
INTGAR(13)	STAT, branch status.
INTGAR(14)	NMETBS, non-metered end bus number (I, J or K).
INTGAR(15)	WN1BUS, winding one side bus number (I, J or K).
INTGAR(16)	WN2BUS, winding two side bus number (I, J or K).
INTGAR(17)	WN3BUS, winding three side bus number (I, J or K).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real REALAR(53) Is an array of 53 elements.

REALAR(1)	R1-2, nominal bus one to two transformer resistance.
REALAR(2)	X1-2, nominal bus one to two transformer reactance.
REALAR(3)	R3-1, nominal bus three to one transformer resistance.
REALAR(4)	X3-1, nominal bus three to one transformer reactance.
REALAR(5)	R2-3, nominal bus two to three transformer resistance.
REALAR(6)	X2-3, nominal bus two to three transformer reactance.
REALAR(7)	SBS1-2, winding one to two base MVA.
REALAR(8)	SBS2-3, winding two to three base MVA.
REALAR(9)	SBS3-1, winding three to one base MVA.
REALAR(10)	MAG1, magnetizing conductance or no-load losses.
REALAR(11)	MAG2, magnetizing susceptance or exciting current.
REALAR(12)	WINDV1, winding one ratio or voltage.
REALAR(13)	NOMV1, winding one nominal voltage.
REALAR(14)	ANG1, winding one phase shift angle.
REALAR(15)	RATA1, winding one rating set A line rating.
REALAR(16)	RATB1, winding one rating set B line rating.
REALAR(17)	RATC1, winding one rating set C line rating.
REALAR(18)	WINDV2, winding two ratio or voltage.
REALAR(19)	NOMV2, winding two nominal voltage.
REALAR(20)	ANG2, winding two phase shift angle.
REALAR(21)	RATA2, winding two rating set A line rating.
REALAR(22)	RATB2, winding two rating set B line rating.
REALAR(23)	RATC2, winding two rating set C line rating.
REALAR(24)	WINDV3, winding three ratio or voltage.
REALAR(25)	NOMV3, winding three nominal voltage.
REALAR(26)	ANG3, winding three phase shift angle.
REALAR(27)	RATA3, winding three rating set A line rating.
REALAR(28)	RATB3, winding three rating set B line rating.
REALAR(29)	RATC3, winding three rating set C line rating.
REALAR(30)	F1, first owner fraction.
REALAR(31)	F2, second owner fraction.
REALAR(32)	F3, third owner fraction.
REALAR(33)	F4, fourth owner fraction.
REALAR(34)	RMA1, winding one ratio/angle high limit.
REALAR(35)	RMI1, winding one ratio/angle low limit.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	REALAR(36)	VMA1, voltage or flow upper limit.
	REALAR(37)	VMI1, voltage or flow lower limit.
	REALAR(38)	CR1, load drop compensating resistance.
	REALAR(39)	CX1, load drop compensating reactance.
	REALAR(40)	VMSTAR, star bus voltage magnitude.
	REALAR(41)	ANSTAR, star bus voltage angle.
	REALAR(42)	actual bus one to two resistance (**returned**).
	REALAR(43)	actual bus one to two reactance (**returned**).
	REALAR(44)	actual bus three to one resistance (**returned**).
	REALAR(45)	actual bus three to one reactance (**returned**).
	REALAR(46)	nominal winding one resistance (**returned**).
	REALAR(47)	nominal winding one reactance (**returned**).
	REALAR(48)	nominal winding two resistance (**returned**).
	REALAR(49)	nominal winding two reactance (**returned**).
	REALAR(50)	nominal winding three resistance (**returned**).
	REALAR(51)	nominal winding three reactance (**returned**).
	REALAR(52)	actual winding one resistance (**returned**).
	REALAR(53)	actual winding one reactance (**returned**).
Character NAME*12	Is the transformer name.	
Integer IERR	Is the error code.	
	IERR = 0	no error occurred
	IERR > 0	error occurred
	IERR = 1	bus not found
	IERR = 2	circuit identifier is more than two characters
	IERR = 3	invalid CW, CZ and/or CM value
	IERR = 4	CW, CZ and/or CM >1 but no base voltage at one or more buses
	IERR = 5	three different buses must be specified
	IERR = 6	blank circuit identifier
	IERR = 7	circuit identifier may not start with ’&’
	IERR = 8	three-winding transformer table full
	IERR = 9	bus table full
	IERR = 10	not enough branch table entries available
	IERR = 11	not enough room in two winding transformer table
	IERR = 12	not enough room in branch ownership table
	IERR = 13	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IERR = -1 warnings – one or more of:

- Invalid set of winding buses.
- Invalid transformer status.
- Invalid non-metered end bus.
- Winding side buses interchanged.
- Losses are too big relative to total impedance at winding base MVA.
- Magnitude of winding impedance is less than 0.00001.
- Controlled bus not found.
- Invalid adjustment control mode flag.
- Any two-winding transformer warnings.

Integer NERR Is the number of entries in PBUF02 containing error messages.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.63 THREE_WND_IMPED_CHNG_3

This API routine is the third release of the three-winding transformer data specification function. Use this API routine to modify the impedance data of an existing three-winding transformer in the working case (see *PSS®E Program Operation Manual*, Non-Transformer Branch Data).

Batch command syntax:

```
BAT_THREE_WND_IMPED_CHNG_3  I  J  K  'CKT'  INTGAR(1)...INTGAR(12)
REALAR(1)...REALAR(17)  'CHARAR(1)'  'CHARAR(2)'
```

Python syntax:

```
ierr,realaro = three_wnd_imped_chng_3(i, j, k, ckt, intgar, realari,
charar)
```

Fortran syntax:

```
CALL THREE_WINDING_IMPEDANCE_CHNG_3(I, J, K, CKT, INTGAR, REALAR,
CHARAR, IERR)
```

where:

Integer I	Is the bus number of one of the buses (input; no default allowed).
Integer J	Is the bus number one of the other buses (input; no default allowed).
Integer K	Is the bus number of the third bus (input; no default allowed).
Character CKT*2	Is the transformer circuit identifier (input; ’1’ by default).
Integer INTGAR(12)	Is an array of twelve elements (input). See the THREE_WND_IMPED_DATA_3 API routine for details.
Real REALAR(23)	Is an array of 23 elements (first seventeen are input values; last six are output values). See the THREE_WND_IMPED_DATA_3 API routine for details.
Real REALARI(17)	Is an array of seventeen elements corresponding to input values REALAR(1) through REALAR(17) above.
Real REALARO(6)	Is an array of six elements corresponding to output values REALAR(18) through REALAR(23) above.
Character CHARAR(2)*12	Is an array of two elements (input). See the THREE_WND_IMPED_DATA_3 API routine for details.
Integer IERR	Is the error code (output).

IERR = 0	no error occurred
IERR > 0	fatal error occurred
IERR = 1	bus not found
IERR = 2	circuit identifier is more than two characters
IERR = 3	invalid CW, CZ and/or CM value
IERR = 4	CW, CZ and/or CM >1 but at least one endpoint bus has no base voltage
IERR = 13	three-winding transformer not found
IERR = 14	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IERR = -1 data error, warning and information messages; one or more of:

- Invalid set of winding buses.
- Invalid CW, CZ and/or CM data value.
- Invalid transformer status.
- Invalid non-metered end bus.
- Winding side buses interchanged.
- Losses are too big relative to total impedance at winding base MVA.
- Magnitude of winding reactance is less than 0.000001.
- Vector group not found.
- Any two-winding transformer warnings.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.64 THREE_WND_IMPED_DATA_3

This API routine is the third release of the three-winding transformer data specification function. Use this API routine to modify the impedance data of an existing three-winding transformer in the working case, or to add a new three-winding transformer to the working case (see *PSS®E Program Operation Manual*, Non-Transformer Branch Data).

Batch command syntax:

```
BAT_THREE_WND_IMPED_DATA_3  I  J  K  'CKT'  INTGAR(1)...INTGAR(12)
REALAR(1)...REALAR(17)  'CHARAR(1)'  'CHARAR(2)'
```

Python syntax:

```
ierr,realaro = three_wnd_imped_data_3(i, j, k, ckt, intgar, realari,
charar)
```

Fortran syntax:

```
CALL THREE_WINDING_IMPEDANCE_API_3(I, J, K, CKT, INTGAR, REALAR,
CHARAR, IERR)
```

where:

Integer I	Is the bus number of one of the buses (input; no default allowed).
Integer J	Is the bus number one of the other buses (input; no default allowed).
Integer K	Is the bus number of the third bus (input; no default allowed).
Character CKT*2	Is the transformer circuit identifier (input; ’1’ by default).
Integer INTGAR(12)	Is an array of twelve elements (input).

INTGAR(1)	O1, first owner number (owner of bus I by default).
INTGAR(2)	O2, second owner number (0 by default).
INTGAR(3)	O3, third owner number (0 by default).
INTGAR(4)	O4, fourth owner number (0 by default).
INTGAR(5)	CW, winding data I/O code (1 by default).
INTGAR(6)	CZ, impedance data I/O code (1 by default).
INTGAR(7)	CM, magnetizing admittance data I/O code (1 by default).
INTGAR(8)	STAT, branch status (1 by default).
INTGAR(9)	NMETBS, non-metered end bus number (I, J or K) (J by default).
INTGAR(10)	WN1BUS, winding one side bus number (I, J or K) (I by default).
INTGAR(11)	WN2BUS, winding two side bus number (I, J or K) (J by default).
INTGAR(12)	WN3BUS, winding three side bus number (I, J or K) (K by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real REALAR(23)	Is an array of 23 elements (first seventeen are input values; last six are output values).	
	REALAR(1)	R1-2, nominal bus one to two transformer resistance (0.0 by default).
	REALAR(2)	X1-2, nominal bus one to two transformer reactance (0.0002 by default).
	REALAR(3)	R2-3, nominal bus two to three transformer resistance (0.0 by default).
	REALAR(4)	X2-3, nominal bus two to three transformer reactance (0.0002 by default).
	REALAR(5)	R3-1, nominal bus three to one transformer resistance (0.0 by default).
	REALAR(6)	X3-1, nominal bus three to one transformer reactance (0.0002 by default).
	REALAR(7)	SBS1-2, winding one to two base MVA (SBASE by default).
	REALAR(8)	SBS2-3, winding two to three base MVA (SBASE by default).
	REALAR(9)	SBS3-1, winding three to one base MVA (SBASE by default).
	REALAR(10)	MAG1, magnetizing conductance or no-load losses (0.0 by default).
	REALAR(11)	MAG2, magnetizing susceptance or exciting current (0.0 by default).
	REALAR(12)	F1, first owner fraction (1.0 by default).
	REALAR(13)	F2, second owner fraction (0.0 by default).
	REALAR(14)	F3, third owner fraction (0.0 by default).
	REALAR(15)	F4, fourth owner fraction (0.0 by default).
	REALAR(16)	VMSTAR, star bus voltage magnitude (1.0 by default).
	REALAR(17)	ANSTAR, star bus voltage angle (0.0 by default).
	REALAR(18)	actual bus one to two resistance (**returned**).
	REALAR(19)	actual bus one to two reactance (**returned**).
	REALAR(20)	actual bus two to three resistance (**returned**).
	REALAR(21)	actual bus two to three reactance (**returned**).
	REALAR(22)	actual bus three to one resistance (**returned**).
	REALAR(23)	actual bus three to one reactance (**returned**).
Real REALARI(17)	Is an array of seventeen elements corresponding to input values REALAR(1) through REALAR(17) as described above.	
Real REALARO(6)	Is an array of six elements corresponding to output values REALAR(18) through REALAR(23) as described above.	
Character CHARAR(2)*12	Is an array of two elements (input).	
	CHARAR(1)	NAME, transformer name (blank by default).
	CHARAR(2)	VGRP, vector group name (blank by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error occurred
IERR > 0	fatal error occurred
IERR = 1	bus not found
IERR = 2	circuit identifier is more than two characters
IERR = 3	invalid CW, CZ and/or CM value
IERR = 4	CW, CZ and/or CM >1 but at least one endpoint bus has no base voltage
IERR = 5	three different buses must be specified
IERR = 6	blank circuit identifier is invalid
IERR = 7	transformer circuit identifier may not start with &, * or @.
IERR = 8	three-winding transformer table is full
IERR = 9	bus table is full
IERR = 10	not enough branch table entries are available
IERR = 11	not enough two-winding transformer table entries are available.
IERR = 12	not enough branch ownership table entries are available.
IERR = 14	prerequisite requirements for API are not met.
IERR = -1	data error, warning and information messages; one or more of: • Invalid set of winding buses. • Invalid CW, CZ and/or CM data value. • Invalid transformer status. • Invalid non-metered end bus. • Winding side buses interchanged. • Losses are too big relative to total impedance at winding base MVA. • Magnitude of winding reactance is less than 0.000001. • Vector group not found. • Any two-winding transformer warnings.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.65 THREE_WND_IMPEDANCE_DATA

This API routine is obsolete. It has been replaced by the API routine THREE_WND_IMPED_DATA_3, and is implemented by a call to the THREE_WND_IMPED_DATA_3 API routine.

Use this API routine to modify the impedance data of an existing three-winding transformer in the working case, or to add a new three-winding transformer to the working case (see *PSS®E Program Operation Manual*, Non-Transformer Branch Data).

Batch command syntax:

```
BAT_THREE_WND_IMPEDANCE_DATA  I  J  K  'CKT'  INTGAR(1)...INTGAR(12)
REALAR(1)...REALAR(17)  'NAME'
```

Python syntax:

```
ierr,realaro = three_wnd_impedance_data(i, j, k, ckt, intgar,
realari, name)
```

Fortran syntax:

```
CALL THREE_WINDING_IMPEDANCE_API_2(I, J, K, CKT, INTGAR, REALAR,
NAME, IERR)
```

where:

Integer I	Is the bus number of one of the buses (input; no default allowed).
Integer J	Is the bus number one of the other buses (input; no default allowed).
Integer K	Is the bus number of the third bus (input; no default allowed).
Character CKT*2	Is the transformer circuit identifier (input; ’1’ by default).
Integer INTGAR(12)	Is an array of twelve elements (input).

INTGAR(1)	O1, first owner number (owner of bus I by default).
INTGAR(2)	O2, second owner number (0 by default).
INTGAR(3)	O3, third owner number (0 by default).
INTGAR(4)	O4, fourth owner number (0 by default).
INTGAR(5)	CW, winding data I/O code (1 by default).
INTGAR(6)	CZ, impedance data I/O code (1 by default).
INTGAR(7)	CM, magnetizing admittance data I/O code (1 by default).
INTGAR(8)	STAT, branch status (1 by default).
INTGAR(9)	NMETBS, non-metered end bus number (I, J or K) (J by default).
INTGAR(10)	WN1BUS, winding one side bus number (I, J or K) (I by default).
INTGAR(11)	WN2BUS, winding two side bus number (I, J or K) (J by default).
INTGAR(12)	WN3BUS, winding three side bus number (I, J or K) (K by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real REALAR(23)	Is an array of 23 elements (first seventeen are input values; last six are output values).	
	REALAR(1)	R1-2, nominal bus one to two transformer resistance (0.0 by default).
	REALAR(2)	X1-2, nominal bus one to two transformer reactance (0.0002 by default).
	REALAR(3)	R2-3, nominal bus two to three transformer resistance (0.0 by default).
	REALAR(4)	X2-3, nominal bus two to three transformer reactance (0.0002 by default).
	REALAR(5)	R3-1, nominal bus three to one transformer resistance (0.0 by default).
	REALAR(6)	X3-1, nominal bus three to one transformer reactance (0.0002 by default).
	REALAR(7)	SBS1-2, winding one to two base MVA (SBASE by default).
	REALAR(8)	SBS2-3, winding two to three base MVA (SBASE by default).
	REALAR(9)	SBS3-1, winding three to one base MVA (SBASE by default).
	REALAR(10)	MAG1, magnetizing conductance or no-load losses (0.0 by default).
	REALAR(11)	MAG2, magnetizing susceptance or exciting current (0.0 by default).
	REALAR(12)	F1, first owner fraction (1.0 by default).
	REALAR(13)	F2, second owner fraction (0.0 by default).
	REALAR(14)	F3, third owner fraction (0.0 by default).
	REALAR(15)	F4, fourth owner fraction (0.0 by default).
	REALAR(16)	VMSTAR, star bus voltage magnitude (1.0 by default).
	REALAR(17)	ANSTAR, star bus voltage angle (0.0 by default).
	REALAR(18)	actual bus one to two resistance (**returned**).
	REALAR(19)	actual bus one to two reactance (**returned**).
	REALAR(20)	actual bus two to three resistance (**returned**).
	REALAR(21)	actual bus two to three reactance (**returned**).
	REALAR(22)	actual bus three to one resistance (**returned**).
	REALAR(23)	actual bus three to one reactance (**returned**).
Real REALARI(17)	Is an array of seventeen elements corresponding to input values REALAR(1) through REALAR(17) as described above.	
Real REALARO(6)	Is an array of six elements corresponding to output values REALAR(18) through REALAR(23) as described above.	
Character NAME*12	Is the transformer name (input; blank by default).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error occurred
IERR > 0	fatal error occurred
IERR = 1	bus not found
IERR = 2	circuit identifier is more than two characters
IERR = 3	invalid CW, CZ and/or CM value
IERR = 4	CW, CZ and/or CM >1 but at least one endpoint bus has no base voltage
IERR = 5	three different buses must be specified
IERR = 6	blank circuit identifier is invalid
IERR = 7	transformer circuit identifier may not start with &, * or @.
IERR = 8	three-winding transformer table is full
IERR = 9	bus table is full
IERR = 10	not enough branch table entries are available
IERR = 11	not enough two-winding transformer table entries are available.
IERR = 12	not enough branch ownership table entries are available.
IERR = 14	prerequisite requirements for API are not met.
IERR = -1	data error, warning and information messages; one or more of: • Invalid set of winding buses. • Invalid CW, CZ and/or CM data value. • Invalid transformer status. • Invalid non-metered end bus. • Winding side buses interchanged. • Losses are too big relative to total impedance at winding base MVA. • Magnitude of winding impedance is less than 0.00001. • Any two-winding transformer warnings.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.66 THREE_WND_WINDING_DATA

This API is obsolete. It has been replaced by the API routine THREE_WND_WINDING_DATA_3, and is implemented by a call to the THREE_WND_WINDING_DATA_3 API routine.

Use this API to modify the data of one winding of an existing three-winding transformer in the working case (see *PSS®E Program Operation Manual*, Non-Transformer Branch Data).

Batch command syntax:

```
BAT_THREE_WND_WINDING_DATA  I  J  K  'CKT'  WARG
INTGAR(1)...INTGAR(5)  REALAR(1)...REALAR(12)
```

Python syntax:

```
ierr,realaro = three_wnd_winding_data(i, j, k, ckt, warg, intgar,
realari)
```

Fortran syntax:

```
CALL THREE_WINDING_WINDING_API_2(I, J, K, CKT, WARG, INTGAR, REALAR,
IERR)
```

where:

Integer I	Is the bus number of one of the buses (input; no default allowed).
Integer J	Is the bus number one of the other buses (input; no default allowed).
Integer K	Is the bus number of the third bus (input; no default allowed).
Character CKT*2	Is the transformer circuit identifier (input; ’1’ by default).
Integer WARG	Is the winding number (1, 2 or 3) (input; no default allowed).
Integer INTGAR(5)	Is an array of five elements (input).
	INTGAR(1) — NTPi, number of tap positions (33 by default).
	INTGAR(2) — TABi, impedance correction table number (0 by default).
	INTGAR(3) — CONTi, controlled bus number (0 by default).
	INTGAR(4) — SICODi, negative for controlled bus on winding bus side (1 by default).
	INTGAR(5) — CODi, adjustment control mode flag (-3 through +3) (0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real REALAR(16) Is an array of sixteen elements (first twelve are input values; last four are output values).

REALAR(1)	WINDVi, winding ratio or voltage (1.0 by default if CW of this transformer is 1 or 3; base voltage of the winding bus by default if CW is 2).
REALAR(2)	NOMVi, winding nominal voltage (0.0 by default).
REALAR(3)	ANGi, winding phase shift angle (0.0 by default).
REALAR(4)	RATAi, winding rating set A line rating (0.0 by default).
REALAR(5)	RATBi, winding rating set B line rating (0.0 by default).
REALAR(6)	RATCi, winding rating set C line rating (0.0 by default).
REALAR(7)	RMAi, winding ratio/angle high limit (1.1 by default).
REALAR(8)	RMIi, winding ratio/angle low limit (0.9 by default).
REALAR(9)	VMAi, winding voltage or flow upper limit (1.1 by default).
REALAR(10)	VMIi, winding voltage or flow lower limit (0.9 by default).
REALAR(11)	CRi, winding load drop compensating resistance (0.0 by default).
REALAR(12)	CXi, winding load drop compensating reactance (0.0 by default).
REALAR(13)	winding nominal resistance (**returned**).
REALAR(14)	winding nominal reactance (**returned**).
REALAR(15)	winding actual resistance (**returned**).
REALAR(16)	winding actual reactance (**returned**).

Real REALARI(12) Is an array of twelve elements corresponding to input values REALAR(1) through REALAR(12) as described above.

Real REALARO(4) Is an array of four elements corresponding to output values REALAR(13) through REALAR(16) as described above.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error occurred.
IERR > 0	fatal error occurred.
IERR = 1	bus not found.
IERR = 2	circuit identifier is more than two characters.
IERR = 3	invalid CW, CZ and/or CM value.
IERR = 4	CW and/or CZ >1 but no base voltage at winding bus.
IERR = 5	three winding transformer not found.
IERR = 6	invalid winding number.
IERR = 7	prerequisite requirements for API are not met.
IERR = -1	data error, warning and information messages; one or more of: • Controlled bus not found. • Invalid adjustment control mode flag. • Any two-winding transformer warnings.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.67 THREE_WND_WINDING_DATA_3

Use this API routine to run the third release of the three-winding transformer winding data function. This API routine is used to modify the data of one winding of an existing three-winding transformer in the working case (see *PSS®E Program Operation Manual*, Non-Transformer Branch Data).

Batch command syntax:

```
BAT_THREE_WND_WINDING_DATA_3  I  J  K  'CKT'  WARG
INTGAR(1)...INTGAR(5)  REALAR(1)...REALAR(12)
```

Python syntax:

```
ierr,realaro = three_wnd_winding_data_3(i, j, k, ckt, warg, intgar,
realari)
```

Fortran syntax:

```
CALL THREE_WINDING_WINDING_API_3(I, J, K, CKT, WARG, INTGAR, REALAR,
IERR)
```

where:

Integer I	Is the bus number of one of the buses (input; no default allowed).
Integer J	Is the bus number one of the other buses (input; no default allowed).
Integer K	Is the bus number of the third bus (input; no default allowed).
Character CKT*2	Is the transformer circuit identifier (input; ’1’ by default).
Integer WARG	Is the winding number (1, 2 or 3) (input; no default allowed).
Integer INTGAR(5)	Is an array of five elements (input).
	INTGAR(1) — NTPi, number of tap positions (33 by default).
	INTGAR(2) — TABi, impedance correction table number (0 by default).
	INTGAR(3) — CONTi, controlled bus number (0 by default).
	INTGAR(4) — SICODi, negative for controlled bus on winding bus side (1 by default).
	INTGAR(5) — CODi, adjustment control mode flag (-3 through +3, -5 or 5) (0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real REALAR(17) Is an array of seventeen elements (first thirteen are input values; last four are output values).

REALAR(1)	WINDVi, winding ratio or voltage (1.0 by default if CW of this transformer is 1 or 3; base voltage of the winding bus by default if CW is 2).
REALAR(2)	NOMVi, winding nominal voltage (0.0 by default).
REALAR(3)	ANGi, winding phase shift angle (0.0 by default).
REALAR(4)	RATAi, winding rating set A line rating (0.0 by default).
REALAR(5)	RATBi, winding rating set B line rating (0.0 by default).
REALAR(6)	RATCi, winding rating set C line rating (0.0 by default).
REALAR(7)	RMAi, winding ratio/angle high limit (1.1 by default).
REALAR(8)	RMIi, winding ratio/angle low limit (0.9 by default).
REALAR(9)	VMAi, winding voltage or flow upper limit (1.1 by default).
REALAR(10)	VMIi, winding voltage or flow lower limit (0.9 by default).
REALAR(11)	CRi, winding load drop compensating resistance (0.0 by default).
REALAR(12)	CXi, winding load drop compensating reactance (0.0 by default).
REALAR(13)	CNXAi, winding connection angle (0.0 by default). Used with adjustment control mode 5 (unsymmetric phase shift control of active power) implemented in PSS®E version 32.1.
REALAR(14)	winding nominal resistance (**returned**).
REALAR(15)	winding nominal reactance (**returned**).
REALAR(16)	winding actual resistance (**returned**).
REALAR(17)	winding actual reactance (**returned**).

Real REALARI(13) Is an array of thirteen elements corresponding to input values REALAR(1) through REALAR(13) as described above.

Real REALARO(4) Is an array of four elements corresponding to output values REALAR(14) through REALAR(17) as described above.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error occurred.
IERR > 0	fatal error occurred.
IERR = 1	bus not found.
IERR = 2	circuit identifier is more than two characters.
IERR = 3	invalid CW, CZ and/or CM value.
IERR = 4	CW and/or CZ >1 but no base voltage at winding bus.
IERR = 5	three winding transformer not found.
IERR = 6	invalid winding number.
IERR = 7	prerequisite requirements for API are not met.
IERR = -1	data error, warning and information messages; one or more of: • Controlled bus not found. • Invalid adjustment control mode flag. • Any two-winding transformer warnings.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.68 TRANSACTION_ADD_MOD

Use this API to add or modify transaction event data in PSS®E working memory.

Batch command syntax:

```
BAT_TRANSACTION_ADD_MOD  EVENT  INTGAR(1)  INTGAR(2)  REALAR(1)
REALAR(2)  'DSCRP'
```

Python syntax:

```
ierr = transaction_add_mod(event, intgar, realar, dscrp)
```

Fortran syntax:

```
CALL TRANSACTION_ADD_MOD(EVENT, INTGAR, REALAR, DSCRP, IERR, NERR)
```

where:

Integer EVENT	Is the transaction event number.	
Integer INTGAR(2)	Is an array of two elements.	
	INTGAR(1)	transaction event status (0 = out-of-service, 1 = in-service).
	INTGAR(2)	transaction event priority.
Real REALAR(2)	Is an array of two elements.	
	REALAR(1)	transaction event magnitude in MW.
	REALAR(2)	transaction event curtailment in MW.
Character DSCRP*24	Is the transaction event description.	
Integer IERR	Is the error code.	
	IERR = 0	no error occurred.
	IERR > 0	error occurred.
	IERR = 2	invalid transaction event number.
	IERR > 2	unable to add or modify transaction event data: error condition detected in data access routine.
	IERR = 10	prerequisite requirements for API are not met.
	IERR = -1	warnings: • None.
Integer NERR	Is the number of error entries in PBUF02 (output).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.69 TRANSFER_CHNG

Use this API routine to modify the data of an inter-area transfer in the working case (see *PSS®E Program Operation Manual*, Zone Data).

Batch command syntax:

```
BAT_TRANSFER_CHNG  APPLYF  ARFROM  ARTO  'TRID'  REALAR(1)
```

Python syntax:

```
ierr = transfer_chng(applyf, arfrom, arto, trid, realar)
```

Fortran syntax:

```
CALL TRANSFER_chng(APPLYF, ARFROM, ARTO, TRID, REALAR, IERR)
```

where:

Integer APPLYF	APPLYF = 1	to modify from and to area desired net interchange entries (DESINT).
	APPLYF = 2	to leave DESINT entries unchanged.
	(Input; no default allowed.)	
Integer ARFROM	Is the from area number (must be an area already defined) (input; no default allowed).	
Integer ARTO	Is the to area number (must be an area already defined) (input; no default allowed).	
Character TRID*1	Is the inter-area transfer identifier (input; ’1’ by default).	
Real REALAR(1)	Is an array of one element (input). See the TRANSFER_DATA API routine for details.	
Integer IERR	Is the error code (output).	
	IERR = 0	no errors occurred
	IERR > 0	fatal error occurred
	IERR = 1	invalid area number
	IERR = 2	inter-area transfer identifier is more than one character
	IERR = 3	invalid APPLYF value
	IERR = 8	area not found
	IERR = 9	inter-area transfer not found
	IERR = 10	prerequisite requirements for API are not met
	IERR = -1	data error, warning and information messages; one or more of: • Changes to desired area net interchanges if APPLYF = 1.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.70 TRANSFER_DATA

Use this API routine to modify the data of an inter-area transfer in the working case, or to add a new inter-area transfer to the working case (see *PSS®E Program Operation Manual*, Zone Data).

Batch command syntax:

```
BAT_TRANSFER_DATA  APPLYF  ARFROM  ARTO  'TRID'  REALAR(1)
```

Python syntax:

```
ierr = transfer_data(applyf, arfrom, arto, trid, realar)
```

Fortran syntax:

```
CALL TRANSFER_API_2(APPLYF, ARFROM, ARTO, TRID, REALAR, IERR)
```

where:

Integer APPLYF	APPLYF = 1	to modify from and to area desired net interchange entries (DESINT).
	APPLYF = 2	to leave DESINT entries unchanged.
	(Input; no default allowed.)	
Integer ARFROM	Is the from area number (must be an area already defined) (input; no default allowed).	
Integer ARTO	Is the to area number (must be an area already defined) (input; no default allowed).	
Character TRID*1	Is the inter-area transfer identifier (input; ’1’ by default).	
Real REALAR(1)	Is an array of one element (input).	
	REALAR(1)	PTRAN, MW comprising this inter-area transfer (0.0 by default).
Integer IERR	Is the error code (output).	
	IERR = 0	no errors occurred
	IERR > 0	fatal error occurred
	IERR = 1	invalid area number
	IERR = 2	inter-area transfer identifier is more than one character
	IERR = 3	invalid APPLYF value
	IERR = 4	invalid inter-area transfer identifier
	IERR = 5	transfer from an area to itself
	IERR = 6	inter-area transfer table is full
	IERR = 7	inter-area transfer already exists
	IERR = 8	area not found
	IERR = 10	prerequisite requirements for API are not met
	IERR = -1	data error, warning and information messages; one or more of: • Changes to desired area net interchanges if APPLYF = 1.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.71 TWO_TERM_DC_CONVR_DATA

Use this API to modify the data of a dc converter of an existing two-terminal dc line in the working case (see *PSS®E Program Operation Manual*, Area Interchange Data). The dc line must have previously been specified via the TWO_TERMINAL_DC_LINE_DATA API.

Batch command syntax:

```
BAT_TWO_TERM_DC_CONVR_DATA  CNVFLG  DC  INTGAR(1)...INTGAR(5)
REALAR(1)...REALAR(11)  'ID'
```

Python syntax:

```
ierr,realaro = two_term_dc_convr_data(cnvflg, dc, intgar, realari,
id)
```

Fortran syntax:

```
CALL TWO_TERMINAL_DC_CONVERTER_API  (CNVFLG, NUM, INTGAR, REALAR, ID,
IERR, NERR)

CALL TWO_TERMINAL_DC_CONVERTER_API_I(CNVFLG, NUM, INTGAR, REALAR, ID,
IERR)

CALL TWO_TERMINAL_DC_CONVERTER_API_C(CNVFLG, NAME, INTGAR, REALAR,
ID, IERR)
```

where:

Integer CNVFLG	Is the converter type (input; no default allowed). CNVFLG = 1 for rectifier data. CNVFLG = 2 for inverter data.
Character*12 or Integer DC	Is the quoted name of the two-terminal dc line for which data for one of its converters is to be specified; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the two-terminal dc line for which data for one of its converters is to be specified (input; no default allowed).
Character*12 NAME	Is the name of the two-terminal dc line for which data for one of its converters is to be specified (input; no default allowed).
Integer INTGAR(5)	Is an array of five elements (input). INTGAR(1) IPR or IPI, converter bus number (no default allowed). INTGAR(2) NBR or NBI, number of bridges in series (0 by default). INTGAR(3) ICR or ICI, zero or bus number of firing angle bus (0 by default). INTGAR(4) IFR or IFI, zero or bus number of winding one side of a two-winding ac transformer (0 by default). INTGAR(5) ITR or ITI, zero or bus number of winding two side of a two-winding ac transformer (0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real REALAR(15) Is an array of fifteen elements (first eleven are input values; last four are output values).

REALAR(1)	ANMNR or ANMNI, minimum firing angle (90.0 by default).
REALAR(2)	ANMXR or ANMXI, maximum firing angle (0.0 by default).
REALAR(3)	RCR or RCI, commutating resistance (0.0 by default).
REALAR(4)	XCR or XCI, commutating reactance (0.0 by default).
REALAR(5)	EBASR or EBASI, primary base ac voltage (0.0 by default).
REALAR(6)	TRR or TRI, transformer ratio (1.0 by default).
REALAR(7)	TAPR or TAPI, tap setting (1.0 by default).
REALAR(8)	TMXR or TMXI, maximum tap setting (1.5 by default).
REALAR(9)	TMNR or TMNI, minimum tap setting (0.51 by default).
REALAR(10)	STPR or STPI, tap step (0.00625 by default).
REALAR(11)	XCAPR or XCAPI, capacitive reactance of CCC type converter (0.0 by default).
REALAR(12)	firing angle (**returned**).
REALAR(13)	dc voltage (kV) (**returned**).
REALAR(14)	real power into converter (PWRCOD) (**returned**).
REALAR(15)	reactive power into converter (PWRCOD) (**returned**).

Real REALARI(11) Is an array of eleven elements corresponding to input values REALAR(1) through REALAR(11) as described above.

Real REALARO(4) Is an array of four elements corresponding to output values REALAR(12) through REALAR(15) as described above.

Character ID*2 Is an ac transformer circuit identifier, IDR or IDI (input; ’1’ by default).

Integer IERR Is the error code (output).

IERR = 0	no errors occurred
IERR > 0	fatal error occurred
IERR = 1	dc line not found
IERR = 2	converter type is not 1 or 2
IERR = 3	transformer circuit identifier is more than two characters
IERR = 4	no converter bus is specified
IERR = 5	converter bus not found
IERR = 6	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IERR = -1 data error, warning and information messages; one or more of:

- Converter bus is dummy bus of a multi-section line.
- Multi-section line grouping deleted.
- Converter maximum angle is less than its minimum angle.
- Converter maximum tap is less than its minimum tap.
- Invalid tap step.
- Commutating bus not found.
- ac transformer bus not found.
- Specified branch not found.
- Specified branch is not a transformer.
- From bus of specified branch is not winding one side bus.

Integer NERR Is the number of entries in PBUF02 containing error messages.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.72 TWO_TERMINAL_DC_LINE_CHNG

Use this API routine to modify the link data of an existing two-terminal dc line in the working case (see *PSS®E Program Operation Manual*, Area Interchange Data). Converter data for the rectifier and inverter are specified via the TWO_TERM_DC_CONVR_DATA API routine.

Batch command syntax:

```
BAT_TWO_TERMINAL_DC_LINE_CHNG  DC  INTGAR(1)  INTGAR(2)
REALAR(1)...REALAR(8)  'METER'
```

Python syntax:

```
ierr,realaro = two_terminal_dc_line_chng(dc, intgar, realari, meter)
```

Fortran syntax:

```
CALL TWO_TERMINAL_DC_LINE_CHNG_I(NUM, INTGAR, REALAR, METER, IERR)

CALL TWO_TERMINAL_DC_LINE_CHNG_C(NAME, INTGAR, REALAR, METER, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the two-terminal dc line for which data is to be specified; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the two-terminal dc line for which data is to be specified (input; no default allowed).
Character*12 NAME	Is the name of the two-terminal dc line for which data is to be specified (input; no default allowed).
Integer INTGAR(2)	Is an array of two elements (input). See the TWO_TERMINAL_DC_LINE_DATA API routine for details.
Real REALAR(9)	Is an array of nine elements (first eight are input values; last one is output value). See the TWO_TERMINAL_DC_LINE_DATA API routine for details.
Real REALARI(8)	Is an array of eight elements corresponding to input values REALAR(1) through REALAR(8) above.
Real REALARO(1)	Is an array of one element corresponding to output value REALAR(9) above.
Character METER*1	Is the metered end flag: 'R' or 'I' (input; 'I' by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no errors occurred
IERR > 0	fatal error occurred
IERR = 2	metered end code is more than one character
IERR = 3	blank dc line name is invalid
IERR = 4	two-terminal dc line not found
IERR = 5	prerequisite requirements for API are not met
IERR = -1	data error, warning and information messages; one or more of: • invalid dc line control mode • invalid dc line resistance • invalid compounding resistance • compounding resistance exceeds line resistance • scheduled voltage must exceed mode switch voltage • invalid metered end code.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.73 TWO_TERMINAL_DC_LINE_DATA

Use this API routine to modify the link data of an existing two-terminal dc line in the working case, or to add a new two-terminal dc line and its link data to the working case (see *PSS®E Program Operation Manual*, Area Interchange Data). Converter data for the rectifier and inverter are specified via the TWO_TERM_DC_CONVR_DATA API routine.

Batch command syntax:

```
BAT_TWO_TERMINAL_DC_LINE_DATA  DC  INTGAR(1)  INTGAR(2)
REALAR(1)...REALAR(8)  'METER'
```

Python syntax:

```
ierr,realaro = two_terminal_dc_line_data(dc, intgar, realari, meter)
```

Fortran syntax:

```
CALL TWO_TERMINAL_DC_LINE_API  (NUM, INTGAR, REALAR, METER, IERR,
NERR)

CALL TWO_TERMINAL_DC_LINE_API_I(NUM, INTGAR, REALAR, METER, IERR)

CALL TWO_TERMINAL_DC_LINE_API_C(NAME, INTGAR, REALAR, METER, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the two-terminal dc line for which data is to be specified; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the two-terminal dc line for which data is to be specified (input; no default allowed).
Character*12 NAME	Is the name of the two-terminal dc line for which data is to be specified (input; no default allowed).
Integer INTGAR(2)	Is an array of two elements (input). INTGAR(1) MDC, control mode (0, 1 or 2) (0 by default). INTGAR(2) CCITMX, CCC Newton solution iteration limit (20 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real REALAR(9)	Is an array of nine elements (first eight are input values; last one is output value). REALAR(1) SETVL, scheduled current (amps) or power (MW) demand (0.0 by default). REALAR(2) VSCHD, scheduled dc voltage (kV) (1.0 by default). REALAR(3) VCMOD, mode switch dc voltage (kV) (0.0 by default). REALAR(4) DELTI, margin (pu, > 0) (0.0 by default). REALAR(5) RDC, dc line resistance (0.0 by default). REALAR(6) RCOMP, compounding resistance (0.0 by default). REALAR(7) DCVMIN, minimum dc voltage (kV) (0.0 by default). REALAR(8) CCCACC, CCC Newton solution acceleration (1.0 by default). REALAR(9) dc line current (amps) (**returned**).
Real REALARI(8)	Is an array of eight elements corresponding to input values REALAR(1) through REALAR(8) as described above.
Real REALARO(1)	Is an array of one element corresponding to output value REALAR(9) as described above.
Character METER*1	Is the metered end flag: 'R' or 'I' (input; 'I' by default).
Integer IERR	Is the error code (output). IERR = 0 no errors occurred IERR > 0 fatal error occurred IERR = 1 two-terminal dc line table is full IERR = 2 metered end code is more than one character IERR = 3 blank dc line name is invalid IERR = 5 prerequisite requirements for API are not met IERR = -1 data error, warning and information messages; one or more of: • invalid dc line control mode • invalid dc line resistance • invalid compounding resistance • compounding resistance exceeds line resistance • scheduled voltage must exceed mode switch voltage • invalid metered end code.
Integer NERR	Is the number of entries in PBUF02 containing error messages.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.74 TWO_WINDING_CHNG_4

This API routine is the fourth release of the two-winding transformer data specification function. Use this API routine to modify the data of an existing two-winding transformer in the working case (see *PSS®E Program Operation Manual*, Owner Data).

Batch command syntax:

```
BAT_TWO_WINDING_CHNG_4  I  J  'CKT'  INTGAR(1)...INTGAR(15)
REALAR(1)...REALAR(24)  'CHARAR(1)'  'CHARAR(2)'
```

Python syntax:

```
ierr,realaro = two_winding_chng_4(i, j, ckt, intgar, realari, charar)
```

Fortran syntax:

```
CALL TWO_WINDING_CHNG_4(I, J, CKT, INTGAR, REALAR, CJARAR, IERR)
```

where:

Integer I	Is the bus number of from bus (input; no default allowed).
Integer J	Is the bus number of to bus (input; no default allowed).
Character CKT*2	Is the circuit identifier (input; ’1’ by default).
Integer INTGAR(15)	Is an array of fifteen elements (input). See the TWO_WINDING_DATA_4 API routine for details.
Real REALAR(26)	Is an array of 26 elements (first twenty-four are input values; last two are output values). See the TWO_WINDING_DATA_4 API routine for details.
Real REALARI(24)	Is an array of 24 elements corresponding to input values REALAR(1) through REALAR(24) above.
Real REALARO(2)	Is an array of two elements corresponding to output values REALAR(25) through REALAR(26) above.
Character CHARAR(2)*12	Is an array of two elements (input). See the TWO_WINDING_DATA_4 API routine for details.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code.

IERR = 0	no errors occurred
IERR > 0	fatal error occurred
IERR = 1	bus not found
IERR = 2	circuit identifier is more than two characters
IERR = 3	branch exists but it is a non-transformer branch
IERR = 10	CW, CZ and/or CM >1 but at least one endpoint bus has no base voltage
IERR = 11	invalid CW, CZ and/or CM value
IERR = 12	two-winding transformer not found
IERR = 13	prerequisite requirements for API are not met.
IERR = -1	data error, warning and information messages; one or more of:

- invalid metered end bus
- invalid transformer status
- invalid owner number
- invalid ownership fraction
- no ownership data specified.
- multi-section line grouping deleted
- branch ownership table is full--only first n owners retained
- owner table is full
- invalid winding 1 side bus number
- invalid CW, CZ and/or CM data value
- controlled bus not found
- invalid adjustment control mode flag
- invalid impedance correction table number
- invalid number of tap positions
- winding nominal voltage is negative
- winding bus base voltage and nominal voltage differ by more than 20%
- invalid winding voltage
- winding voltage and bus base voltage differ by more than 50%
- RMAX is greater than 150% of bus base voltage
- RMIN is less than 50% of bus base voltage
- invalid tap ratio
- tap ratio is outside the band 0.5 to 1.5 pu
- RMAX is greater than 1.5
- RMIN is less than 0.5
- RMAX is less than RMIN
- VMAX is less than VMIN

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

- phase shift angle normalized between -180 and 180
- winding base MVA must be positive
- MAG1 is too big relative to MAG2 at winding base MVA
- magnetizing susceptance MAG2 is positive
- losses are too big relative to total impedance at winding base MVA
- magnitude of reactance is less than 0.000001
- winding 1 and winding 2 sides are interchanged (any two-terminal dc lines controlled by this transformer are logged).
- winding connection angle normalized between -360 and 360
- adjustment control mode flag and the winding connection angle are inconsistent.
- for asymmetric phase shifter, abnormally large or small tap setting results at RMA1 (or RMI1) and CNXA1.
- for asymmetric phase shifter, the specified and derived winding tap settings differ.
- Vector group not found.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.75 TWO_WINDING_DATA

This API routine is obsolete. It has been replaced by the API routine TWO_WINDING_DATA_4, and is implemented by a call to the TWO_WINDING_DATA_4 API routine.

Use this API routine to modify the data of an existing two-winding transformer in the working case, or to add a new two-winding transformer to the working case (see *PSS®E Program Operation Manual*, Owner Data).

Batch command syntax:

```
BAT_TWO_WINDING_DATA  I  J  'CKT'  INTGAR(1)...INTGAR(15)
REALAR(1)...REALAR(23)  'NAME'
```

Python syntax:

```
ierr,realaro = two_winding_data(i, j, ckt, intgar, realari, name)
```

Fortran syntax:

```
CALL TWO_WINDING_API_2(I, J, CKT, INTGAR, REALAR, NAME, IERR)
```

where:

Integer I	Is the bus number of from bus (input; no default allowed).
Integer J	Is the bus number of to bus (input; no default allowed).
Character CKT*2	Is the circuit identifier (input; ’1’ by default).
Integer INTGAR(15)	Is an array of fifteen elements (input).

INTGAR(1)	STAT, branch status (1 by default).
INTGAR(2)	METBUS, metered end bus number (I or J) (I by default).
INTGAR(3)	O1, first owner number (owner of bus I by default).
INTGAR(4)	O2, second owner number (0 by default).
INTGAR(5)	O3, third owner number (0 by default).
INTGAR(6)	O4, fourth owner number (0 by default).
INTGAR(7)	NTP1, number of tap positions (33 by default).
INTGAR(8)	TAB1, impedance correction table number (0 by default).
INTGAR(9)	WN1BUS, winding one side bus number (I or J) (I by default).
INTGAR(10)	CONT1, controlled bus number (0 by default).
INTGAR(11)	SICOD1, negative for controlled bus on winding one side (1 by default).
INTGAR(12)	COD1, adjustment control mode flag (-4 through +4) (0 by default).
INTGAR(13)	CW, winding data I/O code (1 by default).
INTGAR(14)	CZ, impedance data I/O code (1 by default).
INTGAR(15)	CM, magnetizing admittance data I/O code (1 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real REALAR(25) Is an array of 25 elements (first twenty-three are input values; last two are output values).

REALAR(1)	R1-2, nominal transformer resistance (0.0 by default).
REALAR(2)	X1-2, nominal transformer reactance (THRSHZ by default; 0.0001 if THRSHZ = 0.0).
REALAR(3)	SBS1-2, winding base MVA (SBASE by default).
REALAR(4)	WINDV1, winding one ratio/voltage (1.0 by default if CW is 1 or 3; base voltage of bus I by default if CW is 2).
REALAR(5)	NOMV1, winding one nominal voltage (0.0 by default).
REALAR(6)	ANG1, winding one phase shift angle (0.0 by default).
REALAR(7)	WINDV2, winding two ratio/voltage (1.0 by default if CW is 1 or 3; base voltage of bus J by default if CW is 2).
REALAR(8)	NOMV2, winding two nominal voltage (0.0 by default).
REALAR(9)	RATA1, rating set A line rating (0.0 by default).
REALAR(10)	RATB1, rating set B line rating (0.0 by default).
REALAR(11)	RATC1, rating set C line rating (0.0 by default).
REALAR(12)	F1, first owner fraction (1.0 by default).
REALAR(13)	F2, second owner fraction (1.0 by default).
REALAR(14)	F3, third owner fraction (1.0 by default).
REALAR(15)	F4, fourth owner fraction (1.0 by default).
REALAR(16)	MAG1, magnetizing conductance or no-load losses (0.0 by default).
REALAR(17)	MAG2, magnetizing susceptance or exciting current (0.0 by default).
REALAR(18)	RMA1, winding one ratio/angle high limit (1.1 by default).
REALAR(19)	RMI1, winding one ratio/angle low limit (0.9 by default).
REALAR(20)	VMA1, voltage or flow upper limit (1.1 by default).
REALAR(21)	VMI1, voltage or flow lower limit (0.9 by default).
REALAR(22)	CR1, load drop compensating resistance (0.0 by default).
REALAR(23)	CX1, load drop compensating reactance (0.0 by default).
REALAR(24)	actual transformer resistance (**returned**).
REALAR(25)	actual transformer reactance (**returned**).

Real REALARI(23) Is an array of 23 elements corresponding to input values REALAR(1) through REALAR(23) as described above.

Real REALARO(2) Is an array of two elements corresponding to output values REALAR(24) through REALAR(25) as described above.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character NAME*12	Is the transformer name (input; blank by default).	
Integer IERR	Is the error code.	
	IERR = 0	no errors occurred
	IERR > 0	fatal error occurred
	IERR = 1	bus not found
	IERR = 2	circuit identifier is more than two characters
	IERR = 3	branch exists but it is a non-transformer branch
	IERR = 4	branch from a bus to itself
	IERR = 5	blank circuit identifier is invalid
	IERR = 6	circuit identifier may not start with &, * or @
	IERR = 7	branch table is full
	IERR = 8	branch ownership table is full
	IERR = 9	two-winding transformer table is full
	IERR = 10	CW, CZ and/or CM >1 but at least one endpoint bus has no base voltage
	IERR = 11	invalid CW, CZ and/or CM value
	IERR = 13	prerequisite requirements for API are not met.
	IERR = -1	data error, warning and information messages; one or more of: • invalid metered end bus • invalid transformer status • invalid owner number • invalid ownership fraction • no ownership data specified. • multi-section line grouping deleted • branch ownership table is full--only first n owners retained • owner table is full • invalid winding 1 side bus number • invalid CW, CZ and/or CM data value • controlled bus not found • invalid adjustment control mode flag • invalid impedance correction table number • invalid number of tap positions • winding nominal voltage is negative • winding bus base voltage and nominal voltage differ by more than 20% • invalid winding voltage • winding voltage and bus base voltage differ by more than 50% • RMAX is greater than 150% of bus base voltage • RMIN is less than 50% of bus base voltage • invalid tap ratio • tap ratio is outside the band 0.5 to 1.5 pu

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

- RMAX is greater than 1.5
- RMIN is less than 0.5
- RMAX is less than RMIN
- VMAX is less than VMIN
- phase shift angle normalized between -180 and 180
- winding base MVA must be positive
- MAG1 is too big relative to MAG2 at winding base MVA
- losses are too big relative to total impedance at winding base MVA
- magnitude of reactance is less than 0.000001
- winding 1 and winding 2 sides are interchanged (any two-terminal dc lines controlled by this transformer are logged).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.76 TWO_WINDING_DATA_3

This API routine is obsolete. It has been replaced by the API routine TWO_WINDING_DATA_4, and is implemented by a call to the TWO_WINDING_DATA_4, API routine.

This API routine is the third release of the two-winding transformer data specification function. Use this API routine to modify the data of an existing two-winding transformer in the working case, or to add a new two-winding transformer to the working case (see *PSS®E Program Operation Manual*, Owner Data).

Batch command syntax:

```
BAT_TWO_WINDING_DATA_3  I  J  'CKT'  INTGAR(1)...INTGAR(15)
REALAR(1)...REALAR(24)  'NAME'
```

Python syntax:

```
ierr,realaro = two_winding_data_3(i, j, ckt, intgar, realari, name)
```

Fortran syntax:

```
CALL TWO_WINDING_API_3(I, J, CKT, INTGAR, REALAR, NAME, IERR)
```

where:

Integer I	Is the bus number of from bus (input; no default allowed).
Integer J	Is the bus number of to bus (input; no default allowed).
Character CKT*2	Is the circuit identifier (input; ’1’ by default).
Integer INTGAR(15)	Is an array of fifteen elements (input).

INTGAR(1)	STAT, branch status (1 by default).
INTGAR(2)	METBUS, metered end bus number (I or J) (I by default).
INTGAR(3)	O1, first owner number (owner of bus I by default).
INTGAR(4)	O2, second owner number (0 by default).
INTGAR(5)	O3, third owner number (0 by default).
INTGAR(6)	O4, fourth owner number (0 by default).
INTGAR(7)	NTP1, number of tap positions (33 by default).
INTGAR(8)	TAB1, impedance correction table number (0 by default).
INTGAR(9)	WN1BUS, winding one side bus number (I or J) (I by default).
INTGAR(10)	CONT1, controlled bus number (0 by default).
INTGAR(11)	SICOD1, negative for controlled bus on winding one side (1 by default).
INTGAR(12)	COD1, adjustment control mode flag (-5 through +5) (0 by default).
INTGAR(13)	CW, winding data I/O code (1 by default).
INTGAR(14)	CZ, impedance data I/O code (1 by default).
INTGAR(15)	CM, magnetizing admittance data I/O code (1 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real REALAR(26) Is an array of 26 elements (first twenty-four are input values; last two are output values).

REALAR(1)	R1-2, nominal transformer resistance (0.0 by default).
REALAR(2)	X1-2, nominal transformer reactance (THRSHZ by default; 0.0001 if THRSHZ = 0.0).
REALAR(3)	SBS1-2, winding base MVA (SBASE by default).
REALAR(4)	WINDV1, winding one ratio/voltage (1.0 by default if CW is 1 or 3; base voltage of bus I by default if CW is 2).
REALAR(5)	NOMV1, winding one nominal voltage (0.0 by default).
REALAR(6)	ANG1, winding one phase shift angle (0.0 by default).
REALAR(7)	WINDV2, winding two ratio/voltage (1.0 by default if CW is 1 or 3; base voltage of bus J by default if CW is 2).
REALAR(8)	NOMV2, winding two nominal voltage (0.0 by default).
REALAR(9)	RATA1, rating set A line rating (0.0 by default).
REALAR(10)	RATB1, rating set B line rating (0.0 by default).
REALAR(11)	RATC1, rating set C line rating (0.0 by default).
REALAR(12)	F1, first owner fraction (1.0 by default).
REALAR(13)	F2, second owner fraction (1.0 by default).
REALAR(14)	F3, third owner fraction (1.0 by default).
REALAR(15)	F4, fourth owner fraction (1.0 by default).
REALAR(16)	MAG1, magnetizing conductance or no-load losses (0.0 by default).
REALAR(17)	MAG2, magnetizing susceptance or exciting current (0.0 by default).
REALAR(18)	RMA1, winding one ratio/angle high limit (1.1 by default).
REALAR(19)	RMI1, winding one ratio/angle low limit (0.9 by default).
REALAR(20)	VMA1, voltage or flow upper limit (1.1 by default).
REALAR(21)	VMI1, voltage or flow lower limit (0.9 by default).
REALAR(22)	CR1, load drop compensating resistance (0.0 by default).
REALAR(23)	CX1, load drop compensating reactance (0.0 by default).
REALAR(24)	CNXA1, winding connection angle (0.0 by default). Used with adjustment control mode 5 (unsymmatric phase shift control of active power) implemented in PSS®E version 32.1.
REALAR(25)	actual transformer resistance (**returned**).
REALAR(26)	actual transformer reactance (**returned**).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real REALARI(24)	Is an array of 24 elements corresponding to input values REALAR(1) through REALAR(24) as described above.
Real REALARO(2)	Is an array of two elements corresponding to output values REALAR(25) through REALAR(26) as described above.
Character NAME*12	Is the transformer name (input; blank by default).
Integer IERR	Is the error code.

IERR = 0	no errors occurred
IERR > 0	fatal error occurred
IERR = 1	bus not found
IERR = 2	circuit identifier is more than two characters
IERR = 3	branch exists but it is a non-transformer branch
IERR = 4	branch from a bus to itself
IERR = 5	blank circuit identifier is invalid
IERR = 6	circuit identifier may not start with &, * or @
IERR = 7	branch table is full
IERR = 8	branch ownership table is full
IERR = 9	two-winding transformer table is full
IERR = 10	CW, CZ and/or CM >1 but at least one endpoint bus has no base voltage
IERR = 11	invalid CW, CZ and/or CM value
IERR = 13	prerequisite requirements for API are not met.
IERR = -1	data error, warning and information messages; one or more of: • invalid metered end bus • invalid transformer status • invalid owner number • invalid ownership fraction • no ownership data specified. • multi-section line grouping deleted • branch ownership table is full--only first n owners retained • owner table is full • invalid winding 1 side bus number • invalid CW, CZ and/or CM data value • controlled bus not found • invalid adjustment control mode flag • invalid impedance correction table number • invalid number of tap positions • winding nominal voltage is negative • winding bus base voltage and nominal voltage differ by more than 20% • invalid winding voltage • winding voltage and bus base voltage differ by more than 50%

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

- RMAX is greater than 150% of bus base voltage
- RMIN is less than 50% of bus base voltage
- invalid tap ratio
- tap ratio is outside the band 0.5 to 1.5 pu
- RMAX is greater than 1.5
- RMIN is less than 0.5
- RMAX is less than RMIN
- VMAX is less than VMIN
- phase shift angle normalized between -180 and 180
- winding base MVA must be positive
- MAG1 is too big relative to MAG2 at winding base MVA
- magnetizing susceptance MAG2 is positive
- losses are too big relative to total impedance at winding base MVA
- branch reactance is 0.0
- winding 1 and winding 2 sides are interchanged (any two-terminal dc lines controlled by this transformer are logged).
- winding connection angle normalized between -360 and 360
- adjustment control mode flag and the winding connection angle are inconsistent.
- for asymmetric phase shifter, abnormally large or small tap setting results at RMA1 (or RMI1) and CNXA1.
- for asymmetric phase shifter, the specified and derived winding tap settings differ.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.77 TWO_WINDING_DATA_4

This API routine is the fourth release of the two-winding transformer data specification function. Use this API routine to modify the data of an existing two-winding transformer in the working case, or to add a new two-winding transformer to the working case (see *PSS®E Program Operation Manual*, Owner Data).

Batch command syntax:

```
BAT_TWO_WINDING_DATA_4  I  J  'CKT'  INTGAR(1)...INTGAR(15)
REALAR(1)...REALAR(24)  'CHARAR(1)'  'CHARAR(2)'
```

Python syntax:

```
ierr,realaro = two_winding_data_4(i, j, ckt, intgar, realari, charar)
```

Fortran syntax:

```
CALL TWO_WINDING_API_4(I, J, CKT, INTGAR, REALAR, CHARAR, IERR)
```

where:

Integer I	Is the bus number of from bus (input; no default allowed).
Integer J	Is the bus number of to bus (input; no default allowed).
Character CKT*2	Is the circuit identifier (input; ’1’ by default).
Integer INTGAR(15)	Is an array of fifteen elements (input).

INTGAR(1)	STAT, branch status (1 by default).
INTGAR(2)	METBUS, metered end bus number (I or J) (I by default).
INTGAR(3)	O1, first owner number (owner of bus I by default).
INTGAR(4)	O2, second owner number (0 by default).
INTGAR(5)	O3, third owner number (0 by default).
INTGAR(6)	O4, fourth owner number (0 by default).
INTGAR(7)	NTP1, number of tap positions (33 by default).
INTGAR(8)	TAB1, impedance correction table number (0 by default).
INTGAR(9)	WN1BUS, winding one side bus number (I or J) (I by default).
INTGAR(10)	CONT1, controlled bus number (0 by default).
INTGAR(11)	SICOD1, negative for controlled bus on winding one side (1 by default).
INTGAR(12)	COD1, adjustment control mode flag (-5 through +5) (0 by default).
INTGAR(13)	CW, winding data I/O code (1 by default).
INTGAR(14)	CZ, impedance data I/O code (1 by default).
INTGAR(15)	CM, magnetizing admittance data I/O code (1 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real REALAR(26) Is an array of 26 elements (first twenty-four are input values; last two are output values).

REALAR(1)	R1-2, nominal transformer resistance (0.0 by default).
REALAR(2)	X1-2, nominal transformer reactance (THRSHZ by default; 0.0001 if THRSHZ = 0.0).
REALAR(3)	SBS1-2, winding base MVA (SBASE by default).
REALAR(4)	WINDV1, winding one ratio/voltage (1.0 by default if CW is 1 or 3; base voltage of bus I by default if CW is 2).
REALAR(5)	NOMV1, winding one nominal voltage (0.0 by default).
REALAR(6)	ANG1, winding one phase shift angle (0.0 by default).
REALAR(7)	WINDV2, winding two ratio/voltage (1.0 by default if CW is 1 or 3; base voltage of bus J by default if CW is 2).
REALAR(8)	NOMV2, winding two nominal voltage (0.0 by default).
REALAR(9)	RATA1, rating set A line rating (0.0 by default).
REALAR(10)	RATB1, rating set B line rating (0.0 by default).
REALAR(11)	RATC1, rating set C line rating (0.0 by default).
REALAR(12)	F1, first owner fraction (1.0 by default).
REALAR(13)	F2, second owner fraction (1.0 by default).
REALAR(14)	F3, third owner fraction (1.0 by default).
REALAR(15)	F4, fourth owner fraction (1.0 by default).
REALAR(16)	MAG1, magnetizing conductance or no-load losses (0.0 by default).
REALAR(17)	MAG2, magnetizing susceptance or exciting current (0.0 by default).
REALAR(18)	RMA1, winding one ratio/angle high limit (1.1 by default).
REALAR(19)	RMI1, winding one ratio/angle low limit (0.9 by default).
REALAR(20)	VMA1, voltage or flow upper limit (1.1 by default).
REALAR(21)	VMI1, voltage or flow lower limit (0.9 by default).
REALAR(22)	CR1, load drop compensating resistance (0.0 by default).
REALAR(23)	CX1, load drop compensating reactance (0.0 by default).
REALAR(24)	CNXA1, winding connection angle (0.0 by default). Used with adjustment control mode 5 (unsymmatric phase shift control of active power) implemented in PSS®E version 32.1.
REALAR(25)	actual transformer resistance (**returned**).
REALAR(26)	actual transformer reactance (**returned**).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real REALARI(24)	Is an array of 24 elements corresponding to input values REALAR(1) through REALAR(24) as described above.
Real REALARO(2)	Is an array of two elements corresponding to output values REALAR(25) through REALAR(26) as described above.
Character CHARAR(2)*12	Is an array of two elements (input). CHARAR(1) NAME, transformer name (blank by default). CHARAR(2) VGRP, vector group name (blank by default).
Integer IERR	Is the error code. IERR = 0 no errors occurred IERR > 0 fatal error occurred IERR = 1 bus not found IERR = 2 circuit identifier is more than two characters IERR = 3 branch exists but it is a non-transformer branch IERR = 4 branch from a bus to itself IERR = 5 blank circuit identifier is invalid IERR = 6 circuit identifier may not start with &, * or @ IERR = 7 branch table is full IERR = 8 branch ownership table is full IERR = 9 two-winding transformer table is full IERR = 10 CW, CZ and/or CM >1 but at least one endpoint bus has no base voltage IERR = 11 invalid CW, CZ and/or CM value IERR = 13 prerequisite requirements for API are not met. IERR = -1 data error, warning and information messages; one or more of: • invalid metered end bus • invalid transformer status • invalid owner number • invalid ownership fraction • no ownership data specified. • multi-section line grouping deleted • branch ownership table is full--only first n owners retained • owner table is full • invalid winding 1 side bus number • invalid CW, CZ and/or CM data value • controlled bus not found • invalid adjustment control mode flag • invalid impedance correction table number • invalid number of tap positions

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

- winding nominal voltage is negative
- winding bus base voltage and nominal voltage differ by more than 20%
- invalid winding voltage
- winding voltage and bus base voltage differ by more than 50%
- RMAX is greater than 150% of bus base voltage
- RMIN is less than 50% of bus base voltage
- invalid tap ratio
- tap ratio is outside the band 0.5 to 1.5 pu
- RMAX is greater than 1.5
- RMIN is less than 0.5
- RMAX is less than RMIN
- VMAX is less than VMIN
- phase shift angle normalized between -180 and 180
- winding base MVA must be positive
- MAG1 is too big relative to MAG2 at winding base MVA
- magnetizing susceptance MAG2 is positive
- losses are too big relative to total impedance at winding base MVA
- branch reactance is 0.0
- winding 1 and winding 2 sides are interchanged (any two-terminal dc lines controlled by this transformer are logged).
- winding connection angle normalized between -360 and 360
- adjustment control mode flag and the winding connection angle are inconsistent.
- for asymmetric phase shifter, abnormally large or small tap setting results at RMA1 (or RMI1) and CNXA1.
- for asymmetric phase shifter, the specified and derived winding tap settings differ.
- vector group not found

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.78 VSC_DC_LINE_CHNG

Use this API routine to modify the link data of an existing VSC dc line in the working case (see *PSS®E Program Operation Manual*, Voltage Source Converter (VSC) DC Transmission Line Data).

Batch command syntax:

```
BAT_VSC_DC_LINE_CHNG  'NAME'  INTGAR(1)...INTGAR(5)
REALAR(1)...REALAR(5)
```

Python syntax:

```
ierr = vsc_dc_line_chng(name, intgar, realar)
```

Fortran syntax:

```
CALL VSC_DC_LINE_CHNG(NAME, INTGAR, REALAR, IERR)
```

where:

Character NAME*12	Is the non-blank VSC dc line name (input; no default allowed).
Integer INTGAR(5)	Is an array of at least five elements (input). See the VSC_DC_LINE_DATA API routine for details.
Real REALAR(5)	Is an array of at least five elements (input). See the VSC_DC_LINE_DATA API routine for details.
Integer IERR	Is the error code (output). IERR = 0 no errors occurred IERR > 0 fatal error occurred IERR = 2 blank dc line name is invalid IERR = 3 dc line not found IERR = 4 prerequisite requirements for API are not met IERR = -1 data error, warning and information messages; one or more of: • invalid dc line control mode • invalid dc line resistance • invalid owner number • invalid ownership fraction • no ownership data specified • owner table is full.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.79 VSC_DC_CONVERTER_DATA

Use this API to modify the data of a dc converter of an existing VSC dc line in the working case (see *PSS®E Program Operation Manual*, Voltage Source Converter (VSC) DC Transmission Line Data).

Batch command syntax:

```
BAT_VSC_DC_CONVERTER_DATA  'NAME'  IC  INTGAR(1)...INTGAR(4)
REALAR(1)...REALAR(11)
```

Python syntax:

```
ierr = vsc_dc_converter_data(name, ic, intgar, realar)
```

Fortran syntax:

```
CALL VSC_DC_LINE_CONVERTER_API_2(NAME, IC, INTGAR, REALAR, IERR)
```

where:

Character NAME*12	Is the non-blank VSC dc line name (input; no default allowed).	
Integer IC	Is the dc converter index (1 or 2) (input; no default allowed).	
Integer INTGAR(4)	Is the array of at least four elements (input).	
	INTGAR(1)	IBUS, converter bus number (no default allowed).
	INTGAR(2)	TYPE, dc control code (0 = out-of-service, 1 = kV, 2 = MW) (1 by default).
	INTGAR(3)	MODE, ac control code (1 = voltage, 2 = power factor) (1 by default).
	INTGAR(4)	REMOT, remote bus number (0 for this converter to regulate voltage at bus IBUS) (0 by default).
Real REALAR(11)	Is an array of at least eleven elements (input).	
	REALAR(1)	DCSET, dc setpoint (kV or MW) (1.0 by default).
	REALAR(2)	ACSET, ac setpoint (pu voltage or power factor) (1.0 by default).
	REALAR(3)	ALOSS, A loss coefficient (kW) (0.0 by default).
	REALAR(4)	BLOSS, B loss coefficient (kW/amps) (0.0 by default).
	REALAR(5)	MINLOS, minimum converter losses (kW) (0.0 by default).
	REALAR(6)	SMAX, converter ac MVA rating (0.0 by default).
	REALAR(7)	IMAX, converter ac current rating (amps) (0.0 by default).
	REALAR(8)	PWF, power weighting factor fraction (1.0 by default).
	REALAR(9)	MAXQ, reactive power upper limit (Mvar) (9999.0 by default).
	REALAR(10)	MINQ, reactive power lower limit (Mvar) (-9999.0 by default).
	REALAR(11)	RMPCT, percent of contributed reactive power (100.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no errors occurred.
IERR > 0	fatal error occurred.
IERR = 1	invalid converter index.
IERR = 2	no converter bus is specified.
IERR = 3	dc line not found.
IERR = 4	converter bus not found.
IERR = 5	prerequisite requirements for API are not met.
IERR = -1	data error, warning and information messages; one or more of: • Converter bus is dummy bus of a multi-section line • Multi-section line grouping deleted. • Invalid dc control code • Invalid ac control mode • Regulated bus not found • Converter specified as remotely regulating itself. • Type code of regulated bus is not 1 or 2. • dc voltage setpoint is not positive • ac voltage setpoint is outside the band 0.75 to 1.25 pu • ac power factor setpoint is in error (magnitude exceeds 1.0) • Invalid power weighting factor fraction • MAXQ is less than MINQ. • RMPCT is not positive.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.80 VSC_DC_LINE_DATA

Use this API routine to modify the link data of an existing VSC dc line in the working case, or to add a new VSC dc line to the working case (see *PSS®E Program Operation Manual*, Voltage Source Converter (VSC) DC Transmission Line Data).

Batch command syntax:

```
BAT_VSC_DC_LINE_DATA  'NAME'  INTGAR(1)...INTGAR(5)
REALAR(1)...REALAR(5)
```

Python syntax:

```
ierr = vsc_dc_line_data(name, intgar, realar)
```

Fortran syntax:

```
CALL VSC_DC_LINE_API_2(NAME, INTGAR, REALAR, IERR)
```

where:

Character NAME*12	Is the non-blank VSC dc line name (input; no default allowed).	
Integer INTGAR(5)	Is an array of at least five elements (input).	
	INTGAR(1)	MDC, control mode (0 or 1) (0 by default).
	INTGAR(2)	O1, first owner number (1 by default).
	INTGAR(3)	O2, second owner number (0 by default).
	INTGAR(4)	O3, third owner number (0 by default).
	INTGAR(5)	O4, fourth owner number (0 by default).
Real REALAR(5)	Is an array of at least five elements (input).	
	REALAR(1)	RDC, dc line resistance (0.0 by default).
	REALAR(2)	F1, first owner fraction (1.0 by default).
	REALAR(3)	F2, second owner fraction (1.0 by default).
	REALAR(4)	F3, third owner fraction (1.0 by default).
	REALAR(5)	F4, fourth owner fraction (1.0 by default).
Integer IERR	Is the error code (output).	
	IERR = 0	no errors occurred
	IERR > 0	fatal error occurred
	IERR = 1	VSC dc line table is full
	IERR = 2	blank dc line name is invalid
	IERR = 4	prerequisite requirements for API are not met
	IERR = -1	data error, warning and information messages; one or more of: • invalid dc line control mode • invalid dc line resistance • invalid owner number • invalid ownership fraction • no ownership data specified • owner table is full.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2.81 ZONE_DATA

Use this API to modify zone data in the working case (see *PSS®E Program Operation Manual*, Zone Data).

Batch command syntax:

```
BAT_ZONE_DATA  I  'ZONAME'
```

Python syntax:

```
ierr = zone_data(i, zoname)
```

Fortran syntax:

```
CALL ZONE_API_2(I, ZONAME, IERR)
```

where:

Integer I	Is the zone number between 1 and 9999 (input; no default allowed).
Character ZONAME*12	Is the zone name (input; blank by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no errors occurred.
	IERR > 0 — fatal error occurred.
	IERR = 1 — invalid zone number.
	IERR = 2 — zone table is full.
	IERR = 3 — prerequisite requirements for API are not met.
	IERR = -1 — data error, warning and information messages; one or more of: • No equipment is assigned to zone.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

SIEMENS

Chapter 3

Optimal Power Flow

This chapter describes the API routines used in PSS®E Optimal Power Flow. Section 3.1 Optimal Power Flow Program Operation APIs contains API routines corresponding to the operational functions of the PSS®E OPF. Additional information on the functions these commands execute is in the *PSS®E Program Operation Manual*, Chapter 14. Section 3.2 Optimal Power Flow Data Changing APIs contains API routines specific to adding, modifying and deleting optimal power flow data.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1 Optimal Power Flow Program Operation APIs

3.1.1 ADD_DETAILS_TO_OPF_LOG

Use this API to specify or return the value of the OPF solution option to add details of the sensitivity values and Lagrange multipliers.

Batch command syntax:

```
BAT_ADD_DETAILS_TO_OPF_LOG  IVAL
```

Python syntax:

```
ierr = add_details_to_opf_log(ival)
```

or:

```
ierr,ival = add_details_to_opf_log()
```

Fortran syntax:

```
CALL ADD_DETAILS_TO_OPF_LOG(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Indicates whether to append the OPF log file with information on sensitivity values and Lagrange multipliers (input if Batch command or for Python if specified as an argument, or for FORTRAN if IOCODE is 0; else output; 0 by default). IVAL = 0 do not print additional details in log file. IVAL = 1 print additional details.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to the value specified in IVAL. IOCODE = 1 return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid IVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.2 APPLY_ALTERNATE_STEP_SIZE

Use this API to specify or return the value of the OPF solution option for applying an alternate lambda mismatch step size. This option works in conjunction with the OPF_BARRIER_STEP_LENGTH option setting (Section 3.1.25 OPF_BARRIER_STEP_LENGTH), which specifies the alternate barrier step length value.

Batch command syntax:

```
BAT_APPLY_ALTERNATE_STEP_SIZE   IVAL
```

Python syntax:

```
ierr = apply_alternate_step_size(ival)
```

or:

```
ierr,ival = apply_alternate_step_size()
```

Fortran syntax:

```
CALL APPLY_ALTERNATE_STEP_SIZE(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Indicates whether to limit the step size taken during the OPF solution (input if Batch command or for Python if specified as an argument, or for FORTRAN if IOCODE is 0; else output 0 by default). IVAL = 0 do not limit the step size. IVAL = 1 limit the step size.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in IVAL. IOCODE = 1 return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid IVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.3 APPLY_AUTOMATIC_SCALING

Use this API to specify or return the value of the OPF solution option to apply automatic scaling of the objective coefficients and derivatives during the OPF solution.

Batch command syntax:

```
BAT_APPLY_AUTOMATIC_SCALING   IVAL
```

Python syntax:

```
ierr = apply_automatic_scaling(ival)
```

or:

```
ierr,ival = apply_automatic_scaling()
```

Fortran syntax:

```
CALL APPLY_AUTOMATIC_SCALING(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Indicates whether to apply automatic scaling of the objective during the solution (input if Batch command or for Python if specified as an argument, or for FORTRAN if IOCODE is 0; else output 0 by default). IVAL = 0 do not apply automatic scaling of the objective. IVAL = 1 apply automatic scaling of the objective.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in IVAL. IOCODE = 1 return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid IVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.4 BAD_ITER_COARSE_LIMIT

Use this API to specify or return the bad iteration coarse limit value.

Batch command syntax:

```
BAT_BAD_ITER_COARSE_LIMIT   IVAL
```

Python syntax:

```
ierr =  bad_iter_coarse_limit(ival)
```

or:

```
ierr,ival = bad_iter_coarse_limit()
```

Fortran syntax:

```
CALL BAD_ITER_COARSE_LIMIT(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Is the bad iteration coarse limit value (input if Batch command or for Python if specified as an argument, or for FORTRAN if IOCODE is 0; else output 3 by default).
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in IVAL. IOCODE = 1 return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid IVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.5 BAD_ITER_FINE_LIMIT

Use this API to specify or return the bad iteration fine limit value.

Batch command syntax:

```
BAT_BAD_ITER_FINE_LIMIT  IVAL
```

Python syntax:

```
ierr =  bad_iter_fine_limit(ival)
```

or:

```
ierr,ival = bad_iter_fine_limit()
```

Fortran syntax:

```
CALL BAD_ITER_FINE_LIMIT(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Is the bad iteration fine limit value (input if Batch command or for Python if specified as an argument, or for FORTRAN if IOCODE is 0; else output 10 by default).
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in IVAL. IOCODE = 1 return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid IVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.6 CLAMP_NONOPTIMIZED_GENS

Use this API to specify or return the value of the OPF solution option to apply the clamp equation to non-optimized generators. The behavior of the clamp equation may be further customized through the OPF_INITIAL_CLAMP_TOL (Section 3.1.33 OPF_INITIAL_CLAMP_TOL), OPF_FINAL_CLAMP_TOL (Section 3.1.27 OPF_FINAL_CLAMP_TOL) and the OPF_CLAMP_DECAY_FACTOR (Section 3.1.26 OPF_CLAMP_DECAY_FACTOR) option settings.

For additional information on the clamp treatment of non-optimized generators, refer to Clamp Treatment in the *PSS®E Program Operation Manual*.

Batch command syntax:

```
BAT_CLAMP_NONOPTIMIZED_GENS   IVAL
```

Python syntax:

```
ierr = clamp_nonoptimized_gens(ival)
```

or:

```
ierr,ival = clamp_nonoptimized_gens()
```

Fortran syntax:

```
CALL CLAMP_NONOPTIMIZED_GENS(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Indicates whether to apply the clamp equation to nonoptimized generators (input if Batch command or for Python if specified as an argument, or for FORTRAN if IOCODE is 0; else output 0 by default). IVAL = 0 do not clamp nonoptimized generators. IVAL = 1 clamp nonoptimized generators.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in IVAL. IOCODE = 1 return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid IVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.7 CONSTRAIN_INTERFACE_FLOWS

Use this API to specify or return the value of the OPF option to constrain interface flows. The setting of this option affects recognition of any interface flow constraints (Sections 3.2.22 - 3.2.24) and the behavior of the MINIMIZE_INTERFACE_FLOWS (Section 3.1.15 MINIMIZE_INTERFACE_FLOWS) objective function option.

Batch command syntax:

```
BAT_CONSTRAIN_INTERFACE_FLOWS   IVAL
```

Python syntax:

```
ierr = constrain_interface_flows(ival)
```

or:

```
ierr,ival = constrain_interface_flows()
```

Fortran syntax:

```
CALL CONSTRAIN_INTERFACE_FLOWS(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Indicates whether to enable the option to constrain interface flows (input if Batch command or for Python if specified as an argument, or for FORTRAN if IOCODE is 0; else output 0 by default). IVAL = 0 do not constrain interface flows. IVAL = 1 constrain interface flows.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in IVAL. IOCODE = 1 return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid IVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.8 FINAL_OPF_BARRIER_COEFF

Use this API to specify or return the value of the final OPF barrier coefficient. It may be used in conjunction with the INITIAL_OPF_BARRIER_COEFF (Section 3.1.10 INITIAL_OPF_BARRIER_COEFF) value.

Batch command syntax:

```
BAT_FINAL_OPF_BARRIER_COEFF   RVAL
```

Python syntax:

```
ierr =  final_opf_barrier_coeff(rval)
```

or:

```
ierr,rval = final_opf_barrier_coeff()
```

Fortran syntax:

```
CALL FINAL_OPF_BARRIER_COEFF(RVAL, IOCODE, IERR)
```

where:

Real RVAL	Is the final barrier coefficient value (iinput if Batch command or for Python if specified as an argument, or for FORTRAN if IOCODE is 0; else output 0.0001 by default).
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in RVAL. IOCODE = 1 return the current value of the option setting in RVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid RVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.9 IMPORT_ECDI

Use this API to import data from an Economics Dispatch Data File prepared for use with activity ECDI into the OPF data component of working memory. This function removes all existing Generator Dispatch Data, Active Power Dispatch Data, and Cost Curve Data (*PSS®E Program Operation Manual,* Sections 14.15.11 - 14.15.13).

Batch command syntax:

```
BAT_IMPORT_ECDI  FILE
```

Python syntax:

```
ierr = import_ecdi(file)
```

Fortran syntax:

```
CALL IMPORT_ECDI(FILE, IERR)
```

where:

Character FILE*260	Is the name of the Economic Dispatch Data File (input).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	NEWOPF_API error occurred.
	IERR = 2	ECDIAPI error occurred.
	IERR = 3	CSTTBL_QUAD_API error occurred.
	IERR = 4	OPFDSP_TBL_API error occurred.
	IERR = 5	GENDSP_INDV_API error occurred.
	IERR = 6	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.10 INITIAL_OPF_BARRIER_COEFF

Use this API to specify or return the value of the initial OPF barrier coefficient. It may be used in conjunction with the FINAL_OPF_BARRIER_COEFF (Section 3.1.8 FINAL_OPF_BARRIER_COEFF) value.

Batch command syntax:

```
BAT_INITIAL_OPF_BARRIER_COEFF   RVAL
```

Python syntax:

```
ierr =  initial_opf_barrier_coeff(rval)
```

or:

```
ierr,rval = initial_opf_barrier_coeff()
```

Fortran syntax:

```
CALL INITIAL_OPF_BARRIER_COEFF(RVAL, IOCODE, IERR)
```

where:

Real RVAL	Is the initial barrier coefficient value (input if Batch command or for Python if specified as an argument, or for FORTRAN if IOCODE is 0; else output 1.0 by default).
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in RVAL. IOCODE = 1 return the current value of the option setting in RVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid RVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.11 INTERFACE_FLOW_COST_COEFF

Use this API to specify or return the value of the coefficient for the interface flow objective. This API works in conjunction with the MINIMIZE_INTERFACE_FLOWS (Section 3.1.15 MINIMIZE_INTERFACE_FLOWS) objective function option.

Batch command syntax:

```
BAT_INTERFACE_FLOW_COST_COEFF   RVAL
```

Python syntax:

```
ierr =  interface_flow_cost_coeff(rval)
```

or:

```
ierr,rval = interface_flow_cost_coeff()
```

Fortran syntax:

```
CALL INTERFACE_FLOW_COST_COEFF(RVAL, IOCODE, IERR)
```

where:

Real RVAL	Is the value of the interface flow cost coefficient (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0, output if IOCODE is 1; 1.0 by default).
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in RVAL. IOCODE = 1 return the current value of the option setting in RVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid RVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.12 LSTO

Use this API to list OPF working case data in a form suitable for data documentation.

Batch command syntax:

```
BAT_LSTO   SID  ALL  OPT  VREV
```

Python syntax:

```
ierr = lsto(sid, all, opt, vrev)
```

Fortran syntax:

```
CALL LSTONW(SID, ALL, OPT, VREV, IERR)
```

where:

Integer SID — Is a valid subsystem ID from 0 to 11. A subsystem must have been previously defined (input; no default allowed).

Integer ALL — Is the *all buses* or specified subsystem (input; no default allowed).

- ALL = 1 — process all buses.
- ALL = 0 — process only specified buses.

Integer OPT — Is the number corresponding to the category of OPF data that is to be tabulated (input; no default allowed).

= 1, case summary	= 2, bus attribute data
= 3, adjustable bus shunt data	= 4, bus load data
= 5, adjustable bus load table data	= 6, generator dispatch data
= 7, active power dispatch data	= 8, generation reserve data
= 9, generator reactive capability data	= 10, adj. branch reactance data
= 11, linear cost table data	= 12, quadratic cost table data
= 13, polynomial and exponential cost table data	
= 14, period reserve constraint data	= 15, branch flow constraint data
= 16, interface flow constraint data	= 17, linear constraint dependency data
= 18, two-terminal dc line data	= 21, full listing

Integer VREV — Is the option to change voltage output units from the default units to the alternate units (input; no default allowed).

- VREV = 0 — use default voltage units.
- VREV = 1 — use alternate voltage units.

Integer IERR — Is the error code (output).

- IERR = 0 — no error occurred.
- IERR = 1 — invalid SID value or subsystem SID is not defined.
- IERR = 2 — invalid ALL value.
- IERR = 3 — invalid OPT value.
- IERR = 4 — invalid VREV value.
- IERR = 5 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.13 MINIMIZE_ADJ_BUS_SHUNTS

Use this API to specify or return the value of the OPF minimize adjustable bus shunts (var compensation) objective option.

Batch command syntax:

```
BAT_MINIMIZE_ADJ_BUS_SHUNTS   IVAL
```

Python syntax:

```
ierr = minimize_adj_bus_shunts(ival)
```

or:

```
ierr,ival = minimize_adj_bus_shunts()
```

Fortran syntax:

```
CALL MINIMIZE_ADJ_BUS_SHUNTS(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Indicates whether to enable the minimize adjustable bus shunts objective (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 0 by default for input). IVAL = 0 do not enable the minimize adjustable bus shunts objective. IVAL = 1 enable the minimize adjustable bus shunts objective.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in IVAL. IOCODE = 1 return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid IVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.14 MINIMIZE_FUEL_COST

Use this API to specify or return the value of the OPF minimize fuel cost objective option.

Batch command syntax:

```
BAT_MINIMIZE_FUEL_COST   IVAL
```

Python syntax:

```
ierr = minimize_fuel_cost(ival)
```

or:

```
ierr,ival = minimize_fuel_cost()
```

Fortran syntax:

```
CALL MINIMIZE_FUEL_COST(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Indicates whether to enable the minimize fuel cost objective (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 0 by default for input). IVAL = 0 do not enable the minimize fuel cost objective. IVAL = 1 enable the minimize fuel cost objective.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in IVAL. IOCODE = 1 return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid IVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.15 MINIMIZE_INTERFACE_FLOWS

Use this API to specify or return the value of the OPF minimize interface flows objective option.

Batch command syntax:

```
BAT_MINIMIZE_INTERFACE_FLOWS   IVAL
```

Python syntax:

```
ierr = minimize_interface_flows(ival)
```

or:

```
ierr,ival = minimize_interface_flows()
```

Fortran syntax:

```
CALL MINIMIZE_INTERFACE_FLOWS(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Indicates whether to enable the minimize interface flows objective (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 0 by default for input).	
	IVAL = 0	do not enable the minimize interface flows objective.
	IVAL = 1	enable the minimize interface flows objective.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE = 0	set the option setting to value specified in IVAL.
	IOCODE = 1	return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid IVAL value.
	IERR = 2	invalid IOCODE value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.16 MINIMIZE_LOAD_ADJUSTMENTS

Use this API to specify or return the value of the OPF minimize adjustable bus load objective option.

Batch command syntax:

```
BAT_MINIMIZE_LOAD_ADJUSTMENTS   IVAL
```

Python syntax:

```
ierr = minimize_load_adjustments(ival)
```

or:

```
ierr,ival = minimize_load_adjustments()
```

Fortran syntax:

```
CALL MINIMIZE_LOAD_ADJUSTMENTS(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Indicates whether to enable the minimize adjustable bus loads objective (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 0 by default for input). IVAL = 0 do not enable the minimize adjustable bus loads objective. IVAL = 1 enable the minimize adjustable bus loads objective.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in IVAL. IOCODE = 1 return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid IVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.17 MINIMIZE_P_LOSSES

Use this API to specify or return the value of the OPF minimize active power loss objective option.

Batch command syntax:

```
BAT_MINIMIZE_P_LOSSES   IVAL
```

Python syntax:

```
ierr = minimize_p_losses(ival)
```

or:

```
ierr,ival = minimize_p_losses()
```

Fortran syntax:

```
CALL MINIMIZE_P_LOSSES(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Indicates whether to enable the minimize active power loss objective (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 0 by default for input).	
	IVAL = 0	do not enable the minimize active power loss objective.
	IVAL = 1	enable the minimize active power loss objective.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE = 0	set the option setting to value specified in IVAL.
	IOCODE = 1	return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid IVAL value.
	IERR = 2	invalid IOCODE value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.18 MINIMIZE_P_SLACK

Use this API to specify or return the value of the OPF minimize active power slack objective option.

Batch command syntax:

```
BAT_MINIMIZE_P_SLACK   IVAL
```

Python syntax:

```
ierr = minimize_p_slack(ival)
```

or:

```
ierr,ival = minimize_p_slack()
```

Fortran syntax:

```
CALL MINIMIZE_P_SLACK(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Indicates whether to enable the minimize active power slack objective (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 0 by default for input). IVAL = 0 — do not enable the minimize active power slack objective. IVAL = 1 — enable the minimize active power slack objective.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 — set the option setting to value specified in IVAL. IOCODE = 1 — return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output). IERR = 0 — no error. IERR = 1 — invalid IVAL value. IERR = 2 — invalid IOCODE value. IERR = 3 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.19 MINIMIZE_Q_LOSSES

Use this API to specify or return the value of the OPF minimize reactive power loss objective option.

Batch command syntax:

```
BAT_MINIMIZE_Q_LOSSES   IVAL
```

Python syntax:

```
ierr = minimize_q_losses(ival)
```

or:

```
ierr,ival = minimize_q_losses()
```

Fortran syntax:

```
CALL MINIMIZE_Q_LOSSES(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Indicates whether to enable the minimize reactive power loss objective (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 0 by default for input).	
	IVAL = 0	do not enable the minimize reactive power loss objective.
	IVAL = 1	enable the minimize reactive power loss objective.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE = 0	set the option setting to value specified in IVAL.
	IOCODE = 1	return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid IVAL value.
	IERR = 2	invalid IOCODE value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.20 MINIMIZE_Q_SLACK

Use this API to specify or return the value of the OPF minimize reactive power slack objective option.

Batch command syntax:

```
BAT_MINIMIZE_Q_SLACK   IVAL
```

Python syntax:

```
ierr = minimize_q_slack(ival)
```

or:

```
ierr,ival = minimize_q_slack()
```

Fortran syntax:

```
CALL MINIMIZE_Q_SLACK(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Indicates whether to enable the minimize reactive power slack objective (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 0 by default for input). IVAL = 0 do not enable the minimize reactive power slack objective. IVAL = 1 enable the minimize reactive power slack objective.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in IVAL. IOCODE = 1 return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid IVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.21 MINIMIZE_REACTIVE_RESERVE

Use this API to specify or return the value of the OPF minimize reactive reserve objective option.

Batch command syntax:

```
BAT_MINIMIZE_REACTIVE_RESERVE   IVAL
```

Python syntax:

```
ierr = minimize_reactive_reserve(ival)
```

or:

```
ierr,ival = minimize_reactive_reserve()
```

Fortran syntax:

```
CALL MINIMIZE_REACTIVE_RESERVE(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Indicates whether to enable the minimize reactive reserves objective (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 0 by default for input).	
	IVAL = 0	do not enable the minimize reactive reserves objective.
	IVAL = 1	enable the minimize reactive reserves objective.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE = 0	set the option setting to value specified in IVAL.
	IOCODE = 1	return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid IVAL value.
	IERR = 2	invalid IOCODE value.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.22 MINIMIZE_SERIES_COMP

Use this API to specify or return the value of the OPF minimize series compensation (adjustable branch reactance) objective option.

Batch command syntax:

```
BAT_MINIMIZE_SERIES_COMP   IVAL
```

Python syntax:

```
ierr = minimize_series_comp(ival)
```

or:

```
ierr,ival = minimize_series_comp()
```

Fortran syntax:

```
CALL MINIMIZE_SERIES_COMP(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Indicates whether to enable the minimize series compensation objective (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 0 by default for input). IVAL = 0 do not enable the minimize series compensation objective. IVAL = 1 enable the minimize series compensation objective.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in IVAL. IOCODE = 1 return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid IVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.23 NOPF

Use this API to run the OPF solution. Subsystems that have been specified will be optimized to the objectives and tolerances specified, if a solution is achievable.

Batch command syntax:

```
BAT_NOPF   SID  ALL
```

Python syntax:

```
ierr = nopf(sid, all)
```

Fortran syntax:

```
CALL NOPF_SOLVE(SID, ALL, IERR)
```

where:

Integer SID	Is a valid bus subsystem ID from 0 to 11. A bus subsystem must have been previously defined (input; no default allowed).
Integer ALL	Is the *all buses* or specified subsystem flag (input; no default allowed). ALL = 0 process only specified buses. ALL = 1 process all buses.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 error initializing OPF data. IERR = 4 generators are converted. IERR = 5 Topology error: buses exist with bus type code and series element status inconsistencies. IERR = 6 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.24 OPEN_BUS_VOLTAGE_LIMITS

Use this API to specify or return the value of the OPF solution option to open bus voltage limits for initial feasibility.

Batch command syntax:

```
BAT_OPEN_BUS_VOLTAGE_LIMITS   IVAL
```

Python syntax:

```
ierr =  open_bus_voltage_limits(ival)
```

or:

```
ierr,ival = open_bus_voltage_limits()
```

Fortran syntax:

```
CALL OPEN_BUS_VOLTAGE_LIMITS(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Indicates whether to open bus voltage limits for initial feasibility (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 0 by default for input). IVAL = 0 do not open up bus voltage limits. IVAL = 1 open bus voltage limits for initial feasibility.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in IVAL. IOCODE = 1 return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid IVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.25 OPF_BARRIER_STEP_LENGTH

Use this API to specify or return the value of the barrier function step length setting.

Batch command syntax:

```
BAT_OPF_BARRIER_STEP_LENGTH   RVAL
```

Python syntax:

```
ierr =  opf_barrier_step_length(rval)
```

or:

```
ierr,rval = opf_barrier_step_length()
```

Fortran syntax:

```
CALL OPF_BARRIER_STEP_LENGTH(RVAL, IOCODE, IERR)
```

where:

Real RVAL	Is the value of the barrier function step length (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 0.99 by default for input).
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in RVAL. IOCODE = 1 return the current value of the option setting in RVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid RVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.26 OPF_CLAMP_DECAY_FACTOR

Use this API to specify or return the value of the clamp equation decay factor. This API works in conjunction with the API option to CLAMP_NONOPTIMIZED_GENS, and is associated with the APIs for OPF_INITIAL_CLAMP_TOL and OPF_FINAL_CLAMP_TOL.

Batch command syntax:

```
BAT_OPF_CLAMP_DECAY_FACTOR   RVAL
```

Python syntax:

```
ierr =  opf_clamp_decay_factor(rval)
```

or:

```
ierr,rval = opf_clamp_decay_factor()
```

Fortran syntax:

```
CALL OPF_CLAMP_DECAY_FACTOR(RVAL, IOCODE, IERR)
```

where:

Real RVAL	Is the value of the clamp equation decay factor (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 0.1 by default for input).
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in RVAL. IOCODE = 1 return the current value of the option setting in RVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid RVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.27 OPF_FINAL_CLAMP_TOL

Use this API to specify or return the value of the final clamp tolerance. This API works in conjunction with the API option to CLAMP_NONOPTIMIZED_GENS, and is associated with the APIs for OPF_INITIAL_CLAMP_TOL and the OPF_CLAMP_DECAY_FACTOR.

Batch command syntax:

```
BAT_OPF_FINAL_CLAMP_TOL   RVAL
```

Python syntax:

```
ierr =  opf_final_clamp_tol(rval)
```

or:

```
ierr,rval = opf_final_clamp_tol()
```

Fortran syntax:

```
CALL OPF_FINAL_CLAMP_TOL(RVAL, IOCODE, IERR)
```

where:

Real RVAL	Is the value of the final clamp tolerance (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 0.001 by default for input).
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in RVAL. IOCODE = 1 return the current value of the option setting in RVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid RVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.28 OPF_FIX_ALL_GENERATORS

Use this API to specify or return the value of the OPF option to treat all generators as nonoptimized.

Batch command syntax:

```
BAT_OPF_FIX_ALL_GENERATORS   IVAL
```

Python syntax:

```
ierr = opf_fix_all_generators(ival)
```

or:

```
ierr,ival = opf_fix_all_generators()
```

Fortran syntax:

```
CALL OPF_FIX_ALL_GENERATORS(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Indicates whether to enable the option to treat all generators as nonoptimized and fix their settings (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 0 by default for input). IVAL = 0 do not treat all generators as nonoptimized. IVAL = 1 treat all generators as nonoptimized.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in IVAL. IOCODE = 1 return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid IVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.29 OPF_FIX_PHASE_SHIFTERS

Use this API to specify or return the value of the OPF option to fix transformer phase shift angle settings.

Batch command syntax:

```
BAT_OPF_FIX_PHASE_SHIFTERS   IVAL
```

Python syntax:

```
ierr = opf_fix_phase_shifters(ival)
```

or:

```
ierr,ival = opf_fix_phase_shifters()
```

Fortran syntax:

```
CALL OPF_FIX_PHASE_SHIFTERS(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Indicates whether to enable the option to fix transformer phase shift angles (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 0 by default for input). IVAL = 0 do not fix transformer phase shift angle settings. IVAL = 1 fix transformer phase shift angle settings.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in IVAL. IOCODE = 1 return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid IVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.30 OPF_FIX_SWITCHED_SHUNTS

Use this API to specify or return the value of the OPF option to fix switched shunt settings.

Batch command syntax:

```
BAT_OPF_FIX_SWITCHED_SHUNTS   IVAL
```

Python syntax:

```
ierr = opf_fix_switched_shunts(ival)
```

or:

```
ierr,ival = opf_fix_switched_shunts()
```

Fortran syntax:

```
CALL OPF_FIX_SWITCHED_SHUNTS(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Indicates whether to enable the option to fix switched shunt settings (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 0 by default for input). IVAL = 0 do not fix switched shunt settings. IVAL = 1 fix switched shunt settings.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in IVAL. IOCODE = 1 return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid IVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.31 OPF_FIX_TAP_RATIOS

Use this API to specify or return the value of the OPF option to fix transformer tap ratio settings.

Batch command syntax:

```
BAT_OPF_FIX_TAP_RATIOS   IVAL
```

Python syntax:

```
ierr = opf_fix_tap_ratios(ival)
```

or:

```
ierr,ival = opf_fix_tap_ratios()
```

Fortran syntax:

```
CALL OPF_FIX_TAP_RATIOS(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Indicates whether to enable the option to fix transformer tap ratio settings (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 0 by default for input). IVAL = 0 do not fix transformer tap ratio settings. IVAL = 1 fix transformer tap ratio settings.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in IVAL. IOCODE = 1 return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid IVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.32 OPF_FIXED_VOLTAGE_PENALTY

Use this API to specify or return the value of the penalty for fixed voltage excursions.

Batch command syntax:

```
BAT_OPF_FIXED_VOLTAGE_PENALTY   RVAL
```

Python syntax:

```
ierr =  opf_fixed_voltage_penalty(rval)
```

or:

```
ierr,rval = opf_fixed_voltage_penalty()
```

Fortran syntax:

```
CALL OPF_FIXED_VOLTAGE_PENALTY(RVAL, IOCODE, IERR)
```

where:

Real RVAL	Is the value of the penalty for fixed voltage excursions (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 100.0 by default for input).
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in RVAL. IOCODE = 1 return the current value of the option setting in RVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid RVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.33 OPF_INITIAL_CLAMP_TOL

Use this API to specify or return the value of the initial clamp tolerance. This API works in conjunction with the API option to CLAMP_NONOPTIMIZED_GENS, and is associated with the APIs for OPF_FINAL_CLAMP_TOL and the OPF_CLAMP_DECAY_FACTOR.

Batch command syntax:

```
BAT_OPF_INITIAL_CLAMP_TOL   RVAL
```

Python syntax:

```
ierr =  opf_initial_clamp_tol(rval)
```

or:

```
ierr,rval = opf_initial_clamp_tol()
```

Fortran syntax:

```
CALL OPF_INITIAL_CLAMP_TOL(RVAL, IOCODE, IERR)
```

where:

Real RVAL	Is the value of the initial clamp tolerance (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 1.0 by default for input).
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in RVAL. IOCODE = 1 return the current value of the option setting in RVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid RVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.34 OPF_INTERIOR_SHIFT_FACTOR

Use this API to specify or return the value of the interior shift factor multiplier. This value indicates by how much a variable should be moved "more interior" of the upper and lower limits.

Batch command syntax:

```
BAT_OPF_INTERIOR_SHIFT_FACTOR   RVAL
```

Python syntax:

```
ierr =  opf_interior_shift_factor(rval)
```

or:

```
ierr,rval = opf_interior_shift_factor()
```

Fortran syntax:

```
CALL OPF_INTERIOR_SHIFT_FACTOR(RVAL, IOCODE, IERR)
```

where:

Real RVAL	Is the interior shift factor multiplier value (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 0.1 by default for input).
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in RVAL. IOCODE = 1 return the current value of the option setting in RVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid RVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.35 OPF_LAMBDA_TOLERANCE

Use this API to specify or return the value of the lambda tolerance.

Batch command syntax:

```
BAT_OPF_LAMBDA_TOLERANCE   RVAL
```

Python syntax:

```
ierr =  opf_lambda_tolerance(rval)
```

or:

```
ierr,rval = opf_lambda_tolerance()
```

Fortran syntax:

```
CALL OPF_LAMBDA_TOLERANCE(RVAL, IOCODE, IERR)
```

where:

Real RVAL	Is the lambda tolerance value (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 100000.0 by default for input).
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in RVAL. IOCODE = 1 return the current value of the option setting in RVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid RVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.36 OPF_LF_CONTROL_PENALTY

Use this API to specify or return the value of the quadratic penalty for loadflow controls. This penalty is applied to optimized loadflow control variable excursions from their initial condition, in $/pu.

Batch command syntax:

```
BAT_OPF_LF_CONTROL_PENALTY   RVAL
```

Python syntax:

```
ierr =  opf_lf_control_penalty(rval)
```

or:

```
ierr,rval = opf_lf_control_penalty()
```

Fortran syntax:

```
CALL OPF_LF_CONTROL_PENALTY(RVAL, IOCODE, IERR)
```

where:

Real RVAL	Is the value of the quadratic penalty for loadflow controls (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 0.0 by default for input).
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in RVAL. IOCODE = 1 return the current value of the option setting in RVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid RVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.37 OPF_MAX_TAP_RATIO_STEP

Use this API to specify or return the value of the maximum tap ratio step.

Batch command syntax:

```
BAT_OPF_MAX_TAP_RATIO_STEP   RVAL
```

Python syntax:

```
ierr =  opf_max_tap_ratio_step(rval)
```

or:

```
ierr,rval = opf_max_tap_ratio_step()
```

Fortran syntax:

```
CALL OPF_MAX_TAP_RATIO_STEP(RVAL, IOCODE, IERR)
```

where:

Real RVAL	Is the maximum tap ratio step value (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 0.0 by default for input).
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in RVAL. IOCODE = 1 return the current value of the option setting in RVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid RVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.38 OPF_MIN_TAP_RATIO_STEP

Use this API to set or return the value of the minimum tap ratio step.

Batch command syntax:

```
BAT_OPF_MIN_TAP_RATIO_STEP   RVAL
```

Python syntax:

```
ierr =  opf_min_tap_ratio_step(rval)
```

or:

```
ierr,rval = opf_min_tap_ratio_step()
```

Fortran syntax:

```
CALL OPF_MIN_TAP_RATIO_STEP(RVAL, IOCODE, IERR)
```

where:

Real RVAL	Is the minimum tap ratio step value (input if Batch command or for Python if specified as an argument, or for FORTRAN if IOCODE is 0; else output 0.0 by default for input).
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in RVAL. IOCODE = 1 return the current value of the option setting in RVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid RVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.39 OPF_REGULATE_AREA_INT

Use this API to set or return the value of the OPF option to regulate area interchange.

Batch command syntax:

```
BAT_OPF_REGULATE_AREA_INT   IVAL
```

Python syntax:

```
ierr = opf_regulate_area_int(ival)
```

or:

```
ierr,ival = opf_regulate_area_int()
```

Fortran syntax:

```
CALL OPF_REGULATE_AREA_INT(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Indicates whether to enable the option to regulate area interchange (input if Batch command or for Python if specified as an argument, or for FORTRAN if IOCODE is 0; else output 0 by default for input). IVAL = 0 do not regulate area interchange. IVAL = 1 regulate area interchange.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in IVAL. IOCODE = 1 return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid IVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.40 OPF_ROUND_SWITCHED_SHUNTS

Use this API to set or return the value of the OPF solution option to discretize switched shunts.

Batch command syntax:

```
BAT_OPF_ROUND_SWITCHED_SHUNTS   IVAL
```

Python syntax:

```
ierr = opf_round_switched_shunts(ival)
```

or:

```
ierr,ival = opf_round_switched_shunts()
```

Fortran syntax:

```
CALL OPF_ROUND_SWITCHED_SHUNTS(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Indicates whether to discretize switched shunt settings (input if Batch command or for Python if specified as an argument, or for FORTRAN if IOCODE is 0; else output 0 by default for input). IVAL = 0 do not discretize switched shunt settings. IVAL = 1 discretize switched shunt settings.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in IVAL. IOCODE = 1 return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid IVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.41 OPF_ROUND_TAP_RATIOS

Use this API to set or return the value of the OPF solution option to round transformer tap ratio settings.

Batch command syntax:

```
BAT_OPF_ROUND_TAP_RATIOS   IVAL
```

Python syntax:

```
ierr = opf_round_tap_ratios(ival)
```

or:

```
ierr,ival = opf_round_tap_ratios()
```

Fortran syntax:

```
CALL OPF_ROUND_TAP_RATIOS(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Indicates whether to round transformer tap ratio settings (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 0 by default for input). IVAL = 0 do not round transformer tap ratio settings. IVAL = 1 round transformer tap ratio settings.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in IVAL. IOCODE = 1 return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid IVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.42 OPF_SCALE_QGEN_LIMITS

Use this API to set or return the value of the OPF option to scale reactive generation limits in the clamped constraint equation.

Batch command syntax:

```
BAT_OPF_SCALE_QGEN_LIMITS   IVAL
```

Python syntax:

```
ierr = opf_scale_qgen_limits(ival)
```

or:

```
ierr,ival = opf_scale_qgen_limits()
```

Fortran syntax:

```
CALL OPF_SCALE_QGEN_LIMITS(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Indicates whether to scale reactive generation limits in the clamp equation (input if Batch command or for Python if specified as an argument, or for FORTRAN if IOCODE is 0; else output 0 by default for input). IVAL = 0 do not scale reactive generation limits. IVAL = 1 scale reactive generation limits.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in IVAL. IOCODE = 1 return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid IVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.43 OPF_STEP_LENGTH_TOLERANCE

Use this API to set or return the value of the OPF minimum barrier step length tolerance.

Batch command syntax:

```
BAT_OPF_STEP_LENGTH_TOLERANCE   RVAL
```

Python syntax:

```
ierr =  opf_step_length_tolerance(rval)
```

or:

```
ierr,rval = opf_step_length_tolerance()
```

Fortran syntax:

```
CALL OPF_STEP_LENGTH_TOLERANCE(RVAL, IOCODE, IERR)
```

where:

Real RVAL	Is the value of the minimum barrier step length tolerance (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 0.00001 by default for input).
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in RVAL. IOCODE = 1 return the current value of the option setting in RVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid RVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.44 OPF_USE_GENERATOR_VSCHED

Use this API to set or return the value of the OPF option to employ the generator scheduled voltage.

Batch command syntax:

```
BAT_OPF_USE_GENERATOR_VSCHED   IVAL
```

Python syntax:

```
ierr = opf_use_generator_vsched(ival)
```

or:

```
ierr,ival = opf_use_generator_vsched()
```

Fortran syntax:

```
CALL OPF_USE_GENERATOR_VSCHED(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Indicates whether to use the generator scheduled voltage value (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 0 by default for input). IVAL = 0 do not use generator scheduled voltage. IVAL = 1 use generator scheduled voltage.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in IVAL. IOCODE = 1 return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid IVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.45 OPTO

This API is now obsolete. Please use the individual OPF option setting API's de-scribed in this section.

Batch command syntax:

```
BAT_OPTO   SID  ALL  IARY(1) ... IARY(36)  RARY(1) ... RARY(19)  OPFILE
RCFILE
```

Python syntax:

```
ierr = opto(sid, all, iary, rary, opfile, rcfile)
```

Fortran syntax:

```
CALL OPTONW(SID, ALL, IARY, RARY, OPFILE, RCFILE, IERR)
```

where:

Integer SID	Is a valid bus subsystem ID from 0 to 11. A bus subsystem must have been previously defined.	
Integer ALL	Is the *all buses* or specified subsystem flag.	
	ALL = 0	process only specified buses.
	ALL = 1	process all buses.
Integer IARY(36)	Is an array of 36 integers, which are defined as follows:	
	IARY(1)	specifies the fuel cost objective.
	IARY(1) = 0	do not employ minimize fuel cost objective.
	IARY(1) = 1	employ the minimize fuel cost objective.
	IARY(2)	specifies the active power slack bus generation objective.
	IARY(2) = 0	do not employ minimize active power slack bus generation objective.
	IARY(2) = 1	employ the minimize active power slack bus generation objective.
	IARY(3)	specifies the reactive power slack generation objective.
	IARY(3) = 0	do not employ the minimize reactive power slack generation objective.
	IARY(3) = 1	employ the minimize reactive power slack generation objective.
	IARY(4)	specifies the active power loss objective.
	IARY(4) = 0	do not employ the minimize active power loss objective.
	IARY(4) = 1	employ the minimize active power loss objective.
	IARY(5)	specifies the reactive power loss objective.
	IARY(5) = 0	do not employ the minimize reactive power loss objective.
	IARY(5) = 1	employ the minimize reactive power loss objective.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IARY(6)	specifies the adjustable branch reactance (series compensation) objective.
IARY(6) = 0	do not employ the minimize adjustable branch reactance objective.
IARY(6) = 1	employ the minimize adjustable branch reactance objective.
IARY(7)	specifies the adjustable bus shunt (var compensation) objective.
IARY(7) = 0	do not employ the minimize adjustable bus shunt objective.
IARY(7) = 1	employ the minimize adjustable bus shunt objective.
IARY(8)	specifies the adjustable bus load (load shedding/increase) objective.
IARY(8) = 0	do not employ the adjustable bus load objective.
IARY(8) = 1	employ the adjustable bus load objective.
IARY(9)	specifies the interface flow objective.
IARY(9) = 0	do not employ the interface flow objective.
IARY(9) = 1	employ the interface flow objective.
IARY(10)	specifies the reactive generation reserve objective.
IARY(10) = 0	do not employ the reactive generation reserve objective.
IARY(10) = 1	employ the reactive generation reserve objective.
IARY(11)	specifies whether to apply the power flow mismatch tolerance to the dual variable problem.
IARY(11) = 0	apply tolerance to the mismatch equation only.
IARY(11) = 1	apply tolerance to the entire equation.
IARY(12)	specifies whether to apply the clamp equation to nonoptimized generators.
IARY(12) = 0	do not clamp nonoptimized generators.
IARY(12) = 1	clamp nonoptimized generators.
IARY(13)	specifies whether to apply the clamp equation to nonoptimized transformers. (*Not currently functional*)
IARY(13) = 0	do not clamp nonoptimized transformers.
IARY(13) = 1	clamp nonoptimized transformers.
IARY(14)	specifies whether to constrain interface flows.
IARY(14) = 0	do not constrain interface flows.
IARY(14) = 1	constrain interface flows.
IARY(15)	specifies whether to employ automatic scaling of the objective.
IARY(15) = 0	do not employ automatic scaling.
IARY(15) = 1	employ automatic scaling.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IARY(16)	specifies whether to regulate area interchange.
IARY(16) = 0	do not regulate area interchange.
IARY(16) = 1	regulate area interchange.
IARY(17)	specifies whether to fix transformer tap ratio settings.
IARY(17) = 0	do not fix transformer tap ratio settings.
IARY(17) = 1	fix transformer tap ratio settings.
IARY(18)	specifies whether to fix transformer phase shift angles.
IARY(18) = 0	do not fix transformer phase shift angles.
IARY(18) = 1	fix transformer phase shift angles.
IARY(19)	specifies whether to fix switched shunt settings.
IARY(19) = 0	do not fix switched shunt settings.
IARY(19) = 1	fix switched shunt settings.
IARY(20)	specifies whether to treat all generators as nonoptimized.
IARY(20) = 0	do not treat all generators as nonoptimized.
IARY(20) = 1	treat all generators as nonoptimized.
IARY(21)	specifies whether to employ the generator scheduled voltage.
IARY(21) = 0	do not employ generator scheduled voltage.
IARY(21) = 1	use generator scheduled voltage.
IARY(22)	specifies whether to scale reactive generation limits on clamped constraint equations.
IARY(22) = 0	do not scale reactive generation limits.
IARY(22) = 1	scale reactive generation limits.
IARY(23)	specifies whether to round transformer tap settings.
IARY(23) = 0	do not round transformer tap settings.
IARY(23) = 1	round transformer tap settings.
IARY(24)	specifies whether to round switched shunt settings.
IARY(24) = 0	do not round switched shunt settings.
IARY(24) = 1	round switched shunt settings.
IARY(25)	specifies whether to limit the step size taken during solution.
IARY(25) = 0	do not limit the step size.
IARY(25) = 1	limit the step size taken.
IARY(26)	specifies whether to produce an optimization log file.
IARY(26) = 0	do not produce an optimization log file.
IARY(26) = 1	produce an optimization log file.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IARY(27)	specifies whether to include a detailed listing of Lagrange multipliers in the optimization log file. IARY(26) must be set to 1 for this to take effect.
IARY(27) = 0	do not print details.
IARY(27) = 1	print Lagrange multipliers.
IARY(28)	specifies whether to produce a PSS®E raw data change file. (*Not currently functional*)
IARY(28) = 0	do not produce an Read/Change File.
IARY(28) = 1	produce an Read/Change File.
IARY(29)	specifies whether to save the OPF solution option settings to the OPF Options File, PSSOPF.OPT.
IARY(29) = 0	do not save settings.
IARY(29) = 1	save settings to PSSOPF.OPT.
IARY(30)	indicates whether the program should open voltage limits to ensure an initial feasible condition.
IARY(30) = 0	do not have the program open the limits for an initial feasible condition.
IARY(30) = 1	have the program open the limits for an initial feasible condition.
IARY(31)	specifies whether to impose emergency bus voltage limits or normal bus voltage limits.
IARY(31) = 0	impose *normal* voltage limits.
IARY(31) = 1	impose *emergency* bus voltage limits.
IARY(32)	specifies whether to impose emergency branch flow limits or normal branch flow limits.
IARY(32) = 0	impose *normal* branch flow limits.
IARY(32) = 1	impose *emergency* branch flow limits.
IARY(33)	specifies whether to include transformer adjustment data in the Read/Change File (if one is produced according to the setting of IARY(28)). (*Not currently functional*)
IARY(33) = 0	do not include transformer adjustment data.
IARY(33) = 1	include transformer adjustment data.
IARY(34)	specifies the maximum iteration limit (corresponds to ITMXN of the power flow solution parameters).
IARY(35)	specifies the bad iteration coarse limit.
IARY(36)	specifies the bad iteration fine limit.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real RARY(19)	Is an array of 19 real elements, which are defined as follows:	
	RARY(1)	specifies the initial barrier function coefficient.
	RARY(2)	specifies the final barrier function coefficient.
	RARY(3)	specifies the step length blow-up tolerance.
	RARY(4)	specifies the final clamp equation tolerance value.
	RARY(5)	specifies the initial clamp equation tolerance value.
	RARY(6)	specifies the clamp equation decay factor.
	RARY(7)	specifies the cost coefficient for the active power loss objective.
	RARY(8)	specifies the cost coefficient for the reactive power loss objective.
	RARY(9)	specifies the penalty value for power flow controls.
	RARY(10)	specifies the cost coefficient for the interface flow objective.
	RARY(11)	specifies the cost coefficient for the reactive generation reserve objective.
	RARY(12)	specifies the penalty value for fixed voltage.
	RARY(13)	specifies the lambda (Lagrange multiplier) blow-up tolerance.
	RARY(14)	specifies the minimum amount to open up tap settings by.
	RARY(15)	specifies the maximum amount to open up tap settings by.
	RARY(16)	specifies the amount to shift the interior point by during solution.
	RARY(17)	specifies the step length taken for the barrier method.
	RARY(18)	specifies the tolerance value used to determine which bus shunts get added to the Read/Change File as switched shunts (if an RDCH file is produced according to IARY(28) and the minimize added vars objective is employed according to IARY(7)). (*Not currently functional*)
	RARY(19)	specifies the solution convergence tolerance, in MVA (corresponds to TOLN of the power flow solution parameters).
Character OPFILE*260	Is the name of the OPF Optimization Log File that will be produced if IARY(26) is set to 1. It is a good idea to specify the filename in quotes.	
Character RCFILE*260	Is the name of the PTI Read Change Data File that will be produced if IARY(28) is set to 1. It is a good idea to specify the filename in quotes. (*Not currently functional*)	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	invalid IARY value.
	IERR = 4	invalid RARY value.
	IERR = 5	error opening OFILE.
	IERR = 6	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.46 P_LOSSES_COST_COEFF

Use this API to set or return the value of the coefficient for the active power loss objective (OPF Solution Options). This API works in conjunction with the MINIMIZE_P_LOSSES objective function option (Section 3.1.17 MINIMIZE_P_LOSSES).

Batch command syntax:

```
BAT_P_LOSSES_COST_COEFF   RVAL
```

Python syntax:

```
ierr =  p_losses_cost_coeff(rval)
```

or:

```
ierr,rval = p_losses_cost_coeff()
```

Fortran syntax:

```
CALL P_LOSSES_COST_COEFF(RVAL, IOCODE, IERR)
```

where:

Real RVAL	Is the value of the active power loss cost coefficient (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 1.0 by default for input).
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in RVAL. IOCODE = 1 return the current value of the option setting in RVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid RVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.47 PRODUCE_OPF_LOG_FILE

Use this API to set or return the value of the OPF solution option to produce an OPF solution log file, and if so, the name of the log file. This API can be used in conjunction with the ADD_DETAILS_TO_OPF_LOG API if additional details of the Lagrange multipliers are desired.

Batch command syntax:

```
BAT_PRODUCE_OPF_LOG_FILE   IVAL  OFILE
```

Python syntax:

```
ierr = produce_opf_log_file(ival, ofile)
```

or:

```
ierr,ival,ofile = produce_opf_log_file()
```

Fortran syntax:

```
CALL PRODUCE_OPF_LOG_FILE(IVAL, IOCODE, OFILE, IERR)
```

where:

Integer IVAL	Indicates whether to produce an OPF Optimization Log File (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 0 by default for input). IVAL = 0 do not produce an OPF solution log file. IVAL = 1 produce an OPF solution log file.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in IVAL. IOCODE = 1 return the current value of the option setting in IVAL.
Character OFILE*260	Is the OPF Optimization Log output filename (input if Batch command or for Python if specified as an argument, or for FORTRAN if IOCODE is 0; else output; DETAIL.DAT by default for input).
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid IVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.48 Q_LOSSES_COST_COEFF

Use this API to set or return the value of the coefficient for the reactive power loss objective. This API works in conjunction with the API option to MINIMIZE_Q_LOSSES.

Batch command syntax:

```
BAT_Q_LOSSES_COST_COEFF   RVAL
```

Python syntax:

```
ierr =  q_losses_cost_coeff(rval)
```

or:

```
ierr,rval = q_losses_cost_coeff()
```

Fortran syntax:

```
CALL Q_LOSSES_COST_COEFF(RVAL, IOCODE, IERR)
```

where:

Real RVAL	Is the value of the reactive power loss cost coefficient (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 1.0 by default for input).
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in RVAL. IOCODE = 1 return the current value of the option setting in RVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid RVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.49 REACTIVE_RESV_COST_COEFF

Use this API to set or return the value of the coefficient for the reactive reserve objective. This API works in conjunction with the API option to MINIMIZE_REACTIVE_RESERVE.

Batch command syntax:

```
BAT_REACTIVE_RESV_COST_COEFF   RVAL
```

Python syntax:

```
ierr =  reactive_resv_cost_coeff(rval)
```

or:

```
ierr,rval = reactive_resv_cost_coeff()
```

Fortran syntax:

```
CALL REACTIVE_RESV_COST_COEFF(RVAL, IOCODE, IERR)
```

where:

Real RVAL	Is the value of the reactive reserve cost coefficient (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output -1.0 by default for input).
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in RVAL. IOCODE = 1 return the current value of the option setting in RVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid RVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.50 ROPF

Use this API to read an Optimal Power Flow Raw Data File.

Batch command syntax:

```
BAT_ROPF   RFILE
```

Python syntax:

```
ierr = ropf(rfile)
```

Fortran syntax:

```
CALL ROPFNW(RFILE, IERR)
```

where:

Character RFILE*260	Is the Optimal Power Flow Data File name, or '1' if input is coming directly from the terminal or Response file (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 no error.
	IERR = 1 error opening file.
	IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.51 RWOP

Use this API to replicate the OPF data contained in the working case in the form of an Optimal Power Flow Data File.

Batch command syntax:

```
BAT_RWOP   SID  ALL  STATUS(1)...STATUS(4)  OUT  OFILE
```

Python syntax:

```
ierr = rwop(sid, all, status, out, ofile)
```

Fortran syntax:

```
CALL RWOPNW(SID, ALL, STATUS, OUT, OFILE, IERR)
```

where:

Integer SID	Is a valid bus subsystem ID from 0 to 11. A bus subsystem must have been previously defined (input; no default allowed).
Integer ALL	Is the *all buses* or specified buses subsystem flag (input; no default allowed). ALL = 1 process all buses. ALL = 0 process only specified buses.
Integer STATUS(4)	Is an array of four elements (input; no default). They are used to control output options. A value of 0 means the option is not enabled. A value of 1 means the option is enabled. They are as follows: STATUS(1) include isolated buses. STATUS(2) include out-of-service branches. STATUS(3) include subsystem data. STATUS(4) include subsystem tie lines.
Integer OUT	Is the output option selection (input; no default allowed). OUT = 0 direct output to a data file specified by OFILE. OUT = 1 direct output to the progress window.
Character OFILE*260	Is the output filename (input; no default).
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 invalid STATUS(n) value. IERR = 4 invalid OUT value. IERR = 5 error opening OFILE. IERR = 6 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.52 SET_OPF_REPORT_SUBSYSTEM

Use this API to define the subsystem to be used when producing the opf output report. The records contained in the report will be limited to just the equipment residing within the defined subsystem. This includes bounded and infeasible reactive power generation, regulated buses, transformer tap ratios, transformer phase shift angles and bounded switched shunts.

Batch command syntax:

```
SET_OPF_REPORT_SUBSYSTEM   OSID  ALL
```

Python syntax:

```
ierr = set_opf_report_subsystem(osid, all)
```

Fortran syntax:

```
CALL SET_OPF_REPORT_SUBSYSTEM(OSID, ALL, IERR)
```

where:

Integer OSID	Indicates the previously defined subsystem number to be used for the opf report. (input; no default allowed)
Integer ALL	Is the *all buses* or specified subsystem flag. (input; no default allowed) ALL = 0 process only specified buses. ALL = 1 process all buses.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.53 USE_DUAL_CRITERIA

Use this API to set or return the value of the OPF solution option to apply the power flow mismatch tolerance to the dual variable problem.

Batch command syntax:

```
BAT_USE_DUAL_CRITERIA   IVAL
```

Python syntax:

```
ierr = use_dual_criteria(ival)
```

or:

```
ierr,ival = use_dual_criteria()
```

Fortran syntax:

```
CALL USE_DUAL_CRITERIA(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Indicates whether to apply the power flow mismatch tolerance to the dual variable problem (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 0 by default for input). IVAL = 0 apply tolerance to the mismatch equation only. IVAL = 1 apply tolerance to the entire equation.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in IVAL. IOCODE = 1 return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid IVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.54 USE_EMERGENCY_FLOW_LIMITS

Use this API to set or return the value of the OPF solution option to impose the *emergency* flow limits instead of the normal flow limits.

Batch command syntax:

```
BAT_USE_EMERGENCY_FLOW_LIMITS   IVAL
```

Python syntax:

```
ierr =  use_emergency_flow_limits(ival)
```

or:

```
ierr,ival = use_emergency_flow_limits()
```

Fortran syntax:

```
CALL USE_EMERGENCY_FLOW_LIMITS(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Indicates whether to use the emergency flow limits instead of the normal bus voltage limits (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 0 by default for input). IVAL = 0 do not impose the emergency flow limits (normal flow limits will be imposed). IVAL = 1 impose emergency flow limits.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in IVAL. IOCODE = 1 return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid IVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.55 USE_EMERGENCY_VOLT_LIMITS

Use this API to set or return the value of the OPF solution option to impose the *emergency* bus voltage limits instead of the normal bus voltage limits.

Batch command syntax:

```
BAT_USE_EMERGENCY_VOLT_LIMITS   IVAL
```

Python syntax:

```
ierr =  use_emergency_volt_limits(ival)
```

or:

```
ierr,ival = use_emergency_volt_limits()
```

Fortran syntax:

```
CALL USE_EMERGENCY_VOLT_LIMITS(IVAL, IOCODE, IERR)
```

where:

Integer IVAL	Indicates whether to use the emergency bus voltage limits instead of the normal bus voltage limits (input if Batch command or for Python if specified as an argument, or for Fortran if IOCODE is 0; else output 0 by default for input). IVAL = 0 do not impose the emergency bus voltage limits (normal voltage limits will be imposed). IVAL = 1 impose emergency bus voltage limits.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE = 0 set the option setting to value specified in IVAL. IOCODE = 1 return the current value of the option setting in IVAL.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid IVAL value. IERR = 2 invalid IOCODE value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.1.56 WRITE_OPF_OPTIONS_FILE

Use this API to save the PSS®E OPF solution option settings to the PSSOPF.OPT file.

Batch command syntax:

```
BAT_WRITE_OPF_OPTIONS_FILE
```

Python syntax:

```
ierr =  write_opf_options_file()
```

Fortran syntax:

```
CALL WRITE_OPF_OPTIONS_FILE(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2 Optimal Power Flow Data Changing APIs

3.2.1 NEWOPF

Use this API to initialize the working case with default OPF data for all bus oriented data records. It must be called prior to introducing any OPF data records. It has no effect if OPF data already exists in the working case.

Batch command syntax:

```
BAT_NEWOPF
```

Python syntax:

```
ierr = newopf()
```

Fortran syntax:

```
CALL NEWOPF_API(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.2 OPF_ADJBRX_INDV

Use this API to add an individual adjustable branch reactance data record to the working case. It may also be used to modify an existing adjustable branch reactance data record.

Batch command syntax:

```
BAT_OPF_ADJBRX_INDV   EIBUS  EJBUS  CKTID  INTGAR(1)...INTGAR(3)
REALAR(1)...REALAR(4)
```

Python syntax:

```
ierr = opf_adjbrx_indv(eibus, ejbus, cktid, intgar, realar)
```

Fortran syntax:

```
CALL OPF_ADJBRX_INDV(EIBUS, EJBUS, CKTID, INTGAR, REALAR, IERR)
```

where:

Integer EIBUS	Is the from bus number (input; no default allowed).
Integer EJBUS	Is the to bus number (input; no default allowed).
Character*2 CKTID	Is the circuit identifier (input; no default allowed).
Integer INTGAR(3)	Is an array of three integer elements (input; default values indicated below). The value of each is as follows:
	INTGAR(1) is the cost curve type. *This value is not currently used by the program.*
	INTGAR(1) = 1 polynomial and exponential cost curve.
	INTGAR(1) = 2 linear cost curve.
	INTGAR(1) = 3 quadratic cost curve.
	INTGAR(2) is the adjustable branch reactance status (default of 1 if new record, current value otherwise).
	INTGAR(2) = 0 out of service.
	INTGAR(2) = 1 in service.
	INTGAR(3) is the cost table cross reference number. *This value is not currently used by the program.*
Real REALAR(4)	Is an array of four real elements (input; default values indicated below). The value of each is as follows:
	REALAR(1) is the reactance multiplier in pu (default value of 1.0 if new record, current value otherwise).
	REALAR(2) is the maximum reactance multiplier in pu (default value of 1.0 if new record, current value otherwise).
	REALAR(3) is the minimum reactance multiplier in pu (default value of 1.0 if new record, current value otherwise).
	REALAR(4) is the cost scale coefficient in $/pu (default value of 1.0 if new record, current value otherwise).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid bus number.
IERR = 2	specified branch does not exist.
IERR = 3	branch reactance table full.
IERR = 4	specified branch cannot be a transformer.
IERR = 5	prerequisite requirements for API are not met.
IERR = -1	non-fatal error occurred. Warning messages are as follows: • Invalid cost curve type specified • Invalid branch reactance status specified. • Maximum branch reactance multiplier less than minimum multiplier • Branch reactance multiplier cannot be less than 0.1. • Branch reactance multiplier cannot be set to zero.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.3 OPF_ADJBRX_SUBSYS

Use this API to add or modify all OPF branch reactance data records in a specified subsystem of the working case.

Batch command syntax:

```
BAT_OPF_ADJBRX_SUBSYS   SID  ALL  INTGAR(1)...INTGAR(3)
REALAR(1)...REALAR(4)  USEVAL(1)...USEVAL(4)
```

Python syntax:

```
ierr = opf_adjbrx_subsys(sid, all, intgar, realar, useval)
```

Fortran syntax:

```
CALL OPF_ADJBRX_SUBSYS(SID, ALL, INTGAR, REALAR, USEVAL, IERR
```

where:

Integer SID	Is a valid subsystem ID from 0 to 11. A subsystem must have been previously defined. (input; no default allowed)	
Integer ALL	Is the *all buses* or specified subsystem flag. (input; no default allowed)	
	ALL = 0	process only specified buses.
	ALL = 1	process all buses.
Integer INTGAR(3)	Is an array of three integer elements (input; default values indicated below). The value of each is as follows:	
	INTGAR(1)	is the cost curve type. *This value is not currently used by the program.*
	INTGAR(1) = 1	polynomial and exponential cost curve.
	INTGAR(1) = 2	linear cost curve.
	INTGAR(1) = 3	quadratic cost curve.
	INTGAR(2)	is the adjustable branch reactance status (default value of 1 for new records, current values otherwise).
	INTGAR(2) = 0	out of service.
	INTGAR(2) = 1	in service.
	INTGAR(3)	is the cost table cross reference number. *This value is not currently used by the program.*
Real REALAR(4)	Is an array of four real elements (input; default values indicated below). The value of each is as follows:	
	REALAR(1)	is the reactance multiplier in pu (default value of 1.0 for new records, current value otherwise).
	REALAR(2)	is the maximum reactance multiplier in pu (default value of 1.0 for new records, current value otherwise).
	REALAR(3)	is the minimum reactance multiplier in pu (default value of 1.0 for new records, current value otherwise).
	REALAR(4)	is the cost scale coefficient in $/pu (default value of 1.0 for new records, current value otherwise).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer USEVAL(4) Is an array of four integer elements (input; default treatment same as value of 0). The value of each element is defined as follows:

USEVAL(1)	indicates how the value of REALAR(1) should be used:
USEVAL(1) = 0	if the value of REALAR(1) is to be used as defined.
USEVAL(1) = 1	if the value of REALAR(1) is to be *multiplied by* the present value of the branch reactance multiplier.
USEVAL(1) = 2	if the value of REALAR(1) is to be *added to* the present value of the branch reactance multiplier.
USEVAL(2)	indicates how the value of REALAR(2) should be used:
USEVAL(2) = 0	if the value of REALAR(2) is to be used as defined.
USEVAL(2) = 1	if the value of REALAR(2) is to be *multiplied by* the present value of the maximum branch reactance multiplier.
USEVAL(2) = 2	if the value of REALAR(2) is to be *added to* the present value of the maximum branch reactance multiplier.
USEVAL(3)	indicates how the value of REALAR(3) should be used:
USEVAL(3) = 0	if the value of REALAR(3) is to be used as defined.
USEVAL(3) = 1	if the value of REALAR(3) is to be *multiplied by* the present value of the minimum branch reactance multiplier.
USEVAL(3) = 2	if the value of REALAR(3) is to be *added to* the present value of the minimum branch reactance multiplier.
USEVAL(4)	indicates how the value of REALAR(4) should be used:
USEVAL(4) = 0	if the value of REALAR(4) is to be used as defined.
USEVAL(4) = 1	if the value of REALAR(4) is to be *multiplied by* the present value of the cost scale coefficient.
USEVAL(4) = 2	if the value of REALAR(4) is to be *added to* the present value of the cost scale coefficient.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid SID value or subsystem SID is not defined.
IERR = 2	invalid ALL value.
IERR = 3	adjustable branch reactance table is full.
IERR = 4	prerequisite requirements for API are not met.
IERR = -1	non-fatal error occurred. Warning messages are as follows: • Invalid cost curve type specified • Invalid branch reactance status specified. • Maximum branch reactance multiplier less than minimum multiplier • Branch reactance multiplier cannot be less than 0.1. • Branch reactance multiplier cannot be set to zero.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.4 OPF_ADJLOAD_TBL

Use this API to add an individual adjustable bus load table record to the working case. It may also be used to modify an existing adjustable bus load table record.

Batch command syntax:

```
BAT_OPF_ADJLOAD_TBL   TBL  INTGAR(1)...INTGAR(3)
REALAR(1)...REALAR(7)
```

Python syntax:

```
ierr = opf_adjload_tbl(tbl, intgar, realar)
```

Fortran syntax:

```
CALL OPF_ADJLOAD_TBL(TBL, INTGAR, REALAR, IERR)
```

where:

Integer TBL	Is the adjustable bus load table number (input; no default allowed).
Integer INTGAR(3)	Is an array of three integer elements (input; default values indicated below). The value of each is as follows:
	INTGAR(1) is the cost curve type. *This value is not currently used by the program.*
	INTGAR(1) = 1 polynomial and exponential cost curve.
	INTGAR(1) = 2 linear cost curve.
	INTGAR(1) = 3 quadratic cost curve.
	INTGAR(2) is the adjustable bus load status (default of 1 if new record, current value otherwise).
	INTGAR(2) = 0 out of service.
	INTGAR(2) = 1 in service.
	INTGAR(3) is the cost table cross reference number. *This value is not currently used by the program.*
Real REALAR(7)	Is an array of seven real elements (input; default values indicated below). The value of each is as follows:
	REALAR(1) is the load multiplier (default value of 1.0 if new record, current value otherwise).
	REALAR(2) is the maximum load multiplier (default value of 1.0 if new record, current value otherwise).
	REALAR(3) is the minimum load multiplier (default value of 1.0 if new record, current value otherwise).
	REALAR(4) is the load ratio multiplier. *This value is not currently used by the program.*
	REALAR(5) is the maximum load ratio multiplier. *This value is not currently used by the program.*
	REALAR(6) is the minimum load ratio multiplier. *This value is not currently used by the program.*
	REALAR(7) is the load cost scale coefficient (default value of 1.0 if new record, current value otherwise).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error occurred
IERR = 1	adjustable bus load table is full
IERR = 2	prerequisite requirements for API are not met
IERR = -1	non-fatal error occurred. Warning messages are as follows: • invalid cost curve type specified • invalid load table status specified • Maximum load multiplier less than 0.0 • Minimum load multiplier less than 0.0 • maximum load multiplier less than minimum load multiplier • maximum load ratio multiplier less than minimum load ratio multiplier.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.5 OPF_ADJVAR_INDV

Use this API to add an individual OPF adjustable bus shunt data record to the working case. It may also be used to modify an existing adjustable bus shunt data record.

Batch command syntax:

```
BAT_OPF_ADJVAR_INDV   EIBUS   SHNTID   INTGAR(1)...INTGAR(3)
REALAR(1)...REALAR(4)
```

Python syntax:

```
ierr = opf_adjvar_indv(eibus, shntid, intgar, realar)
```

Fortran syntax:

```
CALL OPF_ADJVAR_INDV(EIBUS, SHNTID, INTGAR, REALAR, IERR)
```

where:

Integer EIBUS	Is the bus number (input; no default allowed).
Character*2 SHNTID	Is the bus shunt identifier (input; no default allowed).
Integer INTGAR(3)	Is an array of three integer elements (input; default values indicated below). The value of each is as follows.
	INTGAR(1) is the cost curve type. *This value is not currently used by the program.*
	INTGAR(1) = 1 polynomial and exponential cost curve.
	INTGAR(1) = 2 linear cost curve.
	INTGAR(1) = 3 quadratic cost curve.
	INTGAR(2) is the status (default value of 1 if new record, current value otherwise).
	INTGAR(2) = 0 out of service.
	INTGAR(2) = 1 in service.
	INTGAR(3) is the cost table cross reference number. *This value is not currently used by the program.*
Real REALAR(4)	Is an array of four real elements (input; default values indicated below). The value of each is as follows:
	REALAR(1) is the initial bus shunt susceptance in Mvar at nominal voltage (default value of 0.0 if new record, current value otherwise).
	REALAR(2) is the maximum bus shunt susceptance in Mvar (default value of 0.0 if new record, current value otherwise).
	REALAR(3) is the minimum bus shunt susceptance in Mvar (default value of 0.0 if new record, current value otherwise).
	REALAR(4) is the cost scale coefficient in $/Mvar (default value of 1.0 if new record, current value otherwise).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	adjustable bus shunt identifier is more than two characters
IERR = 2	invalid bus number specified.
IERR = 3	blank adjustable bus shunt identifier is invalid
IERR = 4	adjustable bus shunt table is full
IERR = 5	prerequisite requirements for API are not met
IERR = -1	non-fatal error occurred. Possible warning messages are as follows: • Invalid cost curve type specified. • Invalid adjustable bus shunt status specified. • Maximum shunt susceptance less than minimum shunt susceptance.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.6 OPF_ADJVAR_SUBSYS

Use this API to add or modify all OPF adjustable bus shunt records in a specified subsystem of the working case.

Batch command syntax:

```
BAT_OPF_ADJVAR_SUBSYS   SID  ALL  AUTOADD  SHNTID
INTGAR(1)...INTGAR(4)  REALAR(1)...REALAR(4)   USEVAL(1)...USEVAL(4)
```

Python syntax:

```
ierr = opf_adjvar_subsys(sid, all, autoadd, shntid, intgar, realar,
useval)
```

Fortran syntax:

```
CALL OPF_ADJVAR_SUBSYS(SID, ALL, AUTOADD, SHNTID, INTGAR, REALAR,
USEVAL, IERR)
```

where:

Integer SID	Is a valid subsystem ID from 0 to 11. A subsystem must have been previously defined. (input; no default allowed)
Integer ALL	Is the *all buses* or specified subsystem flag (input; no default allowed). ALL = 0 process only specified buses. ALL = 1 process all buses.
Integer AUTOADD	Indicates whether new records can be added (input; default treatment same as value of 3). AUTOADD = 0 do not add a new var record if it does not already exist. AUTOADD = 1 add a new var record if it does not already exist. AUTOADD = 2 add a separate shunt record at the bus. AUTOADD = 3 only modify existing bus shunt records that correspond to the shunt identifier given in SHNTID.
Character*2 SHNTID	Is the adjustable bus shunt identifier (input; default of '1').
Integer INTGAR(4)	Is an array of four integer elements (input; default values indicated below). The value of each is as follows: INTGAR(1) is the cost curve type. *This value is not currently used by the program.* INTGAR(1) = 1 polynomial and exponential cost curve. INTGAR(1) = 2 piece-wise linear cost curve. INTGAR(1) = 3 piece-wise quadratic cost curve. INTGAR(2) is the status (default value of 1 for new records, current value otherwise). INTGAR(2) = 0 out of service. INTGAR(2) = 1 in service. INTGAR(3) is the cost table cross reference number. *This value is not currently used by the program.*

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	INTGAR(4)	indicates which bus types should be processed (default treatment same as value of 0).
	INTGAR(4) = 0	process all buses in subsystem.
	INTGAR(4) = 1	process only Type 1 buses in subsystem.
	INTGAR(4) = 2	process only Type 2 and 3 buses in subsystem.
Real REALAR(4)	Is an array of four real elements (input; default values indicated below). The value of each is as follows:	
	REALAR(1)	is the bus shunt susceptance in Mvar at nominal voltage (default value of 0.0 for new records, current values used for existing ones).
	REALAR(2)	is the maximum bus shunt susceptance in Mvar (default value of 0.0 for new records, current value otherwise).
	REALAR(3)	is the minimum bus shunt susceptance in Mvar (default value of 0.0 for new records, current value otherwise).
	REALAR(4)	is the cost scale coefficient in $/Mvar (default value of 1.0 for new records, current value otherwise).
Integer USEVAL(4)	Is an array of four integer elements (input; default treatment same as value of 0). The value of each element is defined as follows:	
	USEVAL(1)	indicates how the value of REALAR(1) should be used:
	USEVAL(1) = 0	if the value of REALAR(1) is to be used as defined.
	USEVAL(1) = 1	if the value of REALAR(1) is to be *multiplied by* the present value of the bus shunt susceptance.
	USEVAL(1) = 2	if the value of REALAR(1) is to be *added to* the present value of the bus shunt susceptance.
	USEVAL(2)	indicates how the value of REALAR(2) should be used:
	USEVAL(2) = 0	if the value of REALAR(2) is to be used as defined.
	USEVAL(2) = 1	if the value of REALAR(2) is to be *multiplied by* the present value of the maximum bus shunt susceptance.
	USEVAL(2) = 2	if the value of REALAR(2) is to be *added to* the present value of the maximum bus shunt susceptance.
	USEVAL(3)	indicates how the value of REALAR(3) should be used:
	USEVAL(3) = 0	if the value of REALAR(3) is to be used as defined.
	USEVAL(3) = 1	if the value of REALAR(3) is to be *multiplied by* the present value of the minimum bus shunt susceptance.
	USEVAL(3) = 2	if the value of REALAR(3) is to be *added to* the present value of the minimum bus shunt susceptance.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	USEVAL(4)	indicates how the value of REALAR(4) should be used:
	USEVAL(4) = 0	if the value of REALAR(4) is to be used as defined.
	USEVAL(4) = 1	if the value of REALAR(4) is to be *multiplied by* the present value of the cost scale coefficient.
	USEVAL(4) = 2	if the value of REALAR(4) is to be *added to* the present value of the cost scale coefficient.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	adjustable bus shunt identifier is more than two characters
	IERR = 2	invalid SID value or subsystem SID is not defined
	IERR = 3	invalid ALL value
	IERR = 4	blank adjustable bus shunt identifier is invalid
	IERR = 5	adjustable bus shunt table is full
	IERR = 6	prerequisite requirements for API are not met
	IERR = -1	non-fatal error occurred. Possible warning messages are as follows: • Invalid cost curve type specified. • Invalid adjustable bus shunt status specified. • Maximum shunt susceptance less than minimum shunt susceptance.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.7 OPF_APDSP_TBL

Use this API to add an individual active power dispatch table record to the working case. It may also be used to modify an existing active power dispatch table record.

Batch command syntax:

```
BAT_OPF_APDSP_TBL   TBL  INTGAR(1)...INTGAR(3)   REALAR(1)...REALAR(3)
```

Python syntax:

```
ierr = opf_apdsp_tbl(tbl, intgar, realar)
```

Fortran syntax:

```
CALL OPF_APDSP_TBL(TBL, INTGAR, REALAR, IERR)
```

where:

Integer TBL	Is the active power dispatch table number (input; no default allowed).	
Integer INTGAR(3)	Is an array of three integer elements (input; default values indicated below). The value of each is as follows:	
	INTGAR(1)	is the cost curve type (default value of 1 if new record, current value otherwise).
	INTGAR(1) = 1	polynomial and exponential cost curve.
	INTGAR(1) = 2	linear cost curve.
	INTGAR(1) = 3	quadratic cost curve.
	INTGAR(2)	is the active power dispatch table status (default value of 1 if new record, current value otherwise).
	INTGAR(2) = 0	out of service.
	INTGAR(2) = 1	in service.
	INTGAR(3)	is the cost table cross reference number (default value of 0, current value otherwise).
Real REALAR(3)	Is an array of three real elements (input; default values indicated below). The value of each is as follows:	
	REALAR(1)	is the maximum active power generation, in MW (default value of 9999.0, current value otherwise).
	REALAR(2)	is the minimum active power generation, in MW (default value of -9999.0, current value otherwise).
	REALAR(3)	is the fuel cost scale coefficient (default value of 1.0, current value otherwise).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error occurred
IERR = 1	active power dispatch table is full
IERR = 2	prerequisite requirements for API are not met
IERR = -1	non-fatal error occurred. Warning messages are as follows: • invalid cost curve type specified • invalid active power dispatch table status specified • maximum active power generation less than minimum active power generation.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.8 OPF_BRFLW_3WT_INDV

Use this API to add a flow constraint record for an individual winding of a three-winding transformer in the working case. It may also be used to modify an existing three-winding transformer flow constraint record.

Batch command syntax:

```
BAT_OPF_BRFLW_3WT_INDV  EIBUS  EJBUS  EKBUS  CKTID  BFLWID  INTGAR(1)
INTGAR(2)  REALAR(1)...REALAR(5)
```

Python syntax:

```
ierr = opf_brflw_3wt_indv(eibus, ejbus, ekbus, cktid, bflwid, intgar,
realar)
```

Fortran syntax:

```
CALL OPF_BRFLW_3WT_INDV(EIBUS, EJBUS, EKBUS, CKTID, BFLWID, INTGAR,
REALAR, IERR)
```

where:

Integer EIBUS	Is the from bus number (input; no default allowed). The value of EIBUS indicates the from bus of the winding for which the flow constraint is being introduced.
Integer EJBUS	Is the to bus number (input; no default allowed).
Integer EKBUS	Is the third bus number (input; no default allowed.
Character*2 CKTID	Is the circuit identifier (input; no default allowed).
Character*1 BFLWID	Is the branch flow constraint identifier (input; no default allowed).
Integer INTGAR(2)	Is an array of two integer elements (input; default values indicated below). The value of each is as follows:
	INTGAR(1) is the branch flow type (default value of 4 if new record, current value otherwise).
	INTGAR(1) = 1 MW.
	INTGAR(1) = 2 Mvar.
	INTGAR(1) = 3 MVA.
	INTGAR(1) = 4 Ampere.
	INTGAR(2) is the limit type imposed on the branch flow limits (default value of 1 if new record, current value otherwise).
	INTGAR(2) = 0 reporting only.
	INTGAR(2) = 1 hard limit.
	INTGAR(2) = 2 linear soft limit.
	INTGAR(2) = 3 quadratic soft limit.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real REALAR(5) Is an array of five real elements (input; default values indicated below). The value of each is as follows:

REALAR(1)	is the maximum normal flow limit in physical units (default value of 0.0 for new record, current value otherwise).
REALAR(2)	is the minimum normal flow limit in physical units (default value of 0.0 for new record, current value otherwise).
REALAR(3)	is the maximum emergency flow limit in physical units (default value of 0.0 for new record, current value otherwise).
REALAR(4)	is the minimum emergency flow limit in physical units (default value of 0.0 for new record, current value otherwise).
REALAR(5)	is the soft limit penalty weight (default value of 1.0 for new record, current value otherwise).

Integer IERR Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid bus number specified.
IERR = 2	specified three-winding transformer does not exist.
IERR = 3	specified winding of three-winding transformer not found.
IERR = 4	branch flow constraint table is full.
IERR = 5	prerequisite requirements for API are not met.
IERR = -1	non-fatal error occurred. Possible warning messages are: • Invalid branch flow type specified. • Invalid branch flow limit type specified. • Maximum normal flow limit is less than the minimum normal flow limit. • Maximum emergency flow limit is less than the minimum emergency flow limit.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.9 OPF_BRFLW_BRN_INDV

Use this API to add an individual *branch or two-winding transformer* flow constraint record to the working case. It may also be used to modify an existing branch or two-winding transformer flow constraint.

Batch command syntax:

```
BAT_OPF_BRFLW_BRN_INDV   EIBUS  EJBUS  CKTID  BFLWID  INTGAR(1)
INTGAR(2)  REALAR(1)...REALAR(5)
```

Python syntax:

```
ierr = opf_brflw_brn_indv(eibus, ejbus, cktid, bflwid, intgar,
realar)
```

Fortran syntax:

```
CALL OPF_BRFLW_BRN_INDV(EIBUS, EJBUS, CKTID, BFLWID, INTGAR, REALAR,
IERR)
```

where:

Integer EIBUS	Is the from bus number (input; no default allowed).
Integer EJBUS	Is the to bus number (input; no default allowed).
Character*2 CKTID	Is the circuit identifier (input; no default allowed).
Character*1 BFLWID	Is the branch flow constraint identifier (input; no default allowed).
Integer INTGAR(2)	Is an array of two integer elements (input; default values indicated below). The value of each is as follows:
	INTGAR(1) is the branch flow type (default value of 4 if new record, current value otherwise).
	INTGAR(1) = 1 MW.
	INTGAR(1) = 2 Mvar.
	INTGAR(1) = 3 MVA.
	INTGAR(1) = 4 Ampere.
	INTGAR(2) is the limit type imposed on the branch flow limits (default value of 1 if new record, current value otherwise).
	INTGAR(2) = 0 reporting only.
	INTGAR(2) = 1 hard limit.
	INTGAR(2) = 2 linear soft limit.
	INTGAR(2) = 3 quadratic soft limit.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real REALAR(5) Is an array of five real elements (input; default values indicated below). The value of each is as follows:

- REALAR(1) is the maximum normal flow limit in physical units (default value of 0.0 if new record, current value otherwise).
- REALAR(2) is the minimum normal flow limit in physical units (default value of 0.0 if new record, current value otherwise).
- REALAR(3) is the maximum emergency flow limit in physical units (default value of 0.0 if new record, current value otherwise).
- REALAR(4) is the minimum emergency flow limit in physical units (default value of 0.0 if new record, current value otherwise).
- REALAR(5) is the soft limit penalty weight (default value of 1.0 if new record, current value otherwise).

Integer IERR Is the error code (output).

- IERR = 0 no error occurred.
- IERR = 1 invalid bus number specified.
- IERR = 2 specified branch does not exist.
- IERR = 3 branch flow constraint table is full.
- IERR = 4 prerequisite requirements for API are not met.
- IERR = -1 non-fatal error occurred.
  Possible warning messages are:
  - Invalid branch flow type specified.
  - Invalid branch flow limit type specified.
  - Maximum normal flow limit is less than the minimum normal flow limit.
  - Maximum emergency flow limit is less than the minimum emergency flow limit.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.10 OPF_BRFLW_SUBSYS

Use this API to add or modify all branch flow data records in a specified subsystem of the working case.

Batch command syntax:

```
BAT_OPF_BRFLW_SUBSYS   SID  ALL  AUTOADD  BFLWID
INTGAR(1)...INTGAR(3)  REALAR(1)...REALAR(5)  USEVAL(1)...USEVAL(5)
```

Python syntax:

```
ierr = opf_brflw_subsys(sid, all, autoadd, bflwid, intgar, realar,
useval)
```

Fortran syntax:

```
CALL OPF_BRFLW_SUBSYS(SID, ALL, AUTOADD, BFLWID, INTGAR, REALAR,
USEVAL, IERR)
```

where:

Integer SID	Is a valid subsystem ID from 0 to 11. A subsystem must have been previously defined (input; no default allowed).
Integer ALL	Is the *all buses* or specified subsystem flag (input; no default allowed). ALL = 0 process only specified buses. ALL = 1 process all buses.
Integer AUTOADD	Indicates whether new records can be added (input; no default allowed). AUTOADD = 0 do not add a new branch flow record if it does not already exist. AUTOADD = 1 add a new branch flow record if it does not already exist. AUTOADD = 2 only modify existing branch flow records that correspond to the flow identifier given in BFLWID.
Character*1 BFLWID	Is the branch flow identifier (input; no default allowed).
Integer INTGAR(3)	Is an array of three integer elements (input; default values indicated below). The value of each is as follows: INTGAR(1) is the branch flow type (default value of 4 for new records, current value otherwise). INTGAR(1) = 1 MW. INTGAR(1) = 2 Mvar. INTGAR(1) = 3 MVA. INTGAR(1) = 4 Ampere.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	INTGAR(2)	is the limit type imposed on the branch flow limits (default value of 1 for new records, current value otherwise).
	INTGAR(2) = 0	reporting only.
	INTGAR(2) = 1	hard limit.
	INTGAR(2) = 2	linear soft limit.
	INTGAR(2) = 3	quadratic soft limit.
	INTGAR(3)	is the default rate setting (default value of 0).
	INTGAR(3) = 0	use the limits that have been defined.
	INTGAR(3) = 1	base flow limits on RATEA.
	INTGAR(3) = 2	base flow limits on RATEB.
	INTGAR(3) = 3	base flow limits on RATEC.
Real REALAR(5)	Is an array of five real elements (input; default values indicated below). The value of each is as follows:	
	REALAR(1)	is the maximum normal flow limit in physical units (default value of 0.0 for new records, current value otherwise).
	REALAR(2)	is the minimum normal flow limit in physical units (default value of 0.0 for new records, current value otherwise).
	REALAR(3)	is the maximum emergency flow limit in physical units (default value of 0.0 for new records, current value otherwise).
	REALAR(4)	is the minimum emergency flow limit in physical units (default value of 0.0 for new records, current value otherwise).
	REALAR(5)	is the soft limit penalty weight (default value of 1.0 for new records, current value otherwise).
Integer USEVAL(5)	Is an array of five integer elements (input; default treatment same as value of 0). The value of each element is defined as follows:	
	USEVAL(1)	indicates how the value of REALAR(1) should be used:
	USEVAL(1) = 0	if the value of REALAR(1) is to be used as defined.
	USEVAL(1) = 1	if the value of REALAR(1) is to be *multiplied by* the present value of the maximum normal flow limit.
	USEVAL(1) = 2	if the value of REALAR(1) is to be *added to* the present value of the maximum normal flow limit.
	USEVAL(2)	indicates how the value of REALAR(2) should be used:
	USEVAL(2) = 0	if the value of REALAR(2) is to be used as defined.
	USEVAL(2) = 1	if the value of REALAR(2) is to be *multiplied by* the present value of the minimum normal flow limit.
	USEVAL(2) = 2	if the value of REALAR(2) is to be *added to* the present value of the minimum normal flow limit.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

USEVAL(3)	indicates how the value of REALAR(3) should be used:
USEVAL(3) = 0	if the value of REALAR(3) is to be used as defined.
USEVAL(3) = 1	if the value of REALAR(3) is to be *multiplied by* the present value of the maximum emergency flow limit.
USEVAL(3) = 2	if the value of REALAR(3) is to be *added to* the present value of the maximum emergency flow limit.
USEVAL(4)	indicates how the value of REALAR(4) should be used:
USEVAL(4) = 0	if the value of REALAR(4) is to be used as defined.
USEVAL(4) = 1	if the value of REALAR(4) is to be *multiplied by* the present value of the minimum emergency flow limit.
USEVAL(4) = 2	if the value of REALAR(4) is to be *added to* the present value of the minimum emergency flow limit.
USEVAL(5)	indicates how the value of PENWGT should be used:
USEVAL(5) = 0	if the value of PENWGT is to be used as defined.
USEVAL(5) = 1	if the value of PENWGT is to be *multiplied by* the present value of the soft limit penalty weight.
USEVAL(5) = 2	if the value of PENWGT is to be *added to* the present value of the soft limit penalty weight.

Integer IERR — Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid SID value or subsystem SID is not defined.
IERR = 2	invalid ALL value.
IERR = 3	branch flow constraint table is full.
IERR = 4	prerequisite requirements for API are not met.
IERR = -1	non-fatal error occurred. Possible warning messages are: • Invalid flow type specified. • Invalid branch flow limit type specified. • Maximum normal flow limit is less than the minimum normal flow limit. • Maximum emergency flow limit is less than the minimum emergency flow limit.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.11 OPF_BUS_INDV

Use this API to add OPF bus voltage magnitude data to the working case.

Batch command syntax:

```
BAT_OPF_BUS_INDV   EIBUS   INTGAR(1)   INTGAR(2)   REALAR(1)...REALAR(5)
```

Python syntax:

```
ierr = opf_bus_indv(eibus, intgar, realar)
```

Fortran syntax:

```
CALL OPF_BUS_INDV(EIBUS, INTGAR, REALAR, IERR)
```

where:

Integer EIBUS	Is the bus number (input; no default allowed).
Integer INTGAR(2)	Is an array of two integer elements (input; default values indicated below). The value of each is as follows:
	INTGAR(1) is the limit type imposed on the bus voltage limits (default is the current value; initialized to 1).
	INTGAR(1) = 0 reporting only.
	INTGAR(1) = 1 hard limit.
	INTGAR(1) = 2 linear soft limit.
	INTGAR(1) = 3 quadratic soft limit.
	INTGAR(2) indicates what to do with the defined voltage limits (default treatment same as value of 0).
	INTGAR(2) = 0 leave voltage limits as defined by REALAR(1) and REALAR(2).
	INTGAR(2) = 1 fix normal voltage limit at present voltage values, ignoring any values defined by REALAR(1) and REALAR(2).
	INTGAR(2) = 2 allow normal voltage limits to open up if they exceed the bounds defined by REALAR(1) or REALAR(2).
	INTGAR(2)=3 set normal and emergency OPF voltage limits to the normal and emergency voltage limits as defined in the network bus data.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real REALAR(5) Is an array of five real elements (input; default values indicated below). The value of each is as follows:

- REALAR(1) is the normal maximum voltage limit in pu (default is the current value; initialized to 9999.0).
- REALAR(2) is the normal minimum voltage limit in pu (default is the current value; initialized to -9999.0).
- REALAR(3) is the emergency maximum voltage limit in pu (default is the current value; initialized to 9999.0).
- REALAR(4) is the emergency minimum voltage limit in pu (default is the current value; initialized to -9999.0).
- REALAR(5) is the soft-limit penalty weight (default is the current value; initialized to 1.0).

Integer IERR Is the error code (output).

- IERR = 0 no error occurred.
- IERR = 1 invalid bus number specified.
- IERR = 2 prerequisite requirements for API are not met.
- IERR = -1 non-fatal error occurred.
  Warning messages are as follows:
  - Invalid limit type specified.
  - Maximum normal voltage limit less than minimum limit.
  - Maximum emergency voltage limit less than minimum limit.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.12 OPF_BUS_SUBSYS

Use this API to add OPF bus attribute data for all records in a specified subsystem of the working case.

Batch command syntax:

```
BAT_OPF_BUS_SUBSYS   SID  ALL  INTGAR(1)...INTGAR(3)
REALAR(1)...REALAR(5)  USEVAL(1)...USEVAL(5)
```

Python syntax:

```
ierr = opf_bus_subsys(sid, all, intgar, realar, useval)
```

Fortran syntax:

```
CALL OPF_BUS_SUBSYS(SID, ALL, INTGAR, REALAR, USEVAL, IERR)
```

where:

Integer SID	Is a valid subsystem ID from 0 to 11. A subsystem must have been previously defined (input; no default allowed).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; no default allowed).	
	ALL = 0	process only specified buses.
	ALL = 1	process all buses, use existing data values.
	ALL = 2	process all buses, use default data values.
Integer INTGAR(3)	Is an array of three integer elements (input; default values indicated below). The values of each are as follows:	
	INTGAR(1)	is the limit type imposed on the bus voltage limits (default is the current value; initialized to 1).
	INTGAR(1) = 0	reporting only.
	INTGAR(1) = 1	hard limit.
	INTGAR(1) = 2	linear soft limit.
	INTGAR(1) = 3	quadratic soft limit.
	INTGAR(2)	indicates what to do with the defined voltage limits (default treatment same as value of 0).
	INTGAR(2) = 0	leave voltage limits as defined by REALAR(1) and REALAR(2).
	INTGAR(2) = 1	fix normal voltage limit at present voltage values, ignoring any values defined by REALAR(1) and REALAR(2).
	INTGAR(2) = 2	allow normal voltage limits to open up if they exceed the bounds defined by REALAR(1) or REALAR(2).
	INTGAR(2)=3	set normal and emergency OPF voltage limits to the normal and emergency voltage limits as defined in the network bus data.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	INTGAR(3)	indicates which bus types should be processed (default treatment same as value of 0).
	INTGAR(3) = 0	all buses in subsystem.
	INTGAR(3) = 1	only Type 1 buses in subsystem.
	INTGAR(3) = 2	only Type 2 and 3 buses in subsystem.
Real REALAR(5)	Is an array of five real elements. The value of each is as follows:	
	REALAR(1)	is the normal maximum voltage limit in pu (default is the current value; initialized to 9999.0).
	REALAR(2)	is the normal minimum voltage limit in pu (default is the current value; initialized to -9999.0).
	REALAR(3)	is the emergency maximum voltage limit in pu (default is the current value; initialized to 9999.0).
	REALAR(4)	is the emergency minimum voltage limit in pu (default is the current value; initialized to -9999.0).
	REALAR(5)	is the soft-limit penalty weight (default is the current value; initialized to 1.0).
Integer USEVAL(5)	Is an array of five integer elements (input; default treatment same as value of 0). The value of each element is defined as follows:	
	USEVAL(1)	indicates how the value of REALAR(1) should be used:
	USEVAL(1) = 0	if the value of REALAR(1) is to be used as defined.
	USEVAL(1) = 1	if the value of REALAR(1) is to be *multiplied by* the present value of the normal bus voltage minimum limit.
	USEVAL(1) = 2	if the value of REALAR(1) is to be *added to* the present value of the normal bus voltage minimum limit.
	USEVAL(2)	indicates how the value of REALAR(2) should be used:
	USEVAL(2) = 0	if the value of REALAR(2) is to be used as defined.
	USEVAL(2) = 1	if the value of REALAR(2) is to be *multiplied by* the present value of the normal bus voltage maximum limit.
	USEVAL(2) = 2	if the value of REALAR(2) is to be *added to* the present value of the normal bus voltage maximum limit.
	USEVAL(3)	indicates how the value of REALAR(3) should be used:
	USEVAL(3) = 0	if the value of REALAR(3) is to be used as defined.
	USEVAL(3) = 1	if the value of REALAR(3) is to be *multiplied by* the present value of the emergency bus voltage minimum limit.
	USEVAL(3) = 2	if the value of REALAR(3) is to be *added to* the present value of the emergency bus voltage minimum limit.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	USEVAL(4)	indicates how the value of REALAR(4) should be used:
	USEVAL(4) = 0	if the value of REALAR(4) is to be used as defined.
	USEVAL(4) = 1	if the value of REALAR(4) is to be *multiplied by* the present value of the emergency bus voltage maximum limit.
	USEVAL(4) = 2	if the value of REALAR(4) is to be *added to* the present value of the emergency bus voltage maximum limit.
	USEVAL(5)	indicates how the value of PENWGT should be used:
	USEVAL(5) = 0	if the value of PENWGT is to be used as defined.
	USEVAL(5) = 1	if the value of PENWGT is to be *multiplied by* the present value of the soft limit penalty weight.
	USEVAL(5) = 2	if the value of PENWGT is to be *added to* the present value of the soft limit penalty weight.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	prerequisite requirements for API are not met.
	IERR = -1	non-fatal error occurred. Warning messages are as follows: • Invalid limit type specified. • Maximum normal voltage limit less than minimum limit. • Maximum emergency voltage limit less than minimum limit.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.13 OPF_CSTTBL_LIN

Use this API to add a linear cost curve table record to the working case. It may also be used to modify an existing linear cost curve table record.

Batch command syntax:

```
BAT_OPF_CSTTBL_LIN   TBL  LABL  NPRS

X1, Y1

...

XNPRS, YNPRS
```

Python syntax:

```
ierr = opf_csttbl_lin(tbl, labl, nprs, xy)
```

Fortran syntax:

```
CALL OPF_CSTTBL_LIN(TBL, LABL, NPRS, XY, IERR)
```

where:

Integer TBL	Is the linear cost curve table number (input; no default allowed).
Character*12 LABL	Is the cost table description of, at most, 12 characters (input; default of ' ').
Integer NPRS	Is the number of (X_i, Y_i) coordinate pairs being specified (input; no default allowed).
Real XY(2,NPRS)	Is the list of (X_i, Y_i) coordinate pairs from 1 to NPRS (input; default values indicated below). X_i is the control variable value (default of 0.0) Y_i is the total cost or energy consumption value (default of 0.0).
Integer IERR	Is the error code (output). IERR = 0 no error occurred IERR = 1 cost table data table is full IERR = 2 exceeded maximum number of allowable cost curve data pairs IERR = 3 number of data pairs must not be less than zero IERR = 4 incremental costs are not monotonically increasing IERR = 5 MW values are not monotonically increasing IERR = 6 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.14 OPF_CSTTBL_POLY

Use this API to add a polynomial and exponential cost curve table record to the working case. It may also be used to modify an existing polynomial and exponential cost curve table record.

Batch command syntax:

```
BAT_OPF_CSTTBL_POLY   TBL  LABL  COST  LINCST  QUACST  EXPCST  EXPON
```

Python syntax:

```
ierr = opf_csttbl_poly(tbl, labl, cost, lincst, quacst, expcst,
expon)
```

Fortran syntax:

```
CALL OPF_CSTTBL_POLY(TBL, LABL, COST, LINCST, QUACST, EXPCST, EXPON,
IERR)
```

where:

Integer TBL	Is the polynomial and exponential cost curve table number (input; no default allowed).
Character*12 LABL	Is the cost table description of, at most, 12 characters (input; default of ' ').
Real COST	Is the cost integration constant (input; default of 0.0).
Real LINCST	Is the linear cost coefficient value (input; default of 1.0).
Real QUACST	Is the quadratic cost coefficient value (input; default of 1.0).
Real EXPCST	Is the exponential cost coefficient value (input; default of 0.0).
Real EXPON	Is the exponent of the exponential cost coefficient (input; default of 0.0).
Integer IERR	Is the error code (output). IERR = 0 no error occurred IERR = 1 cost table data table full IERR = 2 exceeded maximum number of allowable cost curve data values IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.15 OPF_CSTTBL_QUAD

Use this API to add a quadratic cost curve table record to the working case. It may also be used to modify an existing quadratic cost curve table record.

Batch command syntax:

```
BAT_OPF_CSTTBL_QUAD   TBL  LABL  COST  NPRS

X1, Y1

...

XNPRS, YNPRS
```

Python syntax:

```
ierr = opf_csttbl_quad(tbl, labl, cost, nprs, xy)
```

Fortran syntax:

```
CALL OPF_CSTTBL_QUAD(TBL, LABL, COST, NPRS, XY, IERR)
```

where:

Integer TBL	Is the quadratic cost curve table number (input; no default allowed).	
Character*12 LABL	Is the cost table description of, at most, 12 characters (input; default of ' ').	
Real COST	Is the cost integration constant (input; default of 0.0).	
Integer NPRS	Is the number of (X_i, Y_i) coordinate pairs to be specified (input; no default).	
Real XY(2,NPRS)	Is the list of (X_i, Y_i) coordinate pairs from 1 to NPRS (input; default values indicated below).	
	X_i	is the control variable value (default of 0.0).
	Y_i	is the incremental cost or energy consumption value (default of 0.0).
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	cost table data table is full.
	IERR = 2	exceeded maximum number of allowable cost curve data pairs.
	IERR = 3	number of data pairs must not be negative.
	IERR = 4	incremental costs are not monotonically increasing.
	IERR = 5	MW values are not monotonically increasing.
	IERR = 6	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.16 OPF_GEN_RCAP_INDV

Use this API to add an individual generation reactive capability data record to the working case. It may also be used to modify an existing generation reactive capability data record.

Batch command syntax:

```
BAT_OPF_GEN_RCAP_INDV   EIBUS   GENID   RCSTAT   REALAR(1)...REALAR(5)
```

Python syntax:

```
ierr = opf_gen_rcap_indv(eibus, genid, rcstat, realar)
```

Fortran syntax:

```
CALL OPF_GEN_RCAP_INDV(EIBUS, GENID, RCSTAT, REALAR, IERR)
```

where:

Integer EIBUS	Is the bus number (input; no default allowed).
Character*2 GENID	Is the machine identifier (input; default of ' ').
Integer RCSTAT	Is the reactive capability limit status (input; default of 4): RCSTAT = 0 out of service. RCSTAT = 1 in service, no reactive generation limits. RCSTAT = 2 in service, inhibiting field increase. RCSTAT = 3 in service, inhibiting field decrease. RCSTAT = 4 in service, fixed Efd. RCSTAT = 5 in service, apply reactive generation limits.
Real REALAR(5)	Is an array of five real elements (input; default values indicated below). The value of each is as follows: REALAR(1) is the direct-axis synchronous reactance in pu on Mbase (default of 1.0 if new record, current value otherwise). REALAR(2) is the stator current limit in pu on Mbase (default of 1.0 if new record, current value otherwise). REALAR(3) is the rated lagging power factor (default of 1.0 if new record, current value otherwise). REALAR(4) is the rated leading power factor (default of 1.0 if new record, current value otherwise). REALAR(5) is the maximum reactive absorption in pu on Mbase (default of 1.0 if new record, current value otherwise).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid bus number specified.
	IERR = 2	specified machine does not exist.
	IERR = 3	generator reactive capability data table is full.
	IERR = 4	prerequisite requirements for API are not met.
	IERR = -1	non-fatal error occurred. Possible warning messages are: • Invalid reactive capability status specified.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.17 OPF_GEN_RCAP_SUBSYS

Use this API to add or modify all OPF generator reactive capability records in a specified subsystem of the working case.

Batch command syntax:

```
BAT_OPF_GEN_RCAP_SUBSYS   SID  ALL  RCSTAT  REALAR(1)...REALAR(5)
USEVAL(1)...USEVAL(5)
```

Python syntax:

```
ierr = opf_gen_rcap_subsys(sid, all, rcstat, realar, useval)
```

Fortran syntax:

```
CALL OPF_GEN_RCAP_SUBSYS(SID, ALL, RCSTAT, REALAR, USEVAL, IERR)
```

where:

Integer SID	Is a valid subsystem ID from 0 to 11. A subsystem must have been previously defined (input; no default allowed).
Integer ALL	Is the *all buses* or specified subsystem flag (input; no default allowed). ALL = 0 process only specified buses. ALL = 1 process all buses.
Real REALAR(5)	Is an array of five real elements (input; default values indicated below). The value of each is as follows. REALAR(1) is the direct-axis synchronous reactance in pu on Mbase (default value of 1.0 for new records, current value otherwise). REALAR(2) is the stator current limit in pu on Mbase (default value of 1.0 for new records, current value otherwise). REALAR(3) is the rated lagging power factor (default value of 1.0 for new records, current value otherwise). REALAR(4) is the rated leading power factor (default value of 1.0 for new records, current value otherwise). REALAR(5) is the maximum reactive absorption in pu on Mbase (default value of 1.0 for new records, current value otherwise).
Integer RCSTAT	Is the reactive capability limit status (input; default value of 4 for new records, current value otherwise): RCSTAT = 0 out of service. RCSTAT = 1 in service, no reactive generation limits. RCSTAT = 2 in service, inhibiting field increase. RCSTAT = 3 in service, inhibiting field decrease. RCSTAT = 4 in service, fixed Efd. RCSTAT = 5 in service, apply reactive generation limits.
Integer USEVAL(5)	Is an array of five integer elements (input; default treatment same as value of 0). The value of each element is defined as follows:

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

USEVAL(1)	indicates how the value of REALAR(1) should be used:
USEVAL(1) = 0	if the value of REALAR(1) is to be used as defined.
USEVAL(1) = 1	if the value of REALAR(1) is to be *multiplied by* the present value of the direct-axis synchronous reactance.
USEVAL(1) = 2	if the value of REALAR(1) is to be *added to* the present value of the direct-axis synchronous reactance.
USEVAL(2)	indicates how the value of REALAR(2) should be used:
USEVAL(2) = 0	if the value of REALAR(2) is to be used as defined.
USEVAL(2) = 1	if the value of REALAR(2) is to be *multiplied by* the present value of the stator current limit.
USEVAL(2) = 2	if the value of REALAR(2) is to be *added to* the present value of the stator current limit.
USEVAL(3)	indicates how the value of REALAR(3) should be used:
USEVAL(3) = 0	if the value of REALAR(3) is to be used as defined.
USEVAL(3) = 1	if the value of REALAR(3) is to be *multiplied by* the present value of the rated lagging power factor.
USEVAL(3) = 2	if the value of REALAR(3) is to be *added to* the present value of the rated lagging power factor.
USEVAL(4)	indicates how the value of REALAR(4) should be used:
USEVAL(4) = 0	if the value of REALAR(4) is to be used as defined.
USEVAL(4) = 1	if the value of REALAR(4) is to be *multiplied by* the present value of the rated leading power factor.
USEVAL(4) = 2	if the value of REALAR(4) is to be *added to* the present value of the rated leading power factor.
USEVAL(5)	indicates how the value of REALAR(5) should be used:
USEVAL(5) = 0	if the value of REALAR(5) is to be used as defined.
USEVAL(5) = 1	if the value of REALAR(5) is to be *multiplied by* the present value of the maximum reactive absorption.
USEVAL(5) = 2	if the value of REALAR(5) is to be *added to* the present value of the maximum reactive absorption.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	generator reactive capability data table is full.
	IERR = 4	prerequisite requirements for API are not met.
	IERR = -1	non-fatal error occurred. Possible warning messages are: • Invalid reactive capability status specified.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.18 OPF_GENDSP_INDV

Use this API to add generator dispatch data to an individual record in the working case.

Batch command syntax:

```
BAT_OPF_GENDSP_INDV   EIBUS  GENID  APDTBL  DSPTCH
```

Python syntax:

```
ierr = opf_gendsp_indv(eibus, genid, apdtbl, dsptch)
```

Fortran syntax:

```
CALL OPF_GENDSP_INDV(EIBUS, GENID, APDTBL, DSPTCH, IERR)
```

where:

Integer EIBUS	Is the bus number (input; no default allowed).
Character*2 GENID	Is the machine identifier (input; no default allowed).
Integer APDTBL	Is the active power dispatch table cross reference number (input; default of 0 if new record, current value otherwise).
Real DSPTCH	Is the dispatch fraction (input; default value of 1.0 if new record, current value otherwise).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 invalid bus number specified.
	IERR = 2 specified machine does not exist.
	IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.19 OPF_GENDSP_SUBSYS

Use this API to add OPF generator dispatch data to all records in a specified subsystem of the working case.

Batch command syntax:

```
BAT_OPF_GENDSP_SUBSYS   SID  ALL  APDTBL  DSPTCH  USEVAL
```

Python syntax:

```
ierr = opf_gendsp_subsys(sid, all, apdtbl, dsptch, useval)
```

Fortran syntax:

```
CALL OPF_GENDSP_SUBSYS(SID, ALL, APDTBL, DSPTCH, USEVAL, IERR)
```

where:

Integer SID	Is a valid subsystem ID from 0 to 11. A subsystem must have been previously defined (input; no default allowed).
Integer ALL	Is the *all buses* or specified subsystem flag (input; no default allowed). ALL = 0 process only specified buses. ALL = 1 process all buses, use existing data values. ALL = 2 process all buses, use default data values.
Integer APDTBL	Is the active power dispatch table cross reference number (input; default of 0 for new record, current value otherwise).
Real DSPTCH	Is the dispatch fraction (input; default of 1.0 for new record, current value otherwise).
Integer USEVAL	Indicates how the value of DSPTCH should be applied (input; default treatment same as value of 0). USEVAL = 0 if the value of DSPTCH is to be used as defined. USEVAL = 1 if the value of DSPTCH is to be *multiplied by* the present value of the dispatch fraction. USEVAL = 2 if the value of DSPTCH is to be *added to* the present value of the dispatch fraction.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.20 OPF_GENRSV_INDV

Use this API to add an individual generator reserve data record to the working case. It may also be used to modify an existing generator reserve data record.

Batch command syntax:

```
BAT_OPF_GENRSV_INDV   EIBUS  GENID  REALAR(1)  REALAR(2)
```

Python syntax:

```
ierr = opf_genrsv_indv(eibus, genid, realar)
```

Fortran syntax:

```
CALL OPF_GENRSV_INDV(EIBUS, GENID, REALAR, IERR)
```

where:

Integer EIBUS	Is the bus number (input; no default allowed).
Character*2 GENID	Is the machine identifier (input; no default allowed).
Real REALAR(2)	Is an array of two real elements (input; default values indicated below). The value of each is as follows: REALAR(1) is the unit ramp rate in MW/minute (default value of 9999.0 if new record, current value otherwise). REALAR(2) is the unit capability in MW (default value of 9999.0 if new record, current value otherwise).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid bus number specified. IERR = 2 specified machine does not exist. IERR = 3 generator reserve unit table full. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.21 OPF_GENRSV_SUBSYS

Use this API to add or modify all OPF generator reserve records in a specified subsystem of the working case.

Batch command syntax:

```
BAT_OPF_GENRSV_SUBSYS   SID  ALL  REALAR(1)  REALAR(2)  USEVAL(1)
USEVAL(2)
```

Python syntax:

```
ierr = opf_genrsv_subsys(sid, all, realar, useval)
```

Fortran syntax:

```
CALL OPF_GENRSV_SUBSYS(SID, ALL, REALAR, USEVAL, IERR)
```

where:

Integer SID	Is a valid subsystem ID from 0 to 11. A subsystem must have been previously defined (input; no default allowed).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; no default allowed).	
	ALL = 0	process only specified buses.
	ALL = 1	process all buses.
Real REALAR(2)	Is an array of two real elements (input; default values indicated below). The value of each is as follows:	
	REALAR(1)	is the unit ramp rate in MW/minute (default value of 9999.0 for new records, current value used otherwise).
	REALAR(2)	is the unit capability in MW (default value of 9999.0 for new records, current value used otherwise).
Integer USEVAL(2)	Is an array of two integer elements (input; default treatment same as value of 0). The value of each element is defined as follows:	
	USEVAL(1)	indicates how the value of REALAR(1) should be used:
	USEVAL(1) = 0	if the value of REALAR(1) is to be used as defined.
	USEVAL(1) = 1	if the value of REALAR(1) is to be *multiplied by* the present value of the unit ramp rate.
	USEVAL(1) = 2	if the value of REALAR(1) is to be *added to* the present value of the unit ramp rate.
	USEVAL(2)	indicates how the value of REALAR(2) should be used:
	USEVAL(2) = 0	if the value of REALAR(2) is to be used as defined.
	USEVAL(2) = 1	if the value of REALAR(2) is to be *multiplied by* the present value of the unit capability.
	USEVAL(2) = 2	if the value of REALAR(2) is to be *added to* the present value of the unit capability.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	generator reserve unit table is full.
	IERR = 4	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.22 OPF_INTFLW_3WT

Use this API to add a participating three-winding transformer winding to a specified interface flow constraint in the working case.

Batch command syntax:

```
BAT_OPF_INTFLW_3WT   IFID  EIBUS  EJBUS  EKBUS  CKTID
```

Python syntax:

```
ierr = opf_intflw_3wt(ifid, eibus, ejbus, ekbus, cktid)
```

Fortran syntax:

```
CALL OPF_INTFLW_3WT(IFID, EIBUS, EJBUS, EKBUS, CKTID, IERR)
```

where:

Integer IFID	Is the interface flow identifier (input; no default allowed).
Integer EIBUS	Is the from bus number (input; no default allowed). The value of EIBUS indicates the from bus of the winding for which the flow constraint is being introduced.
Integer EJBUS	Is the to bus number (input; no default allowed).
Integer EKBUS	Is the third bus number (input; no default allowed).
Character*2 CKTID	Is the circuit identifier (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 participating branch table is full. IERR = 2 invalid interface flow identifier specified. IERR = 3 invalid bus number specified. IERR = 4 specified three-winding transformer does not exist. IERR = 5 specified winding is already participating in the interface. IERR = 6 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.23 OPF_INTFLW_BRN

Use this API to add a participating branch to a specified interface flow constraint in the working case.

Batch command syntax:

```
BAT_OPF_INTFLW_BRN   IFID  EIBUS  EJBUS  CKTID
```

Python syntax:

```
ierr = opf_intflw_brn(ifid, eibus, ejbus, cktid)
```

Fortran syntax:

```
CALL OPF_INTFLW_BRN(IFID, EIBUS, EJBUS, CKTID, IERR)
```

where:

Integer IFID	Is the interface flow identifier (input; no default allowed).
Integer EIBUS	Is the from bus number (input; no default allowed).
Integer EJBUS	Is the to bus number (input; no default allowed).
Character*2 CKTID	Is the circuit identifier (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 — no error occurred.
	IERR = 1 — participating branch table is full.
	IERR = 2 — invalid interface flow identifier specified.
	IERR = 3 — invalid bus number specified.
	IERR = 4 — specified branch does not exist.
	IERR = 5 — specified branch is already participating in the interface.
	IERR = 6 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.24 OPF_INTFLW_MAIN

Use this API to add an interface flow constraint record to the working case. It may also be used to modify an existing interface flow constraint record.

This API must be invoked prior to adding any participating branch or three-winding transformer flows to the interface flow constraint.

Batch command syntax:

```
BAT_OPF_INTFLW_MAIN   IFID  LABL  INTGAR(1)  INTGAR(2)
REALAR(1)...REALAR(3)
```

Python syntax:

```
ierr = opf_intflw_main(ifid, labl, intgar, realar)
```

Fortran syntax:

```
CALL OPF_INTFLW_MAIN(IFID, LABL, INTGAR, REALAR, IERR)
```

where:

Integer IFID	Is the interface flow identifier (input; no default allowed).
Character*32 LABL	Is the interface flow description made up of, at most, 32 characters (input; default of ' ').
Integer INTGAR(2)	Is an array of two integer elements (input; default values indicated below). The values of each are as follows:
	INTGAR(1) is the interface flow type (default value of 1 if new record, current value otherwise).
	INTGAR(1) = 1 MW.
	INTGAR(1) = 2 Mvar.
	INTGAR(2) is the interface flow limit type (default value of 1 if new record, current value otherwise).
	INTGAR(2) = 0 reporting only.
	INTGAR(2) = 1 hard limit.
	INTGAR(2) = 2 linear soft limit.
	INTGAR(2) = 3 quadratic soft limit.
Real REALAR(3)	Is an array of three real elements (input; default values indicated below). The values of each are as follows:
	REALAR(1) is the maximum interface flow, in physical units (default value of 0.0 if new record, current value otherwise).
	REALAR(2) is the minimum interface flow, in physical units (default value of 0.0 if new record, current value otherwise).
	REALAR(3) is the soft limit penalty weight (default value of 1.0 if new record, current value otherwise).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	interface flow constraint table is full.
IERR = 2	prerequisite requirements for API are not met.
IERR = -1	non-fatal error occurred. Possible warning messages are: • Invalid flow type specified. • Invalid interface flow limit type specified. • Maximum interface flow is less than the minimum interface flow.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.25 OPF_INTFLW_RESET

Use this API to remove all interface branches associated with a particular interface flow constraint in the working case.

Batch command syntax:

```
BAT_OPF_INTFLW_RESET   IFID
```

Python syntax:

```
ierr = opf_intflw_reset(ifid)
```

Fortran syntax:

```
CALL OPF_INTFLW_RESET(IFID, IERR)
```

where:

Integer IFID	Is the interface flow identifier (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 invalid interface flow identifier specified.
	IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.26 OPF_LNCEQN_ADJLOAD

Use this API to add a participating adjustable bus load variable to a specified linear constraint dependency equation in the working case. It may also be used to modify an existing record.

Batch command syntax:

```
BAT_OPF_LNCEQN_ADJLOAD   IQID  ITBL  COEFF
```

Python syntax:

```
ierr = opf_lnceqn_adjload(iqid, itbl, coeff)
```

Fortran syntax:

```
CALL OPF_LNCEQN_ADJLOAD(IQID, ITBL, COEFF, IERR)
```

where:

Integer IQID	Is the linear constraint equation identification number (input; no default allowed).
Integer ITBL	Is the adjustable bus load table identifier (input; no default allowed).
Real COEFF	Is the value of the coefficient associated with the equation variable (input; default of 1.0 if new record, current value otherwise).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 specified linear constraint equation does not exist. IERR = 2 participating equation variable table is full. IERR = 3 specified adjustable bus load table does not exist. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.27 OPF_LNCEQN_ADJVAR

Use this API to add a participating adjustable bus shunt variable to a specified linear constraint dependency equation in the working case. It may also be used to modify an existing record.

Batch command syntax:

```
BAT_OPF_LNCEQN_ADJVAR   IQID  EIBUS  SHNTID  COEFF
```

Python syntax:

```
ierr = opf_lnceqn_adjvar(iqid, eibus, shntid, coeff)
```

Fortran syntax:

```
CALL OPF_LNCEQN_ADJVAR(IQID, EIBUS, SHNTID, COEFF, IERR)
```

where:

Integer IQID	Is the linear constraint equation identification number (input; no default allowed).
Integer EIBUS	Is the bus number (input; no default allowed).
Character*2 SHNTID	Is the bus shunt identifier (input; no default allowed).
Real COEFF	Is the value of the coefficient associated with the equation variable (input; default of 1.0 if new record, current value otherwise).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 specified linear constraint equation does not exist.
	IERR = 2 participating equation variable table is full.
	IERR = 3 invalid bus number specified.
	IERR = 4 specified adjustable bus shunt does not exist.
	IERR = 5 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.28 OPF_LNCEQN_BRFLOW

Use this API to add a participating branch flow variable to a specified linear constraint dependency equation in the working case. It may also be used to modify an existing record.

Batch command syntax:

```
BAT_OPF_LNCEQN_BRFLOW   IQID  EIBUS  EJBUS  EKBUS  CKTID  FLWID
COEFF
```

Python syntax:

```
ierr = opf_lnceqn_brflow(iqid, eibus, ejbus, ekbus, cktid, flwid,
coeff)
```

Fortran syntax:

```
CALL OPF_LNCEQN_BRFLOW(IQID, EIBUS, EJBUS, EKBUS, CKTID, FLWID,
COEFF, IERR)
```

where:

Integer IQID	Is the linear constraint equation identification number (input; no default allowed).
Integer EIBUS	Is the from bus number (input; no default allowed).
Integer EJBUS	Is the to bus number (input; no default allowed).
Integer EKBUS	Is the third bus number for a three-winding transformer and zero for a two-winding transformer (input; no default allowed).
Character*2 CKTID	Is the circuit identifier (input; no default allowed).
Character*1 FLWID	Is the branch flow identifier (input; no default allowed).
Real COEFF	Is the value of the coefficient associated with the equation variable (input; default value of 1.0 if new record, current value otherwise).
Integer IERR	Is the error code (output).
	IERR = 0 — no error occurred.
	IERR = 1 — specified linear constraint equation does not exist.
	IERR = 2 — participating equation variable table is full.
	IERR = 3 — invalid bus number specified.
	IERR = 4 — specified branch/transformer does not exist.
	IERR = 5 — specified three-winding transformer does not exist.
	IERR = 6 — flow constraint does not exist at the specified branch.
	IERR = 7 — flow constraint does not exist at the specified three-winding transformer.
	IERR = 8 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.29 OPF_LNCEQN_INTFLOW

Use this API to add a participating interface flow constraint variable to a specified linear constraint dependency equation in the working case. It may also be used to modify an existing record.

Batch command syntax:

```
BAT_OPF_LNCEQN_INTFLOW   IQID  ITBL  COEFF
```

Python syntax:

```
ierr = opf_lnceqn_intflow(iqid, itbl, coeff)
```

Fortran syntax:

```
CALL OPF_LNCEQN_INTFLOW_API(IQID, ITBL, COEFF, IERR)
```

where:

Integer IQID	Is the linear constraint equation identification number (input; no default allowed).
Integer ITBL	Is the interface flow identifier (input; no default allowed).
Real COEFF	Is the value of the coefficient associated with the equation variable (input; default value of 1.0 if new record, current value otherwise).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 specified linear constraint equation does not exist. IERR = 2 participating equation variable table is full. IERR = 3 specified interface identifier does not exist. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.30 OPF_LNCEQN_MAIN

Use this API to add a linear constraint dependency record to the working case. It may also be used to modify an existing linear constraint dependency record.

This API must be invoked prior to adding any participating linear constraint equation variable records to the specified linear constraint dependency equation.

Batch command syntax:

```
BAT_OPF_LNCEQN_MAIN   IQID  LABL  REALAR(1)  REALAR(2)
```

Python syntax:

```
ierr = opf_lnceqn_main(iqid, labl, realar)
```

Fortran syntax:

```
CALL OPF_LNCEQN_MAIN(IQID, LABL, REALAR, IERR)
```

where:

Integer IQID	Is the linear constraint equation identification number (input; no default allowed).
Character*12 LABL	Is the linear constraint description made up of, at most, 12 characters (input; default of ' ').
Real REALAR(2)	Is an array of two real elements (input; default values indicated below). The values of each are as follows: REALAR(1) is the maximum constraint slack value (default value of 1.0 if new record, current value otherwise). REALAR(2) is the minimum constraint slack value (default value of 1.0 if new record, current value otherwise).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 linear constraint table is full. IERR = 2 prerequisite requirements for API are not met. IERR = -1 non-fatal error occurred. Possible warning messages are: • Maximum linear constraint slack is less than the minimum linear constraint slack.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.31 OPF_LNCEQN_PGEN

Use this API to add a participating active power dispatch variable to a specified linear constraint dependency equation in the working case. It may also be used to modify an existing record.

Batch command syntax:

```
BAT_OPF_LNCEQN_PGEN   IQID  ITBL  COEFF
```

Python syntax:

```
ierr = opf_lnceqn_pgen(iqid, itbl, coeff)
```

Fortran syntax:

```
CALL OPF_LNCEQN_PGEN(IQID, ITBL, COEFF, IERR)
```

where:

Integer IQID	Is the linear constraint equation identification number (input; no default allowed).
Integer ITBL	Is the active power dispatch table number (input; no default allowed).
Real COEFF	Is the value of the coefficient associated with the equation variable (input; default value of 1.0 if new record, current value otherwise).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 specified linear constraint equation does not exist. IERR = 2 participating equation variable table is full. IERR = 3 invalid active power dispatch table number specified. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.32 OPF_LNCEQN_QGEN

Use this API to add a participating reactive power dispatch variable of a specified linear constraint dependency equation in the working case. It may also be used to modify an existing record.

Batch command syntax:

```
BAT_OPF_LNCEQN_QGEN   IQID  EIBUS  GENID  COEFF
```

Python syntax:

```
ierr = opf_lnceqn_qgen(iqid, eibus, genid, coeff)
```

Fortran syntax:

```
CALL OPF_LNCEQN_QGEN(IQID, EIBUS, GENID, COEFF, IERR)
```

where:

Integer IQID	Is the linear constraint equation identification number (input; no default allowed).
Integer EIBUS	Is the bus number (input; no default allowed).
Character*2 GENID	Is the machine identifier (input; no default allowed).
Real COEFF	Is the value of the coefficient associated with the equation variable (input; default value of 1.0 if new record, current value otherwise).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 specified linear constraint equation does not exist.
	IERR = 2 participating equation variable table is full.
	IERR = 3 specified bus does not exist.
	IERR = 4 specified machine does not exist at bus.
	IERR = 5 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.33 OPF_LNCEQN_RESET

Use this API to remove all participating variables from a particular OPF linear constraint dependency equation in the working case.

Batch command syntax:

```
BAT_OPF_LNCEQN_RESET   IQID
```

Python syntax:

```
ierr = opf_lnceqn_reset(iqid)
```

Fortran syntax:

```
CALL OPF_LNCEQN_RESET_API(IQID, IERR)
```

where:

Integer IQID	Is the linear constraint equation identification number (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 Invalid linear constraint identifier specified. IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.34 OPF_LNCEQN_SWSHUNT

Use this API to add a participating switched shunt variable to a specified linear constraint dependency equation in the working case. It may also be used to modify an existing record.

Batch command syntax:

```
BAT_OPF_LNCEQN_SWSHUNT   IQID  EIBUS  COEFF
```

Python syntax:

```
ierr = opf_lnceqn_swshunt(iqid, eibus, coeff)
```

Fortran syntax:

```
CALL OPF_LNCEQN_SWSHUNT(IQID, EIBUS, COEFF, IERR)
```

where:

Integer IQID	Is the linear constraint equation identification number (input; no default allowed).
Integer EIBUS	Is the bus number (input; no default allowed).
Real COEFF	Is the value of the coefficient associated with the equation variable (input; default of 1.0 if new record, current value otherwise).
Integer IERR	Is the error code (output).
	IERR = 0 — no error occurred.
	IERR = 1 — specified linear constraint equation does not exist.
	IERR = 2 — participating equation variable table is full.
	IERR = 3 — invalid bus number specified.
	IERR = 4 — specified bus does not have any switched shunt devices.
	IERR = 5 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.35 OPF_LNCEQN_TRAN

Use this API to add a participating transformer control variable to a specified linear constraint dependency equation in the working case. It may also be used to modify an existing record.

Batch command syntax:

```
BAT_OPF_LNCEQN_TRAN   IQID  EIBUS  EJBUS  EKBUS  CKTID  COEFF
```

Python syntax:

```
ierr = opf_lnceqn_tran(iqid, eibus, ejbus, ekbus, cktid, coeff)
```

Fortran syntax:

```
CALL OPF_LNCEQN_TRAN(IQID, EIBUS, EJBUS, EKBUS, CKTID, COEFF, IERR)
```

where:

Integer IQID	Is the linear constraint equation identification number (input; no default allowed).
Integer EIBUS	Is the from bus number (input; no default allowed).
Integer EJBUS	Is the to bus number (input; no default allowed).
Integer EKBUS	Is the third bus number if a three-winding transformer is specified and zero for a two-winding transformer (input; no default allowed).
Character*2 CKTID	Is the circuit identifier (input; no default allowed).
Real COEFF	Is the value of the coefficient associated with the equation variable (input; default value of 1.0 if new record, current value otherwise).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 specified linear constraint equation does not exist.
	IERR = 2 participating equation variable table is full.
	IERR = 3 invalid bus number specified.
	IERR = 3 specified branch/transformer does not exist.
	IERR = 4 specified three-winding transformer does not exist.
	IERR = 5 specified branch is not a transformer.
	IERR = 6 specified branch is not a three-winding transformer.
	IERR = 8 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.36 OPF_LNCEQN_VANG

Use this API to add a participating bus voltage angle variable to a specified linear constraint dependency equation in the working case. It may also be used to modify an existing record.

Batch command syntax:

```
BAT_OPF_LNCEQN_VANG   IQID  EIBUS  COEFF
```

Python syntax:

```
ierr = opf_lnceqn_vang(iqid, eibus, coeff)
```

Fortran syntax:

```
CALL OPF_LNCEQN_VANG(IQID, EIBUS, COEFF, IERR)
```

where:

Integer IQID	Is the linear constraint equation identification number (input; no default allowed).
Integer EIBUS	Is the bus number (input; no default).
Real COEFF	Is the value of the coefficient associated with the equation variable (input; default value of 1.0 if new record, current value otherwise).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 specified linear constraint equation does not exist. IERR = 2 participating equation variable table is full. IERR = 3 invalid bus number specified. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.37 OPF_LNCEQN_VMAG

Use this API to add a participating bus voltage magnitude variable to a specified linear constraint dependency equation in the working case. It may also be used to modify an existing record.

Batch command syntax:

```
BAT_OPF_LNCEQN_VMAG   IQID  EIBUS  COEFF
```

Python syntax:

```
ierr = opf_lnceqn_vmag(iqid, eibus, coeff)
```

Fortran syntax:

```
CALL OPF_LNCEQN_VMAG(IQID, EIBUS, COEFF, IERR)
```

where:

Integer IQID	Is the linear constraint equation identification number (input; no default allowed).
Integer EIBUS	Is the bus number (input; no default allowed).
Real COEFF	Is the value of the coefficient associated with the equation variable (input; default value of 1.0 if new record, current value otherwise).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 specified linear constraint equation does not exist. IERR = 2 participating equation variable table is full. IERR = 3 invalid bus number specified. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.38 OPF_LOAD_INDV

Use this API to add an individual OPF bus load data record to the working case. It may also be used to modify an existing bus load data record.

Batch command syntax:

```
BAT_OPF_LOAD_INDV   EIBUS  LOADID  INTGAR
```

Python syntax:

```
ierr = opf_load_indv(eibus, loadid, intgar)
```

Fortran syntax:

```
CALL OPF_LOAD_INDV(EIBUS, LOADID, INTGAR, IERR)
```

where:

Integer EIBUS	Is the bus number (input; no default allowed).
Character*2 LOADID	Is the bus load identifier (input; no default allowed).
Integer INTGAR	Is the adjustable bus load table cross reference number (input; default value of 0 if new record, current value otherwise).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid bus number specified. IERR = 2 specified load does not exist. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.39 OPF_LOAD_SUBSYS

Use this API to add or modify all OPF bus load data records in a specified subsystem of the working case.

Batch command syntax:

```
BAT_OPF_LOAD_SUBSYS   SID  ALL  INTGAR(1)  INTGAR(2)
```

Python syntax:

```
ierr = opf_load_subsys(sid, all, intgar)
```

Fortran syntax:

```
CALL OPF_LOAD_SUBSYS(SID, ALL, INTGAR, IERR)
```

where:

Integer SID	Is a valid subsystem ID from 0 to 11. A subsystem must have been previously defined (input; no default allowed).
Integer ALL	Is the *all buses* or specified subsystem flag (input; no default allowed).
	ALL = 0 — process only specified buses.
	ALL = 1 — process all buses, use existing data values.
	ALL = 2 — process all buses, use default data values.
Integer INTGAR(2)	Is an array of two integer elements (input; default values indicated below). The value of each is as follows:
	INTGAR(1) — is the bus load table cross reference number (default value of 0, current value otherwise).
	INTGAR(2) — indicates which bus types should be processed (default treatment same as value of 0).
	INTGAR(2) = 0 — all buses in subsystem.
	INTGAR(2) = 1 — only Type 1 buses in subsystem.
	INTGAR(2) = 2 — only Type 2 and 3 buses in subsystem.
Integer IERR	Is the error code (output).
	IERR = 0 — no error occurred.
	IERR = 1 — invalid SID value or subsystem SID is not defined.
	IERR = 2 — invalid ALL value.
	IERR = 3 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.40 OPF_PERRSV_GEN

Use this API to add a participating generation reserve unit to a previously defined period reserve constraint in the working case.

Batch command syntax:

```
BAT_OPF_PERRSV_GEN   RSVID  EIBUS  GENID
```

Python syntax:

```
ierr = opf_perrsv_gen(rsvid, eibus, genid)
```

Fortran syntax:

```
CALL OPF_PERRSV_GEN(RSVID, EIBUS, GENID, IERR)
```

where:

Integer RSVID	Is the period reserve identifier (1 through 15) (input; no default allowed).
Integer EIBUS	Is the bus number (input; no default allowed).
Character*2 GENID	Is the machine identifier (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 — no error occurred.
	IERR = 1 — invalid period reserve identifier specified.
	IERR = 2 — invalid bus number specified.
	IERR = 3 — specified machine does not exist.
	IERR = 4 — specified unit does not have any generator reserve data defined.
	IERR = 5 — specified unit is already participating in the reserve constraint.
	IERR = 6 — generator reserve participating unit table full.
	IERR = 7 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.41 OPF_PERRSV_MAIN

Use this API to add a period reserve constraint record to the working case. It may also be used to modify an existing period reserve constraint record.

This API must be invoked prior to adding any participating generator reserve units (via `OPF_PERRSV_GEN`) to a specified period reserve constraint.

Batch command syntax:

```
BAT_OPF_PERRSV_MAIN   RSVID  STATUS  REALAR(1)  REALAR(2)
```

Python syntax:

```
ierr = opf_perrsv_main(rsvid, status, realar)
```

Fortran syntax:

```
CALL OPF_PERRSV_MAIN(RSVID, STATUS, REALAR, IERR)
```

where:

Integer RSVID	Is a valid period reserve identifier (1 through 15) (input; no default allowed).
Integer STATUS	Is the status of the period reserve constraint (default value of 1 if new record, current value otherwise). STATUS = 0 out of service. STATUS = 1 in service.
Real REALAR(2)	Is an array of two real elements (input; default values indicated below). The value for each is as follows. REALAR(1) is the reserve limit, in MW (default value of 0.0 if new record, current value otherwise). REALAR(2) is the period, in minutes (default value of 9999.0 if new record, current value otherwise).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid period reserve identifier specified. IERR = 2 prerequisite requirements for API are not met. IERR = -1 non-fatal error occurred. Possible warning messages are: • Invalid period reserve status specified.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.42 OPF_PERRSV_RESET

Use this API to remove all generation reserve units associated with a particular period reserve constraint in the working case.

Batch command syntax:

```
BAT_OPF_PERRSV_RESET   RSVID
```

Python syntax:

```
ierr = opf_perrsv_reset(rsvid)
```

Fortran syntax:

```
CALL OPF_PERRSV_RESET(RSVID, IERR)
```

where:

Integer RSVID	Is the period reserve identifier (1 through 15) (input; no default allowed).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid period reserve identifier specified.
	IERR = 2	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.43 PURGE_ALL_OPF_DATA

Use this API to purge and re-initialize all OPF data records in the working case. Unlike the NEWOPF_API call, any OPF data in the working case will be purged and re-initialized.

Batch command syntax:

```
BAT_PURGE_ALL_OPF_DATA
```

Python syntax:

```
ierr = purge_all_opf_data()
```

Fortran syntax:

```
CALL PURGE_ALL_OPF_DATA(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.44 PURGE_OPF_ADJBRX_INDV

Use this API to purge an individual adjustable branch reactance data record from the working case.

Batch command syntax:

```
BAT_PURGE_OPF_ADJBRX_INDV   EIBUS  EJBUS  CKTID
```

Python syntax:

```
ierr = purge_opf_adjbrx_indv (eibus, ejbus, cktid)
```

Fortran syntax:

```
CALL PURGE_OPF_ADJBRX_INDV_2(EIBUS, EJBUS, CKTID, IERR)
```

where:

Integer EIBUS	Is the from bus number (input; no default allowed).
Integer EJBUS	Is the to bus number (input; no default allowed).
Character*2 CKTID	Is the circuit identifier (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 invalid bus number specified.
	IERR = 2 specified branch does not exist.
	IERR = 3 prerequisite requirements for API are not met.
	IERR = 4 specified branch cannot be a transformer.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.45 PURGE_OPF_ADJBRX_SUBSYS

Use this API to purge all OPF branch reactance data records from a specified subsystem of the working case.

Batch command syntax:

```
BAT_PURGE_OPF_ADJBRX_SUBSYS   SID  ALL
```

Python syntax:

```
ierr = purge_opf_adjbrx_subsys(sid, all)
```

Fortran syntax:

```
CALL PURGE_OPF_ADJBRX_SUBSYS_2(SID, ALL, IERR)
```

where:

Integer SID	Is a valid subsystem ID from 0 to 11. A subsystem must have been previously defined (input; no default allowed).
Integer ALL	Is the *all buses* or specified subsystem flag (input; no default allowed). ALL = 0 process only specified buses. ALL = 1 process all buses.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.46 PURGE_OPF_ADJLOAD_TBL

Use this API to purge an individual adjustable bus load table from the working case.

Batch command syntax:

```
BAT_PURGE_OPF_ADJLOAD_TBL   TBL
```

Python syntax:

```
ierr = purge_opf_adjload_tbl(tbl)
```

Fortran syntax:

```
CALL PURGE_OPF_ADJLOD_TBL_2(TBL, IERR)
```

where:

Integer TBL	Is the adjustable bus load table number (input; no default allowed).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred
	IERR = 1	adjustable bus load table does not exist
	IERR = 2	load table still referenced by bus load(s) -- unable to delete
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.47 PURGE_OPF_ADJVAR_INDV

Use this API to purge an individual OPF adjustable bus shunt data record from the working case.

Batch command syntax:

```
BAT_PURGE_OPF_ADJVAR_INDV   EIBUS  SHNTID
```

Python syntax:

```
ierr = purge_opf_adjvar_indv(eibus, shntid)
```

Fortran syntax:

```
CALL PURGE_OPF_ADJVAR_INDV_2(EIBUS, SHNTID, IERR)
```

where:

Integer EIBUS	Is the bus number (input; no default allowed).
Character*2 SHNTID	Is the bus shunt identifier (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 — no error occurred.
	IERR = 1 — adjustable bus shunt identifier is more than two characters.
	IERR = 2 — invalid bus number specified.
	IERR = 3 — blank adjustable bus shunt identifier is invalid.
	IERR = 4 — unable to allocate sufficient memory.
	IERR = 5 — OPF adjustable bus shunt at bus not found.
	IERR = 6 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.48 PURGE_OPF_ADJVAR_SUBSYS

Use this API to purge all OPF adjustable bus shunt records from a specified subsystem of the working case.

Batch command syntax:

```
BAT_PURGE_OPF_ADJVAR_SUBSYS   SID  ALL  SHNTID
```

Python syntax:

```
ierr = purge_opf_adjvar_subsys(sid, all, shntid)
```

Fortran syntax:

```
CALL PURGE_OPF_ADJVAR_SUBSYS_2(SID, ALL, SHNTID, IERR)
```

where:

Integer SID	Is a valid subsystem ID from 0 to 11. A subsystem must have been previously defined (input; no default allowed).
Integer ALL	Is the *all buses* or specified subsystem flag (input; no default allowed). ALL = 0 process only specified buses. ALL = 1 process all buses.
Character*2 SHNTID	Is the adjustable bus shunt identifier (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 adjustable bus shunt identifier is more than two characters. IERR = 2 invalid SID value or subsystem SID is not defined. IERR = 3 invalid ALL value. IERR = 4 unable to allocate sufficient memory. IERR = 5 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.49 PURGE_OPF_APDSP_TBL

Use this API to purge an individual active power dispatch table record from the working case.

Batch command syntax:

```
BAT_PURGE_OPF_APDSP_TBL   TBL
```

Python syntax:

```
ierr = purge_opf_apdsp_tbl(tbl)
```

Fortran syntax:

```
CALL PURGE_OPF_APDSP_TBL_2(TBL, IERR)
```

where:

Integer TBL	Is the active power dispatch table number (input; no default allowed).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	active power dispatch table does not exist.
	IERR = 2	active power dispatch table still referenced -- unable to delete.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.50 PURGE_OPF_BRFLW_3WT

Use this API to purge an individual flow constraint record of a three-winding transformer from the working case.

Batch command syntax:

```
BAT_PURGE_OPF_BRFLW_3WT   EIBUS  EJBUS  EKBUS  CKTID  BFLWID
```

Python syntax:

```
ierr = purge_opf_brflw_3wt(eibus, ejbus, ekbus, cktid, bflwid)
```

Fortran syntax:

```
CALL PURGE_OPF_BRFLW_3WT_2(EIBUS, EJBUS, EKBUS, CKTID, BFLWID, IERR)
```

where:

Integer EIBUS	Is the from bus number (input; no default allowed). The value of EIBUS indicates the from bus of the winding from which the flow constraint is being purged.
Integer EJBUS	Is the to bus number (input; no default allowed).
Integer EKBUS	Is the third bus number (input; no default allowed).
Character*2 CKTID	Is the circuit identifier (input; no default allowed).
Character*1 BFLWID	Is the branch flow constraint identifier (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid bus number specified. IERR = 2 specified three-winding transformer does not exist. IERR = 3 specified winding of three-winding transformer not found. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.51 PURGE_OPF_BRFLW_BRN

Use this API to purge an individual branch or two-winding transformer flow constraint record from the working case.

Batch command syntax:

```
BAT_PURGE_OPF_BRFLW_BRN   EIBUS  EJBUS  CKTID  BFLWID
```

Python syntax:

```
ierr = purge_opf_brflw_brn(eibus, ejbus, cktid, bflwid)
```

Fortran syntax:

```
CALL PURGE_OPF_BRFLW_BRN_2(EIBUS, EJBUS, CKTID, BFLWID, IERR)
```

where:

Integer EIBUS	Is the from bus number (input; no default allowed).
Integer EJBUS	Is the to bus number (input; no default allowed).
Character*2 CKTID	Is the circuit identifier (input; no default allowed).
Character*1 BFLWID	Is the branch flow constraint identifier (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 invalid bus number specified.
	IERR = 2 specified branch does not exist.
	IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.52 PURGE_OPF_BRFLW_SUBSYS

Use this API to purge all branch flow data records in a specified subsystem in the working case.

Batch command syntax:

```
BAT_PURGE_OPF_BRFLW_SUBSYS   SID  ALL  BFLWID
```

Python syntax:

```
ierr = purge_opf_brflw_subsys(sid, all, bflwid)
```

Fortran syntax:

```
CALL PURGE_OPF_BRFLW_SUBSYS_2(SID, ALL, BFLWID, IERR)
```

where:

Integer SID	Is a valid subsystem ID from 0 to 11. A subsystem must have been previously defined (input; no default allowed).
Integer ALL	Is the *all buses* or specified subsystem flag (input; no default allowed). ALL = 0 process only specified buses. ALL = 1 process all buses.
Character*1 BFLWID	Is the branch flow identifier (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.53 PURGE_OPF_BUS_INDV

Use this API to purge (re-initialize) OPF bus voltage magnitude data for an individual record in the working case.

Batch command syntax:

```
BAT_PURGE_OPF_BUS_INDV   EIBUS
```

Python syntax:

```
ierr = purge_opf_bus_indv(eibus)
```

Fortran syntax:

```
CALL PURGE_OPF_BUS_INDV_2(EIBUS, IERR)
```

where:

Integer EIBUS	Is the bus number (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 invalid bus number specified.
	IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.54 PURGE_OPF_BUS_SUBSYS

Use this API to purge (re-initialize) OPF bus attribute data for all records in a specified subsystem of the working case.

Batch command syntax:

```
BAT_PURGE_OPF_BUS_SUBSYS   SID  ALL
```

Python syntax:

```
ierr = purge_opf_bus_subsys(sid, all)
```

Fortran syntax:

```
CALL PURGE_OPF_BUS_SUBSYS_2(SID, ALL, IERR)
```

where:

Integer SID	Is a valid subsystem ID from 0 to 11. A subsystem must have been previously defined (input; no default allowed).
Integer ALL	Is the *all buses* or specified subsystem flag (input; no default allowed). ALL = 0 process only specified buses. ALL = 1 process all buses, use existing data values. ALL = 2 process all buses, use default data values.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.55 PURGE_OPF_CSTTBL_LIN

Use this API to purge a linear cost curve record from the working case.

Batch command syntax:

```
BAT_PURGE_OPF_CSTTBL_LIN   TBL
```

Python syntax:

```
ierr = purge_opf_csttbl_lin(tbl)
```

Fortran syntax:

```
CALL PURGE_OPF_CSTTBL_LIN_2(TBL, IERR)
```

where:

Integer TBL	Is the linear cost curve table number (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred
	IERR = 1 specified linear cost table does not exist
	IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.56 PURGE_OPF_CSTTBL_POLY

Use this API to purge a polynomial and exponential cost curve record from the working case.

Batch command syntax:

```
BAT_PURGE_OPF_CSTTBL_POLY   TBL
```

Python syntax:

```
ierr = purge_opf_csttbl_poly(tbl)
```

Fortran syntax:

```
CALL PURGE_OPF_CSTTBL_POLY_2(TBL, IERR)
```

where:

Integer TBL	Is the polynomial and exponential cost curve table number (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred
	IERR = 1 specified polynomial cost table does not exist
	IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.57 PURGE_OPF_CSTTBL_QUAD

Use this API to purge a quadratic cost curve record from the working case.

Batch command syntax:

```
BAT_PURGE_OPF_CSTTBL_QUAD   TBL
```

Python syntax:

```
ierr = purge_opf_csttbl_quad(tbl)
```

Fortran syntax:

```
CALL PURGE_OPF_CSTTBL_QUAD_2(TBL, IERR)
```

where:

Integer TBL	Is the quadratic cost curve table number (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred
	IERR = 1 specified quadratic cost table does not exist
	IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.58 PURGE_OPF_GEN_RCAP_INDV

Use this API to purge an individual generation reactive capability data record from the working case.

Batch command syntax:

```
BAT_PURGE_OPF_GEN_RCAP_INDV   EIBUS  GENID
```

Python syntax:

```
ierr = purge_opf_gen_rcap_indv(eibus, genid)
```

Fortran syntax:

```
CALL PURGE_OPF_GEN_RCAP_INDV_2(EIBUS, GENID, IERR)
```

where:

Integer EIBUS	Is the bus number (input; no default allowed).	
Character*2 GENID	Is the machine identifier (input; no default allowed).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid bus number specified.
	IERR = 2	specified machine does not exist.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.59 PURGE_OPF_GEN_RCAP_SUBSYS

Use this API to purge all OPF generator reactive capability records in a specified subsystem from the working case.

Batch command syntax:

```
BAT_PURGE_OPF_GEN_RCAP_SUBSYS   SID  ALL
```

Python syntax:

```
ierr = purge_opf_gen_rcap_subsys(sid, all)
```

Fortran syntax:

```
CALL PURGE_OPF_GEN_RCAP_SUBSYS_2(SID, ALL, IERR)
```

where:

Integer SID	Is a valid subsystem ID from 0 to 11. A subsystem must have been previously defined (input; no default allowed).
Integer ALL	Is the *all buses* or specified subsystem flag (input; no default allowed). ALL = 0 process only specified buses. ALL = 1 process all buses.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.60 PURGE_OPF_GENDSP_INDV

Use this API to purge generator dispatch data for an individual record from the working case.

Batch command syntax:

```
BAT_PURGE_OPF_GENDSP_INDV   EIBUS  GENID
```

Python syntax:

```
ierr = purge_opf_gendsp_indv(eibus, genid)
```

Fortran syntax:

```
CALL PURGE_OPF_GENDSP_INDV_2(EIBUS, GENID, IERR)
```

where:

Integer EIBUS	Is the bus number (input; no default allowed).
Character*2 GENID	Is the machine identifier (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 invalid bus number specified.
	IERR = 2 specified machine does not exist.
	IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.61 PURGE_OPF_GENDSP_SUBSYS

Use this API to purge generator dispatch data for all records from a specified subsystem of the working case.

Batch command syntax:

```
BAT_PURGE_OPF_GENDSP_SUBSYS   SID  ALL
```

Python syntax:

```
ierr = purge_opf_gendsp_subsys(sid, all)
```

Fortran syntax:

```
CALL PURGE_OPF_GENDSP_SUBSYS_2(SID, ALL, IERR)
```

where:

Integer SID	Is a valid subsystem ID from 0 to 11. A subsystem must have been previously defined (input; no default allowed).
Integer ALL	Is the *all buses* or specified subsystem flag (input; no default allowed). ALL = 0 process only specified buses. ALL = 1 process all buses, use existing data values. ALL = 2 process all buses, use default data values.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.62 PURGE_OPF_GENRSV_INDV

Use this API to purge an individual generator reserve data record from the working case.

Batch command syntax:

```
BAT_PURGE_OPF_GENRSV_INDV   EIBUS  GENID
```

Python syntax:

```
ierr = purge_opf_genrsv_indv(eibus, genid)
```

Fortran syntax:

```
CALL PURGE_OPF_GENRSV_INDV_2(EIBUS, GENID, IERR)
```

where:

Integer EIBUS	Is the bus number (input; no default allowed).	
Character*2 GENID	Is the machine identifier (input; no default allowed).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid bus number specified.
	IERR = 2	specified machine does not exist.
	IERR = 3	unable to allocate sufficient memory.
	IERR = 4	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.63 PURGE_OPF_GENRSV_SUBSYS

Use this API to purge all OPF generator reserve records in a specified subsystem from the working case.

Batch command syntax:

```
BAT_PURGE_OPF_GENRSV_SUBSYS   SID  ALL
```

Python syntax:

```
ierr = purge_opf_genrsv_subsys(sid, all)
```

Fortran syntax:

```
CALL PURGE_OPF_GENRSV_SUBSYS_2(SID, ALL, IERR)
```

where:

Integer SID	Is a valid subsystem ID from 0 to 11. A subsystem must have been previously defined (input; no default allowed).
Integer ALL	Is the *all buses* or specified subsystem flag (input; no default allowed). ALL = 0 process only specified buses. ALL = 1 process all buses.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 unable to allocate sufficient memory. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.64 PURGE_OPF_INTFLW

Use this API to purge an interface flow constraint record, including all of its participating branches, from the working case.

Batch command syntax:

```
BAT_PURGE_OPF_INTFLW   IFID
```

Python syntax:

```
ierr = purge_opf_intflw(ifid)
```

Fortran syntax:

```
CALL PURGE_OPF_INTFLW_2(IFID, IERR)
```

where:

Integer IFID	Is the interface flow identifier (input; no default allowed).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	interface flow identifier not found.
	IERR = 2	unable to remove interface flow record due to participation in linear constraint record.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.65 PURGE_OPF_INTFLW_BRN

Use this API to remove a participating branch from a specified interface flow constraint in the working case.

Batch command syntax:

```
BAT_PURGE_OPF_INTFLW_BRN   IFID  EIBUS  EJBUS  CKTID
```

Python syntax:

```
ierr = purge_opf_intflw_brn(ifid, eibus, ejbus, cktid)
```

Fortran syntax:

```
CALL PURGE_OPF_INTFLW_BRN(IFID, EIBUS, EJBUS, CKTID, IERR)
```

where:

Integer IFID	Is the interface flow identifier (input; no default allowed).
Integer EIBUS	Is the from bus number (input; no default allowed).
Integer EJBUS	Is the to bus number (input; no default allowed).
Character*2 CKTID	Is the circuit identifier (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 invalid interface flow identifier specified.
	IERR = 2 invalid bus number specified.
	IERR = 3 specified branch does not exist.
	IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.66 PURGE_OPF_INTFLW_3WT

Use this API to remove a participating three-winding transformer winding from a specified interface flow constraint in the working case.

Batch command syntax:

```
BAT_PURGE_OPF_INTFLW_3WT   IFID  EIBUS  EJBUS  EKBUS  CKTID
```

Python syntax:

```
ierr = purge_opf_intflw_3wt(ifid, eibus, ejbus, ekbus, cktid)
```

Fortran syntax:

```
CALL PURGE_OPF_INTFLW_3WT(IFID, EIBUS, EJBUS, EKBUS, CKTID, IERR)
```

where:

Integer IFID	Is the interface flow identifier (input; no default allowed).
Integer EIBUS	Is the from bus number (input; no default allowed). The value of EIBUS indicates the from bus of the winding to be removed from the flow constraint.
Integer EJBUS	Is the to bus number (input; no default allowed).
Integer EKBUS	Is the third bus number (input; no default allowed).
Character*2 CKTID	Is the circuit identifier (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid interface flow identifier specified. IERR = 2 invalid bus number specified. IERR = 3 specified three-winding transformer does not exist. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.67 PURGE_OPF_LNCEQN

Use this API to purge a linear constraint dependency record, including all of its participating variables, from the working case.

Batch command syntax:

```
BAT_PURGE_OPF_LNCEQN   IQID
```

Python syntax:

```
ierr = purge_opf_lnceqn(iqid)
```

Fortran syntax:

```
CALL PURGE_OPF_LNCEQN_2(IQID, IERR)
```

where:

Integer IQID	Is the linear constraint equation identification number (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 linear constraint equation number not found. IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.68 PURGE_OPF_LNCEQN_ADJLOAD

Use this API to remove a participating adjustable bus load variable from a specified linear constraint dependency equation in the working case.

Batch command syntax:

```
BAT_PURGE_OPF_LNCEQN_ADJLOAD   IQID  ITBL
```

Python syntax:

```
ierr = purge_opf_lnceqn_adjload(iqid, itbl)
```

Fortran syntax:

```
CALL PURGE_OPF_LNCEQN_ADJLOAD(IQID, ITBL, IERR)
```

where:

Integer IQID	Is the linear constraint equation identification number (input; no default allowed).
Integer ITBL	Is the adjustable bus load table identifier (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred IERR = 1 specified linear constraint equation does not exist IERR = 2 specified adjustable bus load table does not exist. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.69 PURGE_OPF_LNCEQN_ADJVAR

Use this API to remove a participating adjustable bus shunt variable from a specified linear constraint dependency equation in the working case.

Batch command syntax:

```
BAT_PURGE_OPF_LNCEQN_ADJVAR   IQID  EIBUS  SHNTID
```

Python syntax:

```
ierr = purge_opf_lnceqn_adjvar(iqid, eibus, shntid)
```

Fortran syntax:

```
CALL PURGE_OPF_LNCEQN_ADJVAR(IQID, EIBUS, SHNTID, COEFF, IERR)
```

where:

Integer IQID	Is the linear constraint equation identification number (input; no default allowed).
Integer EIBUS	Is the bus number (input; no default allowed).
Character*2 SHNTID	Is the bus shunt identifier (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 specified linear constraint equation does not exist.
	IERR = 2 invalid bus number specified.
	IERR = 3 specified adjustable bus shunt does not exist.
	IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.70 PURGE_OPF_LNCEQN_BRFLOW

Use this API to remove a participating branch flow variable from a specified linear constraint dependency equation in the working case.

Batch command syntax:

```
BAT_PURGE_OPF_LNCEQN_BRFLOW   IQID  EIBUS  EJBUS  EKBUS  CKTID  FLWID
```

Python syntax:

```
ierr = purge_opf_lnceqn_brflow(iqid, eibus, ejbus, ekbus, cktid,
flwid)
```

Fortran syntax:

```
CALL PURGE_OPF_LNCEQN_BRFLOW(IQID, EIBUS, EJBUS, EKBUS, CKTID, FLWID,
IERR)
```

where:

Integer IQID	Is the linear constraint equation identification number (input; no default allowed).
Integer EIBUS	Is the from bus number (input; no default allowed).
Integer EJBUS	Is the to bus number (input; no default allowed).
Integer EKBUS	Is the third bus number for a three-winding transformer and zero for a two-winding transformer (input; no default allowed).
Character*2 CKTID	Is the circuit identifier (input; no default allowed).
Character*1 FLWID	Is the branch flow identifier (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 — no error occurred.
	IERR = 1 — specified linear constraint equation does not exist.
	IERR = 2 — invalid bus number specified.
	IERR = 3 — specified branch/transformer does not exist.
	IERR = 4 — specified three-winding transformer does not exist.
	IERR = 5 — flow constraint does not exist at the specified branch.
	IERR = 6 — flow constraint does not exist at the specified three-winding transformer.
	IERR = 7 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.71 PURGE_OPF_LNCEQN_INTFLOW

Use this API to remove a participating interface flow constraint variable from a specified linear constraint dependency equation in the working case.

Batch command syntax:

```
BAT_PURGE_OPF_LNCEQN_INTFLOW   IQID  ITBL
```

Python syntax:

```
ierr = purge_opf_lnceqn_intflow(iqid, itbl)
```

Fortran syntax:

```
CALL PURGE_OPF_LNCEQN_INTFLOW(IQID, ITBL, IERR)
```

where:

Integer IQID	Is the linear constraint equation identification number (input; no default allowed).
Integer ITBL	Is the interface flow identifier (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 specified linear constraint equation does not exist. IERR = 2 specified interface identifier does not exist. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.72 PURGE_OPF_LNCEQN_PGEN

Use this API to remove a participating active power dispatch variable from a specified linear constraint dependency equation in the working case.

Batch command syntax:

```
BAT_PURGE_OPF_LNCEQN_PGEN   IQID  ITBL
```

Python syntax:

```
ierr = purge_opf_lnceqn_pgen(iqid, itbl)
```

Fortran syntax:

```
CALL PURGE_OPF_LNCEQN_PGEN(IQID, ITBL, IERR)
```

where:

Integer IQID	Is the linear constraint equation identification number (input; no default allowed).
Integer ITBL	Is the active power dispatch table number (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 specified linear constraint equation does not exist. IERR = 2 invalid active power dispatch table number specified. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.73 PURGE_OPF_LNCEQN_QGEN

Use this API to remove a participating reactive power dispatch variable from a specified linear constraint dependency equation in the working case.

Batch command syntax:

```
BAT_PURGE_OPF_LNCEQN_QGEN   IQID  EIBUS  GENID
```

Python syntax:

```
ierr = purge_opf_lnceqn_qgen(iqid, eibus, genid)
```

Fortran syntax:

```
CALL PURGE_OPF_LNCEQN_QGEN(IQID, EIBUS, GENID, IERR)
```

where:

Integer IQID	Is the linear constraint equation identification number (input; no default allowed).
Integer EIBUS	Is the bus number (input; no default allowed).
Character*2 GENID	Is the machine identifier (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 specified linear constraint equation does not exist.
	IERR = 2 specified bus does not exist.
	IERR = 3 specified machine does not exist at bus.
	IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.74 PURGE_OPF_LNCEQN_SWSHUNT

Use this API to remove a participating switched shunt variable from a specified linear constraint dependency equation in the working case.

Batch command syntax:

```
BAT_PURGE_OPF_LNCEQN_SWSHUNT   IQID  EIBUS
```

Python syntax:

```
ierr = purge_opf_lnceqn_swshunt(iqid, eibus)
```

Fortran syntax:

```
CALL PURGE_OPF_LNCEQN_SWSHUNT(IQID, EIBUS, IERR)
```

where:

Integer IQID	Is the linear constraint equation identification number (input; no default allowed).
Integer EIBUS	Is the bus number (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 — no error occurred.
	IERR = 1 — specified linear constraint equation does not exist.
	IERR = 2 — invalid bus number specified.
	IERR = 3 — specified bus does not have any switched shunt devices.
	IERR = 4 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.75 PURGE_OPF_LNCEQN_TRAN

Use this API to remove a participating transformer control variable from a specified linear constraint dependency equation in the working case.

Batch command syntax:

```
BAT_PURGE_OPF_LNCEQN_TRAN   IQID  EIBUS  EJBUS  EKBUS  CKTID
```

Python syntax:

```
ierr = purge_opf_lnceqn_tran(iqid, eibus, ejbus, ekbus, cktid)
```

Fortran syntax:

```
CALL PURGE_OPF_LNCEQN_TRAN(IQID, EIBUS, EJBUS, EKBUS, CKTID, IERR)
```

where:

Integer IQID	Is the linear constraint equation identification number (input; no default allowed).
Integer EIBUS	Is the from bus number (input; no default allowed).
Integer EJBUS	Is the to bus number (input; no default allowed).
Integer EKBUS	Is the third bus number if a three-winding transformer is specified and zero for a two-winding transformer (input; no default allowed).
Character*2 CKTID	Is the circuit identifier (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 — no error occurred.
	IERR = 1 — specified linear constraint equation does not exist.
	IERR = 2 — invalid bus number specified.
	IERR = 3 — specified branch/transformer does not exist.
	IERR = 4 — specified three-winding transformer does not exist.
	IERR = 5 — specified branch is not a transformer.
	IERR = 6 — specified branch is not a three-winding transformer.
	IERR = 7 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.76 PURGE_OPF_LNCEQN_VANG

Use this API to remove a participating bus voltage angle variable from a specified linear constraint dependency equation in the working case.

Batch command syntax:

```
BAT_PURGE_OPF_LNCEQN_VANG   IQID  EIBUS
```

Python syntax:

```
ierr = purge_opf_lnceqn_vang(iqid, eibus)
```

Fortran syntax:

```
CALL PURGE_OPF_LNCEQN_VANG(IQID, EIBUS, IERR)
```

where:

Integer IQID	Is the linear constraint equation identification number (input; no default allowed).
Integer EIBUS	Is the bus number (input; no default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 specified linear constraint equation does not exist. IERR = 2 invalid bus number specified. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.77 PURGE_OPF_LNCEQN_VMAG

Use this API to remove a participating bus voltage magnitude variable from a specified linear constraint dependency equation in the working case.

Batch command syntax:

```
BAT_PURGE_OPF_LNCEQN_VMAG   IQID  EIBUS
```

Python syntax:

```
ierr = purge_opf_lnceqn_vmag(iqid, eibus)
```

Fortran syntax:

```
CALL PURGE_OPF_LNCEQN_VMAG(IQID, EIBUS, IERR)
```

where:

Integer IQID	Is the linear constraint equation identification number (input; no default allowed).
Integer EIBUS	Is the bus number (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 specified linear constraint equation does not exist. IERR = 2 invalid bus number specified. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.78 PURGE_OPF_LOAD_INDV

Use this API to purge (re-initialize) an individual OPF bus load data record from the working case.

Batch command syntax:

```
BAT_PURGE_OPF_LOAD_INDV   EIBUS  LOADID
```

Python syntax:

```
ierr = purge_opf_load_indv(eibus, loadid)
```

Fortran syntax:

```
CALL PURGE_OPF_LOAD_INDV_2(EIBUS, LOADID, IERR)
```

where:

Integer EIBUS	Is the bus number (input; no default allowed).	
Character*2 LOADID	Is the bus load identifier (input; no default allowed).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid bus number specified.
	IERR = 2	specified load does not exist.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.79 PURGE_OPF_LOAD_SUBSYS

Use this API to purge (re-initialize) all OPF bus load data records in a specified subsystem of the working case.

Batch command syntax:

```
BAT_PURGE_OPF_LOAD_SUBSYS   SID  ALL
```

Python syntax:

```
ierr = purge_opf_load_subsys(sid, all)
```

Fortran syntax:

```
CALL PURGE_OPF_LOAD_SUBSYS_2(SID, ALL, IERR)
```

where:

Integer SID	Is a valid subsystem ID from 0 to 11. A subsystem must have been previously defined (input; no default allowed).
Integer ALL	Is the *all buses* or specified subsystem flag (input; no default allowed). ALL = 0 process only specified buses. ALL = 1 process all buses.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.80 PURGE_OPF_PERRSV

Use this API to purge a period reserve constraint record and its association with participating reserve units from the working case.

Batch command syntax:

```
BAT_PURGE_OPF_PERRSV   RSVID
```

Python syntax:

```
ierr = purge_opf_perrsv(rsvid)
```

Fortran syntax:

```
CALL PURGE_OPF_PERRSV_2(RSVID, IERR)
```

where:

Integer RSVID	Is a valid period reserve identifier (1 through 15) (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid period reserve identifier specified. IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

3.2.81 PURGE_OPF_PERRSV_GEN

Use this API to remove a participating period reserve unit from a specified period reserve constraint.

Batch command syntax:

```
BAT_PURGE_OPF_PERRSV_GEN   RSVID  EIBUS  GENID
```

Python syntax:

```
ierr = purge_opf_perrsv_gen(rsvid, eibus, genid)
```

Fortran syntax:

```
CALL PURGE_OPF_PERRSV_GEN(RSVID, EIBUS, GENID, IERR)
```

where:

Integer RSVID	Is a valid period reserve identifier (1 through 15) (input; no default allowed).
Integer EIBUS	Is the bus number (input; no default allowed).
Character*2 GENID	Is the machine identifier (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 — no error occurred.
	IERR = 1 — period reserve constraint does not exist.
	IERR = 2 — bus not found.
	IERR = 3 — machine not found.
	IERR = 4 — machine does not have generator reserve data defined.
	IERR = 5 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

This page intentionally left blank.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

SIEMENS

Chapter 4

Dynamics Simulation

This chapter describes the API routines used in PSS®E Dynamics Simulation. Section 4.1 Dynamic Simulation Function APIs contains API routines that handle the majority of dynamics simulation functions. Sections 4.3 through 4.8 contain API routines specific to the handling of plant related models, load related models, line relay models, dc line models, and FACTS device models, respectively. Section 4.10 Auxiliary Signal Model APIs contains API routines specific to the handling of models that inject supplementary signals into dc lines and FACTS devices (auxiliary signal models). Section 4.13 Disturbance APIs contains API routines for applying and removing disturbances during dynamics simulations.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1 Dynamic Simulation Function APIs

4.1.1 ADDMODELFOLDER

This API is obsolete and performs no function.

Batch command syntax:

```
BAT_ADDMODELFOLDER  PATHNAME
```

Python syntax:

```
ierr = addmodelfolder(pathname)
```

Fortran syntax:

```
CALL ADDMODELFOLDER(PATHNAME, IERR)
```

where:

Character PATHNAME*260	Is the name of the path (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 2 no room to add folder.
	IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.2 ADDMODELLIBRARY

Use this API to add a library to the list to be searched for library models.

Batch command syntax:

```
BAT_ADDMODELLIBRARY  LIBRARYNAME
```

Python syntax:

```
ierr = addmodellibrary(libraryname)
```

Fortran syntax:

```
CALL ADDMODELLIBRARY(LIBRARYNAME, IERR)
```

where:

Character LIBRARYNAME*260	Is the name of the library of model programs (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 — no error occurred.
	IERR = 1 — Library not found.
	IERR = 2 — Library load failed.
	IERR = 3 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.3 ASTR

Use this API to calculate and replicates the state variable system matrices in the form required by the Linear Dynamic Analysis Program, LSYSAN (activity ASTR).

Batch command syntax:

```
BAT_ASTR  OPTION  MAXDERIV  PERTURB  NSTATE
MINSTATE(1)...MINSTATE(NSTATE)   MAXSTATE(1)...MAXSTATE(NSTATE)
NOUTPUT  MINCHAN(1)...MINCHAN(NOUTPUT)  MAXCHAN(1)...MAXCHAN(NOUTPUT)
NINPUT  INTYPE(1)...INTYPE(NINPUT) INPERT(1)...INPERT(NINPUT)
INBUS(1)...INBUS(NINPUT)  INMCID(1)...INMCID(NINPUT)
INIDENT(1)...INIDENT(NINPUT)  OUTFILE
```

Python syntax:

```
ierr = astr(option, maxderiv, perturb, nstate, minstate, maxstate,
noutput, minchan, maxchan, ninput, intype, inpert, inbus, inmcid,
inident, outfile)
```

Fortran syntax:

```
CALL ASTRAPI(OPTION, MAXDERIV, PERTURB, NSTATE, MINSTATE, MAXSTATE,
NOUTPUT, MINCHAN, MAXCHAN, NINPUT, INTYPE, INPERT, INBUS, INMCID,
INIDENT, OUTFILE, IERR)
```

where:

Integer OPTION	Is the network solution convergence monitor option (input; 0 by default): OPTION = 1 automatically print the convergence monitor. OPTION = 0 print the convergence monitor only if it is enabled via the CM interrupt control code.
Real MAXDERIV	Is the largest derivative threshold for considering that the initialization reflects a steady-state condition (input; 0.01 by default).
Real PERTURB	Is the amount by which the specified state variables are to be perturbed (input; 0.01 by default).
Integer NSTATE	Is the number of state variable ranges defining those state variables that are to be perturbed; 0 to include all STATEs (input; 0 by default).
Integer MINSTATE(NSTATE)	Is a list of starting STATE indices defining the ranges of state variables to be perturbed; ignored if NSTATE = 0 (input; no default allowed).
Integer MAXSTATE(NSTATE)	Is a list of ending STATE indices defining the ranges of state variables to be perturbed; ignored if NSTATE = 0 (input; no default allowed).
Integer NOUTPUT	Is the number of output channel ranges defining those output channels that are to be included in the system output vector; 0 to use the first 'NCHAN' or 50, whichever is less (input; 0 by default).
Integer MINCHAN(NOUTPUT)	Is a list of starting output channel indices defining the ranges of output channels to be included in the system output vector; ignored if NOUTPUT = 0 (input; no default allowed).
Integer MAXCHAN(NOUTPUT)	Is a list of ending output channel indices defining the ranges of output channels to be included in the system output vector; ignored if NOUTPUT = 0 (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer NINPUT	Is the number of system input variables that are to be perturbed (input; 0 by default).
Integer INTYPE(NINPUT)	Is a list of input variable type codes; ignored if NINPUT = 0 (input; no default allowed). 1=EFD, generator main field voltage (pu). 2=PMECH, turbine mechanical power (pu on MBASE). 3=VOTHSG, stabilizer output signal (pu). 4=VREF, voltage regulator voltage setpoint (pu). 5=VAR
Real INPERT(NINPUT)	Is a list of perturbation factors to be applied to those system input variables that are to be perturbed; ignored if NINPUT = 0 (input; PERTURB by default).
Integer INBUS(NINPUT)	Is a list of buses (when the corresponding INTYPE is 1 through 4) and/or VAR indices (when the corresponding INTYPE is 5) used to define those system input variables that are to be perturbed; ignored if NINPUT = 0 (input; no default allowed).
Character INMCID(NINPUT)*2	Is a list of machine identifiers used to define those system input variables that are to be perturbed (when the corresponding INTYPE is 1 through 4; ignored when the corresponding INTYPE is 5); ignored if NINPUT = 0 (input; ’1’ by default).
Character INIDENT(NINPUT)*32	Is a list of text strings used to identify those system input variables that are to be perturbed; ignored if NINPUT = 0. If a blank identifier is specified, the ASTR API provides an appropriate identifier (input; blank by default).
Character OUTFILE*260	Is the name of the Matrix Output File into which the results of this API’s calculation are to be placed; this file may then be used as input to the LSYSAN program (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 generators are not converted. IERR = 2 invalid OPTION value. IERR = 3 invalid MAXDERIV value. IERR = 4 invalid PERTURB value. IERR = 5 derivative threshold exceeded--not in steady state. IERR = 6 invalid code of input to be perturbed. IERR = 7 invalid perturbation amount. IERR = 8 machine is out-of-service. IERR = 9 invalid VAR index. IERR = 10 attempting to select more STATEs than are in the system model. IERR = 11 too many channels selected. IERR = 12 Prior initialization modified the loads--pick up original converted case. IERR = 13 error opening OUTFILE. IERR = 14 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.4 AULIST

Use this API to list auxiliary signal models that inject signals into dc lines or FACTS devices connected to subsystem buses, along with their storage locations in the dynamics data arrays.

Batch command syntax:

```
BAT_AULIST  SID  ALL  STATUS
```

Python syntax:

```
ierr = aulist(sid, all, status)
```

Fortran syntax:

```
CALL AULIST(SID, ALL, STATUS, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Integer STATUS	Is the active/bypassed flag (input; 0 by default). STATUS = 0 active and bypassed models. STATUS = 1 active models only. STATUS = 2 bypassed models only.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 invalid STATUS value. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.5 CHANGE_CHANNEL_OUT_FILE

Use this API to modify the dynamic simulation channel output filename in dynamics working memory (see *PSS®E Program Operation Manual*, activity ALTR).

Batch command syntax:

```
BAT_CHANGE_CHANNEL_OUT_FILE  OUTFILE
```

Python syntax:

```
ierr = change_channel_out_file(outfile)
```

Fortran syntax:

```
CALL CHANGE_CHANNEL_OUT_FILE(OUTFILE, IERR)
```

where:

Character OUTFILE*260	Is the name of the Channel Output File (blank to bypass recording of the output channel values in a Channel Output File) (input; blank by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no errors occurred.
	IERR > 0 — error occurred.
	IERR = 1 — error opening OUTFILE.
	IERR = 2 — prerequisite requirements for API are not met.
	IERR = -1 — warnings:
	• None.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.6 CHANGE_CON

Use this API to change the value of an element of the CON array. The element is specified relative to the beginning of the CON array.

Batch command syntax:

```
BAT_CHANGE_CON  NUMBER  NEWVAL
```

Python syntax:

```
ierr = change_con(number, newval)
```

Fortran syntax:

```
CALL CHANGE_CON(NUMBER, NEWVAL, IERR)
```

where:

Integer NUMBER	Is the CON number or array index (must be between 1 and the number of CONs allowed at the present size level of PSS®E) (input; no default allowed).
Real NEWVAL	Is the new value to be assigned to CON(NUMBER) (input; unchanged by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid CON index. IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.7 CHANGE_ICON

Use this API to change the value of an element of the ICON array (an integer ICON). The element is specified relative to the beginning of the ICON array.

Batch command syntax:

```
BAT_CHANGE_ICON  NUMBER  NEWVAL
```

Python syntax:

```
ierr = change_icon(number, newval)
```

Fortran syntax:

```
CALL CHANGE_ICON(NUMBER, NEWVAL, IERR)
```

where:

Integer NUMBER	Is the ICON number or array index (must be between 1 and the number of ICONs allowed at the present size level of PSS®E) (input; no default allowed).
Integer NEWVAL	Is the new value to be assigned to ICON(NUMBER) (input; unchanged by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid ICON index. IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.8 CHANGE_CHRICN

Use this API to change the value of an element of the CHRICN array (a character ICON). The element is specified relative to the beginning of the CHRICN array.

Batch command syntax:

```
BAT_CHANGE_CHRICN  NUMBER  'NEWVAL'
```

Python syntax:

```
ierr = change_chricn(number, newval)
```

Fortran syntax:

```
CALL CHANGE_CHRICN(NUMBER, NEWVAL, IERR)
```

where:

Integer NUMBER	Is the CHRICN number or array index (must be between 1 and the number of ICONs allowed at the present size level of PSS®E) (input; no default allowed).
Character NEWVAL*2	Is the new value to be assigned to CHRICN(NUMBER) (input; blank by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid CHRICN index. IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.9 CHANGE_GREF

Use this API to change the value of the element of the GREF (governor reference) array associated with a specified machine.

Batch command syntax:

```
BAT_CHANGE_GREF  IBUS  'ID'  NEWVAL
```

Python syntax:

```
ierr = change_gref(ibus, id, newval)
```

Fortran syntax:

```
CALL CHANGE_GREF(IBUS, ID, NEWVAL, IERR)
```

where:

Integer IBUS	Is the plant bus number (input; no default allowed).
Character ID*2	Is the machine identifier (input; '1' by default).
Real NEWVAL	Is the new value to be assigned to the GREF array entry associated with machine ID at bus IBUS (input; unchanged by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error occurred.
	IERR > 0 — error occurred.
	IERR = 1 — bus not found.
	IERR = 2 — machine not found.
	IERR = 3 — prerequisite requirements for API are not met.
	IERR = -1 — warnings – one or more of: • No turbine-governor model at this machine • Turbine-governor model at this machine is bypassed.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.10 CHANGE_STATE

Use this API to change the value of an element of the STATE array, as well as the values of the corresponding time derivative (DSTATE) and the associated memory cell used in the integration algorithm (STORE). The element is specified relative to the beginning of the STATE array.

> **Note**: Although this API may be used to modify the value of a state variable (along with its time derivative and integration memory cell) used in state-space and response test simulations, this ability is rarely required in normal dynamic simulation work. Unless some very special purpose is being served, this API should not be used.
>
> **Note**: This API should **never** be used during extended term simulations.

Batch command syntax:

```
BAT_CHANGE_STATE  NUMBER  NEWVAL(1)...NEWVAL(3)
```

Python syntax:

```
ierr = change_state(number, newval)
```

Fortran syntax:

```
CALL CHANGE_STATE(NUMBER, NEWVAL, IERR)
```

where:

Integer NUMBER	Is the STATE number or array index (must be between 1 and the number of STATEs allowed at the present size level of PSS®E) (input; no default allowed).	
Real NEWVAL(3)	Is an array of three elements (input):	
	NEWVAL(1)	the new value to be assigned to STATE(NUMBER) (unchanged by default).
	NEWVAL(2)	the new value to be assigned to DSTATE(NUMBER) (unchanged by default).
	NEWVAL(3)	the new value to be assigned to STORE(NUMBER) (unchanged by default).
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid STATE index.
	IERR = 2	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.11 CHANGE_SWSREF

Use this API to change the value of the element of the SWREF array associated with a specified switched shunt.

Batch command syntax:

```
BAT_CHANGE_SWSREF  IBUS  NEWVAL
```

Python syntax:

```
ierr = change_swsref(ibus, newval)
```

Fortran syntax:

```
CALL CHANGE_SWSREF(IBUS, NEWVAL, IERR)
```

where:

Integer IBUS	Is the switched shunt bus number (input; no default allowed).
Real NEWVAL	Is the new value to be assigned to the SWSREF array entry associated with switched shunt at bus IBUS (input; unchanged by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error occurred.
	IERR > 0 — error occurred.
	IERR = 1 — bus not found.
	IERR = 2 — switched shunt not found.
	IERR = 3 — prerequisite requirements for API are not met.
	IERR = -1 — warnings – one or more of: • No table model for switched shunt at this bus. • Switched Shunt Dynamic Model at this switched shunt is bypassed.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.12 CHANGE_VAR

Use this API to change the value of an element of the VAR array. The element is specified relative to the beginning of the VAR array.

Batch command syntax:

```
BAT_CHANGE_VAR  NUMBER  NEWVAL
```

Python syntax:

```
ierr = change_var(number, newval)
```

Fortran syntax:

```
CALL CHANGE_VAR(NUMBER, NEWVAL, IERR)
```

where:

Integer NUMBER	Is the VAR number or array index (must be between 1 and the number of VARs allowed at the present size level of PSS®E) (input; no default allowed).
Real NEWVAL	Is the new value to be assigned to VAR(NUMBER) (input; unchanged by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid VAR index. IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.13 CHANGE_VREF

Use this API to change the value of the element of the VREF array associated with a specified machine.

Batch command syntax:

```
BAT_CHANGE_VREF  IBUS  'ID'  NEWVAL
```

Python syntax:

```
ierr = change_vref(ibus, id, newval)
```

Fortran syntax:

```
CALL CHANGE_VREF(IBUS, ID, NEWVAL, IERR)
```

where:

Integer IBUS	Is the plant bus number (input; no default allowed).
Character ID*2	Is the machine identifier (input; '1' by default).
Real NEWVAL	Is the new value to be assigned to the VREF array entry associated with machine ID at bus IBUS (input; unchanged by default).
Integer IERR	Is the error code (output). IERR = 0 — no error occurred. IERR > 0 — error occurred. IERR = 1 — bus not found. IERR = 2 — machine not found. IERR = 3 — prerequisite requirements for API are not met. IERR = -1 — warnings – one or more of: • No excitation system model at this machine • Excitation system model at this machine is bypassed.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.14 DCLIST

Use this API to list dc line models connected to subsystem buses along with their storage locations in the dynamics data arrays.

Batch command syntax:

```
BAT_DCLIST  SID  ALL  STATUS(1)  STATUS(2)
```

Python syntax:

```
ierr = dclist(sid, all, status)
```

Fortran syntax:

```
CALL DCLIST(SID, ALL, STATUS, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Integer STATUS(2)	Is a two-element array (input). STATUS(1) the active/bypassed flag (0 by default). STATUS(1) = 0 list active and bypassed models. STATUS(1) = 1 list active models only. STATUS(1) = 2 list bypassed models only. STATUS(2) the dc line model type (1 by default). STATUS(2) = 1 two-terminal dc line models. STATUS(2) = 2 multi-terminal dc line models. STATUS(2) = 3 VSC dc line models.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 invalid STATUS value. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.15 DELMODELFOLDER

This API is obsolete and performs no function.

Batch command syntax:

```
BAT_DELMODELFOLDER  FOLDERINDEX
```

Python syntax:

```
ierr = delmodelfolder(folderindex)
```

Fortran syntax:

```
CALL DELMODELFOLDER(FOLDERINDEX, IERR)
```

where:

Integer FOLDERINDEX	Is the index (ordinal position, starting with 1) of the folder to be removed from the list (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 folder index out of range. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.16 DLST

Use this API to tabulate specified portions of one or more of the PSS®E dynamics data storage arrays (activity DLST).

Batch command syntax:

```
BAT_DLST  NTYPE  ATYPE(1)...ATYPE(NTYPE)   FIRST(1)...FIRST(NTYPE)
LAST(1)...LAST(NTYPE)
```

Python syntax:

```
ierr = dlst(ntype, atype, first, last)
```

Fortran syntax:

```
CALL DLSTAPI(NTYPE, ATYPE, FIRST, LAST, IERR)
```

where:

Integer NTYPE	Is the number of array portion specifications to be supplied; must be between 1 and 6, inclusive (input; no default allowed).
Integer ATYPE(NTYPE)	Is a list of PSS®E array type codes. No more than one occurrence of each of the valid ATYPE values is allowed in each use of this API (input; no default allowed). ATYPE = 1 for CONs. ATYPE = 2 for STATEs. ATYPE = 3 for VARs. ATYPE = 4 for ICONs and CHRICNs. ATYPE = 5 for output channels. ATYPE = 6 for CRT plot channels.
Integer FIRST(NTYPE)	Is a list of starting indices in the data array(s) specified by the corresponding ATYPE entry (input; 1 by default).
Integer LAST(NTYPE)	Is a list of ending indices in the data array(s) specified by the corresponding ATYPE entry (input; *next available - 1* by default for ATYPE = 1 through 5; 6 by default for ATYPE = 6).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid NTYPE value. IERR = 2 invalid ATYPE value (entry skipped). IERR = 3 ATYPE value previously specified (entry skipped). IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.17 DOCU

Use this API to tabulate the data associated with equipment models referenced in the user's simulation setup (activity DOCU).

Batch command syntax:

```
BAT_DOCU  SID  ALL  STATUS(1)...STATUS(3)
```

Python syntax:

```
ierr = docu(sid, all, status)
```

Fortran syntax:

```
CALL DOCUNW(SID, ALL, STATUS, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer STATUS(3) Is a three-element array (input).

STATUS(1)	processing mode (0 by default).
STATUS(1) = 0	reporting mode.
STATUS(1) = 1	data checking mode.
STATUS(2)	models to process (3 by default).
STATUS(2) = 0	process all CONEC models.
STATUS(2) = 1	process all CONET models.
STATUS(2) = 2	process all CHAN models.
STATUS(2) = 3	process models by subsystem.
STATUS(2) = 4	process Miscellaneious, "other," type models

STATUS(3) value depends on the value of STATUS(2)

If STATUS(2) is equal to 4:

STATUS(3) = 0	Miscellaneous "Other" model. DOCU for every model instance
> 0	Miscellaneous "Other" model. DOCU for the specified model instance

If STATUS(2) is equal to 3, then STATUS(3) is the type of model for which data is to be processed (1 by default).

= 1, all models	= 2, all plant models
= 3, generator models	= 4, compensator models
= 5, stabilizer models	
= 6, minimum excitation limiter models	
= 7, maximum excitation limiter models	
= 8, excitation system models	= 9, turbine governor models
= 10, load characteristic models	= 11, load relay models
= 12, line relay models	= 13, auxiliary signal models
= 14, two-terminal dc line models	= 15, multi-terminal dc line models
= 16, VSC dc line models	= 17, FACTS device models
= 18, CONEC models	= 19, CONET models
= 20, CHAN models	
= 21, turbine load controller	= 22, switched shunt models
= 23, all wind models	= 24, wind generator models
= 25, wind electrical control models	
= 26, wind mechanical system models	
= 27, wind pitch models	= 28, wind aerodynamic models
= 29, wind gust models	= 30, wind auxiliary signal models
= 31, induction machine models	= 32, branch device models
= 33, two-winding transformer device models	
= 34, three-winding transformer models	
= 35, protection models associated with machines	
= 36, "other" models associated with machines	
= 37, "other" models associated with bus	
= 38, "other" models associated with two-terminal dc lines	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	invalid STATUS value.
	IERR = 4	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.18 DROPMODELLIBRARY

Use this API to remove a library from the list being searched for library models.

Batch command syntax:

```
BAT_DROPMODELLIBRARY  LIBRARYNAME
```

Python syntax:

```
ierr = dropmodellibrary(libraryname)
```

Fortran syntax:

```
CALL DROPMODELLIBRARY(LIBRARYNAME, IERR)
```

where:

Character LIBRARYNAME*260	Is the name of the library of model programs (input; no default allowed).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	Library not found.
	IERR = 2	There are still models in use from this library; unload not done.
	IERR = 3	Library unload failed.
	IERR = 4	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.19 DROPMODELPROGRAM

Use this API to remove a model from the list of known library models; library models are added to this list the first time that they are called.

Batch command syntax:

```
BAT_DROPMODELPROGRAM  PROGRAMNAME
```

Python syntax:

```
ierr = dropmodelprogram(programname)
```

Fortran syntax:

```
CALL DROPMODELPROGRAM(PROGRAMNAME, IERR)
```

where:

Character PROGRAMNAME*260	Is the name of the model program (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 Program not found. IERR = 2 Program unload failed. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.20 DYDA

Use this API to replicate dynamics model data in the form of a Dynamics Data Input file (activity DYDA).

Batch command syntax:

```
BAT_DYDA  SID  ALL  STATUS(1)...STATUS(3)  OUT  OFILE
```

Python syntax:

```
ierr = dyda(sid, all, status, out, ofile)
```

Fortran syntax:

```
CALL DYDANW(SID, ALL, STATUS, OUT, OFILE, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Integer STATUS(3)	Is a three-element array (input). STATUS(1) models to process (2 by default). STATUS(1) = 0 process all CONEC models. STATUS(1) = 1 process all CONET models. STATUS(1) = 2 process models by subsystem. STATUS(1) = 3 process Miscellaneous other type models. STATUS(2) value depends on the value of STATUS(1)

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

If STATUS(1) is equal to 3:

STATUS(2)=0	Miscellaneous, "Other," model DYDA for every model instance.
>0	Miscellaneous, "Other," model DYDA for the specificed model instance.

If STATUS(1) is equal to 2, then STATUS(2) is the type of model for which data is to be processed (1 by default).

= 1, all models	= 2, all plant models
= 3, generator models	= 4, compensator models
= 5, stabilizer models	= 6, minimum excitation limiter models
= 7, maximum excitation limiter models	
= 8, excitation system models	= 9, turbine governor models
= 10, load characteristic models	= 11, load relay models
= 12, line relay models	= 13, auxiliary signal models
= 14, two-terminal dc line models	= 15, multi-terminal dc line models
= 16, VSC dc line models	= 17, FACTS device models
= 18, CONEC models	= 19, CONET models
= 20, turbine load controller	= 21, switched shunt models
= 22, all wind models	= 23, wind generator models
= 24, wind electrical control models	= 25, wind mechanical system models
= 26, wind pitch models	= 27, wind aerodynamic models
= 28, wind gust models	= 29, wind auxiliary signal models
= 30, induction machine models	= 31, branch device models
= 32, two-winding transformer device models	
= 33, three-winding transformer models	
= 34, protection models associated with machines	
= 35, “other” models associated with machines	
= 36, “other” models associated with bus	
= 37, “other” models associated with two-terminal dc lines	

STATUS(3) is the load model output option (0 by default).

STATUS(3) = 0	include all subsystem load models attached to loads in the specified subsystem.
STATUS(3) = 1	exclude subsystem load models attached to loads in the specified subsystem except when subsystem number is explicitly in the list.

Integer OUT Is the output destination selection (input; 1 by default).

OUT = 0	direct output to a data file specified by OFILE.
OUT = 1	direct output to the progress window.

Character OFILE*260 Is the name of the file to which data records are to be written (input; no default allowed when OUT is 0).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	invalid STATUS value.
	IERR = 4	invalid OUT value.
	IERR = 5	error opening OFILE.
	IERR = 6	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.21 DYNAMICS_SOLUTION_PARAMS

This API is obsolete. It has been replaced by the APIs DYNAMICS_SOLUTION_PARAM_2 to change the integer and real dynamic solution parameters and CHANGE_CHANNEL_OUT_FILE to change the channel output filename. Use this API to modify the dynamic simulation solution parameters in dynamics working memory.

Batch command syntax:

```
BAT_DYNAMICS_SOLUTION_PARAMS  INTGAR(1)...INTGAR(8)
REALAR(1)...REALAR(8)  OUTFILE
```

Python syntax:

```
ierr = dynamics_solution_params(intgar, realar, outfile)
```

Fortran syntax:

```
CALL DYNAMICS_SOLUTION_PARAMETERS(INTGAR, REALAR, OUTFILE, IERR)
```

where:

Integer INTGAR(8)	Is an array of eight elements (input; unchanged by default).	
	INTGAR(1)	network solution maximum number of iterations.
	INTGAR(2)	number of output channels being monitored; must be less than or equal to the number of output channels defined.
	INTGAR(3)	number of state variables in use in the simulation model; should not normally be changed by the user.
	INTGAR(4)	next available location in the CON array.
	INTGAR(5)	next available location in the STATE array.
	INTGAR(6)	next available location in the VAR array.
	INTGAR(7)	next available location in the ICON and CHRICN arrays.
	INTGAR(8)	next available location in the output channel arrays.
Real REALAR(8)	Is an array of eight elements (input; unchanged by default).	
	REALAR(1)	acceleration sfactor used in the network solution.
	REALAR(2)	convergence tolerance used in the network solution.
	REALAR(3)	simulation time step (DELT).
	REALAR(4)	filter time constant used in calculating bus frequency deviations.
	REALAR(5)	intermediate simulation mode time step threshold used in extended term simulations.
	REALAR(6)	large (island frequency) simulation mode time step threshold used in extended term simulations.
	REALAR(7)	large (island frequency) simulation mode acceleration factor used in extended term simulations.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	REALAR(8)	large (island frequency) simulation mode convergence tolerance used in extended term simulations.
Character OUTFILE*260	Is the name of the Channel Output File (blank to bypass recording of the output channel values in a Channel Output File) (input; blank by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no errors occurred.
	IERR > 0	error occurred.
	IERR = 1	error opening OUTFILE.
	IERR = 2	prerequisite requirements for API are not met.
	IERR = -1	warnings: • None.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.22 DYNAMICS_SOLUTION_PARAM_2

Use this API to modify the dynamic simulation solution parameters (except the channel output filename) in dynamics working memory (see *PSS®E Program Operation Manual*, activity ALTR).

Batch command syntax:

```
BAT_DYNAMICS_SOLUTION_PARAM_2  INTGAR(1)...INTGAR(8)
REALAR(1)...REALAR(8)
```

Python syntax:

```
ierr = dynamics_solution_param_2(intgar, realar)
```

Fortran syntax:

```
CALL DYNAMICS_SOLUTION_PARAMETER_2(INTGAR, REALAR, IERR)
```

where:

Integer INTGAR(8)	Is an array of eight elements (input; unchanged by default).	
	INTGAR(1)	network solution maximum number of iterations.
	INTGAR(2)	number of output channels being monitored; must be less than or equal to the number of output channels defined.
	INTGAR(3)	number of state variables in use in the simulation model; should not normally be changed by the user.
	INTGAR(4)	next available location in the CON array.
	INTGAR(5)	next available location in the STATE array.
	INTGAR(6)	next available location in the VAR array.
	INTGAR(7)	next available location in the ICON and CHRICN arrays.
	INTGAR(8)	next available location in the output channel arrays.
Real REALAR(8)	Is an array of eight elements (input; unchanged by default).	
	REALAR(1)	acceleration sfactor used in the network solution.
	REALAR(2)	convergence tolerance used in the network solution.
	REALAR(3)	simulation time step (DELT).
	REALAR(4)	filter time constant used in calculating bus frequency deviations.
	REALAR(5)	intermediate simulation mode time step threshold used in extended term simulations.
	REALAR(6)	large (island frequency) simulation mode time step threshold used in extended term simulations.
	REALAR(7)	large (island frequency) simulation mode acceleration factor used in extended term simulations.
	REALAR(8)	large (island frequency) simulation mode convergence tolerance used in extended term simulations.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no errors occurred.
	IERR > 0	error occurred.
	IERR = 1	prerequisite requirements for API are not met.
	IERR = -1	warnings: • None.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.23 DYNAMICSMODE

Use this API to return PSS®E to dynamics mode following the use of the POWERFLOWMODE API while in dynamics mode. (The POWERFLOWMODE API would normally be followed by the use of one or more of the power flow data editor APIs).

Batch command syntax:

```
BAT_DYNAMICSMODE  OPT
```

Python syntax:

```
ierr = dynamicsmode(opt)
```

Fortran syntax:

```
CALL DSMODE(OPT, IERR)
```

where:

Logical OPT	Is the factorize admittance matrix flag (input; .FALSE. by default). OPT = .TRUE. (or T or 1 in batch command and Python) to factorize the admittance matrix. OPT = .FALSE. (or F or 0 in batch command and Python) to omit the matrix factorization.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 PSS®E was started up at its powerflow entry point. IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.24 DYRE_ADD

Use this API to read a Dynamics Model Raw Data File and append the model references specified in its data records to the simulation data already contained in dynamics working memory (activity DYRE,ADD).

Batch command syntax:

```
BAT_DYRE_ADD  STARTINDX(1)...STARTINDX(4)  DYREFILE  CONECFILE
CONETFILE
```

Python syntax:

```
ierr = dyre_add(startindx, dyrefile, conecfile, conetfile)
```

Fortran syntax:

```
CALL DYRE_ADD(STARTINDX, DYREFILE, CONECFILE, CONETFILE, IERR)
```

where:

Integer STARTINDX(4)	Is an array of four elements specifying starting locations in the dynamics data storage arrays (input). STARTINDX(1) starting CON index (the next available CON by default). STARTINDX(2) starting STATE index (the next available STATE by default). STARTINDX(3) starting VAR index (the next available VAR by default). STARTINDX(4) starting ICON index (the next available ICON by default).
Character DYREFILE*260	Is the name of the Dynamics Model Raw Data File (input; no default allowed).
Character CONECFILE*260	Is the name of the output file for CONEC additions; blank to route data to the progress area (input; blank by default).
Character CONETFILE*260	Is the name of the output file for CONET additions; blank to route data to the progress area (input; blank by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred IERR = 1 generators are not converted IERR = 2 invalid STARTINDX value IERR = 3 machine model connection tables full--use pack plant model tables function IERR = 4 error opening output file DYREFILE IERR = 5 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.25 DYRE_NEW

Use this API to clear dynamics working memory, read a Dynamics Data File, and place the model references specified on its data records into dynamics working memory (activity DYRE). It optionally creates a command file for compiling the CONEC and CONET subroutines.

Batch command syntax:

```
BAT_DYRE_NEW  STARTINDX(1)...STARTINDX(4)  DYREFILE  CONECFILE
CONETFILE  COMPILFIL
```

Python syntax:

```
ierr = dyre_new(startindx,dyrefile,conecfile,conetfile,compilfil)
```

Fortran syntax:

```
CALL DYRE_NEW(STARTINDX, DYREFILE, CONECFILE, CONETFILE, COMPILFIL,
IERR)
```

where:

Integer STARTINDX(4)	Is an array of four elements specifying starting locations in the dynamics data storage arrays (input). STARTINDX(1) starting CON index (1 by default). STARTINDX(2) starting STATE index (1 by default). STARTINDX(3) starting VAR index (1 by default). STARTINDX(4) starting ICON index (1 by default).
Character DYREFILE*260	Is the name of the Dynamics Model Raw Data File (input; no default allowed).
Character CONECFILE*260	Is the name of CONEC output file; blank for output to the progress area (input; blank by default).
Character CONETFILE*260	Is the name of CONET output file; blank for output to the progress area (input; blank by default).
Character COMPILFIL*260	Is the name of file containing commands to compile the CONEC and CONET output files; blank for none (input; blank by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 generators are not converted. IERR = 2 invalid STARTINDX value. IERR = 4 error opening output file DYREFILE. IERR = 5 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.26 ERUN

Use this API to calculate PSS®E state-space dynamic simulations of excitation system response tests (activity ERUN).

Batch command syntax:

```
BAT_ERUN  TPAUSE  NPRT  NPLT  CRTPLT
```

Python syntax:

```
ierr = erun(tpause, nprt, nplt, crtplt)
```

Fortran syntax:

```
CALL ERUNAPI(TPAUSE, NPRT, NPLT, CRTPLT, IERR)
```

where:

Real TPAUSE	Is the value of simulated time at which the simulation should next pause (input; no default allowed).
Integer NPRT	Is the number of time steps between the printing of the output channel values (input; unchanged by default).
Integer NPLT	Is the number of time steps between the writing of the output channel values to the current Channel Output File (input; unchanged by default).
Integer CRTPLT	Is the number of time steps between the plotting of those output channel values that have been designated as CRT output channels (input; unchanged by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 activity ESTR needs to be executed. IERR = 2 error opening the current Channel Output File. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.27 ESTR_OPEN_CIRCUIT_TEST

Use this API to initialize a PSS®E dynamic simulation for excitation system open circuit response simulations (i.e., in preparation for activity ERUN) and to specify the Channel Output File into which the output channel values are to be recorded during the simulation (activity ESTR).

Batch command syntax:

```
BAT_ESTR_OPEN_CIRCUIT_TEST  OPTION  VDELTA  OUTFILE
```

Python syntax:

```
ierr = estr_open_circuit_test(option, vdelta, outfile)
```

Fortran syntax:

```
CALL ESTR_OPEN_CIRCUIT_TEST(OPTION, VDELTA, OUTFILE, IERR)
```

where:

Integer OPTION	Is the bus flag indicating those buses to be tested (input; 0 by default). OPTION = 0 — run test on all in-service machines with excitation system models. OPTION = — bus number for which machines are to be tested.
Real VDELTA	Is the per unit voltage reference step applied at TIME = 0.0 (input; 0.05 by default).
Character OUTFILE*260	Is the name of the Channel Output File (blank to bypass recording of the output channel values in a Channel Output File) (input; blank by default).
Integer IERR	Is the error code (output). IERR = 0 — no error occurred. IERR = 1 — bus not found. IERR = 2 — error opening OUTFILE. IERR = 3 — generators are not "converted". IERR = 4 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.28 ESTR_RESPONSE_RATIO_TEST

Use this API to initialize a PSS®E dynamic simulation for excitation system response ratio test simulations (i.e., in preparation for activity ERUN) and to specify the Channel Output File into which the output channel values are to be recorded during the simulation (activity ESTR).

Batch command syntax:

```
BAT_ESTR_RESPONSE_RATIO_TEST  OPTION  PFDEFALT  NMACHS
BUSARY(1)...BUSARY(NMACHS)   IDARY(1)...IDARY(NMACHS)
PFARY(1)...PFARY(NMACHS)  OUTFILE
```

Python syntax:

```
ierr = estr_response_ratio_test(option, pfdefalt, nmachs, busary,
idary, pfary, outfile)
```

Fortran syntax:

```
CALL ESTR_RESPONSE_RATIO_TEST(OPTION, PFDEFALT, NMACHS, BUSARY,
IDARY, PFARY, OUTFILE, IERR)
```

where:

Integer OPTION	Is the bus flag indicating those buses to be tested (input; 0 by default). OPTION = 0 — run test on all in-service machines with excitation system models. OPTION = — bus number for which machines are to be tested.
Real PFDEFALT	Is the default power factor at which machines are initialized (input; 0.95 by default).
Integer NMACHS	Is the number of machines to be initialized at other power factors (input; 0 by default).
Integer BUSARY(NMACHS)	Is a list of bus numbers identifying machines to be initialized at other power factors; ignored if NMACHS = 0 (input; no default allowed).
Character IDARY(NMACHS)*2	Is a list of machine identifiers identifying machines to be initialized at other power factors; ignored if NMACHS = 0 (input; ’1’ by default).
Real PFARY(NMACHS)	Is a list of power factors; ignored if NMACHS = 0. Machine IDARY(n) at bus BUSARY(n) is initialized to the power factor specified in PFARY(n) (input; PFDEFALT by default).
Character OUTFILE*260	Is the name of the Channel Output File (blank to bypass recording of the output channel values in a Channel Output File) (input; blank by default).
Integer IERR	Is the error code (output). IERR = 0 — no error occurred. IERR = 1 — bus not found. IERR = 2 — invalid power factor. IERR = 3 — error opening OUTFILE. IERR = 4 — generators are not "converted". IERR = 5 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.29 FCLIST

Use this API to list FACTS device models connected to subsystem buses along with their storage locations in the dynamics data arrays.

Batch command syntax:

```
BAT_FCLIST  SID  ALL  STATUS
```

Python syntax:

```
ierr = fclist(sid, all, status)
```

Fortran syntax:

```
CALL FCLIST(SID, ALL, STATUS, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Integer STATUS	Is the active/bypassed flag (input; 0 by default). STATUS = 0 active and bypassed models. STATUS = 1 active models only. STATUS = 2 bypassed models only.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 invalid STATUS value. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.30 GRUN

Use this API to calculate PSS®E state-space dynamic simulations of turbine-governor response tests (activity GRUN).

Batch command syntax:

```
BAT_GRUN  TPAUSE  NPRT  NPLT  CRTPLT
```

Python syntax:

```
ierr = grun(tpause, nprt, nplt, crtplt)
```

Fortran syntax:

```
CALL GRUNAPI(TPAUSE, NPRT, NPLT, CRTPLT, IERR)
```

where:

Real TPAUSE	Is the value of simulated time at which the simulation should next pause (input; no default allowed).
Integer NPRT	Is the number of time steps between the printing of the output channel values (input; unchanged by default).
Integer NPLT	Is the number of time steps between the writing of the output channel values to the current Channel Output File (input; unchanged by default).
Integer CRTPLT	Is the number of time steps between the plotting of those output channel values that have been designated as CRT output channels (input; unchanged by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 activity GSTR needs to be executed. IERR = 2 error opening the current Channel Output File. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.31 GSTR

Use this API to initialize a PSS®E dynamic simulation for governor response simulations (i.e., in preparation for activity GRUN) and to specify the Channel Output File into which the output channel values are to be recorded during the simulation (activity GSTR).

Batch command syntax:

```
BAT_GSTR  OPTION  PINITIAL  PDELTA  OUTFILE
```

Python syntax:

```
ierr = gstr(option, pinitial, pdelta, outfile)
```

Fortran syntax:

```
CALL GSTRAPI(OPTION, PINITIAL, PDELTA, OUTFILE, IERR)
```

where:

Integer OPTION	Is the bus flag indicating those buses to be tested (input; 0 by default). OPTION = 0 run test on all in-service machines with turbine-governor models. OPTION = bus number for which machines are to be tested.
Real PINITIAL	Is the initial per unit loading of all machines being tested (input; 0.8 by default).
Real PDELTA	Is the per unit loading step applied at TIME = 0.0 (input; 0.1 by default).
Character OUTFILE*260	Is the name of the Channel Output File (blank to bypass recording of the output channel values in a Channel Output File) (input; blank by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 bus not found. IERR = 2 error opening OUTFILE. IERR = 3 generators are not, "converted". IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.32 INCREMENT_GREF

Use this API to add a specified value to the value of the element of the GREF (governor reference) array associated with a specified machine.

Batch command syntax:

```
BAT_INCREMENT_GREF  IBUS  'ID'  NEWVAL
```

Python syntax:

```
ierr = increment_gref(ibus, id, newval)
```

Fortran syntax:

```
CALL INCREMENT_GREF(IBUS, ID, NEWVAL, IERR)
```

where:

Integer IBUS	Is the plant bus number (input; no default allowed).
Character ID*2	Is the machine identifier (input; '1' by default).
Real NEWVAL	Is the new value to be added to the GREF array entry associated with machine ID at bus IBUS (input; 0.0 by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 error occurred. IERR = 1 bus not found. IERR = 2 machine not found. IERR = 3 prerequisite requirements for API are not met. IERR = -1 warnings – one or more of: • No turbine-governor model at this machine • Turbine-governor model at this machine is bypassed.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.33 INCREMENT_SWSREF

Use this API to add a specified value to the value of the element of the SWSREF array associated with a switched shunt.

Batch command syntax:

```
BAT_INCREMENT_SWSREF  IBUS  NEWVAL
```

Python syntax:

```
ierr = increment_swsref(ibus, newval)
```

Fortran syntax:

```
CALL INCREMENT_SWSREF(IBUS, NEWVAL, IERR)
```

where:

Integer IBUS	Is the switched shunt bus number (input; no default allowed).
Real NEWVAL	Is the new value to be added to the SWSREF array entry associated with switched shunt at bus IBUS (input; 0.0 by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error occurred.
	IERR > 0 — error occurred.
	IERR = 1 — bus not found.
	IERR = 2 — switched shunt not found.
	IERR = 3 — prerequisite requirements for API are not met.
	IERR = -1 — warnings – one or more of: • No table model for switched shunt at this bus • Switched Shunt Dynamic Model at this switched shunt is bypassed.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.34 INCREMENT_VREF

Use this API to add a specified value to the value of the element of the VREF array associated with a specified machine.

Batch command syntax:

```
BAT_INCREMENT_VREF  IBUS  'ID'  NEWVAL
```

Python syntax:

```
ierr = increment_vref(ibus, id, newval)
```

Fortran syntax:

```
CALL INCREMENT_VREF(IBUS, ID, NEWVAL, IERR)
```

where:

Integer IBUS	Is the plant bus number (input; no default allowed).
Character ID*2	Is the machine identifier (input; '1' by default).
Real NEWVAL	Is the new value to be added to the VREF array entry associated with machine ID at bus IBUS (input; 0.0 by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error occurred.
	IERR > 0 — error occurred.
	IERR = 1 — bus not found.
	IERR = 2 — machine not found.
	IERR = 3 — prerequisite requirements for API are not met.
	IERR = -1 — warnings – one or more of: • No excitation system model at this machine • Excitation system model at this machine is bypassed.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.35 LDLIST

Use this API to list load models connected to subsystem loads along with their storage locations in the dynamics data arrays.

Batch command syntax:

```
BAT_LDLIST  SID  ALL  STATUS(1)  STATUS(2)
```

Python syntax:

```
ierr = ldlist(sid, all, status)
```

Fortran syntax:

```
CALL LDLIST(SID, ALL, STATUS, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).
	ALL = 1 process all buses.
	ALL = 0 process only buses in subsystem SID.
Integer STATUS(2)	Is a two-element array (input).
	STATUS(1) the active/bypassed flag (0 by default).
	STATUS(1) = 0 active and bypassed models.
	STATUS(1) = 1 active models only.
	STATUS(1) = 2 bypassed models only.
	STATUS(2) the model connection report option (0 by default).
	STATUS(2) = 0 list models connected to loads.
	STATUS(2) = 1 list loads connected to model.
Integer IERR	Is the error code (output).
	IERR = 0 no error.
	IERR = 1 invalid SID value or subsystem SID is not defined.
	IERR = 2 invalid ALL value.
	IERR = 3 invalid STATUS value.
	IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.36 MLST

Use this API to list plant models connected to subsystem machines along with their storage locations in the dynamics data arrays (activity MLST).

Batch command syntax:

```
BAT_MLST  SID  ALL  STATUS(1)...STATUS(3)
```

Python syntax:

```
ierr = mlst(sid, all, status)
```

Fortran syntax:

```
CALL MLSTNW(SID, ALL, STATUS, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all buses.
	ALL = 0	process only buses in subsystem SID.
Integer STATUS(3)	Is a three-element array (input).	
	STATUS(1)	the connected/unconnected flag (1 by default).
	STATUS(1) = 0	list all unconnected models.
	STATUS(1) = 1	list connected subsystem models.
	STATUS(2)	the active/bypassed flag (0 by default).
	STATUS(2) = 0	active and bypassed models.
	STATUS(2) = 1	active models only.
	STATUS(2) = 2	bypassed models only.
	STATUS(3)	the plant model type (1 by default).
	STATUS(3) = 1	all plant models.
	STATUS(3) = 2	generator models.
	STATUS(3) = 3	compensator models.
	STATUS(3) = 4	stabilizer models.
	STATUS(3) = 5	minimum excitation models.
	STATUS(3) = 6	maximum excitation models.
	STATUS(3) = 7	excitation system models.
	STATUS(3) = 8	governor models.
	STATUS(3) = 9	turbine load controller models.
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid SID value or subsystem SID is not defined.
	IERR = 2	invalid ALL value.
	IERR = 3	invalid STATUS value.
	IERR = 4	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.37 MOVEMODELFOLDER

This API is obsolete and performs no function.

Batch command syntax:

```
BAT_MOVEMODELFOLDER  FOLDERINDEX NEWINDEX
```

Python syntax:

```
ierr = movemodelfolder(folderindex, newindex)
```

Fortran syntax:

```
CALL MOVEMODELFOLDER(FOLDERINDEX, NEWINDEX, IERR)
```

where:

Integer FOLDERINDEX	Is the index (ordinal position, starting with 1) of the folder to be moved (input; no default allowed).
Integer NEWINDEX	Is the index (ordinal position, starting with 1) where the folder currently in position FOLDERINDEX should be located (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 folder index out of range. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.38 MRUN

Use this API to calculate PSS®E extended term dynamic simulations (activity MRUN).

Batch command syntax:

```
BAT_MRUN  OPTION  TPAUSE  NPRT  NPLT  CRTPLT
```

Python syntax:

```
ierr = mrun(option, tpause, nprt, nplt, crtplt)
```

Fortran syntax:

```
CALL MRUNAPI(OPTION, TPAUSE, NPRT, NPLT, CRTPLT, IERR)
```

where:

Integer OPTION	Is the network solution convergence monitor option (input; 0 by default). OPTION = 1 automatically print the convergence monitor. OPTION = 0 print the convergence monitor only if it is enabled via the CM interrupt control code.
Real TPAUSE	Is the value of simulated time at which the simulation should next pause (input; no default allowed).
Integer NPRT	Is the number of time steps between the printing of the output channel values (input; unchanged by default).
Integer NPLT	Is the number of time steps between the writing of the output channel values to the current Channel Output File (input; unchanged by default).
Integer CRTPLT	Is the number of time steps between the plotting of those output channel values that have been designated as CRT output channels (input; unchanged by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 long time step mode threshold is at or below intermediate mode threshold. IERR = 2 activity MSTR needs to be executed. IERR = 3 invalid OPTION value. IERR = 4 generators are not converted. IERR = 5 error opening the current Channel Output File. IERR = 6 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.39 MSTR

Use this API to initialize a PSS®E dynamic simulation for extended term simulations (i.e., in preparation for activity MRUN) and to specify the Channel Output File into which the output channel values are to be recorded during the dynamic simulation (activity MSTR).

Batch command syntax:

```
BAT_MSTR  OPTION  OUTFILE
```

Python syntax:

```
ierr = mstr(option, outfile)
```

Fortran syntax:

```
CALL MSTRAPI(OPTION, OUTFILE, IERR)
```

where:

Integer OPTION	Is the network solution convergence monitor option (input; 0 by default). OPTION = 1 automatically print the convergence monitor. OPTION = 0 print the convergence monitor only if it is enabled via the CM interrupt control code.
Character OUTFILE*260	Is the name of the Channel Output File (blank to bypass recording of the output channel values in a Channel Output File) (input; blank by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 long time step mode threshold is at or below intermediate mode threshold. IERR = 2 DELT is greater than intermediate time step mode threshold. IERR = 3 generators are not converted. IERR = 4 invalid OPTION value. IERR = 5 Prior initialization modified the loads--pick up original converted case. IERR = 6 error opening OUTFILE. IERR = 7 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.40 POWERFLOWMODE

Use this API to switch PSS®E from dynamics mode to power flow mode. (This API would normally be followed by the use of one or more of the power flow data editor APIs, and then the DYNAMIC-SMODE API).

Batch command syntax.

```
BAT_POWERFLOWMODE
```

Python syntax:

```
ierr = powerflowmode()
```

Fortran syntax:

```
CALL PFMODE(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.41 PSAS

Use this API to convert a PSAS Command File into a PSS®E Response File (activity PSAS).

Batch command syntax:

```
BAT_PSAS  INPFILE  OUTFILE
```

Python syntax:

```
ierr,respfil = psas(inpfile, outfile)
```

Fortran syntax:

```
CALL PSASAPI(INPFILE, OUTFILE, RESPFIL, IERR)
```

where:

Character INPFILE*260	Is the name of the PSAS Command file (input; no default allowed).
Character OUTFILE*260	Is the name of output Response File; blank to have PSAS assign the filename (input; blank by default).
Character RESPFIL*260	Is the pathname of the Response File constructed by PSAS (output).
Integer IERR	Is the error code (output).
	IERR = 0 — no error occurred.
	IERR = 1 — error opening INPFILE.
	IERR = 2 — error opening OUTFILE.
	IERR = 3 — errors detected while reading the PSAS Command Data input file.
	IERR = 4 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.42 RLLIST

Use this API to list line relay models connected to subsystem buses along with their storage locations in the dynamics data arrays.

Batch command syntax:

```
BAT_RLLIST  SID  ALL  STATUS
```

Python syntax:

```
ierr = rllist(sid, all, status)
```

Fortran syntax:

```
CALL RLLIST(SID, ALL, STATUS, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Integer STATUS	Is the active/bypassed flag (input; 0 by default). STATUS = 0 active and bypassed models. STATUS = 1 active models only. STATUS = 2 bypassed models only.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 invalid STATUS value. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.43 RSTR

Use this API to read a dynamics Snapshot File into PSS®E working memory (activity RSTR).

Batch command syntax:

```
BAT_RSTR  SFILE
```

Python syntax:

```
ierr = rstr(sfile)
```

Fortran syntax:

```
CALL RSTRNW(SFILE, IERR)
```

where:

Character SFILE*260	Is the Snapshot File; '*' to restore the most recently accessed Snapshot File in the current execution of PSS®E (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 no error.
	IERR = 1 error opening SFILE.
	IERR = 2 error reading from SFILE.
	IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.44 RUN

Use this API to calculate PSS®E state-space dynamic simulations (activity RUN).

Batch command syntax:

```
BAT_RUN  OPTION  TPAUSE  NPRT  NPLT  CRTPLT
```

Python syntax:

```
ierr = run(option, tpause, nprt, nplt, crtplt)
```

Fortran syntax:

```
CALL RUNAPI(OPTION, TPAUSE, NPRT, NPLT, CRTPLT, IERR)
```

where:

Integer OPTION	Is the network solution convergence monitor option (input; 0 by default). OPTION = 1 automatically print the convergence monitor. OPTION = 0 print the convergence monitor only if it is enabled via the CM interrupt control code.
Real TPAUSE	Is the value of simulated time at which the simulation should next pause (input; no default allowed).
Integer NPRT	Is the number of time steps between the printing of the output channel values (input; unchanged by default).
Integer NPLT	Is the number of time steps between the writing of the output channel values to the current Channel Output File (input; unchanged by default).
Integer CRTPLT	Is the number of time steps between the plotting of those output channel values that have been designated as CRT output channels (input; unchanged by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 activity STRT needs to be executed. IERR = 2 invalid OPTION value. IERR = 3 generators are not converted. IERR = 4 error opening the current Channel Output File. IERR = 5 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.45 RWDY

Use this API to replicate portions of dynamics model data in the form of either a Breaker Duty Data File, Unit Inertia and Governor Data File, or PSSPLT Relay Characteristic Data File (activity RWDY).

Batch command syntax:

```
BAT_RWDY  OPTIONS(1)...OPTIONS(3)  OUT  OFILE
```

Python syntax:

```
ierr = rwdy(options, out, ofile)
```

Fortran syntax:

```
CALL RWDYAPI(OPTIONS, OUT, OFILE, IERR)
```

where:

Integer OPTIONS(3)	Is a three-element array indicating output options (input).	
	OPTIONS(1)	is the type of file to be output (1 by default):
	OPTIONS(1) = 1	Breaker Duty Data File.
	OPTIONS(1) = 2	Unit Inertia and Governor Data File.
	OPTIONS(1) = 3	PSSPLT Relay Characteristic Data File.
	OPTIONS(2)	is the machine limits flag for activity INLF (ignored unless OPTIONS(1) = 2) (0 by default).
	OPTIONS(2) = 0	include machine limits.
	OPTIONS(2) = 1	suppress machine limits.
	OPTIONS(3)	is a flag indicating the treatment of machines without governor models in activity INLF (ignored unless OPTIONS(1) = 2) (0 by default):
	OPTIONS(3) = 0	use the working case limits.
	OPTIONS(3) = 1	treat them as non-dispatchable.
Integer OUT	Is the output destination selection (input; 1 by default).	
	OUT = 0	direct output to a data file specified by OFILE.
	OUT = 1	direct output to the progress window.
Character OFILE*260	Is the name of the file to which data records are to be written (input; blank by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid OPTIONS value.
	IERR = 2	invalid OUT value.
	IERR = 3	error opening OFILE.
	IERR = 4	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.46 SET_GENANG

This API is obsolete. This API has been replaced by SET_GENANG_2.
Use this API to enable or disable the simulation option setting that scans for generators for which the angle differs from the angular average by more than a specified threshold.
Batch command syntax:

```
BAT_SET_GENANG  STATUS  ANGLE
```

Python syntax:

```
ierr = set_genang(status, angle)
```

or:

```
ierr,status,angle = set_genang()
```

Fortran syntax:

```
CALL SET_GENANG(STATUS, ANGLE, IOCODE, IERR)
```

where:

Integer STATUS	Is the value of the option setting (input if batch command or ICODE is 0 or negative; unchanged by default) (output if ICODE is positive). STATUS = 1 enable scanning for generators exceeding angle threshold. STATUS = anything else, disable scanning option.
Real ANGLE	Is the value of the deviation threshold (input if batch command or ICODE is 0 or negative; unchanged by default) (output if ICODE is positive).
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE ≤ 0 set the option setting to the value specified in STATUS and the threshold to the value specified in ANGLE. IOCODE > 0 return the current value of the option setting in STATUS and the threshold in ANGLE.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid angle threshold--must be >0.0. IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.47 SET_GENANG_2

Use this API to enable or disable the simulation option setting that scans for generators for which the angle differs from the angular average by more than a specified threshold. Scanning, if enabled, will be done for all machines whose machine MVA is greater than the specified value
Batch command syntax:

```
BAT_SET_GENANG_2  STATUS  ANGLE    MBASE
```

Python syntax:

```
ierr = set_genang_2(status, angle, mbase)
```

or:

```
ierr,status,angle,mbase = set_genang_2()
```

Fortran syntax:

```
CALL SET_GENANG_2(STATUS, ANGLE, MBASE, IOCODE, IERR)
```

where:

Integer STATUS	Is the value of the option setting (input if batch command or ICODE is 0 or negative; unchanged by default) (output if ICODE is positive). STATUS = 1 enable scanning for generators exceeding angle threshold. STATUS = anything else, disable scanning option.
Real ANGLE	Is the value of the deviation threshold (input if batch command or ICODE is 0 or negative; unchanged by default) (output if ICODE is positive).
Real MBASE	Is the value of the machine size (MVA) - (input if batch command or ICODE is 0 or negative; unchanged by default) (output if ICODE is positive). All machines whose MVAR ratings are above this threshold will be scanned.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE ≤ 0 set the option setting to the value specified in STATUS and the threshold values specified in ANGLE and MBASE. IOCODE > 0 return the current value of the option setting in STATUS and the threshold values in ANGLE and MBASE

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid angle threshold--must be > 0.0.
	IERR = 2	invalid angle threshold--must be ≥ 0.0.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.48 SET_GENPWR

Use this API to enable or disable the simulation option setting that scans for generators for which the mechanical power differs from its electrical power by more than a specified threshold.

Batch command syntax:

```
BAT_SET_GENPWR  STATUS  POWER
```

Python syntax:

```
ierr = set_genpwr(status, power)
```

or:

```
ierr,status,power = set_genpwr()
```

Fortran syntax:

```
CALL SET_GENPWR(STATUS, POWER, IOCODE, IERR)
```

where:

Integer STATUS	Is the value of the option setting (input if batch command or ICODE is 0 or negative; unchanged by default) (output if ICODE is positive).	
	STATUS = 1	enable scanning for generators exceeding power unbalance threshold.
	STATUS =	anything else, disable scanning option.
Real POWER	Is the value of the power unbalance threshold (input if batch command or ICODE is 0 or negative; unchanged by default) (output if ICODE is positive).	
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE ≤ 0	set the option setting to the value specified in STATUS and the threshold to the value specified in POWER.
	IOCODE > 0	return the current value of the option setting in STATUS and the threshold in POWER.
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	invalid power unbalance threshold--must be >0.0 and <10.0.
	IERR = 2	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.49 SET_NETFRQ

Use this API to enable or disable the simulation option setting that models the frequency dependence of network parameters.

Batch command syntax:

```
BAT_SET_NETFRQ  STATUS
```

Python syntax:

```
ierr = set_netfrq(status)
```

or:

```
ierr,status = set_netfrq()
```

Fortran syntax:

```
CALL SET_NETFRQ(STATUS, IOCODE, IERR)
```

where:

Integer STATUS	Is the value of the option setting (input if batch command or ICODE is 0 or negative; unchanged by default) (output if ICODE is positive).	
	STATUS = 1	enable network frequency dependence.
	STATUS =	anything else, disable network frequency dependence.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed).	
	IOCODE ≤ 0	set the option setting to the value specified in STATUS.
	IOCODE > 0	return the current value of the option setting in STATUS.
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.50 SET_NEXT_CHANNEL

Use this API to assign the next available channel index value.

Batch command syntax:

```
BAT_SET_NEXT_CHANNEL  IVAL
```

Python syntax:

```
ierr = set_next_channel(ival)
```

Fortran syntax:

```
CALL SET_NEXT_CHANNEL(IVAL, IERR)
```

where:

Integer IVAL	Value to assign to the next available channel index.	
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid channel index.
	IERR = 2	index is beyond the maximum allowed.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.51 SET_NEXT_ICON

Use this API to assign the next available icon index value.

Batch command syntax:

```
BAT_SET_NEXT_ICON  IVAL
```

Python syntax:

```
ierr = set_next_icon(ival)
```

Fortran syntax:

```
CALL SET_NEXT_ICON(IVAL, IERR)
```

where:

Integer IVAL	Value to assign to the next available icon index.	
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid channel index.
	IERR = 2	index is beyond the maximum allowed.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.52 SET_NEXT_VAR

Use this API to assign the next available VAR index value.

Batch command syntax:

```
BAT_SET_NEXT_VAR  IVAL
```

Python syntax:

```
ierr = set_next_var(ival)
```

Fortran syntax:

```
CALL SET_NEXT_VAR(IVAL, IERR)
```

where:

Integer IVAL	Value to assign to the next available var index.	
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid channel index.
	IERR = 2	index is beyond the maximum allowed.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.53 SET_OSSCAN

Use this API to enable or disable the simulation option setting that scans for out-of-step conditions using a generic relay.

Batch command syntax:

```
BAT_SET_OSSCAN  STATUS  TRIP
```

Python syntax:

```
ierr = set_osscan(status, trip)
```

or:

```
ierr,status,trip = set_osscan()
```

Fortran syntax:

```
CALL SET_OSSCAN(STATUS, TRIP, IOCODE, IERR)
```

where:

Integer STATUS	Is the value of the option setting (input if batch command or ICODE is 0 or negative; unchanged by default) (output if ICODE is positive). STATUS = 1 enable out-of-step scanning. STATUS = anything else, disable out-of-step scanning.
Integer TRIP	Is the trip flag (input; unchanged by default). TRIP = 1 trip out-of-step circuits. TRIP = anything else, monitor only.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE ≤ 0 set the option setting to the value specified in STATUS and the trip flag to the value specified in TRIP. IOCODE > 0 return the current value of the option setting in STATUS and the trip flag in TRIP.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.54 SET_RELANG

Use this API to enable or disable the simulation option setting that expresses the ANGLE array relative to a designated reference angle.

Batch command syntax:

```
BAT_SET_RELANG  SWITCH  IBUSEX  'ID'
```

Python syntax:

```
ierr = set_relang(switch, ibusex, id)
```

or:

```
ierr,switch,ibusex,id = set_relang()
```

Fortran syntax:

```
CALL SET_RELANG(SWITCH, IBUSEX, ID, IOCODE, IERR)
```

where:

Integer SWITCH	Is the value of the option setting (input if batch command or ICODE is 0 or negative; unchanged by default) (output if ICODE is positive). SWITCH = 1 enable relative angle calculation. SWITCH = anything else, disable relative angle calculation (use absolute angles).
Integer IBUSEX	Is the type of relative angle calculation to be applied (input if batch command or ICODE is 0 or negative; unchanged by default) (output if ICODE is positive). IBUSEX > 0 use machine ID at bus IBUSEX as the reference machine. IBUSEX = 0 use system average angle. IBUSEX = -1 use system weighted average angle.
Character ID*2	Is the machine identifier (input if batch command or ICODE is 0 or negative; '1' by default) (output if ICODE is positive).
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE ≤ 0 set the option setting to the value specified in SWITCH and the relative angle parameters to the values specified in IBUSEX and ID. IOCODE > 0 return the current value of the option setting in SWITCH and the relative angle parameters in IBUSEX and ID.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid IBUSEX value. IERR = 2 bus not found. IERR = 3 machine not found. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.55 SET_RELSCN

Use this API to enable or disable the simulation option setting that scans branches using a generic branch relay.

Batch command syntax:

```
BAT_SET_RELSCN  STATUS
```

Python syntax:

```
ierr = set_relscn(status)
```

or:

```
ierr,status = set_relscn()
```

Fortran syntax:

```
CALL SET_RELSCN(STATUS, IOCODE, IERR)
```

where:

Integer STATUS	Is the value of the option setting (input if batch command or ICODE is 0 or negative; unchanged by default) (output if ICODE is positive). STATUS = 1 enable relay scanning. STATUS = anything else, disable relay scanning.
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE ≤ 0 set the option setting to the value specified in STATUS. IOCODE > 0 return the current value of the option setting in STATUS.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.56 SET_VLTSCN

Use this API to enable or disable the simulation option setting that scans buses for high or low values of voltage magnitude.

Batch command syntax:

```
BAT_SET_VLTSCN  STATUS  VHI  VLO
```

Python syntax:

```
ierr = set_vltscn(status, vhi, vlo)
```

or:

```
ierr,status,vhi,vlo = set_vltscn()
```

Fortran syntax:

```
CALL SET_VLTSCN(STATUS, VHI, VLO, IOCODE, IERR)
```

where:

Integer STATUS	Is the value of the option setting (input if batch command or ICODE is 0 or negative; unchanged by default) (output if ICODE is positive). STATUS = 1 enable voltage scanning. STATUS = anything else, disable voltage scanning.
Real VHI	Is the high voltage threshold (input if batch command or ICODE is 0 or negative; unchanged by default) (output if ICODE is positive).
Real VLO	Is the low voltage threshold (input if batch command or ICODE is 0 or negative; unchanged by default) (output if ICODE is positive).
Integer IOCODE	Indicates the operation mode of the API (input; no default allowed). IOCODE ≤ 0 set the option setting to the value specified in STATUS and the voltage limits to VHI and VLO. IOCODE > 0 return the current value of the option setting in STATUS and the voltage limits in VHI and VLO.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid specification of VHI and VLO. IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.57 SET_VOLT_VIOL_SUBSYS_FLAG

This API is used to set the flag to denote the bus subsystem to be used for voltage violation (voltage recovery and voltage dip) checks.

Batch command syntax:

```
BAT_SET_VOLT_VIOL_SUBSYS_FLAG    ALL
```

Python syntax:

```
ierr = set_volt_viol_subsys_flag (all)

or

ierr,all = set_volt_viol_subsys_flag (all)
```

Fortran syntax:

```
CALL SET_VOLT_VIOL_SUBSYS_FLAG (ALL, ICODE, IERR)
```

where:

Integer ALL	Is the flag to set the subsystem definition for performing voltage violation check: (input if batch command or ICODE is 0 or negative; unchanged by default) (output if ICODE is positive). ALL = 1 Do voltage violation check on all buses ALL = 0 Use existing bus subsystem definition
Integer ICODE	Indicates the operation mode of API (input; no default allowed). ICODE ≤ 0 set the option setting to the value specified in ALL. ICODE > 0 return the current value of the option setting in ALL.
Integer IERR	Is the error code (output). IERR = 0 no error IERR = 1 Invalid ALL value IERR = 2 prerequisite requirements for API not met

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.58 SET_VOLTAGE_DIP_CHECK

This API is used to enable or disable the simulation option setting which scans buses for voltage dip (back-swing).

Batch command syntax:

```
BAT_ SET_VOLTAGE_DIP_CHECK   VDPCHK   VDPTHR   VDPDUR
```

Python syntax:

```
ierr = set_voltage_dip_check (vdpchk, vdpthr, vdpdur)

or

ierr, set_voltage_dip_check (vdpchk, vdpthr, vdpdur)
```

Fortran syntax:

```
CALL SET_VOLTAGE_DIP_CHECK (VDPCHK, VDPTHR, VDPDUR, ICODE, IERR)
```

where:

Integer VDPCHK	Is the flag to set voltage dip check: (input if command or ICODE is 0 or negative; unchanged by default). (output if ICODE is positive.) VDPCHK = 1 enable voltage dip check VDPCHK ≠ 1 disable voltage dip check
Real VDPTHR	Is the voltage dip threshold in per unit: (input if batch command or ICODE is 0 or negative; unchanged by default). (output if ICODE is positive.) If the voltage swings-back below this threshold a timer is imitated to check for the dip duration.
Real VDPDUR	Is the voltage dip duration in seconds: (input if batch command or ICODE is 0 or negative; unchanged by default). (output if ICODE is positive.)
Integer ICODE	Indicates the operation mode of API (input; no default allowed). ICODE ≤ 0 set the option setting to the value specified in VDPCHK, VDPTHR, and VDPDUR. ICODE > 0 return the current value of the option setting in VDPCHK, VDPTHR, and VDPDUR.
Integer IERR	Is the error code (output). IERR = 0 no error IERR = 1 prerequisite requirements for API not met

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.59 SET_VOLTAGE_REC_CHECK

This API is used to enable or disable the simulation option setting which scans buses for voltage recovery (primary and secondary recovery).

Batch command syntax:

```
BAT_ SET_VOLTAGE_REC_CHECK   VPRCHK   VSRCHK   VPRTHR   VPRTIM
VSRTHR   VSRTIM
```

Python syntax:

```
ierr = set_voltage_rec_check (vprchk, vsrchk, vprthr, vprtim, vsrthr,
vsrtim)

or

ierr,set_voltage_rec_check (vprchk, vsrchk, vprthr, vprtim, vsrthr,
vsrtim)
```

Fortran Syntax:

```
CALL SET_VOLTAGE_REC_CHECK (VPRCHK, VSRCHK, VPRTHR, VPRTIM, VSRTHR
VSRTIM, ICODE, IERR)
```

where:

Integer VPRCHK	Is the flag to set voltage primary recovery check: (input if batch command or ICODE is 0 or negative; unchanged by default). (output if ICODE is positive.) VPRCHK = 1 enable primary recovery check VPRCHK ≠ 1 disable primary recovery check
Integer VSRCHK	Is the flag to set voltage secondary recovery check: (input if batch command or ICODE is 0 or negative; unchanged by default). (output if ICODE is positive.) This flag can be set only if the primary recovery check is ON. VSRCHK = 1 enable secondary recovery check VSRCHK ≠ 1 disable secondary recovery check
Real VPRTHR	Is the voltage primary recovery threshold in per unit: (input if batch command or ICODE is 0 or negative; unchanged by default). (output if ICODE is positive.)
Real VPRTIM	Is the voltage primary recovery time in seconds: (input if batch command or ICODE is 0 or negative; unchanged by default). (output if ICODE is positive.)
Real VSRTHR	Is the voltage secondary recovery threshold in per unit: (input if batch command or ICODE is 0 or negative; unchanged by default). (output if ICODE is positive.) VSRTHR has to be greater than VPRTHR.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real VSRTIM	Is the voltage secondary recovery time in seconds: (input if batch command or ICODE is 0 or negative; unchanged by default). (output if ICODE is positive.) VSRTIM has to be greater than VPRTIM.
Integer ICODE	Indicates the operation mode of API (input; no default allowed). ICODE ≤ 0 — set the option setting to the value specified in VPRCHK, VSRCHK, VPRTHR, VPRTIM, VSRTHR and VSRTIM. ICODE > 0 — return the current value of the option setting in VPRCHK, VSRCHK, VPRTHR, VPRTIM, VSRTHR and VSRTIM.
Integer IERR	Is the error code (output). IERR = 0; no error IERR = 1 — invalid specification of VPRTHR and VSRTHR IERR = 2 — invalid specification of VPRTIM and VSRTIM IERR = 3 — prerequisite requirements for API not met

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.60 SETMODELFOLDER

This API is obsolete and performs no function.

Batch command syntax:

```
BAT_SETMODELFOLDER  PATHNAME FOLDERINDEX
```

Python syntax:

```
ierr = setmodelfolder(pathname, folderindex)
```

Fortran syntax:

```
CALL SETMODELFOLDER(PATHNAME, FOLDERINDEX, IERR)
```

where:

Character PATHNAME*260	Is the name of the path (input; no default allowed).
Integer FOLDERINDEX	Is the index (ordinal position, starting with 1) in the list that should be set to value of PATHNAME. If the current length of the list is X, then FOLDERINDEX may be no larger than X+1 (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 folder index out of range. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.61 SETMODELFOLDERS

This API is obsolete and performs no function.

Batch command syntax:

```
BAT_SETMODELFOLDERS  SIZFOLDERS  PATHNAMES(1)...PATHNAMES(SIZFOLDERS)
```

Python syntax:

```
ierr = setmodelfolders(sizfolders, pathnames)
```

Fortran syntax:

```
CALL SETMODELFOLDERS(SIZFOLDERS, PATHNAMES, IERR)
```

where:

Integer SIZFOLDERS	Is the number of folder to set (input; 0 by default).
Character PATHNAMES(SIZFOLDERS)*260	Is an array of SIZFOLDERS elements that contains the folders to place in the list (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.62 SIZE_DS

Use this API to obtain a summary of utilization in the general purpose dynamics data storage arrays as well as in the various model connection and allocation tables contained in dynamics working memory.

Batch command syntax:

```
BAT_SIZE_DS
```

Python syntax:

```
ierr = size_ds()
```

Fortran syntax:

```
CALL SIZE_DS(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.63 SNAP

Use this API to save PSS®E dynamics working memory into a Snapshot File (activity SNAP).

Batch command syntax:

```
BAT_SNAP  STATUS(1)...STATUS(5)  SFILE
```

Python syntax:

```
ierr = snap(status, sfile)
```

Fortran syntax:

```
CALL SNAPNW(STATUS, SFILE, IERR)
```

where:

Integer STATUS(5)	Is a five-element array (input). For each entry, -1 may be specified to indicate that the number of elements of the corresponding array(s) to save in the Snapshot File is 1 through the *next available - 1*. STATUS(1) number of CONs to save (next available-1 by default). STATUS(2) number of STATEs to save (next available-1 by default). STATUS(3) number of VARs to save (next available-1 by default). STATUS(4) number of ICONs to save (next available-1 by default). STATUS(5) number of output channels to save (next available-1 by default).
Character SFILE*260	Is the Snapshot File; '*' to use the most recently accessed Snapshot File in the current execution of PSS®E (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid STATUS value. IERR = 2 error opening SFILE. IERR = 3 error writing to SFILE. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.64 SRRS

Use this API to load a Dynamics Snapshot Raw Data File into PSS®E working memory (activity SRRS). It optionally creates a command file for compiling the CONEC and CONET subroutines associated with the Source Form Snapshot.

Batch command syntax:

```
BAT_SRRS  SRRSFILE  CONECFILE  CONETFILE  COMPILFIL
```

Python syntax:

```
ierr = srrs(srrsfile, conecfile, conetfile, compilfil)
```

Fortran syntax:

```
CALL SRRSAPI(SRRSFILE, CONECFILE, CONETFILE, COMPILFIL, IERR)
```

where:

Character SRRSFILE*260	Is the name of the Dynamics Snapshot Raw Data File (input; no default allowed).
Character CONECFILE*260	Is the name of CONEC file to be used in the compiling command file; blank for no CONEC file (input; blank by default).
Character CONETFILE*260	Is the name of CONET file to be used in the compiling command file; blank for no CONET file (input; blank by default).
Character COMPILFIL*260	Is the name of file containing commands to compile the CONEC and CONET files; blank for none (input; blank by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — error opening SRRSFILE.
	IERR = 2 — file SRRSFILE is not in Dynamics Snapshot Raw Data File format.
	IERR = 3 — snapshot is too big for the current size level of PSS®E.
	IERR = 4 — error reading from SRRSFILE.
	IERR = 5 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.65 STRT

Use this API to initialize a PSS®E dynamic simulation for state-space simulations (i.e., in preparation for activity RUN) and to specify the Channel Output File into which the output channel values are to be recorded during the dynamic simulation (activity STRT).

Batch command syntax:

```
BAT_STRT  OPTION  OUTFILE
```

Python syntax:

```
ierr = strt(option, outfile)
```

Fortran syntax:

```
CALL STRTAPI(OPTION, OUTFILE, IERR)
```

where:

Integer OPTION	Is the network solution convergence monitor option (input; 0 by default) OPTION = 1 automatically print the convergence monitor. OPTION = 0 print the convergence monitor only if it is enabled via the CM interrupt control code.
Character OUTFILE*260	Is the name of the Channel Output File (blank to bypass recording of the output channel values in a Channel Output File) (input; blank by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 generators are not converted. IERR = 2 invalid OPTION value. IERR = 3 Prior initialization modified the loads--pick up original converted case. IERR = 4 error opening OUTFILE. IERR = 5 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.66 SWSLIST

Use this API to list switched shunt models connected to subsystem buses along with their storage locations in the dynamics data arrays.

Batch command syntax:

```
BAT_SWSLIST  SID  ALL  STATUS
```

Python syntax:

```
ierr = swslist(sid, all, status)
```

Fortran syntax:

```
CALL SWSLIST(SID, ALL, STATUS, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default). ALL = 1 process all buses. ALL = 0 process only buses in subsystem SID.
Integer STATUS	Is the active/bypassed flag (input; 0 by default). STATUS = 0 active and bypassed models. STATUS = 1 active models only. STATUS = 2 bypassed models only.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid SID value or subsystem SID is not defined. IERR = 2 invalid ALL value. IERR = 3 invalid STATUS value. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.67 TRIG_VOLT_VIOLATION_CHECK

This API is used to trigger voltage violation (voltage recovery, voltage dip) checks.

Batch command syntax:

```
BAT_ TRIG_VOLT_VIOLATION_CHECK STATUS
```

Python syntax:

```
ierr = trig_volt_violation_check (status)
```

Fortran Syntax:

```
CALL TRIG_VOLT_VIOLATION_CHECK (STATUS, ICODE, IERR)
```

where:

Integer STATUS	Is the value of the option setting.	
	STATUS = 1	trigger voltage violation
	STATUS ≠ 1	disable voltage violation check
Integer ICODE	Indicates the operation mode of the API (input; no default allowed).	
	ICODE ≤ 0	set the option setting to the value specified in STATUS
	ICODE > 0	return the current value of the option setting in STATUS
Integer IERR	Is the error code (output).	
	IERR = 0	no error
	IERR = 1	prerequisite requirements for API not met

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.1.68 WNLIST

Use this API to list wind models connected to subsystem buses along with their storage locations in the dynamics data arrays.

Batch command syntax:

```
BAT_WNLIST  SID  ALL  STATUS
```

Python syntax:

```
ierr = wnlist(sid, all, status)
```

Fortran syntax:

```
CALL WNLIST(SID, ALL, STATUS, IERR)
```

where:

Integer SID	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).
	ALL = 1 process all buses.
	ALL = 0 process only buses in subsystem SID.
Integer STATUS(2)	Is a two-element array (input).
	STATUS(1) the active/bypassed flag (0 by default).
	STATUS(1) = 0 active and bypassed models.
	STATUS(1) = 1 active models only.
	STATUS(1) = 2 bypassed models only.
	STATUS(2) the wind model type (1 by default).
	STATUS(2) = 1 all wind models.
	STATUS(2) = 2 wind generator models.
	STATUS(2) = 3 wind electrical models.
	STATUS(2) = 4 wind mechanical models.
	STATUS(2) = 5 wind pitch models.
	STATUS(2) = 6 wind aerodynamic models.
	STATUS(2) = 7 wind gust models.
	STATUS(2) = 8 wind auxiliary signal models.
Integer IERR	Is the error code (output).
	IERR = 0 no error.
	IERR = 1 invalid SID value or subsystem SID is not defined.
	IERR = 2 invalid ALL value.
	IERR = 3 invalid STATUS value.
	IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.2 Output Channel APIs

4.2.1 General Output Channel Related Functions

4.2.1.1 CHSB

Use this API to specify, on a subsystem basis, the simulation variables to monitor during dynamic simulation runs (activity CHSB).

Batch command syntax:

```
BAT_CHSB  SID  ALL  STATUS(1)...STATUS(6)
```

Python syntax:

```
ierr = chsb(sid, all, status)
```

Fortran syntax:

```
CALL CHSBNW(SID, ALL, STATUS, IERR)
```

where:

Integer SID	Is a valid area, bus, owner, or zone subsystem identifier. Valid subsystem identifiers range from 0 to 11. Area, bus, owner, or zone subsystem SID must have been previously defined (input; 0 by default).	
Integer ALL	Is the *all buses* or specified subsystem flag (input; 1 by default).	
	ALL = 1	process all buses.
	ALL = 0	process only buses in subsystem SID.
Integer STATUS(6)	Is an array of six elements (input).	
	STATUS(1)	is the starting channel index, or -1 for the next available (next available by default).
	STATUS(2)	is the starting VAR index, or -1 for the next available (next available by default).
	STATUS(3)	is the starting ICON index, or -1 for the next available (next available by default).
	STATUS(4)	used to indicate the type of quantities to be placed in output channels (1 by default).
	STATUS(4) = 1	machine, bus, load, branch quantities by bus subsystem.
	STATUS(4) = 2	tie quantity by area subsystem.
	STATUS(4) = 3	tie quantity by zone subsystem.
	STATUS(4) = 4	power totals for an area subsystem.
	STATUS(4) = 5	power totals for a zone subsystem.
	STATUS(4) = 6	power totals for an owner subsystem.
	STATUS(4) = 7	power totals for all buses.
	STATUS(4) = 8	machine angle statistics.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

STATUS(5)	quantity code (1 by default).

If STATUS(4) = 1:

STATUS(5) = 1	ANGLE, machine relative rotor angle (degrees).
STATUS(5) = 2	PELEC, machine electrical power (pu on SBASE).
STATUS(5) = 3	QELEC, machine reactive power.
STATUS(5) = 4	ETERM, machine terminal voltage (pu).
STATUS(5) = 5	EFD, generator main field voltage (pu).
STATUS(5) = 6	PMECH, turbine mechanical power (pu on MBASE).
STATUS(5) = 7	SPEED, machine speed deviation from nominal (pu).
STATUS(5) = 8	XADIFD, machine field current (pu).
STATUS(5) = 9	ECOMP, voltage regulator compensated voltage (pu).
STATUS(5) = 10	VOTHSG, stabilizer output signal (pu).
STATUS(5) = 11	VREF, voltage regulator voltage setpoint (pu).
STATUS(5) = 12	BSFREQ, bus pu frequency deviations.
STATUS(5) = 13	VOLT, bus pu voltages (complex).
STATUS(5) = 14	voltage and angle
STATUS(5) = 15	flow (P).
STATUS(5) = 16	flow (P and Q).
STATUS(5) = 17	flow (MVA).
STATUS(5) = 18	apparent impedance (R and X).
STATUS(5) = 21	ITERM.
STATUS(5) = 22	machine apparent impedance
STATUS(5) = 23	VUEL, minimum excitation limiter output signal (pu).
STATUS(5) = 24	VOEL, maximum excitation limiter output signal (pu).
STATUS(5) = 25	PLOAD.
STATUS(5) = 26	QLOAD.
STATUS(5) = 27	GREF, turbine governor reference.
STATUS(5) = 28	LCREF, turbine load control reference.
STATUS(5) = 29	WVLCTY, wind velocity (m/s).
STATUS(5) = 30	WTRBSP, wind turbine rotor speed deviation (pu).
STATUS(5) = 31	WPITCH, pitch angle (degrees).
STATUS(5) = 32	WAEROT, aerodynamic torque (pu on MBASE).
STATUS(5) = 33	WROTRV, rotor voltage (pu on MBASE).
STATUS(5) = 34	WROTRI, rotor current (pu on MBASE).
STATUS(5) = 35	WPCMND, active power command from wind control (pu on MBASE).
STATUS(5) = 36	WQCMND, reactive power command from wind control (pu on MBASE).
STATUS(5) = 37	WAUXSG, output of wind auxiliary control (pu on MBASE).

If STATUS(4) = 4, 5, 6, 7 or 8.
STATUS(5) is ignored.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

If STATUS(4) = 4, 5, 6, 7 or 8.
STATUS(5) is ignored.

STATUS(6)	out-of-service equipment option (0 by default).
STATUS(6) = 0	omit out-of-service equipment.
STATUS(6) = 1	include out-of-service equipment.

Integer IERR — Is the error code (output).

IERR = 0	no error occurred.
IERR = 1	invalid SID value or subsystem SID is not defined.
IERR = 2	invalid ALL value.
IERR = 3	invalid STATUS value.
IERR = 4	have just assigned channel n, which is the highest channel number.
IERR = 5	trying to assign locations beyond the end of the VAR or ICON array.
IERR = 6	machine angle statistics are already assigned to channels.
IERR = 7	n monitoring models specified--pointer table full.
IERR = 8	next channel is greater than the maximum allowed channel number.
IERR = 9	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.2.1.2 CHANGE_CHANNEL

Use this API to change the channel address and the channel identifier of a specified output channel.

Batch command syntax:

```
BAT_CHANGE_CHANNEL  CHNARG  ADRARG  IDNARG
```

Python syntax:

```
ierr = change_channel(chnarg,adrarg, idnarg)
```

Fortran syntax:

```
CALL CHANGE_CHANNEL (CHNARG, ADRARG, IDNARG, IERR)
```

where:

Integer CHNARG	Is channel number for which the channel address and/or channel identifier has to be changed (input; no default allowed)
Integer ADRARG	Is channel address (input; unchanged by default).
Character IDNARG*32	Is the channel identifier (input; unchanged by default).
Integer IERR	Is the error code (output). IERR = 0 — no error. IERR = 1 — invalid channel number. IERR = 2 — channel number is at or beyond the next available channel. IERR = 3 — invalid channel address. IERR = 4 — prerequisite requirements for API are not met. IERR = -1 — warnings – one or more of: • channel number is larger than NCHAN

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.2.1.3 CRT_PLOT_CHANNEL

Use this API to assign one of the main simulation output channels to a designated CRT plot channel.

Batch command syntax:

```
BAT_CRT_PLOT_CHANNEL  CRTCHAN  CHANNEL  CMIN  CMAX
```

Python syntax:

```
ierr = crt_plot_channel (crtchan, channel, cmin, cmax)
```

Fortran syntax:

```
CALL CRT_PLOT_CHANNEL(CRTCHAN, CHANNEL, CMIN, CMAX, IERR)
```

where:

Integer CRTCHAN	Is the CRT plot channel number (1 through 6) (input; no default allowed).
Integer CHANNEL	Is the number of the main simulation output channel to be assigned to CRT channel number CRTCHAN (must be between 1 and the number of output channels allowed at the present size level of PSS®E), or 0 to disable CRT plot channel CRTCHAN (input; unchanged by default).
Real CMIN	Is the minimum plotting scale value for this CRT plot channel (input; unchanged by default).
Real CMAX	Is the maximum plotting scale value for this CRT plot channel (input; unchanged by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error occurred.
	IERR > 0 — error occurred.
	IERR = 1 — invalid CRTCHAN value.
	IERR = 2 — invalid CHANNEL value.
	IERR = 3 — CMIN is not less than CMAX.
	IERR = 4 — prerequisite requirements for API are not met.
	IERR = -1 — warnings – one or more of:
	• CHANNEL is larger than NCHAN.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.2.1.4 DELETE_ALL_PLOT_CHANNELS

Use this API to delete all the plot channels in the working case. Use of this API will reset the number of plot channels to zero, and set the next channel to 1.

Batch command syntax:

```
BAT_DELETE_ALL_PLOT_CHANNELS
```

Python syntax:

```
ierr = delete_all_plot_channels()
```

Fortran syntax:

```
CALL_DELETE_ALL_PLOT_CHANNELS(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.2.1.5 DMPC

Use this API to replicate the output channel specifications from dynamics working memory in the form of a Response File suitable for transferring them to another Snapshot that models essentially the same system (activity DMPC).

Batch command syntax:

```
BAT_DMPC  OPTIONS(1)...OPTIONS(4)  OUT  OFILE
```

Python syntax:

```
ierr = dmpc(options, out, ofile)
```

Fortran syntax:

```
CALL DMPCAPI(OPTIONS, OUT, OFILE, IERR)
```

where:

Integer OPTIONS(4)	Is an array of four elements (input).	
	OPTIONS(1)	is the channel selection code (1 by default).
	OPTIONS(1) = 0	skip channels containing STATEs or VARs not associated with channel monitoring models.
	OPTIONS(1) = 1	include all channels.
	OPTIONS(2)	is the starting channel code (-1 by default).
	OPTIONS(2) < 0	start at next available channel.
	OPTIONS(2) = 0	keep existing channel numbers.
	OPTIONS(2) > 0	starting channel number.
	OPTIONS(3)	is the starting VAR code for channel monitoring models (-1 by default).
	OPTIONS(3) < 0	start at next available VAR.
	OPTIONS(3) = 0	keep existing VAR indices.
	OPTIONS(3) > 0	starting VAR index.
	OPTIONS(4)	is the starting ICON code for channel monitoring models (-1 by default).
	OPTIONS(4) < 0	start at next available ICON.
	OPTIONS(4) = 0	keep existing ICON indices.
	OPTIONS(4) > 0	starting ICON index.
Integer OUT	Is the output destination selection (input; 1 by default).	
	OUT = 0	direct output to a data file specified by OFILE.
	OUT = 1	direct output to the progress window.
Character OFILE*260	Is the name of the file to which data records are to be written (input; no default allowed when OUT is 0).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid OPTIONS(1) value.
	IERR = 2	starting channel number is greater than the largest channel number allowed.
	IERR = 3	starting VAR index is greater than the largest VAR index allowed.
	IERR = 4	starting ICON index is greater than the largest ICON index allowed.
	IERR = 5	invalid OUT value.
	IERR = 6	error opening file OFILE.
	IERR = 7	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.2.1.6 LIST_CHANNEL_MODELS

Use this API to tabulate the output channel monitoring models referenced in the dynamic model.

Batch command syntax:

```
BAT_LIST_CHANNEL_MODELS  STATUS
```

Python syntax:

```
ierr = list_channel_models(status)
```

Fortran syntax:

```
CALL LIST_CHANNEL_MODELS(STATUS, IERR)
```

where:

Integer STATUS	Is the option to indicate the category of channel monitoring models to be listed (input; 0 by default). STATUS = 0 all output channel monitoring models. STATUS = 1 only models monitoring equipment that exists in the power flow working case. STATUS = 2 only models monitoring equipment that is not present in the power flow working case.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR = 1 invalid STATUS value. IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.2.1.7 REMOVE_UNUSED_CHAN_MODELS

Use this API to delete from the output channel monitoring model table those model references indicating equipment that is not present in the power flow working case.

Batch command syntax:

```
BAT_REMOVE_UNUSED_CHAN_MODELS
```

Python syntax:

```
ierr = remove_unused_chan_models()
```

Fortran syntax:

```
CALL REMOVE_UNUSED_CHAN_MODELS(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.2.2 Assign Machine Quantities to Output Channels

4.2.2.1 MACHINE_APP_R_X_CHANNEL

Use this API to add a pair of output channels containing the apparent impedance of the system, as seen from the terminals of a specified machine, along with a corresponding call to the GENTMZ monitoring model.

Batch command syntax:

```
BAT_MACHINE_APP_R_X_CHANNEL  STATUS(1)...STATUS(4)  'ID'  'IDENT(1)'
'IDENT(2)'
```

Python syntax:

```
ierr = machine_app_r_x_channel(status, id, ident)
```

Fortran syntax:

```
CALL MACHINE_APP_R_X_CHANNEL(STATUS, ID, IDENT, IERR)
```

where:

Integer STATUS(4)	Is an array of four elements (input).	
	STATUS(1)	is the starting channel index, or -1 for the next available (next available by default).
	STATUS(2)	is the starting VAR index, or -1 for the next available (next available by default).
	STATUS(3)	is the starting ICON index, or -1 for the next available (next available by default).
	STATUS(4)	is the number of the bus to which the machine for which the apparent impedance is to be placed in output channels is connected (no default allowed).
Character ID*2	Is the machine identifier (input; '1' by default).	
Character IDENT(2)*32	Is an array of two identifiers to be assigned to the two output channels. If a blank channel identifier is specified, the API generates an appropriate channel identifier (input; blank by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid STATUS value.
	IERR = 2	starting channel number is greater than the largest channel number allowed.
	IERR = 3	starting VAR index is greater than the largest VAR index allowed.
	IERR = 4	starting ICON index is greater than the largest ICON index allowed.
	IERR = 5	model GENTMZ needs 2 VARs but the last one exceeds the largest VAR index allowed.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IERR = 6	model GENTMZ needs 2 ICONs but the last one exceeds the largest ICON index allowed.
IERR = 7	the maximum number of channel monitoring models has already been specified.
IERR = 8	model GENTMZ needs 2 channels but the last one exceeds the largest channel number allowed.
IERR = 9	bus not found.
IERR = 10	machine not found.
IERR = 11	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.2.2.2 MACHINE_ARRAY_CHANNEL

Use this API to add an output channel containing a plant related model variable of a designated type for a specified machine.

Batch command syntax:

```
BAT_MACHINE_ARRAY_CHANNEL  STATUS(1)...STATUS(3)  'ID'  'IDENT'
```

Python syntax:

```
ierr = machine_array_channel(status, id, ident)
```

Fortran syntax:

```
CALL MACHINE_ARRAY_CHANNEL(STATUS, ID, IDENT, IERR)
```

where:

Integer STATUS(3)	Is an array of three elements (input).	
	STATUS(1)	is the starting channel index, or -1 for the next available (next available by default).
	STATUS(2)	used to indicate the quantity to be placed in an output channel (1 by default).
	STATUS(2) = 1	ANGLE, machine relative rotor angle (degrees).
	STATUS(2) = 2	PELEC, machine electrical power (pu on SBASE)
	STATUS(2) = 3	QELEC, machine reactive power.
	STATUS(2) = 4	ETERM, machine terminal voltage (pu).
	STATUS(2) = 5	EFD, generator main field voltage (pu).
	STATUS(2) = 6	PMECH, turbine mechanical power (pu on MBASE).
	STATUS(2) = 7	SPEED, machine speed deviation from nominal (pu).
	STATUS(2) = 8	XADIFD, machine field current (pu).
	STATUS(2) = 9	ECOMP, voltage regulator compensated voltage (pu).
	STATUS(2) = 10	VOTHSG, stabilizer output signal (pu).
	STATUS(2) = 11	VREF, voltage regulator voltage setpoint (pu).
	STATUS(2) = 12	VUEL, minimum excitation limiter output signal (pu).
	STATUS(2) = 13	VOEL, maximum excitation limiter output signal (pu).
	STATUS(2) = 14	GREF, turbine governor reference.
	STATUS(2) = 15	LCREF, turbine load control reference.
	STATUS(2) = 16	WVLCTY, wind velocity (m/s).
	STATUS(2) = 17	WTRBSP, wind turbine rotor speed deviation (pu).
	STATUS(2) = 18	WPITCH, pitch angle (degrees).
	STATUS(2) = 19	WAEROT, aerodynamic torque (pu on MBASE).
	STATUS(2) = 20	WROTRV, rotor voltage (pu on MBASE).
	STATUS(2) = 21	WROTRI, rotor current (pu on MBASE).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	STATUS(2) = 22	WPCMND, active power command from wind control (pu on MBASE).
	STATUS(2) = 23	WQCMND, reactive power command from wind control (pu on MBASE).
	STATUS(2) = 24	WAUXSG, output of wind auxiliary control (pu on MBASE).
	STATUS(3)	is the number of the bus to which the machine for which the quantity is to be placed in an output channel is connected (no default allowed).
Character ID*2	Is the machine identifier (input; ’1’ by default).	
Character IDENT*32	Is the identifier to be assigned to the output channel. If a blank channel identifier is specified, the API generates an appropriate channel identifier (input; blank by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid STATUS value.
	IERR = 2	starting channel number is greater than the largest channel number allowed.
	IERR = 3	bus not found.
	IERR = 4	machine not found.
	IERR = 5	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.2.2.3 MACHINE_ITERM_CHANNEL

Use this API to add an output channel containing the terminal current of a specified machine, along with a corresponding call to the GENTMC monitoring model.

Batch command syntax:

```
BAT_MACHINE_ITERM_CHANNEL  STATUS(1)...STATUS(4)  'ID'  'IDENT'
```

Python syntax:

```
ierr = machine_iterm_channel(status, id, ident)
```

Fortran syntax:

```
CALL MACHINE_ITERM_CHANNEL(STATUS, ID, IDENT, IERR)
```

where:

Integer STATUS(4)	Is an array of four elements (input).	
	STATUS(1)	is the starting channel index, or -1 for the next available (next available by default).
	STATUS(2)	is the starting VAR index, or -1 for the next available (next available by default).
	STATUS(3)	is the starting ICON index, or -1 for the next available (next available by default).
	STATUS(4)	is the number of the bus to which the machine for which the terminal current is to be placed in an output channel is connected (no default allowed).
Character ID*2	Is the machine identifier (input; '1' by default).	
Character IDENT*32	Is the identifier to be assigned to the output channel. If a blank channel identifier is specified, the API generates an appropriate channel identifier (input; blank by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid STATUS value.
	IERR = 2	starting channel number is greater than the largest channel number allowed.
	IERR = 3	starting VAR index is greater than the largest VAR index allowed.
	IERR = 4	starting ICON index is greater than the largest ICON index allowed.
	IERR = 5	model GENTMC needs 2 VARs but the last one exceeds the largest VAR index allowed.
	IERR = 6	model GENTMC needs 2 ICONs but the last one exceeds the largest ICON index allowed.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IERR = 7	the maximum number of channel monitoring models has already been specified.
IERR = 8	bus not found.
IERR = 9	machine not found.
IERR = 10	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.2.3 Assign Branch Quantities to Output Channels

4.2.3.1 BRANCH_APP_R_X_CHANNEL

Use this API to add a pair of output channels containing the apparent impedance, as seen at the from bus of a specified branch, along with a corresponding call to the RELAY2 monitoring model.

Batch command syntax:

```
BAT_BRANCH_APP_R_X_CHANNEL  STATUS(1)...STATUS(5)  'ID'  'IDENT(1)'
'IDENT(2)'
```

Python syntax:

```
ierr = branch_app_r_x_channel(status, id, ident)
```

Fortran syntax:

```
CALL BRANCH_APP_R_X_CHANNEL(STATUS, ID, IDENT, IERR)
```

where:

Integer STATUS(5)	Is an array of five elements (input).	
	STATUS(1)	is the starting channel index, or -1 for the next available (next available by default).
	STATUS(2)	is the starting VAR index, or -1 for the next available (next available by default).
	STATUS(3)	is the starting ICON index, or -1 for the next available (next available by default).
	STATUS(4)	is the number of the from bus of the branch for which apparent impedance is to be placed in output channels (no default allowed).
	STATUS(5)	is the number of the to bus of the branch for which apparent impedance is to be placed in output channels (no default allowed).
Character ID*2	Is the circuit identifier (input; '1' by default).	
Character IDENT(2)*32	Is an array of two identifiers to be assigned to the two output channels. If a blank channel identifier is specified, the API generates an appropriate channel identifier (input; blank by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid STATUS value.
	IERR = 2	starting channel number is greater than the largest channel number allowed.
	IERR = 3	starting VAR index is greater than the largest VAR index allowed.
	IERR = 4	starting ICON index is greater than the largest ICON index allowed.
	IERR = 5	model RELAY2 needs 3 ICONs but the last one exceeds the largest ICON index allowed.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IERR = 6	the maximum number of channel monitoring models has already been specified.
IERR = 7	model RELAY2 needs 2 VARs but the last one exceeds the largest VAR index allowed.
IERR = 8	model RELAY2 needs 2 channels but the last one exceeds the largest channel number allowed.
IERR = 9	bus not found.
IERR = 10	branch not found.
IERR = 11	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.2.3.2 BRANCH_MVA_CHANNEL

Use this API to add an output channel containing the MVA flow at the from bus of a specified branch, along with a corresponding call to the FLOW1 monitoring model.

Batch command syntax:

```
BAT_BRANCH_MVA_CHANNEL  STATUS(1)...STATUS(5)  'ID'  'IDENT'
```

Python syntax:

```
ierr = branch_mva_channel(status, id, ident)
```

Fortran syntax:

```
CALL BRANCH_MVA_CHANNEL(STATUS, ID, IDENT, IERR)
```

where:

Integer STATUS(5)	Is an array of five elements (input).	
	STATUS(1)	is the starting channel index, or -1 for the next available (next available by default).
	STATUS(2)	is the starting VAR index, or -1 for the next available (next available by default).
	STATUS(3)	is the starting ICON index, or -1 for the next available (next available by default).
	STATUS(4)	is the number of the from bus of the branch for which MVA flow is to be placed in an output channel (no default allowed).
	STATUS(5)	is the number of the to bus of the branch for which MVA is to be placed in an output channel (no default allowed).
Character ID*2	Is the circuit identifier (input; '1' by default).	
Character IDENT*32	Is the identifier to be assigned to the output channel. If a blank channel identifier is specified, the API generates an appropriate channel identifier (input; blank by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid STATUS value.
	IERR = 2	starting channel number is greater than the largest channel number allowed.
	IERR = 3	starting VAR index is greater than the largest VAR index allowed.
	IERR = 4	starting ICON index is greater than the largest ICON index allowed.
	IERR = 5	model FLOW1 needs 3 ICONs but the last one exceeds the largest ICON index allowed.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IERR = 6	the maximum number of channel monitoring models has already been specified.
IERR = 7	bus not found.
IERR = 8	branch not found.
IERR = 9	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.2.3.3 BRANCH_P_AND_Q_CHANNEL

Use this API to add a pair of output channels containing the active and reactive power flow at the from bus of a specified branch, along with a corresponding call to the FLOW1 monitoring model.

Batch command syntax:

```
BAT_BRANCH_P_AND_Q_CHANNEL  STATUS(1)...STATUS(5)  'ID'  'IDENT(1)'
'IDENT(2)'
```

Python syntax:

```
ierr = branch_p_and_q_channel(status, id, ident)
```

Fortran syntax:

```
CALL BRANCH_P_AND_Q_CHANNEL(STATUS, ID, IDENT, IERR)
```

where:

Integer STATUS(5)	Is an array of five elements (input).	
	STATUS(1)	is the starting channel index, or -1 for the next available (next available by default).
	STATUS(2)	is the starting VAR index, or -1 for the next available (next available by default).
	STATUS(3)	is the starting ICON index, or -1 for the next available (next available by default).
	STATUS(4)	is the number of the from bus of the branch for which active and reactive power flows are to be placed in output channels (no default allowed).
	STATUS(5)	is the number of the to bus of the branch for which active and reactive power flows are to be placed in output channels (no default allowed).
Character ID*2	Is the circuit identifier (input; '1' by default).	
Character IDENT(2)*32	Is an array of two identifiers to be assigned to the two output channels. If a blank channel identifier is specified, the API generates an appropriate channel identifier (input; blank by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid STATUS value.
	IERR = 2	starting channel number is greater than the largest channel number allowed.
	IERR = 3	starting VAR index is greater than the largest VAR index allowed.
	IERR = 4	starting ICON index is greater than the largest ICON index allowed.
	IERR = 5	model FLOW1 needs 3 ICONs but the last one exceeds the largest ICON index allowed.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IERR = 6	the maximum number of channel monitoring models has already been specified.
IERR = 7	model FLOW1 needs 2 VARs but the last one exceeds the largest VAR index allowed.
IERR = 8	model FLOW1 needs 2 channels but the last one exceeds the largest channel number allowed.
IERR = 9	bus not found.
IERR = 10	branch not found.
IERR = 11	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.2.3.4 BRANCH_P_CHANNEL

Use this API to add an output channel containing the active power flow at the from bus of a specified branch, along with a corresponding call to the FLOW1 monitoring model.

Batch command syntax:

```
BAT_BRANCH_P_CHANNEL  STATUS(1)...STATUS(5)  'ID'  'IDENT'
```

Python syntax:

```
ierr = branch_p_channel(status, id, ident)
```

Fortran syntax:

```
CALL BRANCH_P_CHANNEL(STATUS, ID, IDENT, IERR)
```

where:

Integer STATUS(5)	Is an array of five elements (input).	
	STATUS(1)	is the starting channel index, or -1 for the next available (next available by default).
	STATUS(2)	is the starting VAR index, or -1 for the next available (next available by default).
	STATUS(3)	is the starting ICON index, or -1 for the next available (next available by default).
	STATUS(4)	is the number of the from bus of the branch for which active power flow is to be placed in an output channel (no default allowed).
	STATUS(5)	is the number of the to bus of the branch for which active power flow is to be placed in an output channel (no default allowed).
Character ID*2	Is the circuit identifier (input; '1' by default).	
Character IDENT*32	Is the identifier to be assigned to the output channel. If a blank channel identifier is specified, the API generates an appropriate channel identifier (input; blank by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid STATUS value.
	IERR = 2	starting channel number is greater than the largest channel number allowed.
	IERR = 3	starting VAR index is greater than the largest VAR index allowed.
	IERR = 4	starting ICON index is greater than the largest ICON index allowed.
	IERR = 5	model FLOW1 needs 3 ICONs but the last one exceeds the largest ICON index allowed.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IERR = 6	the maximum number of channel monitoring models has already been specified.
IERR = 7	bus not found.
IERR = 8	branch not found.
IERR = 9	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.2.3.5 THREE_WND_APP_R_X_CHANNEL

Use this API to add a pair of output channels containing the apparent impedance as seen at the from bus of a specified three-winding transformer, along with a corresponding call to the RELAY3 monitoring model.

Batch command syntax:

```
BAT_THREE_WND_APP_R_X_CHANNEL  STATUS(1)...STATUS(6)  'ID'
'IDENT(1)'  'IDENT(2)'
```

Python syntax:

```
ierr = three_wnd_app_r_x_channel(status, id, ident)
```

Fortran syntax:

```
CALL THREE_WINDING_APP_R_X_CHANNEL(STATUS, ID, IDENT, IERR)
```

where:

Integer STATUS(6)	Is an array of six elements (input).	
	STATUS(1)	is the starting channel index, or -1 for the next available (next available by default).
	STATUS(2)	is the starting VAR index, or -1 for the next available (next available by default).
	STATUS(3)	is the starting ICON index, or -1 for the next available (next available by default).
	STATUS(4)	is the number of the from bus of the three-winding transformer for which apparent impedance is to be placed in output channels (no default allowed).
	STATUS(5)	is the number of one of the to buses of the three-winding transformer for which apparent impedance is to be placed in output channels (no default allowed).
	STATUS(6)	is the number of the third bus of the three-winding transformer for which apparent impedance is to be placed in output channels (no default allowed).
Character ID*2	Is the circuit identifier (input; '1' by default).	
Character IDENT(2)*32	Is an array of two identifiers to be assigned to the two output channels. If a blank channel identifier is specified, the API generates an appropriate channel identifier (input; blank by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid STATUS value.
	IERR = 2	starting channel number is greater than the largest channel number allowed.
	IERR = 3	starting VAR index is greater than the largest VAR index allowed.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IERR = 4	starting ICON index is greater than the largest ICON index allowed.
IERR = 5	model RELAY3 needs 4 ICONs but the last one exceeds the largest ICON index allowed.
IERR = 6	the maximum number of channel monitoring models has already been specified.
IERR = 7	model RELAY3 needs 2 VARs but the last one exceeds the largest VAR index allowed.
IERR = 8	model RELAY3 needs 2 channels but the last one exceeds the largest channel number allowed.
IERR = 9	bus not found.
IERR = 10	branch not found.
IERR = 11	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.2.3.6 THREE_WND_MVA_CHANNEL

Use this API to add an output channel containing the MVA flow at the from bus of a specified three-winding transformer, along with a corresponding call to the FLOW3 monitoring model.

Batch command syntax:

```
BAT_THREE_WND_MVA_CHANNEL  STATUS(1)...STATUS(6)  'ID'  'IDENT'
```

Python syntax:

```
ierr = three_wnd_mva_channel(status, id, ident)
```

Fortran syntax:

```
CALL THREE_WINDING_MVA_CHANNEL(STATUS, ID, IDENT, IERR)
```

where:

Integer STATUS(6)	Is an array of six elements (input).	
	STATUS(1)	is the starting channel index, or -1 for the next available (next available by default).
	STATUS(2)	is the starting VAR index, or -1 for the next available (next available by default).
	STATUS(3)	is the starting ICON index, or -1 for the next available (next available by default).
	STATUS(4)	is the number of the from bus of the three-winding transformer for which MVA flow is to be placed in an output channel (no default allowed).
	STATUS(5)	is the number of one of the to buses of the three-winding transformer for which MVA flow is to be placed in an output channel (no default allowed).
	STATUS(6)	is the number of the third bus of the three-winding transformer for which MVA flow is to be placed in an output channel (no default allowed).
Character ID*2	Is the circuit identifier (input; '1' by default).	
Character IDENT*32	Is the identifier to be assigned to the output channel. If a blank channel identifier is specified, the API generates an appropriate channel identifier (input; blank by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid STATUS value.
	IERR = 2	starting channel number is greater than the largest channel number allowed.
	IERR = 3	starting VAR index is greater than the largest VAR index allowed.
	IERR = 4	starting ICON index is greater than the largest ICON index allowed.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IERR = 5	model FLOW3 needs 4 ICONs but the last one exceeds the largest ICON index allowed.
IERR = 6	the maximum number of channel monitoring models has already been specified.
IERR = 7	bus not found.
IERR = 8	branch not found.
IERR = 9	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.2.3.7 THREE_WND_P_AND_Q_CHANNEL

Use this API to add a pair of output channels containing the active and reactive power flow at the from bus of a specified three-winding transformer, along with a corresponding call to the FLOW3 monitoring model.

Batch command syntax:

```
BAT_THREE_WND_P_AND_Q_CHANNEL  STATUS(1)...STATUS(6)  'ID'
'IDENT(1)'  'IDENT(2)'
```

Python syntax:

```
ierr = three_wnd_p_and_q_channel(status, id, ident)
```

Fortran syntax:

```
CALL THREE_WINDING_P_AND_Q_CHANNEL(STATUS, ID, IDENT, IERR)
```

where:

Integer STATUS(6)	Is an array of six elements (input).	
	STATUS(1)	is the starting channel index, or -1 for the next available (next available by default).
	STATUS(2)	is the starting VAR index, or -1 for the next available (next available by default).
	STATUS(3)	is the starting ICON index, or -1 for the next available (next available by default).
	STATUS(4)	is the number of the from bus of the three-winding transformer for which active and reactive power flows are to be placed in output channels (no default allowed).
	STATUS(5)	is the number of one of the to buses of the three-winding transformer for which active and reactive power flows are to be placed in output channels (no default allowed).
	STATUS(6)	is the number of the third bus of the three-winding transformer for which active and reactive power flows are to be placed in output channels (no default allowed).
Character ID*2	Is the circuit identifier (input; '1' by default).	
Character IDENT(2)*32	Is an array of two identifiers to be assigned to the two output channels. If a blank channel identifier is specified, the API generates an appropriate channel identifier (input; blank by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid STATUS value.
	IERR = 2	starting channel number is greater than the largest channel number allowed.
	IERR = 3	starting VAR index is greater than the largest VAR index allowed.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IERR = 4	starting ICON index is greater than the largest ICON index allowed.
IERR = 5	model FLOW3 needs 4 ICONs but the last one exceeds the largest ICON index allowed.
IERR = 6	the maximum number of channel monitoring models has already been specified.
IERR = 7	model FLOW3 needs 2 VARs but the last one exceeds the largest VAR index allowed.
IERR = 8	model FLOW3 needs 2 channels but the last one exceeds the largest channel number allowed.
IERR = 9	bus not found.
IERR = 10	branch not found.
IERR = 11	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.2.3.8 THREE_WND_P_CHANNEL

Use this API to add an output channel containing the active power flow at the from bus of a specified three-winding transformer, along with a corresponding call to the FLOW3 monitoring model.

Batch command syntax:

```
BAT_THREE_WND_P_CHANNEL  STATUS(1)...STATUS(6)  'ID'  'IDENT'
```

Python syntax:

```
ierr = three_wnd_p_channel(status, id, ident)
```

Fortran syntax:

```
CALL THREE_WINDING_P_CHANNEL(STATUS, ID, IDENT, IERR)
```

where:

Integer STATUS(6)	Is an array of six elements (input).	
	STATUS(1)	is the starting channel index, or -1 for the next available (next available by default).
	STATUS(2)	is the starting VAR index, or -1 for the next available (next available by default).
	STATUS(3)	is the starting ICON index, or -1 for the next available (next available by default).
	STATUS(4)	is the number of the from bus of the three-winding transformer for which active power flow is to be placed in an output channel (no default allowed).
	STATUS(5)	is the number of one of the to buses of the three-winding transformer for which active power flow is to be placed in an output channel (no default allowed).
	STATUS(6)	is the number of the third bus of the three-winding transformer for which active power flow is to be placed in an output channel (no default allowed).
Character ID*2	Is the circuit identifier (input; '1' by default).	
Character IDENT*32	Is the identifier to be assigned to the output channel. If a blank channel identifier is specified, the API generates an appropriate channel identifier (input; blank by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid STATUS value.
	IERR = 2	starting channel number is greater than the largest channel number allowed.
	IERR = 3	starting VAR index is greater than the largest VAR index allowed.
	IERR = 4	starting ICON index is greater than the largest ICON index allowed.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IERR = 5	model FLOW3 needs 4 ICONs but the last one exceeds the largest ICON index allowed.
IERR = 6	the maximum number of channel monitoring models has already been specified.
IERR = 7	bus not found.
IERR = 8	branch not found.
IERR = 9	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.2.4 Assign Other Quantities to Output Channels

4.2.4.1 BUS_FREQUENCY_CHANNEL

Use this API to add an output channel containing the per unit frequency deviation at a specified bus.

Batch command syntax:

```
BAT_BUS_FREQUENCY_CHANNEL  STATUS(1)  STATUS(2)  'IDENT'
```

Python syntax:

```
ierr = bus_frequency_channel(status, ident)
```

Fortran syntax:

```
CALL BUS_FREQUENCY_CHANNEL(STATUS, IDENT, IERR)
```

where:

Integer STATUS(2)	Is an array of two elements (input).	
	STATUS(1)	is the starting channel index, or -1 for the next available (next available by default).
	STATUS(2)	is the number of the bus for which frequency deviation is to be placed in an output channel is connected (no default allowed).
Character IDENT*32	Is the identifier to be assigned to the output channel. If a blank channel identifier is specified, the API generates an appropriate channel identifier (input; blank by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid STATUS value.
	IERR = 2	starting channel number is greater than the largest channel number allowed.
	IERR = 3	bus not found.
	IERR = 4	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.2.4.2 LOAD_ARRAY_CHANNEL

Use this API to add an output channel containing the active or reactive component of a designated load.

Batch command syntax:

```
BAT_LOAD_ARRAY_CHANNEL  STATUS(1)...STATUS(3)  'ID'  'IDENT'
```

Python syntax:

```
ierr = load_array_channel(status, id, ident)
```

Fortran syntax:

```
CALL LOAD_ARRAY_CHANNEL(STATUS, ID, IDENT, IERR)
```

where:

Integer STATUS(3)	Is an array of three elements (input).	
	STATUS(1)	is the starting channel index, or -1 for the next available (next available by default).
	STATUS(2)	used to indicate the quantity to be placed in an output channel (1 by default).
	STATUS(2) = 1	PLOAD.
	STATUS(2) = 2	QLOAD.
	STATUS(3)	is the number of the bus to which the load for which quantity is to be placed in an output channel is connected (no default allowed).
Character ID*2	Is the load identifier (input; '1' by default).	
Character IDENT*32	Is the identifier to be assigned to the output channel. If a blank channel identifier is specified, the API generates an appropriate channel identifier (input; blank by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid STATUS value.
	IERR = 2	starting channel number is greater than the largest channel number allowed.
	IERR = 3	bus not found.
	IERR = 4	load not found.
	IERR = 5	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.2.4.3 STATE_CHANNEL

Use this API to add an output channel containing the value of a specified STATE.

Batch command syntax:

```
BAT_STATE_CHANNEL  STATUS(1)  STATUS(2)  'IDENT'
```

Python syntax:

```
ierr = state_channel(status, ident)
```

Fortran syntax:

```
CALL STATE_CHANNEL(STATUS, IDENT, IERR)
```

where:

Integer STATUS(2)	Is an array of two elements (input). STATUS(1) is the starting channel index, or -1 for the next available (next available by default). STATUS(2) is the index of the STATE for which the value is to be placed in an output channel (no default allowed).
Character IDENT*32	Is the identifier to be assigned to the output channel. If a blank channel identifier is specified, the API generates an appropriate channel identifier (input; blank by default).
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid STATUS value. IERR = 2 starting channel number is greater than the largest channel number allowed. IERR = 3 invalid STATE index. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.2.4.4 VAR_CHANNEL

Use this API to add an output channel containing the value of a specified VAR.

Batch command syntax:

```
BAT_VAR_CHANNEL  STATUS(1)  STATUS(2)  'IDENT'
```

Python syntax:

```
ierr = var_channel(status, ident)
```

Fortran syntax:

```
CALL VAR_CHANNEL(STATUS, IDENT, IERR)
```

where:

Integer STATUS(2)	Is an array of two elements (input).	
	STATUS(1)	is the starting channel index, or -1 for the next available (next available by default).
	STATUS(2)	is the index of the VAR for which the value is to be placed in an output channel (no default allowed).
Character IDENT*32	Is the identifier to be assigned to the output channel. If a blank channel identifier is specified, the API generates an appropriate channel identifier (input; blank by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid STATUS value.
	IERR = 2	starting channel number is greater than the largest channel number allowed.
	IERR = 3	invalid VAR index.
	IERR = 4	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.2.4.5 VOLTAGE_CHANNEL

Use this API to add an output channel containing the voltage magnitude in per unit of a specified bus, along with a corresponding call to the VOLMAG monitoring model.

Batch command syntax:

```
BAT_VOLTAGE_CHANNEL  STATUS(1)...STATUS(4)  'IDENT'
```

Python syntax:

```
ierr = voltage_channel(status, ident)
```

Fortran syntax:

```
CALL VOLTAGE_CHANNEL(STATUS, IDENT, IERR)
```

where:

Integer STATUS(4)	Is an array of four elements (input).	
	STATUS(1)	is the starting channel index, or -1 for the next available (next available by default).
	STATUS(2)	is the starting VAR index, or -1 for the next available (next available by default).
	STATUS(3)	is the starting ICON index, or -1 for the next available (next available by default).
	STATUS(4)	is the number of the bus for which voltage magnitude is to be placed in an output channel (no default allowed).
Character IDENT*32	Is the identifier to be assigned to the output channel. If a blank channel identifier is specified, the API generates an appropriate channel identifier (input; blank by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid STATUS value.
	IERR = 2	starting channel number is greater than the largest channel number allowed.
	IERR = 3	starting VAR index is greater than the largest VAR index allowed.
	IERR = 4	starting ICON index is greater than the largest ICON index allowed.
	IERR = 5	the maximum number of channel monitoring models has already been specified.
	IERR = 6	bus not found.
	IERR = 7	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.2.4.6 VOLTAGE_AND_ANGLE_CHANNEL

Use this API to add a pair of output channels containing the voltage magnitude in per unit and phase angle in degrees of a specified bus, along with a corresponding call to the VOLMAG monitoring model.

Batch command syntax:

```
BAT_VOLTAGE_AND_ANGLE_CHANNEL  STATUS(1)...STATUS(4)  'IDENT(1)'
'IDENT(2)'
```

Python syntax:

```
ierr = voltage_and_angle_channel(status, ident)
```

Fortran syntax:

```
CALL VOLTAGE_AND_ANGLE_CHANNEL(STATUS, IDENT, IERR)
```

where:

Integer STATUS(4)	Is an array of four elements (input).	
	STATUS(1)	is the starting channel index, or -1 for the next available (next available by default).
	STATUS(2)	is the starting VAR index, or -1 for the next available (next available by default).
	STATUS(3)	is the starting ICON index, or -1 for the next available (next available by default).
	STATUS(4)	is the number of the bus for which voltage magnitude and phase angle are to be placed in output channels (no default allowed).
Character IDENT(2)*32	Is an array of two identifiers to be assigned to the two output channels. If a blank channel identifier is specified, the API generates an appropriate channel identifier (input; blank by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid STATUS value.
	IERR = 2	starting channel number is greater than the largest channel number allowed.
	IERR = 3	starting VAR index is greater than the largest VAR index allowed.
	IERR = 4	starting ICON index is greater than the largest ICON index allowed.
	IERR = 5	the maximum number of channel monitoring models has already been specified.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IERR = 6	bus not found.
IERR = 7	model VOLMAG needs 2 VARs but the last one exceeds the largest VAR index allowed.
IERR = 8	model VOLMAG needs 2 channels but the last one exceeds the largest channel number allowed.
IERR = 9	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.3 Plant Related Model APIs

4.3.1 ADD_PLANT_MODEL

Use this API to add a plant related model of a designated type to a specified machine. If a model of the designated type already exists at the machine, it is replaced.

Batch command syntax:

```
BAT_ADD_PLANT_MODEL  IBUS  'ID'  MTYPE  'NAME'  IBUS2 'ID2'  NICN
IDATA  CHDATA  NCON  RDATA
```

Python syntax:

```
ierr = add_plant_model(ibus, id, mtype, name, ibus2, id2, nicn,
idata, chdata, ncon, rdata)
```

Fortran syntax:

```
CALL ADD_PLANT_MODEL(IBUS, ID, MTYPE, NAME, IBUS2, ID2, NICN, IDATA,
CHDATA, NCON, RDATA, IERR)
```

where:

Integer IBUS	Is the plant bus number (input; no default allowed).
Character ID*2	Is the machine identifier (input; blank by default).
Integer MTYPE	Is the model type (input; 1 by default). MTYPE = 1 generator model. MTYPE = 2 compensator model. MTYPE = 3 stabilizer model. MTYPE = 4 minimum excitation limiter model. MTYPE = 5 maximum excitation limiter model. MTYPE = 6 excitation system model. MTYPE = 7 turbine-governor model. MTYPE = 8 turbine-load controller model.
Character NAME*16	Is the model name (input; no default allowed).
Integer IBUS2	Is the bus number of the second machine if this is a two machine model, or 0 for a single machine model (input; 0 by default).
Character ID2*2	Is the machine identifier of the second machine if this is a two machine model, or ' ' for a single machine model (input; blank by default).
Integer NICN	Is the number of ICONs required by the model (input; no default allowed).
Integer IDATA(NICN)	Are the integer ICON data (input; 0 by default). When a character value is intended for an ICON value, the corresponding IDATA entry is ignored and the integer ICON is set to 0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character CHDATA(NICN)*2	Are the character ICON data (input; no default allowed for an ICON for which a character value is to be specified). When an integer value is intended for an ICON value, the corresponding CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).
Integer NCON	Is the number of CONs required by the model (input; no default allowed).
Real RDATA(NCON)	Are the CON data (input; 0.0 by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	machine model connection tables full--use pack plant model tables function
IERR = 2	plant model tables are corrupt
IERR = 3	invalid MTYPE value
IERR = 4	unknown model NAME
IERR = 5	model is not consistent with MTYPE
IERR = 6	# CONs specified is not the number expected
IERR = 7	# ICONs specified is not the number expected
IERR = 8	second machine not found where one is required
IERR = 9	same machine specified as the first and second machines
IERR = 10	maximum number of array elements exceeded
IERR = 11	bus not found
IERR = 12	machine not found
IERR = 13	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.3.2 CHANGE_PLMOD_CHRICN

Use this API to change the value of a character ICON (i.e., an element of the CHRICN array) of a designated plant related model at a specified machine. The corresponding integer ICON is set to 0.

Batch command syntax:

```
BAT_CHANGE_PLMOD_CHRICN  IBUS  'ID'  'NAME'  J  'CHDATA'
```

Python syntax:

```
ierr = change_plmod_chricn(ibus, id, name, j, chdata)
```

Fortran syntax:

```
CALL CHANGE_PLMOD_CHRICN(IBUS, ID, NAME, J, CHDATA, IERR)
```

where:

Integer IBUS	Is the plant bus number (input; no default allowed).
Character ID*2	Is the machine identifier (input; blank by default).
Character NAME*16	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change relative to the start of the block of ICONs assigned to this model (1 for ICON(I), 2 for ICON(I+1), etc.) (input; no default allowed).
Character CHDATA*2	Is the new character ICON value (input; blank by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — unknown model NAME.
	IERR = 2 — model NAME is not a plant-related model.
	IERR = 3 — no <model type> model at this machine.
	IERR = 4 — the <model type> model at this machine is aaa not NAME.
	IERR = 5 — no input ICONs used in model NAME.
	IERR = 6 — ICON number outside of the ICON range of this model.
	IERR = 7 — bus not found.
	IERR = 8 — machine not found.
	IERR = 9 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.3.3 CHANGE_PLMOD_CON

Use this API to change the value of a CON of a designated plant related model at a specified machine.

Batch command syntax:

```
BAT_CHANGE_PLMOD_CON  IBUS  'ID'  'NAME'  J  RDATA
```

Python syntax:

```
ierr = change_plmod_con(ibus, id, name, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_PLMOD_CON(IBUS, ID, NAME, J, RDATA, IERR)
```

where:

Integer IBUS	Is the plant bus number (input; no default allowed).
Character ID*2	Is the machine identifier (input; blank by default).
Character NAME*16	Is the model name (input; no default allowed).
Integer J	Is the CON number to change relative to the start of the block of CONs assigned to this model (1 for CON(J), 2 for CON(J+1), etc.) (input; no default allowed).
Real RDATA	Is the new CON value (input; unchanged by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — unknown model NAME.
	IERR = 2 — model NAME is not a plant-related model.
	IERR = 3 — no <model type> model at this machine.
	IERR = 4 — the <model type> model at this machine is aaa not NAME.
	IERR = 5 — no CONs used in model NAME.
	IERR = 6 — CON number outside of the CON range of this model.
	IERR = 7 — bus not found.
	IERR = 8 — machine not found.
	IERR = 9 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.3.4 CHANGE_PLMOD_DATA

This API is obsolete. It has been replaced by the APIs CHANGE_PLMOD_CON, CHANGE_PLMOD_ICON and CHANGE_PLMOD_CHRICN.

Use this API to change the value of an ICON and/or CON of a plant related model of a designated type at a specified machine.

Batch command syntax:

```
BAT_CHANGE_PLMOD_DATA  IBUS  'ID'  MTYPE  'NAME'  J  RDATA  I  IDATA
'CHDATA'
```

Python syntax:

```
ierr = change_plmod_data(ibus,id,mtype,name,j,rdata,i,idata,chdata)
```

Fortran syntax:

```
CALL
CHANGE_PLMOD_DATA(IBUS,ID,MTYPE,NAME,J,RDATA,I,IDATA,CHDATA,IERR)
```

where:

Integer IBUS	Is the plant bus number (input; no default allowed).
Character ID*2	Is the machine identifier (input; blank by default).
Integer MTYPE	Is the model type (input; 1 by default). MTYPE = 1 generator model. MTYPE = 2 compensator model. MTYPE = 3 stabilizer model. MTYPE = 4 minimum excitation limiter model. MTYPE = 5 maximum excitation limiter model. MTYPE = 6 excitation system model. MTYPE = 7 turbine-governor model. MTYPE = 8 turbine-load controller model.
Character NAME*6	Is the model name (input; no default allowed).
Integer J	Is the CON number to change relative to the start of the block of CONs assigned to this model, or 0 for no CON change (1 for CON(J), 2 for CON(J+1), etc.) (input; 0 by default).
Real RDATA	Is the new CON value (input; unchanged by default).
Integer I	Is the ICON number to change relative to the start of the block of ICONs assigned to this model, or 0 for no ICON change (1 for ICON(I), 2 for ICON(I+1), etc.) (input; 0 by default).
Integer IDATA	Is the new integer ICON value (input; unchanged by default). When a character value is intended for an ICON value, the IDATA entry is ignored and the integer ICON is set to 0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character CHDATA*2	Is the new character ICON value (input; no default allowed if the character ICON is to be changed). When an integer value is intended for an ICON value, the CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).
Integer IERR	Is the error code (output).

IERR = 0	no error.
IERR = 1	invalid MTYPE value.
IERR = 2	unknown model NAME.
IERR = 3	no model of type MTYPE at this machine.
IERR = 4	type MTYPE model at this machine is aaa not xxx.
IERR = 5	CON number exceeds the number of CONs used by the model.
IERR = 6	ICON number exceeds the number of ICONs used by the model.
IERR = 7	neither the CON nor ICON number is positive.
IERR = 8	bus not found.
IERR = 9	machine not found.
IERR = 10	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.3.5 CHANGE_PLMOD_ICON

Use this API to change the value of an integer ICON of a designated plant related model at a specified machine. The corresponding character CHRICN is set to '$$'.

Batch command syntax:

```
BAT_CHANGE_PLMOD_ICON  IBUS  'ID'  'NAME'  J  IDATA
```

Python syntax:

```
ierr = change_plmod_icon(ibus, id, name, j, idata)
```

Fortran syntax:

```
CALL CHANGE_PLMOD_ICON(IBUS, ID, NAME, J, IDATA, IERR)
```

where:

Integer IBUS	Is the plant bus number (input; no default allowed).
Character ID*2	Is the machine identifier (input; blank by default).
Character NAME*16	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change relative to the start of the block of ICONs assigned to this model (1 for ICON(I), 2 for ICON(I+1), etc.) (input; no default allowed).
Integer IDATA	Is the new integer ICON value (input; unchanged by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — unknown model NAME.
	IERR = 2 — model NAME is not a plant-related model.
	IERR = 3 — no <model type> model at this machine.
	IERR = 4 — the <model type> model at this machine is aaa not NAME.
	IERR = 5 — no input ICONs used in model NAME.
	IERR = 6 — ICON number outside of the ICON range of this model.
	IERR = 7 — bus not found.
	IERR = 8 — machine not found.
	IERR = 9 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.3.6 CHANGE_PLMOD_VAR

Use this API to change the value of a VAR of a designated plant related model at a specified machine.

Batch command syntax:

```
BAT_CHANGE_PLMOD_VAR  IBUS  'ID'  'NAME'  J  RDATA
```

Python syntax:

```
ierr = change_plmod_var(ibus, id, name, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_PLMOD_VAR(IBUS, ID, NAME, J, RDATA, IERR)
```

where:

Integer IBUS	Is the plant bus number (input; no default allowed).
Character ID*2	Is the machine identifier (input; blank by default).
Character NAME*16	Is the model name (input; no default allowed).
Integer J	Is the VAR number to change relative to the start of the block of VARs assigned to this model (1 for VAR(J), 2 for VAR(J+1), etc.) (input; no default allowed).
Real RDATA	Is the new VAR value (input; unchanged by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — unknown model NAME.
	IERR = 2 — model NAME is not a plant-related model.
	IERR = 3 — no <model type> model at this machine.
	IERR = 4 — the <model type> model at this machine is aaa not NAME.
	IERR = 5 — no VARs used in model NAME.
	IERR = 6 — VAR number outside of the VAR range of this model.
	IERR = 7 — bus not found.
	IERR = 8 — machine not found.
	IERR = 9 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.3.7 GMB_ADD_PLANT_MODEL

Use this API to add a plant related GMB model to a specified machine. If a model of the designated type already exists at the machine, it is replaced.

Batch command syntax:

```
BAT_GMB_ADD_PLANT_MODEL  IBUS  'ID'  'NAME'  IBUS2 'ID2'  NICN  IDATA
CHDATA  NCON  RDATA
```

Python syntax:

```
ierr = gmb_add_plant_model(ibus, id, name, ibus2, id2, nicn, idata,
chdata, ncon, rdata)
```

Fortran syntax:

```
CALL GMB_ADD_PLANT_MODEL(IBUS, ID, NAME, IBUS2, ID2, NICN, IDATA,
CHDATA, NCON, RDATA, IERR)
```

where:

Integer IBUS	Is the plant bus number (input; no default allowed).
Character ID*2	Is the machine identifier (input; blank by default).
Character NAME*16	Is the model name (input; no default allowed).
Integer IBUS2	Is the bus number of the second machine if this is a two machine model, or 0 for a single machine model (input; 0 by default).
Character ID2*2	Is the machine identifier of the second machine if this is a two machine model, or ' ' for a single machine model (input; blank by default).
Integer NICN	Is the number of ICONs required by the model (input; no default allowed).
Integer IDATA(NICN)	Are the integer ICON data (input; 0 by default). When a character value is intended for an ICON value, the corresponding IDATA entry is ignored and the integer ICON is set to 0.
Character CHDATA(NICN)*2	Are the character ICON data (input; no default allowed for an ICON for which a character value is to be specified). When an integer value is intended for an ICON value, the corresponding CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).
Integer NCON	Is the number of CONs required by the model (input; no default allowed).
Real RDATA(NCON)	Are the CON data (input; 0.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error
IERR = 1	machine model connection tables full--use pack plant model tables function
IERR = 2	plant model tables are corrupt
IERR = 3	unlicensed GMB model
IERR = 4	error reading GMB model
IERR = 5	GMB model is not a plant type model
IERR = 6	# CONs specified is not the number expected
IERR = 7	# ICONs specified is not the number expected
IERR = 8	second machine not found where one is required
IERR = 9	same machine specified as the first and second machines
IERR = 10	maximum number of array elements exceeded
IERR = 11	number of user-defined models exceeds maximum allowed
IERR = 12	bus not found
IERR = 13	machine not found
IERR = 14	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.3.8 PLMOD_CONSISTENCY

Use this API to check consistency among the plant related models referenced at each machine.

Batch command syntax:

```
BAT_PLMOD_CONSISTENCY
```

Python syntax:

```
ierr = plmod_consistency()
```

Fortran syntax:

```
CALL PLMOD_CONSISTENCY(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.3.9 PLMOD_PACK

Use this API to remove entries that are marked as unused from the plant model connection tables and the plant model array allocation tables. It packs the entries remaining in these tables at the front of the various arrays.

Batch command syntax:

```
BAT_PLMOD_PACK
```

Python syntax:

```
ierr = plmod_pack()
```

Fortran syntax:

```
CALL PLMOD_PACK(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.3.10 PLMOD_REMOVE

Use this API to remove a plant related model of a designated type from a specified machine.

Batch command syntax:

```
BAT_PLMOD_REMOVE  IBUS  'ID'  MTYPE
```

Python syntax:

```
ierr = plmod_remove(ibus, id, mtype)
```

Fortran syntax:

```
CALL PLMOD_REMOVE(IBUS, ID, MTYPE, IERR)
```

where:

Integer IBUS	Is the plant bus number (input; no default allowed).
Character ID*2	Is the machine identifier (input; blank by default).
Integer MTYPE	Is the model type (input; 1 by default). MTYPE = 1 generator model. MTYPE = 2 compensator model. MTYPE = 3 stabilizer model. MTYPE = 4 minimum excitation limiter model. MTYPE = 5 maximum excitation limiter model. MTYPE = 6 excitation system model. MTYPE = 7 turbine-governor model. MTYPE = 8 turbine-load controller model.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid MTYPE value. IERR = 2 no model of type MTYPE at this machine. IERR = 3 bus not found. IERR = 4 machine not found. IERR = 5 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.3.11 PLMOD_STATUS

Use this API to change the status of a plant related model of a designated type at a specified machine.

Batch command syntax:

```
BAT_PLMOD_STATUS  IBUS  'ID'  MTYPE  STATUS
```

Python syntax:

```
ierr = plmod_status(ibus, id, mtype, status)
```

Fortran syntax:

```
CALL PLMOD_STATUS(IBUS, ID, MTYPE, STATUS, IERR)
```

where:

Integer IBUS	Is the plant bus number (input; no default allowed).
Character ID*2	Is the machine identifier (input; blank by default).
Integer MTYPE	Is the model type (input; 1 by default). MTYPE = 1 generator model. MTYPE = 2 compensator model. MTYPE = 3 stabilizer model. MTYPE = 4 minimum excitation limiter model. MTYPE = 5 maximum excitation limiter model. MTYPE = 6 excitation system model. MTYPE = 7 turbine-governor model. MTYPE = 8 turbine-load controller model.
Integer STATUS	Is the new model status (input; 1 by default). STATUS = 0 out-of-service. STATUS = 1 in-service.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid MTYPE value. IERR = 2 no model of type MTYPE at this machine. IERR = 3 invalid STATUS value. IERR = 4 bus not found. IERR = 5 machine not found. IERR = 6 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.3.12 PLMOD_UNCONNECTED

Use this API to list or remove from dynamics working memory those plant related models that are assigned to machines that are not present in the current power flow working case (unconnected).

Batch command syntax:

```
BAT_PLMOD_UNCONNECTED  OPT
```

Python syntax:

```
ierr = plmod_unconnected(opt)
```

Fortran syntax:

```
CALL PLMOD_UNCONNECTED(OPT, IERR)
```

where:

Integer OPT	Is the function code (input; 0 by default): OPT = 0 list unconnected plant models. OPT = 1 remove unconnected plant models.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid OPT value. IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.3.13 PLMOD_USER

Use this API to list user-written plant model definitions or to remove user-written plant model definitions that are not assigned to any machines (unused) from the user model definition tables.

Batch command syntax:

```
BAT_PLMOD_USER  OPT
```

Python syntax:

```
ierr = plmod_user(opt)
```

Fortran syntax:

```
CALL PLMOD_USER(OPT, IERR)
```

where:

Integer OPT	Is the function code (input; 0 by default):	
	OPT = 0	list user-written plant model definitions.
	OPT = 1	remove unused user-written plant model definitions.
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid OPT value.
	IERR = 2	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.4 Wind Related Model APIs

4.4.1 ADD_WIND_MODEL

Use this API to add a wind related model of a designated type to a specified wind machine. If a model of the designated type already exists at the machine, it is replaced.

Batch command syntax:

```
BAT_ADD_WIND_MODEL  IBUS  'ID'  MTYPE  'NAME'  NICN  IDATA  CHDATA
NCON  RDATA
```

Python syntax:

```
ierr = add_wind_model(ibus, id, mtype, name, nicn, idata, chdata,
ncon, rdata)
```

Fortran syntax:

```
CALL ADD_WIND_MODEL(IBUS, ID, MTYPE, NAME, NICN, IDATA, CHDATA, NCON,
RDATA, IERR)
```

where:

Integer IBUS	Is the wind machine bus number (input; no default allowed).
Character ID*2	Is the machine identifier (input; blank by default).
Integer MTYPE	Is the model type (input; 1 by default).MTYPE = 1wind generator model. MTYPE = 2 wind electrical control model. MTYPE = 3 wind mechanical system model. MTYPE = 4 wind pitch control model. MTYPE = 5 wind aerodynamic model. MTYPE = 6 wind gust model. MTYPE = 7 wind auxiliary control model.
Character NAME*16	Is the model name (input; no default allowed).
Integer NICN	Is the number of ICONs required by the model (input; no default allowed).
Integer IDATA(NICN)	Are the integer ICON data (input; 0 by default). When a character value is intended for an ICON value, the corresponding IDATA entry is ignored and the integer ICON is set to 0.
Character CHDATA(NICN)*2	Are the character ICON data (input; no default allowed for an ICON for which a character value is to be specified). When an integer value is intended for an ICON value, the corresponding CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).
Integer NCON	Is the number of CONs required by the model (input; no default allowed).
Real RDATA(NCON)	Are the CON data (input; 0.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error
IERR = 1	wind machine model connection tables full--use pack wind model tables function
IERR = 2	wind model tables are corrupt
IERR = 3	invalid MTYPE value
IERR = 4	unknown model NAME
IERR = 5	model is not consistent with MTYPE
IERR = 6	# CONs specified is not the number expected
IERR = 7	# ICONs specified is not the number expected
IERR = 8	maximum number of array elements exceeded
IERR = 9	machine is not designated as a wind machine in power flow
IERR = 10	bus not found
IERR = 11	machine not found
IERR = 12	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.4.2 CHANGE_WNMOD_CHRICN

Use this API to change the value of a character ICON (i.e., an element of the CHRICN array) of a designated wind related model at a specified wind machine. The corresponding integer ICON is set to 0.

Batch command syntax:

```
BAT_CHANGE_WNMOD_CHRICN  IBUS  'ID'  'NAME'  J  'CHDATA'
```

Python syntax:

```
ierr = change_wnmod_chricn(ibus, id, name, j, chdata)
```

Fortran syntax:

```
CALL CHANGE_WNMOD_CHRICN(IBUS, ID, NAME, J, CHDATA, IERR)
```

where:

Integer IBUS	Is the plant bus number (input; no default allowed).
Character ID*2	Is the machine identifier (input; blank by default).
Character NAME*16	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change relative to the start of the block of ICONs assigned to this model (1 for ICON(I), 2 for ICON(I+1), etc.) (input; no default allowed).
Character CHDATA*2	Is the new character ICON value (input; blank by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — unknown model NAME.
	IERR = 2 — model NAME is not a wind model.
	IERR = 3 — no <model type> model at this machine.
	IERR = 4 — the <model type> model at this machine is aaa not NAME.
	IERR = 5 — no input ICONs used in model NAME.
	IERR = 6 — ICON number outside of the ICON range of this model.
	IERR = 7 — machine is not designated as a wind machine in power flow
	IERR = 8 — bus not found
	IERR = 9 — machine not found.
	IERR = 10 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.4.3 CHANGE_WNMOD_CON

Use this API to change the value of a CON of a designated wind related model at a specified wind machine.

Batch command syntax:

```
BAT_CHANGE_WNMOD_CON  IBUS  'ID'  'NAME'  J  RDATA
```

Python syntax:

```
ierr = change_wnmod_con(ibus, id, name, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_WNMOD_CON(IBUS, ID, NAME, J, RDATA, IERR)
```

where:

Integer IBUS	Is the plant bus number (input; no default allowed).
Character ID*2	Is the machine identifier (input; blank by default).
Character NAME*16	Is the model name (input; no default allowed).
Integer J	Is the CON number to change relative to the start of the block of CONs assigned to this model (1 for CON(J), 2 for CON(J+1), etc.) (input; no default allowed).
Real RDATA	Is the new CON value (input; unchanged by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — unknown model NAME.
	IERR = 2 — model NAME is not a wind model.
	IERR = 3 — no <model type> model at this machine.
	IERR = 4 — the <model type> model at this machine is aaa not NAME.
	IERR = 5 — no input CONs used in model NAME.
	IERR = 6 — CON number outside of the CON range of this model.
	IERR = 7 — machine is not designated as a wind machine in power flow
	IERR = 8 — bus not found.
	IERR = 9 — machine not found.
	IERR = 10 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.4.4 CHANGE_WNMOD_ICON

Use this API to change the value of an integer ICON of a designated wind related model at a specified wind machine. The corresponding character CHRICN is set to '$$'.

Batch command syntax:

```
BAT_CHANGE_WNMOD_ICON  IBUS  'ID'  'NAME'  J  IDATA
```

Python syntax:

```
ierr = change_wnmod_icon(ibus, id, name, j, idata)
```

Fortran syntax:

```
CALL CHANGE_WNMOD_ICON(IBUS, ID, NAME, J, IDATA, IERR)
```

where:

Integer IBUS	Is the plant bus number (input; no default allowed).
Character ID*2	Is the machine identifier (input; blank by default).
Character NAME*16	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change relative to the start of the block of ICONs assigned to this model (1 for ICON(I), 2 for ICON(I+1), etc.) (input; no default allowed).
Integer IDATA	Is the new integer ICON value (input; unchanged by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — unknown model NAME.
	IERR = 2 — model NAME is not a wind model.
	IERR = 3 — no <model type> model at this machine.
	IERR = 4 — the <model type> model at this machine is aaa not NAME.
	IERR = 5 — no input ICONs used in model NAME.
	IERR = 6 — ICON number outside of the ICON range of this model.
	IERR = 7 — machine is not designated as a wind machine in power flow
	IERR = 8 — bus not found.
	IERR = 9 — machine not found.
	IERR = 10 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.4.5 CHANGE_WNMOD_VAR

Use this API to change the value of a VAR of a designated wind related model at a specified wind machine.

Batch command syntax:

```
BAT_CHANGE_WNMOD_VAR  IBUS  'ID'  'NAME'  J  RDATA
```

Python syntax:

```
ierr = change_wnmod_var(ibus, id, name, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_WNMOD_VAR(IBUS, ID, NAME, J, RDATA, IERR)
```

where:

Integer IBUS	Is the plant bus number (input; no default allowed).
Character ID*2	Is the machine identifier (input; blank by default).
Character NAME*16	Is the model name (input; no default allowed).
Integer J	Is the VAR number to change relative to the start of the block of VARs assigned to this model (1 for VAR(J), 2 for VAR(J+1), etc.) (input; no default allowed).
Real RDATA	Is the new VAR value (input; unchanged by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	unknown model NAME
IERR = 2	model NAME is not a wind model
IERR = 3	no <model type> model at this machine
IERR = 4	the <model type> model at this machine is aaa not NAME
IERR = 5	no input VARs used in model NAME
IERR = 6	VAR number outside of the VAR range of this model
IERR = 7	machine is not designated as a wind machine in power flow
IERR = 8	bus not found
IERR = 9	machine not found
IERR = 10	prerequisite requirements for API are not met

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.4.6 WNMOD_CONSISTENCY

Use this API to check consistency among the wind related models referenced at each machine.

Batch command syntax:

```
BAT_WNMOD_CONSISTENCY
```

Python syntax:

```
ierr = wnmod_consistency()
```

Fortran syntax:

```
CALL WNMOD_CONSISTENCY(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.4.7 WNMOD_PACK

Use this API to remove entries that are marked as unused from the wind model connection tables and the wind model array allocation tables. It packs the entries remaining in these tables at the front of the various arrays.

Batch command syntax:

```
BAT_WNMOD_PACK
```

Python syntax:

```
ierr = wnmod_pack()
```

Fortran syntax:

```
CALL WNMOD_PACK(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.4.8 WNMOD_REMOVE

Use this API to remove a wind related model of a designated type from a specified wind machine.

Batch command syntax:

```
BAT_WNMOD_REMOVE  IBUS  'ID'  MTYPE
```

Python syntax:

```
ierr = wnmod_remove(ibus, id, mtype)
```

Fortran syntax:

```
CALL WNMOD_REMOVE(IBUS, ID, MTYPE, IERR)
```

where:

Integer IBUS	Is the wind machine bus number (input; no default allowed).
Character ID*2	Is the machine identifier (input; blank by default).
Integer MTYPE	Is the model type (input; 1 by default). MTYPE = 1 wind generator model. MTYPE = 2 wind electrical control model. MTYPE = 3 wind mechanical system model. MTYPE = 4 wind pitch control model. MTYPE = 5 wind aerodynamic model. MTYPE = 6 wind gust model. MTYPE = 7 wind auxiliary control model.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid MTYPE value. IERR = 2 no model of type MTYPE at this machine. IERR = 3 machine is not designated as a wind machine in power flow. IERR = 4 bus not found. IERR = 5 machine not found IERR = 6 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.4.9 WNMOD_STATUS

Use this API to change the status of a wind related model of a designated type at a specified wind machine.

Batch command syntax:

```
BAT_WNMOD_STATUS  IBUS  'ID'  MTYPE  STATUS
```

Python syntax:

```
ierr = wnmod_status(ibus, id, mtype, status)
```

Fortran syntax:

```
CALL WNMOD_STATUS(IBUS, ID, MTYPE, STATUS, IERR)
```

where:

Integer IBUS	Is the plant bus number (input; no default allowed).
Character ID*2	Is the machine identifier (input; blank by default).
Integer MTYPE	Is the model type (input; 1 by default). MTYPE = 1 wind generator model. MTYPE = 2 wind electrical control model. MTYPE = 3 wind mechanical system model. MTYPE = 4 wind pitch control model. MTYPE = 5 wind aerodynamic model. MTYPE = 6 wind gust model. MTYPE = 7 wind auxiliary control model.
Integer STATUS	Is the new model status (input; 1 by default). STATUS = 0 out-of-service. STATUS = 1 in-service.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid MTYPE value. IERR = 2 no model of type MTYPE at this machine. IERR = 3 invalid STATUS value. IERR = 4 machine is not designated as a wind machine in power flow. IERR = 5 bus not found. IERR = 6 machine not found. IERR = 7 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.4.10 WNMOD_UNCONNECTED

Use this API to list or remove from dynamics working memory those wind related models that are assigned to wind machines that are not present in the current power flow working case (unconnected).

Batch command syntax:

```
BAT_WNMOD_UNCONNECTED  OPT
```

Python syntax:

```
ierr = wnmod_unconnected(opt)
```

Fortran syntax:

```
CALL WNMOD_UNCONNECTED(OPT, IERR)
```

where:

Integer OPT	Is the function code (input; 0 by default): OPT = 0 list unconnected wind models. OPT = 1 remove unconnected wind models.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid OPT value. IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.4.11 WNMOD_USER

Use this API to list user-written wind model definitions or to remove user-written wind model definitions that are not assigned to any wind machines (unused) from the user model definition tables.

Batch command syntax:

```
BAT_WNMOD_USER  OPT
```

Python syntax:

```
ierr = wnmod_user(opt)
```

Fortran syntax:

```
CALL WNMOD_USER(OPT, IERR)
```

where:

Integer OPT	Is the function code (input; 0 by default):	
	OPT = 0	list user-written wind model definitions.
	OPT = 1	remove unused user-written wind model definitions.
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid OPT value.
	IERR = 2	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.5 Load Related Model APIs

4.5.1 ADD_LOAD_MODEL

Use this API to add a load related model of a designated type to a specified load or subsystem. If a model of the designated type already exists at the load or subsystem, it is replaced.

Batch command syntax:

```
BAT_ADD_LOAD_MODEL IBUS  'ID'  MTYPE  LTYPE  'NAME'  NICN  IDATA
CHDATA  NCON  RDATA
```

Python syntax:

```
ierr = add_load_model(ibus, id, mtype, ltype, name, nicn, idata,
chdata, ncon, rdata)
```

Fortran syntax:

```
CALL ADD_LOAD_MODEL(IBUS, ID, MTYPE, LTYPE, NAME, NICN, IDATA,
CHDATA, NCON, RDATA, IERR)
```

where:

Integer IBUS	Is the bus, owner, zone or area number. Ignored when MTYPE is 4 (input; no default allowed when MTYPE is less than 4).
Character ID*2	Is the load identifier, or "*" to apply the model to all loads in the subsystem (input; blank by default).
Integer MTYPE	Is the subsystem type (input; 0 by default). MTYPE = 0 bus model. MTYPE = 1 owner model. MTYPE = 2 zone model. MTYPE = 3 area model. MTYPE = 4 all model.
Integer LTYPE	Is the load model type (input; 1 by default). LTYPE = 1 load characteristic model. LTYPE = 2 load relay model.
Character NAME*16	Is the model name (input; no default allowed).
Integer NICN	Is the number of ICONs required by the model (input; no default allowed).
Integer IDATA(NICN)	Are the integer ICON data (input; 0 by default). When a character value is intended for an ICON value, the corresponding IDATA entry is ignored and the integer ICON is set to 0.
Character CHDATA(NICN)*2	Are the character ICON data (input; no default allowed for an ICON for which a character value is to be specified). When an integer value is intended for an ICON value, the corresponding CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer NCON	Is the number of CONs required by the model (input; no default allowed).	
Real RDATA(NCON)	Are the CON data (input; 0.0 by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error
	IERR = 1	load model connection tables full--use pack load model tables function
	IERR = 2	invalid MTYPE value
	IERR = 3	invalid LTYPE value
	IERR = 4	unknown model NAME
	IERR = 5	model is not consistent with LTYPE
	IERR = 6	model is not consistent with MTYPE
	IERR = 7	# CONs specified is not the number expected
	IERR = 8	# ICONs specified is not the number expected
	IERR = 9	maximum number of array elements exceeded
	IERR = 10	type MTYPE load model table is full
	IERR = 11	no loads with the specified load identifier
	IERR = 12	invalid area/owner/zone number
	IERR = 13	no loads with the specified load identifier for specified subsystem
	IERR = 14	no loads at bus
	IERR = 15	bus not found
	IERR = 16	load not found
	IERR = 17	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.5.2 CHANGE_LDMOD_CHRICN

Use this API to change the value of a character ICON (i.e., an element of the CHRICN array) of a load related model at a specified load or subsystem. The corresponding integer ICON is set to 0.

Batch command syntax:

```
BAT_CHANGE_LDMOD_CHRICN  IBUS  'ID'  'NAME'  J  'CHDATA'
```

Python syntax:

```
ierr = change_ldmod_chricn(ibus, id, name, j, chdata)
```

Fortran syntax:

```
CALL CHANGE_LDMOD_CHRICN(IBUS, ID, NAME, J, CHDATA, IERR)
```

where:

Integer IBUS	Is the bus, owner, zone or area number. Ignored when an all type load model is specified (input; no default allowed unless an all type load model is specified).
Character ID*2	Is the load identifier, or "*" if the model is applied to all loads in the subsystem (input; blank by default).
Character NAME*16	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change relative to the start of the block of ICONs assigned to this model (1 for ICON(I), 2 for ICON(I+1), etc.) (input; no default allowed).
Character CHDATA*2	Is the new character ICON value (input; blank by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error
	IERR = 1 — unknown model NAME
	IERR = 2 — model NAME is not a load-related model
	IERR = 3 — no <subsystem type> type <load type> model for this id in tables for <subsystem type> IBUS
	IERR = 4 — no all type <load type> model for this id in tables
	IERR = 5 — <subsystem type> <load type> model for this id for <subsystem type> IBUS is aaa not xxx
	IERR = 6 — all type <load type> model for this id is aaa not xxx
	IERR = 7 — no input ICONs used in model NAME
	IERR = 8 — ICON number outside of the ICON range of this model
	IERR = 9 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.5.3 CHANGE_LDMOD_CON

Use this API to change the value of a CON of a load related model at a specified load or subsystem.

Batch command syntax:

```
BAT_CHANGE_LDMOD_CON  IBUS  'ID'  'NAME'  J  RDATA
```

Python syntax:

```
ierr = change_ldmod_con(ibus, id, name, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_LDMOD_CON(IBUS, ID, NAME, J, RDATA, IERR)
```

where:

Integer IBUS	Is the bus, owner, zone or area number. Ignored when an all type load model is specified (input; no default allowed unless an all type load model is specified).
Character ID*2	Is the load identifier, or "*" if the model is applied to all loads in the subsystem (input; blank by default).
Character NAME*16	Is the model name (input; no default allowed).
Integer J	Is the CON number to change relative to the start of the block of CONs assigned to this model (1 for CON(J), 2 for CON(J+1), etc.) (input; no default allowed).
Real RDATA	Is the new CON value (input; unchanged by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	unknown model NAME
IERR = 2	model NAME is not a load-related model
IERR = 3	no <subsystem type> type <load type> model for this id in tables for <subsystem type> IBUS
IERR = 4	no all type <load type> model for this id in tables
IERR = 5	<subsystem type> <load type> model for this id for <subsystem type> IBUS is aaa not xxx
IERR = 6	all type <load type> model for this id is aaa not xxx
IERR = 7	no CONs used in model NAME
IERR = 8	CON number outside of the CON range of this model
IERR = 9	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.5.4 CHANGE_LDMOD_DATA

This API is obsolete. It has been replaced by the APIs CHANGE_LDMOD_CON, CHANGE_LDMOD_ICON and CHANGE_LDMOD_CHRICN.

Use this API to change the value of an ICON and/or CON of a load related model of a designated type at a specified load or subsystem.

Batch command syntax:

```
BAT_CHANGE_LDMOD_DATA  IBUS  'ID'  MTYPE  LTYPE  'NAME'  J  RDATA  I
IDATA  'CHDATA'
```

Python syntax:

```
ierr =
change_ldmod_data(ibus,id,mtype,ltype,name,j,rdata,i,idata,chdata)
```

Fortran syntax:

```
CALL CHANGE_LDMOD_DATA(IBUS, ID, MTYPE, LTYPE, NAME, J, RDATA, I,
IDATA, CHDATA, IERR)
```

where:

Integer IBUS	Is the bus, owner, zone or area number. Ignored when MTYPE is 4 (input; no default allowed when MTYPE is less than 4).
Character ID*2	Is the load identifier, or "*" if the model is applied to all loads in the subsystem (input; blank by default).
Integer MTYPE	Is the subsystem type (input; 0 by default). MTYPE = 0 bus model. MTYPE = 1 owner model. MTYPE = 2 zone model. MTYPE = 3 area model. MTYPE = 4 all model.
Integer LTYPE	Is the load model type (input; 1 by default). LTYPE = 1 load characteristic model. LTYPE = 2 load relay model.
Character NAME*6	Is the model name (input; no default allowed).
Integer J	Is the CON number to change relative to the start of the block of CONs assigned to this model, or 0 for no CON change (1 for CON(J), 2 for CON(J+1), etc.) (input; 0 by default).
Real RDATA	Is the new CON value (input; unchanged by default).
Integer I	Is the ICON number to change relative to the start of the block of ICONs assigned to this model, or 0 for no ICON change (1 for ICON(I), 2 for ICON(I+1), etc.) (input; 0 by default).
Integer IDATA	Is the new integer ICON value (input; unchanged by default). When a character value is intended for an ICON value, the IDATA entry is ignored and the integer ICON is set to 0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character CHDATA*2	Is the new character ICON value (input; no default allowed if the character ICON is to be changed). When an integer value is intended for an ICON value, the CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	invalid MTYPE value
IERR = 2	invalid LTYPE value
IERR = 3	unknown model NAME
IERR = 4	no all type LTYPE model for this id in tables
IERR = 5	no STYPE type LTYPE model for this id in tables for MTYPE IBUS
IERR = 6	all type LTYPE model for this id is aaa not xxx
IERR = 7	MTYPE type LTYPE model for this id for MTYPE IBUS is aaa not xxx
IERR = 8	CON number exceeds the number of CONs used by the model
IERR = 9	ICON number exceeds the number of input ICONs used by the model
IERR = 10	neither the CON nor ICON number is positive
IERR = 11	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.5.5 CHANGE_LDMOD_ICON

Use this API to change the value of an integer ICON of a load related model at a specified load or subsystem. The corresponding character CHRICN is set to '$$'.

Batch command syntax:

```
BAT_CHANGE_LDMOD_ICON  IBUS  'ID'  'NAME'  J  IDATA
```

Python syntax:

```
ierr = change_ldmod_icon(ibus, id, name, j, idata)
```

Fortran syntax:

```
CALL CHANGE_LDMOD_ICON(IBUS, ID, NAME, J, IDATA, IERR)
```

where:

Integer IBUS	Is the bus, owner, zone or area number. Ignored when an all type load model is specified (input; no default allowed unless an all type load model is specified).
Character ID*2	Is the load identifier, or "*" if the model is applied to all loads in the subsystem (input; blank by default).
Character NAME*16	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change relative to the start of the block of ICONs assigned to this model (1 for ICON(I), 2 for ICON(I+1), etc.) (input; no default allowed).
Integer IDATA	Is the new integer ICON value (input; unchanged by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	unknown model NAME
IERR = 2	model NAME is not a load-related model
IERR = 3	no <subsystem type> type <load type> model for this id in tables for <subsystem type> IBUS
IERR = 4	no all type <load type> model for this id in tables
IERR = 5	<subsystem type> <load type> model for this id for <subsystem type> IBUS is aaa not xxx
IERR = 6	all type <load type> model for this id is aaa not xxx
IERR = 7	no input ICONs used in model NAME
IERR = 8	ICON number outside of the ICON range of this model
IERR = 9	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.5.6 CHANGE_LDMOD_VAR

Use this API to change the value of a VAR of a load related model at a specified load or subsystem.

This API can be used only for bus based load models. It cannot be used for changing VARs of owner, zone, area, and "all" type load models.

Batch command syntax:

```
BAT_CHANGE_LDMOD_VAR  IBUS  'ID'  'NAME'  J  RDATA
```

Python syntax:

```
ierr = change_ldmod_var(ibus, id, name, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_LDMOD_VAR(IBUS, ID, NAME, J, RDATA, IERR)
```

where:

Integer IBUS	Is the bus number.
Character ID*2	Is the load identifier, or "*" if the model is applied to all loads in the subsystem (input; blank by default).
Character NAME*16	Is the model name (input; no default allowed).
Integer J	Is the VAR number to change relative to the start of the block of VARs assigned to this model (1 for VAR(J), 2 for VAR(J+1), etc.) (input; no default allowed).
Real RDATA	Is the new VAR value (input; unchanged by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	unknown model NAME
IERR = 2	model NAME is not a load-related model
IERR = 3	no <subsystem type> type <load type> model for this id in tables for <subsystem type> IBUS
IERR = 4	no "all" type <load type> model for this id in tables
IERR = 5	<subsystem type> <load type> model for this id for <subsystem type> IBUS is aaa not xxx
IERR = 6	"all" type <load type> model for this id is aaa not xxx
IERR = 7	no VARs used in model NAME
IERR = 8	VAR number outside of the VAR range of this model
IERR = 9	VARs of only bus based (load type) models can be changed using this API
IERR = 10	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.5.7 LDMOD_PACK

Use this API to remove entries that are marked as unused from the load model connection tables and the load model array allocation tables. It packs the entries remaining in these tables at the front of the various arrays.

Batch command syntax:

```
BAT_LDMOD_PACK
```

Python syntax:

```
ierr = ldmod_pack()
```

Fortran syntax:

```
CALL LDMOD_PACK(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.5.8 LDMOD_REMOVE

Use this API to remove a load related model of a designated type from a specified load or subsystem.

Batch command syntax:

```
BAT_LDMOD_REMOVE  MTYPE  IBUS  'ID'  LTYPE
```

Python syntax:

```
ierr = ldmod_remove(mtype, ibus, id, ltype)
```

Fortran syntax:

```
CALL LDMOD_REMOVE(MTYPE, IBUS, ID, LTYPE, IERR)
```

where:

Integer MTYPE	Is the subsystem type (input; 0 by default). MTYPE = 0 bus model. MTYPE = 1 owner model. MTYPE = 2 zone model. MTYPE = 3 area model. MTYPE = 4 all model.
Integer IBUS	Is the bus, owner, zone or area number. Ignored when MTYPE is 4 (input; no default allowed when MTYPE is less than 4).
Character ID*2	Is the load identifier, or "*" if the model is applied to all loads in the subsystem (input; blank by default).
Integer LTYPE	Is the load model type (input; 1 by default). LTYPE = 1 load characteristic model. LTYPE = 2 load relay model.
Integer IERR	Is the error code (output). IERR = 0 no error IERR = 1 invalid number specified for IBUS IERR = 2 invalid LTYPE value IERR = 3 invalid MTYPE value IERR = 4 no all type LTYPE model for this id in tables IERR = 5 no MTYPE type LTYPE model for this id in tables for MTYPE IBUS IERR = 6 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.5.9 LDMOD_STATUS

Use this API to change the status of a load related model of a designated type for all subsystem loads at which it is applied.

Batch command syntax:

```
BAT_LDMOD_STATUS  MTYPE  IBUS  'ID'  LTYPE  STATUS
```

Python syntax:

```
ierr = ldmod_status(mtype, ibus, id, ltype, status)
```

Fortran syntax:

```
CALL LDMOD_STATUS(MTYPE, IBUS, ID, LTYPE, STATUS, IERR)
```

where:

Integer MTYPE	Is the subsystem type (input; 0 by default). MTYPE = 0 bus model. MTYPE = 1 owner model. MTYPE = 2 zone model. MTYPE = 3 area model. MTYPE = 4 all model.
Integer IBUS	Is the bus, owner, zone or area number. Ignored when MTYPE is 4 (input; no default allowed when MTYPE is less than 4).
Character ID*2	Is the load identifier, or "*" if the model is applied to all loads in the subsystem (input; blank by default).
Integer LTYPE	Is the load model type (input; 1 by default). LTYPE = 1 load characteristic model. LTYPE = 2 load relay model.
Integer STATUS	Is the new model status (input; 1 by default). STATUS = 0 out-of-service. STATUS = 1 in-service.
Integer IERR	Is the error code (output). IERR = 0 no error IERR = 1 invalid number specified for IBUS IERR = 2 invalid LTYPE value IERR = 3 invalid MTYPE value IERR = 4 no all type LTYPE model for this id in tables IERR = 5 no MTYPE type LTYPE model for this id in tables for MTYPE IBUS IERR = 6 invalid STATUS value IERR = 7 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.5.10 LDMOD_STATUS2

Use this API to change the status of a load related model of a designated type at a specified load.

Batch command syntax:

```
BAT_LDMOD_STATUS2  IBUS  'ID'  LTYPE  STATUS
```

Python syntax:

```
ierr = ldmod_status2(ibus, id, ltype, status)
```

Fortran syntax:

```
CALL LDMOD_STATUS2(IBUS, ID, LTYPE, STATUS, IERR)
```

where:

Integer IBUS	Is the bus number (input; no default allowed).
Character ID*2	Is the load identifier (input; blank by default).
Integer LTYPE	Is the load model type (input; 1 by default). LTYPE = 1 load characteristic model. LTYPE = 2 load relay model.
Integer STATUS	Is the new model status (input; 1 by default). STATUS = 0 out-of-service. STATUS = 1 in-service.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid LTYPE value. IERR = 2 invalid STATUS value. IERR = 3 no LTYPE model at this load. IERR = 4 bus not found. IERR = 5 load not found. IERR = 6 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.5.11 LDMOD_UNCONNECTED

Use this API to list or remove from dynamics working memory those load related models that are assigned to loads that are not present in the current power flow working case (unconnected).

Batch command syntax:

```
BAT_LDMOD_UNCONNECTED OPT
```

Python syntax:

```
ierr = ldmod_unconnected(opt)
```

Fortran syntax:

```
CALL LDMOD_UNCONNECTED(OPT, IERR)
```

where:

Integer OPT	Is the function code (input; 0 by default): OPT = 0 list unconnected load models. OPT = 1 remove unconnected load models.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid OPT value. IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.5.12 LDMOD_USER

Use this API to list user-written load model definitions or to remove user-written load model definitions that are not assigned to any loads (unused) from the user model definition tables.

Batch command syntax:

```
BAT_LDMOD_USER  OPT
```

Python syntax:

```
ierr = ldmod_user(opt)
```

Fortran syntax:

```
CALL LDMOD_USER(OPT, IERR)
```

where:

Integer OPT	Is the function code (input; 0 by default):	
	OPT = 0	list user-written load model definitions.
	OPT = 1	remove unused user-written load model definitions.
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid OPT value.
	IERR = 2	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.6 Line Relay Model APIs

4.6.1 ADD_RELAY_MODEL

Use this API to add a line relay model in the designated relay slot of the from bus end of a specified branch. If a model already exists in the designated relay slot at the from bus end of the branch, it is replaced.

Batch command syntax:

```
BAT_ADD_RELAY_MODEL  IBUS  JBUS  'ID'  RS  'NAME'  NICN  IDATA
CHDATA  NCON  RDATA
```

Python syntax:

```
ierr =
add_relay_model(ibus,jbus,id,rs,name,nicn,idata,chdata,ncon,rdata)
```

Fortran syntax:

```
CALL ADD_RELAY_MODEL(IBUS, JBUS, ID, RS, NAME, NICN, IDATA, CHDATA,
NCON, RDATA, IERR)
```

where:

Integer IBUS	Is the from bus number (input; no default allowed).
Integer JBUS	Is the to bus number (input; no default allowed).
Character ID*2	Is the circuit identifier (input; blank by default).
Integer RS	Is the relay slot (1 or 2) (input; 1 by default).
Character NAME*16	Is the model name (input; no default allowed).
Integer NICN	Is the number of ICONs required by the model (input; no default allowed).
Integer IDATA(NICN)	Are the integer ICON data (input; 0 by default). When a character value is intended for an ICON value, the corresponding IDATA entry is ignored and the integer ICON is set to 0.
Character CHDATA(NICN)*2	Are the character ICON data (input; no default allowed for an ICON for which a character value is to be specified). When an integer value is intended for an ICON value, the corresponding CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).
Integer NCON	Is the number of CONs required by the model (input; no default allowed).
Real RDATA(NCON)	Are the CON data (input; 0.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error
IERR = 1	invalid RS value
IERR = 2	unknown model NAME
IERR = 3	model is not a line relay model
IERR = 4	# CONs specified is not the number expected
IERR = 5	# ICONs specified is not the number expected
IERR = 6	maximum number of array elements exceeded
IERR = 7	line relay model connection tables full--use pack line relay model tables function
IERR = 8	line relay model tables are corrupt
IERR = 9	bus not found
IERR = 10	branch not found
IERR = 11	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.6.2 CHANGE_RLMOD_CHRICN

Use this API to change the value of a character ICON (i.e., an element of the CHRICN array) of the line relay model in the designated relay slot of the from bus end of a specified branch. The corresponding integer ICON is set to 0.

Batch command syntax:

```
BAT_CHANGE_RLMOD_CHRICN  IBUS  JBUS  'ID'  RS  'NAME'  J  'CHDATA'
```

Python syntax:

```
ierr = change_rlmod_chricn(ibus, jbus, id, rs, name, j, chdata)
```

Fortran syntax:

```
CALL CHANGE_RLMOD_CHRICN(IBUS,JBUS,ID,RS,NAME,J,CHDATA,IERR)
```

where:

Integer IBUS	Is the from bus number (input; no default allowed).
Integer JBUS	Is the to bus number (input; no default allowed).
Character ID*2	Is the circuit identifier (input; blank by default).
Integer RS	Is the relay slot (1 or 2) (input; 1 by default).
Character NAME*16	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change relative to the start of the block of ICONs assigned to this model (1 for ICON(I), 2 for ICON(I+1), etc.) (input; no default allowed).
Character CHDATA*2	Is the new character ICON value (input; blank by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — invalid RS value.
	IERR = 2 — unknown model NAME.
	IERR = 3 — no relay model on this branch at the bus IBUS end.
	IERR = 4 — relay model at bus IBUS end is aaa not xxx.
	IERR = 5 — no input ICONs used in model NAME.
	IERR = 6 — ICON number outside of the ICON range of this model.
	IERR = 7 — bus not found.
	IERR = 8 — branch not found.
	IERR = 9 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.6.3 CHANGE_RLMOD_CON

Use this API to change the value of a CON of the line relay model in the designated relay slot of the from bus end of a specified branch.

Batch command syntax:

```
BAT_CHANGE_RLMOD_CON  IBUS  JBUS  'ID'  RS  'NAME'  J  RDATA
```

Python syntax:

```
ierr = change_rlmod_con(ibus, jbus, id, rs, name, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_RLMOD_CON(IBUS, JBUS, ID, RS, NAME, J, RDATA, IERR)
```

where:

Integer IBUS	Is the from bus number (input; no default allowed).
Integer JBUS	Is the to bus number (input; no default allowed).
Character ID*2	Is the circuit identifier (input; blank by default).
Integer RS	Is the relay slot (1 or 2) (input; 1 by default).
Character NAME*16	Is the model name (input; no default allowed).
Integer J	Is the CON number to change relative to the start of the block of CONs assigned to this model (1 for CON(J), 2 for CON(J+1), etc.) (input; no default allowed).
Real RDATA	Is the new CON value (input; unchanged by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — invalid RS value.
	IERR = 2 — unknown model NAME.
	IERR = 3 — no relay model on this branch at the bus IBUS end.
	IERR = 4 — relay model at bus IBUS end is aaa not xxx.
	IERR = 5 — no CONs used in model NAME.
	IERR = 6 — CON number outside of the CON range of this model.
	IERR = 7 — bus not found.
	IERR = 8 — branch not found.
	IERR = 9 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.6.4 CHANGE_RLMOD_DATA

This API is obsolete. It has been replaced by the APIs CHANGE_RLMOD_CON, CHANGE_RLMOD_ICON and CHANGE_RLMOD_CHRICN.

Use this API to change the value of an ICON and/or CON of the line relay model in the designated relay slot of the from bus end of a specified branch.

Batch command syntax:

```
BAT_CHANGE_RLMOD_DATA  IBUS  JBUS  'ID'  RS  'NAME'  J  RDATA  I
IDATA  'CHDATA'
```

Python syntax:

```
ierr = change_rlmod_data(ibus,jbus,id,rs,name,j,rdata,i,idata,chdata)
```

Fortran syntax:

```
CALL CHANGE_RLMOD_DATA(IBUS, JBUS, ID, RS, NAME, J, RDATA, I, IDATA,
CHDATA, IERR)
```

where:

Integer IBUS	Is the from bus number (input; no default allowed).
Integer JBUS	Is the to bus number (input; no default allowed).
Character ID*2	Is the circuit identifier (input; blank by default).
Integer RS	Is the relay slot (1 or 2) (input; 1 by default).
Character NAME*6	Is the model name (input; no default allowed).
Integer J	Is the CON number to change relative to the start of the block of CONs assigned to this model, or 0 for no CON change (1 for CON(J), 2 for CON(J+1), etc.) (input; 0 by default).
Real RDATA	Is the new CON value (input; unchanged by default).
Integer I	Is the ICON number to change relative to the start of the block of ICONs assigned to this model, or 0 for no ICON change (1 for ICON(I), 2 for ICON(I+1), etc.) (input; 0 by default).
Integer IDATA	Is the new integer ICON value (input; unchanged by default). When a character value is intended for an ICON value, the IDATA entry is ignored and the integer ICON is set to 0.
Character CHDATA*2	Is the new character ICON value (input; no default allowed if the character ICON is to be changed). When an integer value is intended for an ICON value, the CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid RS value.
	IERR = 2	unknown model NAME.
	IERR = 3	no relay model on this branch at the bus IBUS end.
	IERR = 4	relay model at bus IBUS end is aaa not xxx.
	IERR = 5	CON number exceeds the number of CONs used by the model.
	IERR = 6	ICON number exceeds the number of ICONs used by the model.
	IERR = 7	neither the CON nor ICON number is positive.
	IERR = 8	bus not found.
	IERR = 9	branch not found.
	IERR = 10	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.6.5 CHANGE_RLMOD_ICON

Use this API to change the value of an integer ICON of the line relay model in the designated relay slot of the from bus end of a specified branch. The corresponding character CHRICN is set to '$$'.

Batch command syntax:

```
BAT_CHANGE_RLMOD_ICON  IBUS  JBUS  'ID'  RS  'NAME'  J  IDATA
```

Python syntax:

```
ierr = change_rlmod_icon(ibus, jbus, id, rs, name, j, idata)
```

Fortran syntax:

```
CALL CHANGE_RLMOD_ICON(IBUS, JBUS, ID, RS, NAME, J, IDATA, IERR)
```

where:

Integer IBUS	Is the from bus number (input; no default allowed).
Integer JBUS	Is the to bus number (input; no default allowed).
Character ID*2	Is the circuit identifier (input; blank by default).
Integer RS	Is the relay slot (1 or 2) (input; 1 by default).
Character NAME*16	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change relative to the start of the block of ICONs assigned to this model (1 for ICON(I), 2 for ICON(I+1), etc.) (input; no default allowed).
Integer IDATA	Is the new integer ICON value (input; unchanged by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — invalid RS value.
	IERR = 2 — unknown model NAME.
	IERR = 3 — no relay model on this branch at the bus IBUS end.
	IERR = 4 — relay model at bus IBUS end is aaa not xxx.
	IERR = 5 — no input ICONs used in model NAME.
	IERR = 6 — ICON number outside of the ICON range of this model.
	IERR = 7 — bus not found.
	IERR = 8 — branch not found.
	IERR = 9 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.6.6 CHANGE_RLMOD_VAR

Use this API to change the value of a VAR of the line relay model in the designated relay slot of the from bus end of a specified branch.

Batch command syntax:

```
BAT_CHANGE_RLMOD_VAR  IBUS  JBUS  'ID'  RS  'NAME'  J  RDATA
```

Python syntax:

```
ierr = change_rlmod_var(ibus, jbus, id, rs, name, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_RLMOD_VAR(IBUS, JBUS, ID, RS, NAME, J, RDATA, IERR)
```

where:

Integer IBUS	Is the from bus number (input; no default allowed).
Integer JBUS	Is the to bus number (input; no default allowed).
Character ID*2	Is the circuit identifier (input; blank by default).
Integer RS	Is the relay slot (1 or 2) (input; 1 by default).
Character NAME*16	Is the model name (input; no default allowed).
Integer J	Is the VAR number to change relative to the start of the block of VARs assigned to this model (1 for VAR(J), 2 for VAR(J+1), etc.) (input; no default allowed).
Real RDATA	Is the new VAR value (input; unchanged by default).
Integer IERR	Is the error code (output).
	IERR = 0 no error.
	IERR = 1 invalid RS value.
	IERR = 2 unknown model NAME.
	IERR = 3 no relay model on this branch at the bus IBUS end.
	IERR = 4 relay model at bus IBUS end is aaa not xxx.
	IERR = 5 no VARs used in model NAME.
	IERR = 6 VAR number outside of the VAR range of this model.
	IERR = 7 bus not found.
	IERR = 8 branch not found.
	IERR = 9 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.6.7 RLMOD_PACK

Use this API to remove entries that are marked as unused from the line relay model connection tables and the line relay model array allocation tables. It packs the entries remaining in these tables at the front of the various arrays.

Batch command syntax:

```
BAT_RLMOD_PACK
```

Python syntax:

```
ierr = rlmod_pack()
```

Fortran syntax:

```
CALL RLMOD_PACK(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.6.8 RLMOD_REMOVE

Use this API to remove the line relay model in the designated relay slot of the from bus end of a specified branch.

Batch command syntax:

```
BAT_RLMOD_REMOVE  IBUS  JBUS 'ID'  RS
```

Python syntax:

```
ierr = rlmod_remove(ibus, jbus, id, rs)
```

Fortran syntax:

```
CALL RLMOD_REMOVE(IBUS, JBUS, ID, RS, IERR)
```

where:

Integer IBUS	Is the from bus number (input; no default allowed).
Integer JBUS	Is the to bus number (input; no default allowed).
Character ID*2	Is the circuit identifier (input; blank by default).
Integer RS	Is the relay slot (1 or 2) (input; 1 by default).
Integer IERR	Is the error code (output).
	IERR = 0 no error.
	IERR = 1 invalid RS value.
	IERR = 2 no relay model on this branch at the bus IBUS end.
	IERR = 3 bus not found.
	IERR = 4 branch not found.
	IERR = 5 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.6.9 RLMOD_STATUS

Use this API to change the status of the line relay model in the designated relay slot of the from bus end of a specified branch.

Batch command syntax:

```
BAT_RLMOD_STATUS  IBUS  JBUS  'ID'  RS  STATUS
```

Python syntax:

```
ierr = rlmod_status(ibus, jbus, id, rs, status)
```

Fortran syntax:

```
CALL RLMOD_STATUS(IBUS, JBUS, ID, RS, STATUS, IERR)
```

where:

Integer IBUS	Is the from bus number (input; no default allowed).
Integer JBUS	Is the to bus number (input; no default allowed).
Character ID*2	Is the circuit identifier (input; blank by default).
Integer RS	Is the relay slot (1 or 2) (input; 1 by default).
Integer STATUS	Is the new model status (input; 1 by default). STATUS = 0 out-of-service. STATUS = 1 in-service.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid RS value. IERR = 2 no relay model on this branch at the bus IBUS end. IERR = 3 invalid STATUS value. IERR = 4 bus not found. IERR = 5 branch not found. IERR = 6 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.6.10 RLMOD_UNCONNECTED

Use this API to list or remove from dynamics working memory those line relay models that are assigned to branches that are not present in the current power flow working case (unconnected).

Batch command syntax:

```
BAT_RLMOD_UNCONNECTED  OPT
```

Python syntax:

```
ierr = rlmod_unconnected(opt)
```

Fortran syntax:

```
CALL RLMOD_UNCONNECTED(OPT, IERR)
```

where:

Integer OPT	Is the function code (input; 0 by default):	
	OPT = 0	list unconnected line relay models.
	OPT = 1	remove unconnected line relay models.
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid OPT value.
	IERR = 2	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.6.11 RLMOD_USER

Use this API to list user-written line relay model definitions or to remove user-written line relay model definitions that are not assigned to any branches (unused) from the user model definition tables.

Batch command syntax:

```
BAT_RLMOD_USER  OPT
```

Python syntax:

```
ierr = rlmod_user(opt)
```

Fortran syntax:

```
CALL RLMOD_USER(OPT, IERR)
```

where:

Integer OPT	Is the function code (input; 0 by default): OPT = 0 list user-written line relay definitions. OPT = 1 remove unused user-written line relay definitions.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid OPT value. IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.7 DC Line Model APIs

4.7.1 ADD_MTDCLINE_MODEL

Use this API to add a multi-terminal dc line model to the specified multi-terminal dc line. If a model already exists at the multi-terminal dc line, it is replaced.

Batch command syntax:

```
BAT_ADD_MTDCLINE_MODEL  DC  'MODEL'  NICN  IDATA  CHDATA  NCON  RDATA
IERR
```

Python syntax:

```
ierr = add_mtdcline_model(dc,model,nicn,idata,chdata,ncon,rdata)
```

Fortran syntax:

```
CALL ADD_MTDCLINE_MODEL(NUM,MODEL,NICN,IDATA,CHDATA,NCON,RDATA,IERR)

CALL
ADD_MTDCLINE_MODEL_I(NUM,MODEL,NICN,IDATA,CHDATA,NCON,RDATA,IERR)

CALL
ADD_MTDCLINE_MODEL_C(NAME,MODEL,NICN,IDATA,CHDATA,NCON,RDATA,IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the multi-terminal dc line to which the model is to be added; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the multi-terminal dc line to which the model is to be added (input; no default allowed).
Character*12 NAME	Is the name of the multi-terminal dc line to which the model is to be added (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer NICN	Is the number of ICONs required by the model (input; no default allowed).
Integer IDATA(NICN)	Are the integer ICON data (input; 0 by default). When a character value is intended for an ICON value, the corresponding IDATA entry is ignored and the integer ICON is set to 0.
Character*2 CHDATA(NICN)	Are the character ICON data (input; no default allowed for an ICON for which a character value is to be specified). When an integer value is intended for an ICON value, the corresponding CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).
Integer NCON	Is the number of CONs required by the model (input; no default allowed).
Real RDATA(NCON)	Are the CON data (input; 0.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error
	IERR = 1	dc line not found
	IERR = 2	multi-terminal dc line model connection tables full--use pack dc line model tables function
	IERR = 3	unknown model name
	IERR = 4	model is not a multi-terminal dc line model
	IERR = 5	# CONs specified is not the number expected
	IERR = 6	# ICONs specified is not the number expected
	IERR = 7	maximum number of array elements exceeded
	IERR = 8	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.7.2 ADD_TTDCLINE_MODEL

Use this API to add a two-terminal dc line model to the specified two-terminal dc line. If a model already exists at the two-terminal dc line, it is replaced.

Batch command syntax:

```
BAT_ADD_TTDCLINE_MODEL  DC  'MODEL'  NICN  IDATA  CHDATA  NCON  RDATA
IERR
```

Python syntax:

```
ierr = add_ttdcline_model(dc,model,nicn,idata,chdata,ncon,rdata)
```

Fortran syntax:

```
CALL ADD_TTDCLINE_MODEL(NUM,MODEL,NICN,IDATA,CHDATA,NCON,RDATA,IERR)

CALL
ADD_TTDCLINE_MODEL_I(NUM,MODEL,NICN,IDATA,CHDATA,NCON,RDATA,IERR)

CALL
ADD_TTDCLINE_MODEL_C(NAME,MODEL,NICN,IDATA,CHDATA,NCON,RDATA,IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the two-terminal dc line to which the model is to be added; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the two-terminal dc line to which the model is to be added (input; no default allowed).
Character*12 NAME	Is the name of the two-terminal dc line to which the model is to be added (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer NICN	Is the number of ICONs required by the model (input; no default allowed).
Integer IDATA(NICN)	Are the integer ICON data (input; 0 by default). When a character value is intended for an ICON value, the corresponding IDATA entry is ignored and the integer ICON is set to 0.
Character*2 CHDATA(NICN)	Are the character ICON data (input; no default allowed for an ICON for which a character value is to be specified). When an integer value is intended for an ICON value, the corresponding CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).
Integer NCON	Is the number of CONs required by the model (input; no default allowed).
Real RDATA(NCON)	Are the CON data (input; 0.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error
	IERR = 1	dc line not found
	IERR = 2	two-terminal dc line model connection tables full--use pack dc line model tables function
	IERR = 3	unknown model name
	IERR = 4	model is not a two-terminal dc line model
	IERR = 5	# CONs specified is not the number expected
	IERR = 6	# ICONs specified is not the number expected
	IERR = 7	maximum number of array elements exceeded
	IERR = 8	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.7.3 ADD_VSDCLINE_MODEL

Use this API to add a VSC dc line model to the specified VSC dc line. If a model already exists at the VSC dc line, it is replaced.

Batch command syntax:

```
BAT_ADD_VSDCLINE_MODEL  'DCNAME'  'MODEL'  NICN  IDATA  CHDATA  NCON
RDATA  IERR
```

Python syntax:

```
ierr = add_vsdcline_model(dcname,model,nicn,idata,chdata,ncon,rdata)
```

Fortran syntax:

```
CALL
ADD_VSDCLINE_MODEL(DCNAME,MODEL,NICN,IDATA,CHDATA,NCON,RDATA,IERR)
```

where:

Character*12 DCNAME	Is the VSC dc line name (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer NICN	Is the number of ICONs required by the model (input; no default allowed).
Integer IDATA(NICN)	Are the integer ICON data (input; 0 by default). When a character value is intended for an ICON value, the corresponding IDATA entry is ignored and the integer ICON is set to 0.
Character*2 CHDATA(NICN)	Are the character ICON data (input; no default allowed for an ICON for which a character value is to be specified). When an integer value is intended for an ICON value, the corresponding CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).
Integer NCON	Is the number of CONs required by the model (input; no default allowed).
Real RDATA(NCON)	Are the CON data (input; 0.0 by default).

Integer IERR	Is the error code (output).
	IERR = 0 no error
	IERR = 1 dc line not found
	IERR = 2 VSC dc line model connection tables full--use pack dc line model tables function
	IERR = 3 unknown model name
	IERR = 4 model is not a VSC dc line model
	IERR = 5 # CONs specified is not the number expected
	IERR = 6 # ICONs specified is not the number expected
	IERR = 7 maximum number of array elements exceeded
	IERR = 8 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.7.4 CHANGE_MTDCLMOD_CHRICN

Use this API to change the value of a character ICON (i.e., an element of the CHRICN array) of the multi-terminal dc line model of a specified multi-terminal dc line. The corresponding integer ICON is set to 0.

Batch command syntax:

```
BAT_CHANGE_MTDCLMOD_CHRICN  DC  'MODEL'  J  'CHDATA'
```

Python syntax:

```
ierr = change_mtdclmod_chricn(dc, model, j, chdata)
```

Fortran syntax:

```
CALL CHANGE_MTDCLMOD_CHRICN(NUM, MODEL, J, CHDATA, IERR)

CALL CHANGE_MTDCLMOD_CHRICN_I(NUM, MODEL, J, CHDATA, IERR)

CALL CHANGE_MTDCLMOD_CHRICN_C(NAME, MODEL, J, CHDATA, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the multi-terminal dc line for which model data is to be changed; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the multi-terminal dc line for which model data is to be changed (input; no default allowed).
Character*12 NAME	Is the name of the multi-terminal dc line for which model data is to be changed (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change relative to the start of the block of ICONs assigned to this model (1 for ICON(I), 2 for ICON(I+1), etc.) (input; no default allowed).
Character*2 CHDATA	Is the new character ICON value (input; blank by default).
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 dc line not found. IERR = 2 unknown model name. IERR = 3 no model at this dc line. IERR = 4 model at this dc line is aaa not xxx. IERR = 5 no input ICONs used in model. IERR = 6 ICON number outside of the ICON range of this model. IERR = 7 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.7.5 CHANGE_MTDCLMOD_CON

Use this API to change the value of a CON of the multi-terminal dc line model of a specified multi-terminal dc line.

Batch command syntax:

```
BAT_CHANGE_MTDCLMOD_CON  DC  'MODEL'  J  RDATA
```

Python syntax:

```
ierr = change_mtdclmod_con(dc, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_MTDCLMOD_CON(NUM, MODEL, J, RDATA, IERR)

CALL CHANGE_MTDCLMOD_CON_I(NUM, MODEL, J, RDATA, IERR)

CALL CHANGE_MTDCLMOD_CON_C(NAME, MODEL, J, RDATA, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the multi-terminal dc line for which model data is to be changed; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the multi-terminal dc line for which model data is to be changed (input; no default allowed).
Character*12 NAME	Is the name of the multi-terminal dc line for which model data is to be changed (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the CON number to change relative to the start of the block of CONs assigned to this model (1 for CON(J), 2 for CON(J+1), etc.) (input; no default allowed).
Real RDATA	Is the new CON value (input; unchanged by default).
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 dc line not found. IERR = 2 unknown model name. IERR = 3 no model at this dc line. IERR = 4 model at this dc line is aaa not xxx. IERR = 5 no CONs used in model. IERR = 6 CON number outside of the CON range of this model. IERR = 7 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.7.6 CHANGE_MTDCLMOD_DATA

This API is obsolete. It has been replaced by the APIs CHANGE_MTDCLMOD_CON, CHANGE_MTDCLMOD_ICON and CHANGE_MTDCLMOD_CHRICN.

Use this API to change the value of an ICON and/or CON of the multi-terminal dc line model of a specified multi-terminal dc line.

Batch command syntax:

```
BAT_CHANGE_MTDCLMOD_DATA  IDC  'NAME'  J  RDATA  I  IDATA  'CHDATA'
```

Python syntax:

```
ierr = change_mtdclmod_data(idc, name, j, rdata, i, idata, chdata)
```

Fortran syntax:

```
CALL CHANGE_MTDCLMOD_DATA(IDC,NAME,J,RDATA,I,IDATA,CHDATA,IERR)
```

where:

Integer IDC	Is the multi-terminal dc line number (input; no default allowed).
Character*6 NAME	Is the model name (input; no default allowed).
Integer J	Is the CON number to change relative to the start of the block of CONs assigned to this model, or 0 for no CON change (1 for CON(J), 2 for CON(J+1), etc.) (input; 0 by default).
Real RDATA	Is the new CON value (input; unchanged by default).
Integer I	Is the ICON number to change relative to the start of the block of ICONs assigned to this model, or 0 for no ICON change (1 for ICON(I), 2 for ICON(I+1), etc.) (input; 0 by default).
Integer IDATA	Is the new integer ICON value (input; unchanged by default). When a character value is intended for an ICON value, the IDATA entry is ignored and the integer ICON is set to 0.
Character*2 CHDATA	Is the new character ICON value (input; no default allowed if the character ICON is to be changed). When an integer value is intended for an ICON value, the CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 1	dc line not found.
	IERR = 2	unknown model name.
	IERR = 3	no model at this dc line.
	IERR = 4	model at this dc line is aaa not xxx.
	IERR = 5	no CONs used in model; or
	IERR = 5	no input ICONs used in model.
	IERR = 6	CON number outside of the CON range of this model; or
	IERR = 6	ICON number outside of the ICON range of this model.
	IERR = 7	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.7.7 CHANGE_MTDCLMOD_ICON

Use this API to change the value of an integer ICON of the multi-terminal dc line model of a specified multi-terminal dc line. The corresponding character CHRICN is set to '$$'.

Batch command syntax:

```
BAT_CHANGE_MTDCLMOD_ICON  DC  'MODEL'  J  IDATA
```

Python syntax:

```
ierr = change_mtdclmod_icon(dc, model, j, idata)
```

Fortran syntax:

```
CALL CHANGE_MTDCLMOD_ICON(NUM, MODEL, J, IDATA, IERR)

CALL CHANGE_MTDCLMOD_ICON_I(NUM, MODEL, J, IDATA, IERR)

CALL CHANGE_MTDCLMOD_ICON_C(NAME, MODEL, J, IDATA, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the multi-terminal dc line for which model data is to be changed; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the multi-terminal dc line for which model data is to be changed (input; no default allowed).
Character*12 NAME	Is the name of the multi-terminal dc line for which model data is to be changed (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change relative to the start of the block of ICONs assigned to this model (1 for ICON(I), 2 for ICON(I+1), etc.) (input; no default allowed).
Integer IDATA	Is the new integer ICON value (input; unchanged by default).
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 dc line not found. IERR = 2 unknown model name. IERR = 3 no model at this dc line. IERR = 4 model at this dc line is aaa not xxx. IERR = 5 no input ICONs used in model. IERR = 6 ICON number outside of the ICON range of this model. IERR = 7 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.7.8 CHANGE_MTDCLMOD_VAR

Use this API to change the value of a VAR of the multi-terminal dc line model of a specified multi-terminal dc line.

Batch command syntax:

```
BAT_CHANGE_MTDCLMOD_VAR  DC  'MODEL'  J  RDATA
```

Python syntax:

```
ierr = change_mtdclmod_var(dc, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_MTDCLMOD_VAR(NUM, MODEL, J, RDATA, IERR)

CALL CHANGE_MTDCLMOD_VAR_I(NUM, MODEL, J, RDATA, IERR)

CALL CHANGE_MTDCLMOD_VAR_C(NAME, MODEL, J, RDATA, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the multi-terminal dc line for which model data is to be changed; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the multi-terminal dc line for which model data is to be changed (input; no default allowed).
Character*12 NAME	Is the name of the multi-terminal dc line for which model data is to be changed (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the VAR number to change relative to the start of the block of VARs assigned to this model (1 for VAR(J), 2 for VAR(J+1), etc.) (input; no default allowed).
Real RDATA	Is the new VAR value (input; unchanged by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — dc line not found.
	IERR = 2 — unknown model name.
	IERR = 3 — no model at this dc line.
	IERR = 4 — model at this dc line is aaa not xxx.
	IERR = 5 — no VARs used in model.
	IERR = 6 — VAR number outside of the VAR range of this model.
	IERR = 7 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.7.9 CHANGE_TTDCLMOD_CHRICN

Use this API to change the value of a character ICON (i.e., an element of the CHRICN array) of the two-terminal dc line model of a specified two-terminal dc line. The corresponding integer ICON is set to 0.

Batch command syntax:

```
BAT_CHANGE_TTDCLMOD_CHRICN  DC  'MODEL'  J  'CHDATA'
```

Python syntax:

```
ierr = change_ttdclmod_chricn(dc, model, j, chdata)
```

Fortran syntax:

```
CALL CHANGE_TTDCLMOD_CHRICN(NUM, MODEL, J, CHDATA, IERR)

CALL CHANGE_TTDCLMOD_CHRICN_I(NUM, MODEL, J, CHDATA, IERR)

CALL CHANGE_TTDCLMOD_CHRICN_C(NAME, MODEL, J, CHDATA, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the two-terminal dc line for which model data is to be changed; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the two-terminal dc line for which model data is to be changed (input; no default allowed).
Character*12 NAME	Is the name of the two-terminal dc line for which model data is to be changed (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change relative to the start of the block of ICONs assigned to this model (1 for ICON(I), 2 for ICON(I+1), etc.) (input; no default allowed).
Character*2 CHDATA	Is the new character ICON value (input; blank by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — dc line not found.
	IERR = 2 — unknown model name.
	IERR = 3 — no model at this dc line.
	IERR = 4 — model at this dc line is aaa not xxx.
	IERR = 5 — no input ICONs used in model.
	IERR = 6 — ICON number outside of the ICON range of this model.
	IERR = 7 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.7.10 CHANGE_TTDCLMOD_CON

Use this API to change the value of a CON of the two-terminal dc line model of a specified two-terminal dc line.

Batch command syntax:

```
BAT_CHANGE_TTDCLMOD_CON  DC  'MODEL'  J  RDATA
```

Python syntax:

```
ierr = change_ttdclmod_con(dc, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_TTDCLMOD_CON(NUM, MODEL, J, RDATA, IERR)

CALL CHANGE_TTDCLMOD_CON_I(NUM, MODEL, J, RDATA, IERR)

CALL CHANGE_TTDCLMOD_CON_C(NAME, MODEL, J, RDATA, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the two-terminal dc line for which model data is to be changed; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the two-terminal dc line for which model data is to be changed (input; no default allowed).
Character*12 NAME	Is the name of the two-terminal dc line for which model data is to be changed (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the CON number to change relative to the start of the block of CONs assigned to this model (1 for CON(J), 2 for CON(J+1), etc.) (input; no default allowed).
Real RDATA	Is the new CON value (input; unchanged by default).
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 dc line not found. IERR = 2 unknown model name. IERR = 3 no model at this dc line. IERR = 4 model at this dc line is aaa not xxx. IERR = 5 no CONs used in model. IERR = 6 CON number outside of the CON range of this model. IERR = 7 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.7.11 CHANGE_TTDCLMOD_DATA

This API is obsolete. It has been replaced by the APIs CHANGE_TTDCLMOD_CON, CHANGE_TTDCLMOD_ICON and CHANGE_TTDCLMOD_CHRICN.

Use this API to change the value of an ICON and/or CON of the two-terminal dc line model of a specified two-terminal dc line.

Batch command syntax:

```
BAT_CHANGE_TTDCLMOD_DATA  IDC  'NAME'  J  RDATA  I  IDATA  'CHDATA'
```

Python syntax:

```
ierr = change_ttdclmod_data(idc, name, j, rdata, i, idata, chdata)
```

Fortran syntax:

```
CALL CHANGE_TTDCLMOD_DATA(IDC,NAME,J,RDATA,I,IDATA,CHDATA,IERR)
```

where:

Integer IDC	Is the two-terminal dc line number (input; no default allowed).
Character*6 NAME	Is the model name (input; no default allowed).
Integer J	Is the CON number to change relative to the start of the block of CONs assigned to this model, or 0 for no CON change (1 for CON(J), 2 for CON(J+1), etc.) (input; 0 by default).
Real RDATA	Is the new CON value (input; unchanged by default).
Integer I	Is the ICON number to change relative to the start of the block of ICONs assigned to this model, or 0 for no ICON change (1 for ICON(I), 2 for ICON(I+1), etc.) (input; 0 by default).
Integer IDATA	Is the new integer ICON value (input; unchanged by default). When a character value is intended for an ICON value, the IDATA entry is ignored and the integer ICON is set to 0.
Character*2 CHDATA	Is the new character ICON value (input; no default allowed if the character ICON is to be changed). When an integer value is intended for an ICON value, the CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	dc line not found.
	IERR = 2	unknown model name.
	IERR = 3	no model at this dc line.
	IERR = 4	model at this dc line is aaa not xxx.
	IERR = 5	no CONs used in model; or
	IERR = 5	no input ICONs used in model.
	IERR = 6	CON number outside of the CON range of this model; or
	IERR = 6	ICON number outside of the ICON range of this model.
	IERR = 7	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.7.12 CHANGE_TTDCLMOD_ICON

Use this API to change the value of an integer ICON of the two-terminal dc line model of a specified two-terminal dc line. The corresponding character CHRICN is set to '$$'.

Batch command syntax:

```
BAT_CHANGE_TTDCLMOD_ICON  DC  'MODEL'  J  IDATA
```

Python syntax:

```
ierr = change_ttdclmod_icon(dc, model, j, idata)
```

Fortran syntax:

```
CALL CHANGE_TTDCLMOD_ICON(NUM, MODEL, J, IDATA, IERR)

CALL CHANGE_TTDCLMOD_ICON_I(NUM, MODEL, J, IDATA, IERR)

CALL CHANGE_TTDCLMOD_ICON_R(NAME, MODEL, J, IDATA, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the two-terminal dc line for which model data is to be changed; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the two-terminal dc line for which model data is to be changed (input; no default allowed).
Character*12 NAME	Is the name of the two-terminal dc line for which model data is to be changed (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change relative to the start of the block of ICONs assigned to this model (1 for ICON(I), 2 for ICON(I+1), etc.) (input; no default allowed).
Integer IDATA	Is the new integer ICON value (input; unchanged by default).
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 dc line not found. IERR = 2 unknown model name. IERR = 3 no model at this dc line. IERR = 4 model at this dc line is aaa not xxx. IERR = 5 no input ICONs used in model. IERR = 6 ICON number outside of the ICON range of this model. IERR = 7 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.7.13 CHANGE_TTDCLMOD_VAR

Use this API to change the value of a VAR of the two-terminal dc line model of a specified two-terminal dc line.

Batch command syntax:

```
BAT_CHANGE_TTDCLMOD_VAR  DC  'MODEL'  J  RDATA
```

Python syntax:

```
ierr = change_ttdclmod_var(dc, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_TTDCLMOD_VAR(NUM, MODEL, J, RDATA, IERR)

CALL CHANGE_TTDCLMOD_VAR_I(NUM, MODEL, J, RDATA, IERR)

CALL CHANGE_TTDCLMOD_VAR_C(NAME, MODEL, J, RDATA, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the two-terminal dc line for which model data is to be changed; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the two-terminal dc line for which model data is to be changed (input; no default allowed).
Character*12 NAME	Is the name of the two-terminal dc line for which model data is to be changed (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the VAR number to change relative to the start of the block of VARs assigned to this model (1 for VAR(J), 2 for VAR(J+1), etc.) (input; no default allowed).
Real RDATA	Is the new VAR value (input; unchanged by default).
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 dc line not found. IERR = 2 unknown model name. IERR = 3 no model at this dc line. IERR = 4 model at this dc line is aaa not xxx. IERR = 5 no VARs used in model. IERR = 6 VAR number outside of the VAR range of this model. IERR = 7 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.7.14 CHANGE_VSDCLMOD_CHRICN

Use this API to change the value of a character ICON (i.e., an element of the CHRICN array) of the VSC dc line model of a specified VSC dc line. The corresponding integer ICON is set to 0.

Batch command syntax:

```
BAT_CHANGE_VSDCLMOD_CHRICN  'DCNAME'  'MODEL'  J  'CHDATA'
```

Python syntax:

```
ierr = change_vsdclmod_chricn(dcname, model, j, chdata)
```

Fortran syntax:

```
CALL CHANGE_VSDCLMOD_CHRICN(DCNAME, MODEL, J, CHDATA, IERR)
```

where:

Character*12 DCNAME	Is the VSC dc line name (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change relative to the start of the block of ICONs assigned to this model (1 for ICON(I), 2 for ICON(I+1), etc.) (input; no default allowed).
Character*2 CHDATA	Is the new character ICON value (input; blank by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — dc line not found.
	IERR = 2 — unknown model name.
	IERR = 3 — no model at this dc line.
	IERR = 4 — model at this dc line is aaa not xxx.
	IERR = 5 — no input ICONs used in model.
	IERR = 6 — ICON number outside of the ICON range of this model.
	IERR = 7 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.7.15 CHANGE_VSDCLMOD_CON

Use this API to change the value of a CON of the VSC dc line model of a specified VSC dc line.

Batch command syntax:

```
BAT_CHANGE_VSDCLMOD_CON  'DCNAME'  'MODEL'  J  RDATA
```

Python syntax:

```
ierr = change_vsdclmod_con(dcname, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_VSDCLMOD_CON(DCNAME, MODEL, J, RDATA, IERR)
```

where:

Character*12 DCNAME	Is the VSC dc line name (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the CON number to change relative to the start of the block of CONs assigned to this model (1 for CON(J), 2 for CON(J+1), etc.) (input; no default allowed).
Real RDATA	Is the new CON value (input; unchanged by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — VSC dc line not found.
	IERR = 2 — unknown model name.
	IERR = 3 — no model at this dc line.
	IERR = 4 — model at this dc line is aaa not xxx.
	IERR = 5 — no CONs used in model.
	IERR = 6 — CON number outside of the CON range of this model.
	IERR = 7 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.7.16 CHANGE_VSDCLMOD_DATA

This API is obsolete. It has been replaced by the APIs CHANGE_VSDCLMOD_CON, CHANGE_VSDCLMOD_ICON and CHANGE_VSDCLMOD_CHRICN.

Use this API to change the value of an ICON and/or CON of the VSC dc line model of a specified VSC dc line.

Batch command syntax:

```
BAT_CHANGE_VSDCLMOD_DATA  'VSDCNM'  'NAME'  J  RDATA  I  IDATA
'CHDATA'
```

Python syntax:

```
ierr = change_vsdclmod_data(vsdcnm,name,j,rdata,i,idata,chdata)
```

Fortran syntax:

```
CALL CHANGE_VSDCLMOD_DATA(VSDCNM,NAME,J,RDATA,I,IDATA,CHDATA,IERR)
```

where:

Character*12 VSDCNM	Is the VSC dc line name (input; no default allowed).
Character*6 NAME	Is the model name (input; no default allowed).
Integer J	Is the CON number to change relative to the start of the block of CONs assigned to this model, or 0 for no CON change (1 for CON(J), 2 for CON(J+1), etc.) (input; 0 by default).
Real RDATA	Is the new CON value (input; unchanged by default).
Integer I	Is the ICON number to change relative to the start of the block of ICONs assigned to this model, or 0 for no ICON change (1 for ICON(I), 2 for ICON(I+1), etc.) (input; 0 by default).
Integer IDATA	Is the new integer ICON value (input; unchanged by default). When a character value is intended for an ICON value, the IDATA entry is ignored and the integer ICON is set to 0.
Character*2 CHDATA	Is the new character ICON value (input; no default allowed if the character ICON is to be changed). When an integer value is intended for an ICON value, the CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	dc line not found.
	IERR = 2	unknown model name.
	IERR = 3	no model at this dc line.
	IERR = 4	model at this dc line is aaa not xxx.
	IERR = 5	no CONs used in model; or
	IERR = 5	no input ICONs used in model.
	IERR = 6	CON number outside of the CON range of this model; or
	IERR = 6	ICON number outside of the ICON range of this model.
	IERR = 7	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.7.17 CHANGE_VSDCLMOD_ICON

Use this API to change the value of an integer ICON of the VSC dc line model of a specified VSC dc line. The corresponding character CHRICN is set to '$$.

Batch command syntax:

```
BAT_CHANGE_VSDCLMOD_ICON  'DCNAME'  'MODEL'  J  IDATA
```

Python syntax:

```
ierr = change_vsdclmod_icon(dcname, model, j, idata)
```

Fortran syntax:

```
CALL CHANGE_VSDCLMOD_ICON(DCNAME, MODEL, J, IDATA, IERR)
```

where:

Character*12 DCNAME	Is the VSC dc line name (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change relative to the start of the block of ICONs assigned to this model (1 for ICON(I), 2 for ICON(I+1), etc.) (input; no default allowed).
Integer IDATA	Is the new integer ICON value (input; unchanged by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — dc line not found.
	IERR = 2 — unknown model name.
	IERR = 3 — no model at this dc line.
	IERR = 4 — model at this dc line is aaa not xxx.
	IERR = 5 — no input ICONs used in model.
	IERR = 6 — ICON number outside of the ICON range of this model.
	IERR = 7 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.7.18 CHANGE_VSDCLMOD_VAR

Use this API to change the value of a VAR of the VSC dc line model of a specified VSC dc line.

Batch command syntax:

```
BAT_CHANGE_VSDCLMOD_VAR  'DCNAME'  'MODEL'  J  RDATA
```

Python syntax:

```
ierr = change_vsdclmod_var(dcname, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_VSDCLMOD_VAR(DCNAME, MODEL, J, RDATA, IERR)
```

where:

Character*12 DCNAME	Is the VSC dc line name (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the VAR number to change relative to the start of the block of VARs assigned to this model (1 for VAR(J), 2 for VAR(J+1), etc.) (input; no default allowed).
Real RDATA	Is the new VAR value (input; unchanged by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — VSC dc line not found.
	IERR = 2 — unknown model name.
	IERR = 3 — no model at this dc line.
	IERR = 4 — model at this dc line is aaa not xxx.
	IERR = 5 — no VARs used in model.
	IERR = 6 — VAR number outside of the VAR range of this model.
	IERR = 7 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.7.19 DCLMOD_PACK

Use this API to remove entries that are marked as unused from the dc line model connection tables and the dc line array allocation tables. It packs the entries remaining in these tables at the front of the various arrays.

Batch command syntax:

```
BAT_DCLMOD_PACK
```

Python syntax:

```
ierr = dclmod_pack()
```

Fortran syntax:

```
CALL DCLMOD_PACK(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.7.20 DCLMOD_UNCONNECTED

Use this API to list or remove from dynamics working memory those dc line models that are assigned to dc lines that are not present in the current power flow working case (unconnected).

Batch command syntax:

```
BAT_DCLMOD_UNCONNECTED  OPT
```

Python syntax:

```
ierr = dclmod_unconnected(opt)
```

Fortran syntax:

```
CALL DCLMOD_UNCONNECTED(OPT, IERR)
```

where:

Integer OPT	Is the function code (input; 0 by default):	
	OPT = 0	list unconnected dc line models.
	OPT = 1	remove unconnected dc line models.
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid OPT value.
	IERR = 2	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.7.21 DCLMOD_USER

Use this API to list user-written dc line model definitions or to remove user-written dc line model definitions that are not assigned to any dc lines (unused) from the user model definition tables.

Batch command syntax:

```
BAT_DCLMOD_USER  OPT
```

Python syntax:

```
ierr = dclmod_user(opt)
```

Fortran syntax:

```
CALL DCLMOD_USER(OPT, IERR)
```

where:

Integer OPT	Is the function code (input; 0 by default):	
	OPT = 0	list user-written dc line model definitions.
	OPT = 1	remove unused user-written dc line model definitions.
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid OPT value.
	IERR = 2	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.7.22 MTDCLMOD_REMOVE

Use this API to remove the multi-terminal dc line model from a specified multi-terminal dc line.

Batch command syntax:

```
BAT_MTDCLMOD_REMOVE  DC
```

Python syntax:

```
ierr = mtdclmod_remove(dc)
```

Fortran syntax:

```
CALL MTDCLMOD_REMOVE(NUM, IERR)

CALL MTDCLMOD_REMOVE_I(NUM, IERR)

CALL MTDCLMOD_REMOVE_C(NAME, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the multi-terminal dc line for which the model is to be removed; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the multi-terminal dc line for which the model is to be removed (input; no default allowed).
Character*12 NAME	Is the name of the multi-terminal dc line for which the model is to be removed (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 dc line not found. IERR = 2 no model at this dc line. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.7.23 MTDCLMOD_STATUS

Use this API to change the status of the multi-terminal dc line model at a specified multi-terminal dc line.

Batch command syntax:

```
BAT_MTDCLMOD_STATUS  DC  STATUS
```

Python syntax:

```
ierr = mtdclmod_status(dc, status)
```

Fortran syntax:

```
CALL MTDCLMOD_STATUS(NUM, STATUS, IERR)

CALL MTDCLMOD_STATUS_I(NUM, STATUS, IERR)

CALL MTDCLMOD_STATUS_R(NAME, STATUS, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the multi-terminal dc line for which the model status is to be changed; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the multi-terminal dc line for which the model status is to be changed (input; no default allowed).
Character*12 NAME	Is the name of the multi-terminal dc line for which the model status is to be changed (input; no default allowed).
Integer STATUS	Is the new model status (input; 1 by default). STATUS = 0 out-of-service. STATUS = 1 in-service.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 dc line not found. IERR = 2 no model at this dc line. IERR = 3 invalid STATUS value. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.7.24 TTDCLMOD_REMOVE

Use this API to remove the two-terminal dc line model from a specified two-terminal dc line.

Batch command syntax:

```
BAT_TTDCLMOD_REMOVE  DC
```

Python syntax:

```
ierr = ttdclmod_remove(dc)
```

Fortran syntax:

```
CALL TTDCLMOD_REMOVE(NUM, IERR)

CALL TTDCLMOD_REMOVE_I(NUM, IERR)

CALL TTDCLMOD_REMOVE_C(NAME, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the two-terminal dc line for which the model is to be removed; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the two-terminal dc line for which the model is to be removed (input; no default allowed).
Character*12 NAME	Is the name of the two-terminal dc line for which the model is to be removed (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 dc line not found. IERR = 2 no model at this dc line. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.7.25 TTDCLMOD_STATUS

Use this API to change the status of the two-terminal dc line model at a specified two-terminal end line.

Batch command syntax:

```
BAT_TTDCLMOD_STATUS  DC  STATUS
```

Python syntax:

```
ierr = ttdclmod_status(dc, status)
```

Fortran syntax:

```
CALL TTDCLMOD_STATUS(NUM, STATUS, IERR)

CALL TTDCLMOD_STATUS_I(NUM, STATUS, IERR)

CALL TTDCLMOD_STATUS_C(NAME, STATUS, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the two-terminal dc line for which the model status is to be changed; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the two-terminal dc line for which the model status is to be changed (input; no default allowed).
Character*12 NAME	Is the name of the two-terminal dc line for which the model status is to be changed (input; no default allowed).
Integer STATUS	Is the new model status (input; 1 by default). STATUS = 0 out-of-service. STATUS = 1 in-service.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 dc line not found. IERR = 2 no model at this dc line. IERR = 3 invalid STATUS value. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.7.26 VSDCLMOD_REMOVE

Use this API to remove the VSC dc line model from a specified VSC dc line.

Batch command syntax:

```
BAT_VSDCLMOD_REMOVE  'DCNAME'
```

Python syntax:

```
ierr = vsdclmod_remove(dcname)
```

Fortran syntax:

```
CALL VSDCLMOD_REMOVE(DCNAME, IERR)
```

where:

Character*12 DCNAME	Is the VSC dc line name (input; no default allowed).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	dc line not found.
	IERR = 2	no model at this dc line.
	IERR = 3	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.7.27 VSDCLMOD_STATUS

Use this API to change the status of the VSC dc line model at a specified VSC dc line.

Batch command syntax:

```
BAT_VSDCLMOD_STATUS  'DCNAME'  STATUS
```

Python syntax:

```
ierr = vsdclmod_status(dcname, status)
```

Fortran syntax:

```
CALL VSDCLMOD_STATUS(DCNAME, STATUS, IERR)
```

where:

Character*12 DCNAME	Is the VSC dc line name (input; no default allowed).
Integer STATUS	Is the new model status (input; 1 by default). STATUS = 0 out-of-service. STATUS = 1 in-service.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 dc line not found. IERR = 2 no model at this dc line. IERR = 3 invalid STATUS value. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.8 FACTS Device Model APIs

4.8.1 ADD_FACTS_DEVICE_MODEL

Use this API to add a FACTS device model to the specified FACTS device. If a model already exists at the FACTS device, it is replaced.

Batch command syntax:

```
BAT_ADD_FACTS_DEVICE_MODEL  FD  'MODEL'  NICN  IDATA  CHDATA  NCON
RDATA  IERR
```

Python syntax:

```
ierr = add_facts_device_model(fd,model,nicn,idata,chdata,ncon,rdata)
```

Fortran syntax:

```
CALL
ADD_FACTS_DEVICE_MODEL(NUM,MODEL,NICN,IDATA,CHDATA,NCON,RDATA,IERR)

CALL ADD_FACTS_DEVICE_MODEL_I(NUM, MODEL, NICN, IDATA, CHDATA, NCON,
RDATA, IERR)

CALL ADD_FACTS_DEVICE_MODEL_C(NAME, MODEL, NICN, IDATA, CHDATA, NCON,
RDATA, IERR)
```

where:

Character*12 or Integer FD	Is the quoted name of the FACTS device to which the model is to be added; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the FACTS device to which the model is to be added (input; no default allowed).
Character*12 NAME	Is the name of the FACTS device to which the model is to be added (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer NICN	Is the number of ICONs required by the model (input; no default allowed).
Integer IDATA(NICN)	Are the integer ICON data (input; 0 by default). When a character value is intended for an ICON value, the corresponding IDATA entry is ignored and the integer ICON is set to 0.
Character*2 CHDATA(NICN)	Are the character ICON data (input; no default allowed for an ICON for which a character value is to be specified). When an integer value is intended for an ICON value, the corresponding CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).
Integer NCON	Is the number of CONs required by the model (input; no default allowed).
Real RDATA(NCON)	Are the CON data (input; 0.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error
	IERR = 1	FACTS device not found
	IERR = 2	unknown model name
	IERR = 3	model is not a FACTS device model
	IERR = 4	# CONs specified is not the number expected
	IERR = 5	# ICONs specified is not the number expected
	IERR = 6	maximum number of array elements exceeded
	IERR = 7	FACTS device model connection tables full; use pack FACTS device model tables function
	IERR = 8	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.8.2 CHANGE_FCTMOD_CHRICN

Use this API to change the value of a character ICON (i.e., an element of the CHRICN array) of the FACTS device model of a specified FACTS device. The corresponding integer ICON is set to 0.

Batch command syntax:

```
BAT_CHANGE_FCTMOD_CHRICN  FD  'MODEL'  J  'CHDATA'
```

Python syntax:

```
ierr = change_fctmod_chricn(fd, model, j, chdata)
```

Fortran syntax:

```
CALL CHANGE_FCTMOD_CHRICN(NUM, MODEL, J, CHDATA, IERR)

CALL CHANGE_FCTMOD_CHRICN_I(NUM, MODEL, J, CHDATA, IERR)

CALL CHANGE_FCTMOD_CHRICN_C(NAME, MODEL, J, CHDATA, IERR)
```

where:

Character*12 or Integer FD	Is the quoted name of the FACTS device for which model data is to be changed; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the FACTS device for which model data is to be changed (input; no default allowed).
Character*12 NAME	Is the name of the FACTS device for which model data is to be changed (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change relative to the start of the block of ICONs assigned to this model (1 for ICON(I), 2 for ICON(I+1), etc.) (input; no default allowed).
Character*2 CHDATA	Is the new character ICON value (input; blank by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — FACTS device not found.
	IERR = 2 — unknown model name.
	IERR = 3 — no model at this FACTS device.
	IERR = 4 — model at this FACTS device is aaa not xxx.
	IERR = 5 — no input ICONs used in model.
	IERR = 6 — ICON number outside of the ICON range of this model.
	IERR = 7 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.8.3 CHANGE_FCTMOD_CON

Use this API to change the value of a CON of the FACTS device model of a specified FACTS device.

Batch command syntax:

```
BAT_CHANGE_FCTMOD_CON  FD  'MODEL'  J  RDATA
```

Python syntax:

```
ierr = change_fctmod_con(fd, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_FCTMOD_CON(NUM, MODEL, J, RDATA, IERR)

CALL CHANGE_FCTMOD_CON_I(NUM, MODEL, J, RDATA, IERR)

CALL CHANGE_FCTMOD_CON_C(NAME, MODEL, J, RDATA, IERR)
```

where:

Character*12 or Integer FD	Is the quoted name of the FACTS device for which model data is to be changed; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the FACTS device for which model data is to be changed (input; no default allowed).
Character*12 NAME	Is the name of the FACTS device for which model data is to be changed (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the CON number to change relative to the start of the block of CONs assigned to this model (1 for CON(J), 2 for CON(J+1), etc.) (input; no default allowed).
Real RDATA	Is the new CON value (input; unchanged by default).
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 FACTS device not found. IERR = 2 unknown model name. IERR = 3 no model at this FACTS device. IERR = 4 model at this FACTS device is aaa not xxx. IERR = 5 no CONs used in model. IERR = 6 CON number outside of the CON range of this model. IERR = 7 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.8.4 CHANGE_FCTMOD_DATA

This API is obsolete. It has been replaced by the APIs CHANGE_FCTMOD_CON, CHANGE_FCTMOD_ICON and CHANGE_FCTMOD_CHRICN.

Use this API to change the value of an ICON and/or CON of the FACTS device model of a specified FACTS device.

Batch command syntax:

```
BAT_CHANGE_FCTMOD_DATA  IDV  'NAME'  J  RDATA  I  IDATA  'CHDATA'
```

Python syntax:

```
ierr = change_fctmod_data(idv, name, j, rdata, i, idata, chdata)
```

Fortran syntax:

```
CALL CHANGE_FCTMOD_DATA(IDV,NAME,J,RDATA,I,IDATA,CHDATA,IERR)
```

where:

Integer IDV	Is the FACTS device number (input; no default allowed).
Character*6 NAME	Is the model name (input; no default allowed).
Integer J	Is the CON number to change relative to the start of the block of CONs assigned to this model, or 0 for no CON change (1 for CON(J), 2 for CON(J+1), etc.) (input; 0 by default).
Real RDATA	Is the new CON value (input; unchanged by default).
Integer I	Is the ICON number to change relative to the start of the block of ICONs assigned to this model, or 0 for no ICON change (1 for ICON(I), 2 for ICON(I+1), etc.) (input; 0 by default).
Integer IDATA	Is the new integer ICON value (input; unchanged by default). When a character value is intended for an ICON value, the IDATA entry is ignored and the integer ICON is set to 0.
Character*2 CHDATA	Is the new character ICON value (input; no default allowed if the character ICON is to be changed). When an integer value is intended for an ICON value, the CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	FACTS device not found.
	IERR = 2	unknown model name.
	IERR = 3	no model at this FACTS device.
	IERR = 4	model at this FACTS device is aaa not xxx.
	IERR = 5	no CONs used in model; or
	IERR = 5	no input ICONs used in model.
	IERR = 6	CON number outside of the CON range of this model; or
	IERR = 6	ICON number outside of the ICON range of this model.
	IERR = 7	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.8.5 CHANGE_FCTMOD_ICON

Use this API to change the value of an integer ICON of the FACTS device model of a specified FACTS device. The corresponding character CHRICN is set to '$$'.

Batch command syntax:

```
BAT_CHANGE_FCTMOD_ICON  FD  'MODEL'  J  IDATA
```

Python syntax:

```
ierr = change_fctmod_icon(fd, model, j, idata)
```

Fortran syntax:

```
CALL CHANGE_FCTMOD_ICON(NUM, MODEL, J, IDATA, IERR)

CALL CHANGE_FCTMOD_ICON_I(NUM, MODEL, J, IDATA, IERR)

CALL CHANGE_FCTMOD_ICON_C(NAME, MODEL, J, IDATA, IERR)
```

where:

Character*12 or Integer FD	Is the quoted name of the FACTS device for which model data is to be changed; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the FACTS device for which model data is to be changed (input; no default allowed).
Character*12 NAME	Is the name of the FACTS device for which model data is to be changed (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change relative to the start of the block of ICONs assigned to this model (1 for ICON(I), 2 for ICON(I+1), etc.) (input; no default allowed).
Integer IDATA	Is the new integer ICON value (input; unchanged by default).
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 FACTS device not found. IERR = 2 unknown model name. IERR = 3 no model at this FACTS device. IERR = 4 model at this FACTS device is aaa not xxx. IERR = 5 no input ICONs used in model. IERR = 6 ICON number outside of the ICON range of this model. IERR = 7 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.8.6 CHANGE_FCTMOD_VAR

Use this API to change the value of a VAR of the FACTS device model of a specified FACTS device.

Batch command syntax:

```
BAT_CHANGE_FCTMOD_VAR  FD  'MODEL'  J  RDATA
```

Python syntax:

```
ierr = change_fctmod_var(fd, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_FCTMOD_VAR(NUM, MODEL, J, RDATA, IERR)

CALL CHANGE_FCTMOD_VAR_I(NUM, MODEL, J, RDATA, IERR)

CALL CHANGE_FCTMOD_VAR_C(NAME, MODEL, J, RDATA, IERR)
```

where:

Character*12 or Integer FD	Is the quoted name of the FACTS device for which model data is to be changed; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the FACTS device for which model data is to be changed (input; no default allowed).
Character*12 NAME	Is the name of the FACTS device for which model data is to be changed (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the VAR number to change relative to the start of the block of VARs assigned to this model (1 for VAR(J), 2 for VAR(J+1), etc.) (input; no default allowed).
Real RDATA	Is the new VAR value (input; unchanged by default).
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 FACTS device not found. IERR = 2 unknown model name. IERR = 3 no model at this FACTS device. IERR = 4 model at this FACTS device is aaa not xxx. IERR = 5 no VARs used in model. IERR = 6 VAR number outside of the VAR range of this model. IERR = 7 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.8.7 FCTMOD_PACK

Use this API to remove entries that are marked as unused from the FACTS device model connection tables and the FACTS device model array allocation tables. It packs the entries remaining in these tables at the front of the various arrays.

Batch command syntax:

```
BAT_FCTMOD_PACK
```

Python syntax:

```
ierr = fctmod_pack()
```

Fortran syntax:

```
CALL FCTMOD_PACK(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.8.8 FCTMOD_REMOVE

Use this API to remove the FACTS device model from a specified FACTS device.

Batch command syntax:

```
BAT_FCTMOD_REMOVE  FD
```

Python syntax:

```
ierr = fctmod_remove(fd)
```

Fortran syntax:

```
CALL FCTMOD_REMOVE(NUM, IERR)

CALL FCTMOD_REMOVE_I(NUM, IERR)

CALL FCTMOD_REMOVE_C(NAME, IERR)
```

where:

Character*12 or Integer FD	Is the quoted name of the FACTS device for which the model is to be removed; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the FACTS device for which the model is to be removed (input; no default allowed).
Character*12 NAME	Is the name of the FACTS device for which the model is to be removed (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 FACTS device not found. IERR = 2 no model at this FACTS device. IERR = 3 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.8.9 FCTMOD_STATUS

Use this API to change the status of the FACTS device model at a specified FACTS device.

Batch command syntax:

```
BAT_FCTMOD_STATUS  FD  STATUS
```

Python syntax:

```
ierr = fctmod_status(fd, status)
```

Fortran syntax:

```
CALL FCTMOD_STATUS(NUM, STATUS, IERR)

CALL FCTMOD_STATUS_I(NUM, STATUS, IERR)

CALL FCTMOD_STATUS_C(NAME, STATUS, IERR)
```

where:

Character*12 or Integer FD	Is the quoted name of the FACTS device for which the model status is to be changed; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the FACTS device for which the model status is to be changed (input; no default allowed).
Character*12 NAME	Is the name of the FACTS device for which the model status is to be changed (input; no default allowed).
Integer STATUS	Is the new model status (input; 1 by default). STATUS = 0 out-of-service. STATUS = 1 in-service.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 FACTS device not found. IERR = 2 no model at this FACTS device. IERR = 3 invalid STATUS value. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.8.10 FCTMOD_UNCONNECTED

Use this API to list or remove from dynamics working memory those FACTS device models that are assigned to FACTS device that are not present in the current power flow working case (unconnected).

Batch command syntax:

```
BAT_FCTMOD_UNCONNECTED  OPT
```

Python syntax:

```
ierr = fctmod_unconnected(opt)
```

Fortran syntax:

```
CALL FCTMOD_UNCONNECTED(OPT, IERR)
```

where:

Integer OPT	Is the function code (input; 0 by default): OPT = 0 list unconnected FACTS device models. OPT = 1 remove unconnected FACTS device models.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid OPT value. IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.8.11 FCTMOD_USER

Use this API to list user-written FACTS device model definitions or to remove user-written FACTS device model definitions that are not assigned to any FACTS devices (unused) from the user model definition tables.

Batch command syntax:

```
BAT_FCTMOD_USER  OPT
```

Python syntax:

```
ierr = fctmod_user(opt)
```

Fortran syntax:

```
CALL FCTMOD_USER(OPT, IERR)
```

where:

Integer OPT	Is the function code (input; 0 by default):	
	OPT = 0	list user-written FACTS device model definitions.
	OPT = 1	remove unused user-written FACTS device model definitions.
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid OPT value.
	IERR = 2	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.8.12 GMB_ADD_FACTS_DEV_MODEL

Use this API to add a GMB FACTS device model to the specified FACTS device. If a model already exists at the FACTS device, it is replaced.

Batch command syntax:

```
BAT_GMB_ADD_FACTS_DEV_MODEL  FD  'MODEL'  NICN  IDATA  CHDATA  NCON
RDATA  IERR
```

Python syntax:

```
ierr = gmb_add_facts_dev_model(fd,model,nicn,idata,chdata,ncon,rdata)
```

Fortran syntax:

```
CALL GMB_ADD_FACTS_DEV_MODEL(NAME,MODEL,NICN,IDATA,CHDATA,NCON,RDATA,
IERR)
```

where:

Character*12 or FD	Is the quoted name of the FACTS device to which the model is to be added; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the FACTS device to which the model is to be added (input; no default allowed).
Character*12 NAME	Is the name of the FACTS device to which the model is to be added (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer NICN	Is the number of ICONs required by the model (input; no default allowed).
Integer IDATA(NICN)	Are the integer ICON data (input; 0 by default). When a character value is intended for an ICON value, the corresponding IDATA entry is ignored and the integer ICON is set to 0.
Character*2 CHDATA(NICN)	Are the character ICON data (input; no default allowed for an ICON for which a character value is to be specified). When an integer value is intended for an ICON value, the corresponding CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).
Integer NCON	Is the number of CONs required by the model (input; no default allowed).
Real RDATA(NCON)	Are the CON data (input; 0.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error
IERR = 1	FACTS device not found
IERR = 2	error reading GMB model name
IERR = 3	model is not a FACTS device model
IERR = 4	# CONs specified is not the number expected
IERR = 5	# ICONs specified is not the number expected
IERR = 6	maximum number of array elements exceeded
IERR = 7	FACTS device model connection tables full; use pack FACTS device model tables function
IERR = 8	unlicensed GMB model name
IERR = 9	number of user-defined models exceeds maximum allowed
IERR = 10	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.9 Switched Shunt Model APIs

4.9.1 ADD_SWSHUNT_MODEL

Use this API to add a switched shunt model to the specified switched shunt device. If a model already exists at the switched shunt, it is replaced.

Batch command syntax:

```
BAT_ADD_SWSHUNT_MODEL  IBUS  'MODEL'  NICN  IDATA  CHDATA  NCON
RDATA  IERR
```

Python syntax:

```
ierr = add_swshunt_model(ibus,model,nicn,idata,chdata,ncon,rdata)
```

Fortran syntax:

```
CALL ADD_SWSHUNT_MODEL(IBUS,MODEL,NICN,IDATA,CHDATA,NCON,RDATA,IERR)
```

where:

Integer IBUS	Is the switched shunt bus number (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer NICN	Is the number of ICONs required by the model (input; no default allowed).
Integer IDATA(NICN)	Are the integer ICON data (input; 0 by default). When a character value is intended for an ICON value, the corresponding IDATA entry is ignored and the integer ICON is set to 0.
Character*2 CHDATA(NICN)	Are the character ICON data (input; no default allowed for an ICON for which a character value is to be specified). When an integer value is intended for an ICON value, the corresponding CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).
Integer NCON	Is the number of CONs required by the model (input; no default allowed).
Real RDATA(NCON)	Are the CON data (input; 0.0 by default).

Integer IERR	Is the error code (output).
	IERR = 0 no error
	IERR = 1 unknown model name
	IERR = 2 model is not a switched shunt model
	IERR = 3 # CONs specified is not the number expected
	IERR = 4 # ICONs specified is not the number expected
	IERR = 5 maximum number of array elements exceeded
	IERR = 6 switched shunt model connection tables full--use pack switched shunt model tables function
	IERR = 7 bus not found
	IERR = 8 switched shunt not found
	IERR = 9 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.9.2 CHANGE_SWSMOD_CHRICN

Use this API to change the value of a character ICON (i.e., an element of the CHRICN array) of the switched shunt model of a specified switched shunt device. The corresponding integer ICON is set to 0.

Batch command syntax:

```
BAT_CHANGE_SWSMOD_CHRICN  IBUS  'MODEL'  J  'CHDATA'
```

Python syntax:

```
ierr = change_swsmod_chricn(ibus, model, j, chdata)
```

Fortran syntax:

```
CALL CHANGE_SWSMOD_CHRICN(IBUS, MODEL, J, CHDATA, IERR)
```

where:

Integer IBUS	Is the bus number of the switched shunt for which model data is to be changed (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change relative to the start of the block of ICONs assigned to this model (1 for ICON(I), 2 for ICON(I+1), etc.) (input; no default allowed).
Character*2 CHDATA	Is the new character ICON value (input; blank by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — unknown model name.
	IERR = 2 — no model at this switched shunt device.
	IERR = 3 — model at this switched shunt is aaa not xxx.
	IERR = 4 — no input ICONs used in model.
	IERR = 5 — ICON number outside of the ICON range of this model.
	IERR = 6 — bus not found.
	IERR = 7 — switched shunt not found.
	IERR = 8 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.9.3 CHANGE_SWSMOD_CON

Use this API to change the value of a CON of the switched shunt model of a specified switched shunt device.

Batch command syntax:

```
BAT_CHANGE_SWSMOD_CON  IBUS  'MODEL'  J  RDATA
```

Python syntax:

```
ierr = change_swsmod_con(ibus, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_SWSMOD_CON(IBUS, MODEL, J, RDATA, IERR)
```

where:

Integer IBUS	Is the bus number of the switched shunt for which model data is to be changed (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the CON number to change relative to the start of the block of CONs assigned to this model (1 for CON(J), 2 for CON(J+1), etc.) (input; no default allowed).
Real RDATA	Is the new CON value (input; unchanged by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — unknown model name.
	IERR = 2 — no model at this switched shunt device.
	IERR = 3 — model at this switched shunt is aaa not xxx.
	IERR = 4 — no input CONs used in model.
	IERR = 5 — CON number outside of the CON range of this model.
	IERR = 6 — bus not found.
	IERR = 7 — switched shunt not found.
	IERR = 8 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.9.4 CHANGE_SWSMOD_ICON

Use this API to change the value of an integer ICON (i.e., an element of the ICON array) of the switched shunt model of a specified switched shunt device. The corresponding character CHRICN is set to '$$'.

Batch command syntax:

```
BAT_CHANGE_SWSMOD_ICON  IBUS  'MODEL'  J  IDATA
```

Python syntax:

```
ierr = change_swsmod_icon(ibus, model, j, idata)
```

Fortran syntax:

```
CALL CHANGE_SWSMOD_ICON(IBUS, MODEL, J, IDATA, IERR)
```

where:

Integer IBUS	Is the bus number of the switched shunt for which model data is to be changed (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change relative to the start of the block of ICONs assigned to this model (1 for ICON(I), 2 for ICON(I+1), etc.) (input; no default allowed).
Integer IDATA	Is the new integer ICON value (input; blank by default).
Integer IERR	Is the error code (output).
	IERR = 0 no error.
	IERR = 1 unknown model name.
	IERR = 2 no model at this switched shunt device.
	IERR = 3 model at this switched shunt is aaa not xxx.
	IERR = 4 no input ICONs used in model.
	IERR = 5 ICON number outside of the ICON range of this model.
	IERR = 6 bus not found.
	IERR = 7 switched shunt not found.
	IERR = 8 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.9.5 CHANGE_SWSMOD_VAR

Use this API to change the value of a VAR of the switched shunt model of a specified switched shunt device.

Batch command syntax:

```
BAT_CHANGE_SWSMOD_VAR  IBUS  'MODEL'  J  RDATA
```

Python syntax:

```
ierr = change_swsmod_var(ibus, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_SWSMOD_VAR(IBUS, MODEL, J, RDATA, IERR)
```

where:

Integer IBUS	Is the bus number of the switched shunt for which model data is to be changed (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer j	Is the VAR number to change relative to the start of the block of VARs assigned to this model (1 for VAR(J), 2 for VAR(J+1), etc.) (input; no default allowed).
Real RDATA	Is the new VAR value (input; unchanged by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	unknown model name
IERR = 2	no model at this switched shunt device
IERR = 3	model at this switched shunt is aaa not xxx
IERR = 4	no input VARs used in model
IERR = 5	VAR number outside of the VAR range of this model
IERR = 6	bus not found
IERR = 7	switched shunt not found
IERR = 8	prerequisite requirements for API are not met

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.9.6 GMB_ADD_SWSHUNT_MODEL

Use this API to add a GMB switched shunt model to the specified switched shunt device. If a model already exists at the switched shunt, it is replaced.

Batch command syntax:

```
BAT_GMB_ADD_SWSHUNT_MODEL  IBUS  'MODEL'  NICN  IDATA  CHDATA  NCON
RDATA  IERR
```

Python syntax:

```
ierr = gmb_add_swshunt_model(ibus,model,nicn,idata,chdata,ncon,rdata)
```

Fortran syntax:

```
CALL
GMB_ADD_SWSHUNT_MODEL(IBUS,MODEL,NICN,IDATA,CHDATA,NCON,RDATA,IERR)
```

where:

Integer IBUS	Is the switched shunt bus number (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer NICN	Is the number of ICONs required by the model (input; no default allowed).
Integer IDATA(NICN)	Are the integer ICON data (input; 0 by default). When a character value is intended for an ICON value, the corresponding IDATA entry is ignored and the integer ICON is set to 0.
Character*2 CHDATA(NICN)	Are the character ICON data (input; no default allowed for an ICON for which a character value is to be specified). When an integer value is intended for an ICON value, the corresponding CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).
Integer NCON	Is the number of CONs required by the model (input; no default allowed).
Real RDATA(NCON)	Are the CON data (input; 0.0 by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	error reading GMB model name
IERR = 2	model is not a switched shunt model
IERR = 3	# CONs specified is not the number expected
IERR = 4	# ICONs specified is not the number expected
IERR = 5	maximum number of array elements exceeded
IERR = 6	switched shunt model connection tables full--use pack switched shunt model tables function
IERR = 7	unlicensed GMB model name
IERR = 8	number of user-defined models exceeds maximum allowed
IERR = 9	bus not found
IERR = 10	switched shunt not found
IERR = 11	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.9.7 SWSMOD_PACK

Use this API to remove entries that are marked as unused from the switched shunt model connection tables and the switched shunt model array allocation tables. It packs the entries remaining in these tables at the front of the various arrays.

Batch command syntax:

```
BAT_SWSMOD_PACK
```

Python syntax:

```
ierr = swsmod_pack()
```

Fortran syntax:

```
CALL SWSMOD_PACK(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.9.8 SWSMOD_REMOVE

Use this API to remove the switched shunt model from a specified switched shunt.

Batch command syntax:

```
BAT_SWSMOD_REMOVE  IBUS
```

Python syntax:

```
ierr = swsmod_remove(ibus)
```

Fortran syntax:

```
CALL SWSMOD_REMOVE(IBUS, IERR)
```

where:

Integer IBUS	Is the bus number of the switched shunt for which the model is to be removed (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 no error.
	IERR = 1 no model at this switched shunt device.
	IERR = 2 Bus not found.
	IERR = 3 switched shunt not found.
	IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.9.9 SWSMOD_STATUS

Use this API to change the status of the switched shunt model at a specified switched shunt device.

Batch command syntax:

```
BAT_SWSMOD_STATUS  IBUS  STATUS
```

Python syntax:

```
ierr = swsmod_status(ibus, status)
```

Fortran syntax:

```
CALL SWSMOD_STATUS(IBUS, STATUS, IERR)
```

where:

Integer IBUS	Is the bus number of the switched shunt for which the model status is to be changed (input; no default allowed).
Integer STATUS	Is the new model status (input; 1 by default). STATUS = 0 out-of-service. STATUS = 1 in-service.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 no model at this switched shunt device. IERR = 2 invalid STATUS value. IERR = 3 Bus not found. IERR = 4 switched shunt not found. IERR = 5 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.9.10 SWSMOD_UNCONNECTED

Use this API to list or remove from dynamics working memory those switched shunt models that are assigned to switched shunt that are not present in the current power flow working case (unconnected).

Batch command syntax:

```
BAT_SWSMOD_UNCONNECTED  OPT
```

Python syntax:

```
ierr = swsmod_unconnected(opt)
```

Fortran syntax:

```
CALL SWSMOD_UNCONNECTED(OPT, IERR)
```

where:

Integer OPT	Is the function code (input; 0 by default): OPT = 0 list unconnected switched shunt models. OPT = 1 remove unconnected switched shunt models.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid OPT value. IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.9.11 SWSMOD_USER

Use this API to list user-written switched shunt model definitions or to remove user-written switched shunt model definitions that are not assigned to any switched shunt devices (unused) from the user model definition tables.

Batch command syntax:

```
BAT_SWSMOD_USER  OPT
```

Python syntax:

```
ierr = swsmod_user(opt)
```

Fortran syntax:

```
CALL SWSMOD_USER(OPT, IERR)
```

where:

Integer OPT	Is the function code (input; 0 by default):	
	OPT = 0	list user-written switched shunt model definitions.
	OPT = 1	remove unused user-written switched shunt model definitions.
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid OPT value.
	IERR = 2	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10 Auxiliary Signal Model APIs

4.10.1 ADD_FCTSAUXSIGNAL_MODEL

Use this API to add a FACTS device auxiliary signal model at the specified injection point of the designated FACTS device. If a model already exists at the specified injection point of the FACTS device, it is replaced.

Batch command syntax:

```
BAT_ADD_FCTSAUXSIGNAL_MODEL  FD  ISGNDX  'MODEL'  NICN  IDATA  CHDATA
NCON  RDATA  IERR
```

Python syntax:

```
ierr = add_fctsauxsignal_model(fd, isgndx, model, nicn, idata,
chdata, ncon, rdata)
```

Fortran syntax:

```
CALL ADD_FCTSAUXSIGNAL_MODEL(NUM, ISGNDX, MODEL, NICN, IDATA, CHDATA,
NCON, RDATA, IERR)

CALL ADD_FCTSAUXSIGNAL_MODEL_I(NUM, ISGNDX, MODEL, NICN, IDATA,
CHDATA, NCON, RDATA, IERR)

CALL ADD_FCTSAUXSIGNAL_MODEL_C(NAME, ISGNDX, MODEL, NICN, IDATA,
CHDATA, NCON, RDATA, IERR)
```

where:

Character*12 or Integer FD	Is the quoted name of the FACTS device receiving the output of the auxiliary signal model; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the FACTS device receiving the output of the auxiliary signal model (input; no default allowed).
Character*12 NAME	Is the name of the FACTS device receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer NICN	Is the number of ICONs required by the model (input; no default allowed).
Integer IDATA(NICN)	Are the integer ICON data (input; 0 by default). When a character value is intended for an ICON value, the corresponding IDATA entry is ignored and the integer ICON is set to 0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character*2 CHDATA(NICN)	Are the character ICON data (input; no default allowed for an ICON for which a character value is to be specified). When an integer value is intended for an ICON value, the corresponding CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).
Integer NCON	Is the number of CONs required by the model (input; no default allowed).
Real RDATA(NCON)	Are the CON data (input; 0.0 by default).
Integer IERR	Is the error code (output).

IERR = 0	no error.
IERR = 1	FACTS device not found.
IERR = 2	invalid ISGNDX value for a FACTS device.
IERR = 3	unknown model name.
IERR = 4	model is not an auxiliary signal model.
IERR = 5	# CONs specified is not the number expected.
IERR = 6	# ICONs specified is not the number expected.
IERR = 7	maximum number of array elements exceeded.
IERR = 8	auxiliary signal model connection tables full--use pack auxiliary signal model tables function.
IERR = 9	this auxiliary signal may be used only with two-terminal dc lines.
IERR = 10	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.2 ADD_MTDCAUXSIGNAL_MODEL

Use this API to add a multi-terminal dc line auxiliary signal model at the specified auxiliary signal index of the designated multi-terminal dc line. If a model already exists at the specified auxiliary signal index of the multi-terminal dc line, it is replaced.

Batch command syntax:

```
BAT_ADD_MTDCAUXSIGNAL_MODEL  DC  ISGNDX  'MODEL'  NICN  IDATA  CHDATA
NCON  RDATA  IERR
```

Python syntax:

```
ierr = add_mtdcauxsignal_model(dc, isgndx, model, nicn, idata,
chdata, ncon, rdata)
```

Fortran syntax:

```
CALL ADD_MTDCAUXSIGNAL_MODEL(NUM, ISGNDX, MODEL, NICN, IDATA, CHDATA,
NCON, RDATA, IERR)

CALL ADD_MTDCAUXSIGNAL_MODEL_I(NUM, ISGNDX, MODEL, NICN, IDATA,
CHDATA, NCON, RDATA, IERR)

CALL ADD_MTDCAUXSIGNAL_MODEL_C(NAME, ISGNDX, MODEL, NICN, IDATA,
CHDATA, NCON, RDATA, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the multi-terminal dc line receiving the output of the auxiliary signal model; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the multi-terminal dc line receiving the output of the auxiliary signal model (input; no default allowed).
Character*12 NAME	Is the name of the multi-terminal dc line receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer NICN	Is the number of ICONs required by the model (input; no default allowed).
Integer IDATA(NICN)	Are the integer ICON data (input; 0 by default). When a character value is intended for an ICON value, the corresponding IDATA entry is ignored and the integer ICON is set to 0.
Character*2 CHDATA(NICN)	Are the character ICON data (input; no default allowed for an ICON for which a character value is to be specified). When an integer value is intended for an ICON value, the corresponding CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).
Integer NCON	Is the number of CONs required by the model (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real RDATA(NCON)	Are the CON data (input; 0.0 by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	dc line not found.
	IERR = 2	invalid ISGNDX value for a multi-terminal dc line.
	IERR = 3	unknown model name.
	IERR = 4	model is not an auxiliary signal model.
	IERR = 5	# CONs specified is not the number expected.
	IERR = 6	# ICONs specified is not the number expected.
	IERR = 7	maximum number of array elements exceeded.
	IERR = 8	auxiliary signal model connection tables full--use pack auxiliary signal model tables function.
	IERR = 9	this auxiliary signal may be used only with two-terminal dc lines.
	IERR = 10	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.3 ADD_TTDCAUXSIGNAL_MODEL

Use this API to add a two-terminal dc line auxiliary signal model at the specified injection point of the designated two-terminal dc line. If a model already exists at the specified injection point of the two-terminal dc line, it is replaced.

Batch command syntax:

```
BAT_ADD_TTDCAUXSIGNAL_MODEL  DC  ISGNDX  'MODEL'  NICN  IDATA  CHDATA
NCON  RDATA  IERR
```

Python syntax:

```
ierr = add_ttdcauxsignal_model(dc, isgndx, model, nicn, idata,
chdata, ncon, rdata)
```

Fortran syntax:

```
CALL ADD_TTDCAUXSIGNAL_MODEL(NUM, ISGNDX, MODEL, NICN, IDATA, CHDATA,
NCON, RDATA, IERR)

CALL ADD_TTDCAUXSIGNAL_MODEL_I(NUM, ISGNDX, MODEL, NICN, IDATA,
CHDATA, NCON, RDATA, IERR)

CALL ADD_TTDCAUXSIGNAL_MODEL_C(NAME, ISGNDX, MODEL, NICN, IDATA,
CHDATA, NCON, RDATA, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the two-terminal dc line receiving the output of the auxiliary signal model; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the two-terminal dc line receiving the output of the auxiliary signal model (input; no default allowed).
Character*12 NAME	Is the name of the two-terminal dc line receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer NICN	Is the number of ICONs required by the model (input; no default allowed).
Integer IDATA(NICN)	Are the integer ICON data (input; 0 by default). When a character value is intended for an ICON value, the corresponding IDATA entry is ignored and the integer ICON is set to 0.
Character*2 CHDATA(NICN)	Are the character ICON data (input; no default allowed for an ICON for which a character value is to be specified). When an integer value is intended for an ICON value, the corresponding CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).
Integer NCON	Is the number of CONs required by the model (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real RDATA(NCON)	Are the CON data (input; 0.0 by default).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	dc line not found.
	IERR = 2	invalid ISGNDX value for a two-terminal dc line.
	IERR = 3	unknown model name.
	IERR = 4	model is not an auxiliary signal model.
	IERR = 5	# CONs specified is not the number expected.
	IERR = 6	# ICONs specified is not the number expected.
	IERR = 7	maximum number of array elements exceeded.
	IERR = 8	auxiliary signal model connection tables full--use pack auxiliary signal model tables function.
	IERR = 10	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.4 ADD_VSDCAUXSIGNAL_MODEL

Use this API to add a VSC dc line auxiliary signal model at the specified injection point of the designated VSC dc line. If a model already exists at the specified injection point of the VSC dc line, it is replaced.

Batch command syntax:

```
BAT_ADD_VSDCAUXSIGNAL_MODEL  'DCNAME'  ISGNDX  'MODEL'  NICN  IDATA
CHDATA  NCON  RDATA  IERR
```

Python syntax:

```
ierr = add_vsdcauxsignal_model(dcname, isgndx, model, nicn, idata,
chdata, ncon, rdata)
```

Fortran syntax:

```
CALL ADD_VSDCAUXSIGNAL_MODEL(DCNAME, ISGNDX, MODEL, NICN, IDATA,
CHDATA, NCON, RDATA, IERR)
```

where:

Character*12 DCNAME	Is the name of the VSC dc line receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer NICN	Is the number of ICONs required by the model (input; no default allowed).
Integer IDATA(NICN)	Are the integer ICON data (input; 0 by default). When a character value is intended for an ICON value, the corresponding IDATA entry is ignored and the integer ICON is set to 0.
Character*2 CHDATA(NICN)	Are the character ICON data (input; no default allowed for an ICON for which a character value is to be specified). When an integer value is intended for an ICON value, the corresponding CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).
Integer NCON	Is the number of CONs required by the model (input; no default allowed).
Real RDATA(NCON)	Are the CON data (input; 0.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	dc line not found.
	IERR = 2	invalid ISGNDX value for a VSC dc line.
	IERR = 3	unknown model name.
	IERR = 4	model is not an auxiliary signal model.
	IERR = 5	# CONs specified is not the number expected.
	IERR = 6	# ICONs specified is not the number expected.
	IERR = 7	maximum number of array elements exceeded.
	IERR = 8	auxiliary signal model connection tables full--use pack auxiliary signal model tables function.
	IERR = 9	this auxiliary signal may be used only with two-terminal dc lines.
	IERR = 10	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.5 AUXMOD_PACK

Use this API to remove entries that are marked as unused from the auxiliary signal model connection tables and the auxiliary signal array allocation tables. It packs the entries remaining in these tables at the front of the various arrays.

Batch command syntax:

```
BAT_AUXMOD_PACK
```

Python syntax:

```
ierr = auxmod_pack()
```

Fortran syntax:

```
CALL AUXMOD_PACK(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.6 AUXMOD_UNCONNECTED

Use this API to list or remove from dynamics working memory those auxiliary signal models that are assigned to dc lines or FACTS devices that are not present in the current power flow working case (unconnected).

Batch command syntax:

```
BAT_AUXMOD_UNCONNECTED  OPT
```

Python syntax:

```
ierr = auxmod_unconnected(opt)
```

Fortran syntax:

```
CALL AUXMOD_UNCONNECTED(OPT, IERR)
```

where:

Integer OPT	Is the function code (input; 0 by default): OPT = 0 list unconnected auxiliary signal models. OPT = 1 remove unconnected auxiliary signal models.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 invalid OPT value. IERR = 2 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.7 AUXMOD_USER

Use this API to list user-written auxiliary signal model definitions or to remove user-written auxiliary signal model definitions that are not assigned to any dc lines or FACTS devices (unused) from the user model definition tables.

Batch command syntax:

```
BAT_AUXMOD_USER  OPT
```

Python syntax:

```
ierr = auxmod_user(opt)
```

Fortran syntax:

```
CALL AUXMOD_USER(OPT, IERR)
```

where:

Integer OPT	Is the function code (input; 0 by default):	
	OPT = 0	list user-written auxiliary signal model definitions.
	OPT = 1	remove unused user-written auxiliary signal model definitions.
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid OPT value.
	IERR = 2	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.8 CHANGE_FCTSAUXMOD_CHRICN

Use this API to change the value of a character ICON (i.e., an element of the CHRICN array) of an auxiliary signal model of a specified FACTS device. The corresponding integer ICON is set to 0.

Batch command syntax:

```
BAT_CHANGE_FCTSAUXMOD_CHRICN  FD  ISGNDX  'MODEL'  J  'CHDATA'
```

Python syntax:

```
ierr = change_fctsauxmod_chricn(fd, isgndx, model, j, chdata)
```

Fortran syntax:

```
CALL CHANGE_FCTSAUXMOD_CHRICN(NUM, ISGNDX, MODEL, J, CHDATA, IERR)

CALL CHANGE_FCTSAUXMOD_CHRICN_I(NUM,ISGNDX,MODEL,J,CHDATA,IERR)

CALL CHANGE_FCTSAUXMOD_CHRICN_C(NAME,ISGNDX,MODEL,J,CHDATA,IERR)
```

where:

Character*12 or Integer FD	Is the quoted name of the FACTS device receiving the output of the auxiliary signal model; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the FACTS device receiving the output of the auxiliary signal model (input; no default allowed).
Character*12 NAME	Is the name of the FACTS device receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change relative to the start of the block of ICONs assigned to this model (1 for ICON(I), 2 for ICON(I+1), etc.) (input; no default allowed).
Character*2 CHDATA	Is the new character ICON value (input; blank by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error.
IERR = 1	FACTS device not found.
IERR = 2	invalid ISGNDX value for a FACTS device.
IERR = 3	unknown model name.
IERR = 4	no auxiliary signal model at this index for this FACTS device.
IERR = 5	model at this index for this FACTS device is aaa not xxx.
IERR = 6	no input ICONs used in model.
IERR = 7	ICON number outside of the ICON range of this model.
IERR = 8	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.9 CHANGE_FCTSAUXMOD_CON

Use this API to change the value of a CON of an auxiliary signal model of a specified FACTS device.

Batch command syntax:

```
BAT_CHANGE_FCTSAUXMOD_CON  FD  ISGNDX  'MODEL'  J  RDATA
```

Python syntax:

```
ierr = change_fctsauxmod_con(fd, isgndx, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_FCTSAUXMOD_CON(NUM, ISGNDX, MODEL, J, RDATA, IERR)

CALL CHANGE_FCTSAUXMOD_CON_I(NUM, ISGNDX, MODEL, J, RDATA, IERR)

CALL CHANGE_FCTSAUXMOD_CON_C(NAME, ISGNDX, MODEL, J, RDATA, IERR)
```

where:

Character*12 or Integer FD	Is the quoted name of the FACTS device receiving the output of the auxiliary signal model; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the FACTS device receiving the output of the auxiliary signal model (input; no default allowed).
Character*12 NAME	Is the name of the FACTS device receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the CON number to change relative to the start of the block of CONs assigned to this model (1 for CON(J), 2 for CON(J+1), etc.) (input; no default allowed).
Real RDATA	Is the new CON value (input; unchanged by default).
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 FACTS device not found. IERR = 2 invalid ISGNDX value for a FACTS device. IERR = 3 unknown model name. IERR = 4 no auxiliary signal model at this index for this FACTS device. IERR = 5 model at this index for this FACTS device is aaa not xxx. IERR = 6 no CONs used in model. IERR = 7 CON number outside of the CON range of this model. IERR = 8 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.10 CHANGE_FCTSAUXMOD_DATA

This API is obsolete. It has been replaced by the APIs CHANGE_FCTSAUXMOD_CON, CHANGE_FCTSAUXMOD_ICON and CHANGE_FCTSAUXMOD_CHRICN.

Use this API to change the value of an ICON and/or CON of an auxiliary signal model of a specified FACTS device.

Batch command syntax:

```
BAT_CHANGE_FCTSAUXMOD_DATA  IDV  ISGNDX  'NAME'  J  RDATA  I  IDATA
'CHDATA'
```

Python syntax:

```
ierr = change_fctsauxmod_data(idv,isgndx,name,j,rdata,i,idata,chdata)
```

Fortran syntax:

```
CALL
CHANGE_FCTSAUXMOD_DATA(IDV,ISGNDX,NAME,J,RDATA,I,IDATA,CHDATA,IERR)
```

where:

Integer IDV	Is the number of the FACTS device receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Character*6 NAME	Is the model name (input; no default allowed).
Integer J	Is the CON number to change relative to the start of the block of CONs assigned to this model, or 0 for no CON change (1 for CON(J), 2 for CON(J+1), etc.) (input; 0 by default).
Real RDATA	Is the new CON value (input; unchanged by default).
Integer I	Is the ICON number to change relative to the start of the block of ICONs assigned to this model, or 0 for no ICON change (1 for ICON(I), 2 for ICON(I+1), etc.) (input; 0 by default).
Integer IDATA	Is the new integer ICON value (input; unchanged by default). When a character value is intended for an ICON value, the IDATA entry is ignored and the integer ICON is set to 0.
Character*2 CHDATA	Is the new character ICON value (input; no default allowed if the character ICON is to be changed). When an integer value is intended for an ICON value, the CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	FACTS device not found.
	IERR = 2	invalid ISGNDX value for a FACTS device.
	IERR = 3	unknown model name.
	IERR = 4	no auxiliary signal model at this index for this FACTS device.
	IERR = 5	model at this index for this FACTS device is aaa not xxx.
	IERR = 6	no CONs used in model; or
	IERR = 6	no input ICONs used in model; or
	IERR = 7	CON number outside of the CON range of this model; or
	IERR = 7	ICON number outside of the ICON range of this model.
	IERR = 8	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.11 CHANGE_FCTSAUXMOD_ICON

Use this API to change the value of an integer ICON of an auxiliary signal model of a specified FACTS device. The corresponding character CHRICN is set to '$$'.

Batch command syntax:

```
BAT_CHANGE_FCTSAUXMOD_ICON  FD  ISGNDX  'MODEL'  J  IDATA
```

Python syntax:

```
ierr = change_fctsauxmod_icon(fd, isgndx, model, j, idata)
```

Fortran syntax:

```
CALL CHANGE_FCTSAUXMOD_ICON(NUM, ISGNDX, MODEL, J, IDATA, IERR)

CALL CHANGE_FCTSAUXMOD_ICON_I(NUM, ISGNDX, MODEL, J, IDATA, IERR)

CALL CHANGE_FCTSAUXMOD_ICON_C(NAME, ISGNDX, MODEL, J, IDATA, IERR)
```

where:

Character*12 or Integer FD	Is the quoted name of the FACTS device receiving the output of the auxiliary signal model; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the FACTS device receiving the output of the auxiliary signal model (input; no default allowed).
Character*12 NAME	Is the name of the FACTS device receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change relative to the start of the block of ICONs assigned to this model (1 for ICON(I), 2 for ICON(I+1), etc.) (input; no default allowed).
Integer IDATA	Is the new integer ICON value (input; unchanged by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 2	FACTS device not found.
	IERR = 3	invalid ISGNDX value for a FACTS device.
	IERR = 4	unknown model name.
	IERR = 5	no auxiliary signal model at this index for this FACTS device.
	IERR = 6	model at this index for this FACTS device is aaa not xxx.
	IERR = 7	no input ICONs used in model.
	IERR = 8	ICON number outside of the ICON range of this model.
	IERR = 9	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.12 CHANGE_FCTSAUXMOD_VAR

Use this API to change the value of a VAR of an auxiliary signal model of a specified FACTS device.

Batch command syntax:

```
BAT_CHANGE_FCTSAUXMOD_VAR  FD  ISGNDX  'MODEL'  J  RDATA
```

Python syntax:

```
ierr = change_fctsauxmod_var(fd, isgndx, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_FCTSAUXMOD_VAR(NUM, ISGNDX, MODEL, J, RDATA, IERR)

CALL CHANGE_FCTSAUXMOD_VAR_I(NUM, ISGNDX, MODEL, J, RDATA, IERR)

CALL CHANGE_FCTSAUXMOD_VAR_C(NAME, ISGNDX, MODEL, J, RDATA, IERR)
```

where:

Character*12 or Integer FD	Is the quoted name of the FACTS device receiving the output of the auxiliary signal model; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the FACTS device receiving the output of the auxiliary signal model (input; no default allowed).
Character*12 NAME	Is the name of the FACTS device receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the VAR number to change relative to the start of the block of VARs assigned to this model (1 for VAR(J), 2 for VAR(J+1), etc.) (input; no default allowed).
Real RDATA	Is the new VAR value (input; unchanged by default).
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 FACTS device not found. IERR = 2 invalid ISGNDX value for a FACTS device. IERR = 3 unknown model name. IERR = 4 no auxiliary signal model at this index for this FACTS device. IERR = 5 model at this index for this FACTS device is aaa not xxx. IERR = 6 no VARs used in model. IERR = 7 VAR number outside of the VAR range of this model. IERR = 8 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.13 CHANGE_MTDCAUXMOD_CHRICN

Use this API to change the value of a character ICON (i.e., an element of the CHRICN array) of an auxiliary signal model of a specified multi-terminal dc line. The corresponding integer ICON is set to 0.

Batch command syntax:

```
BAT_CHANGE_MTDCAUXMOD_CHRICN  DC  ISGNDX  'MODEL'  J  'CHDATA'
```

Python syntax:

```
ierr = change_mtdcauxmod_chricn(dc, isgndx, model, j, chdata)
```

Fortran syntax:

```
CALL CHANGE_MTDCAUXMOD_CHRICN(NUM, ISGNDX, MODEL, J, CHDATA, IERR)

CALL CHANGE_MTDCAUXMOD_CHRICN_I(NUM,ISGNDX,MODEL,J,CHDATA,IERR)

CALL CHANGE_MTDCAUXMOD_CHRICN_C(NAME,ISGNDX,MODEL,J,CHDATA,IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the multi-terminal dc line receiving the output of the auxiliary signal model; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the multi-terminal dc line receiving the output of the auxiliary signal model (input; no default allowed).
Character*12 NAME	Is the name of the multi-terminal dc line receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change relative to the start of the block of ICONs assigned to this model (1 for ICON(I), 2 for ICON(I+1), etc.) (input; no default allowed).
Character*2 CHDATA	Is the new character ICON value (input; blank by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — dc line not found.
	IERR = 2 — invalid ISGNDX value for a multi-terminal dc line.
	IERR = 3 — unknown model name.
	IERR = 4 — no auxiliary signal model at this index for this multi-terminal dc line.
	IERR = 5 — model at this index for this multi-terminal dc line is aaa not xxx.
	IERR = 6 — no input ICONs used in model.
	IERR = 7 — ICON number outside of the ICON range of this model.
	IERR = 8 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.14 CHANGE_MTDCAUXMOD_CON

Use this API to change the value of a CON of an auxiliary signal model of a specified multi-terminal dc line.

Batch command syntax:

```
BAT_CHANGE_MTDCAUXMOD_CON  DC  ISGNDX  'MODEL'  J  RDATA
```

Python syntax:

```
ierr = change_mtdcauxmod_con(dc, isgndx, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_MTDCAUXMOD_CON(NUM, ISGNDX, MODEL, J, RDATA, IERR)

CALL CHANGE_MTDCAUXMOD_CON_I(NUM, ISGNDX, MODEL, J, RDATA, IERR)

CALL CHANGE_MTDCAUXMOD_CON_C(NAME, ISGNDX, MODEL, J, RDATA, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the multi-terminal dc line receiving the output of the auxiliary signal model; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the multi-terminal dc line receiving the output of the auxiliary signal model (input; no default allowed).
Character*12 NAME	Is the name of the multi-terminal dc line receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the CON number to change relative to the start of the block of CONs assigned to this model (1 for CON(J), 2 for CON(J+1), etc.) (input; no default allowed).
Real RDATA	Is the new CON value (input; unchanged by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — dc line not found.
	IERR = 2 — invalid ISGNDX value for a multi-terminal dc line.
	IERR = 3 — unknown model name.
	IERR = 4 — no auxiliary signal model at this index for this multi-terminal dc line.
	IERR = 5 — model at this index for this multi-terminal dc line is aaa not xxx.
	IERR = 6 — no CONs used in model.
	IERR = 7 — CON number outside of the CON range of this model.
	IERR = 8 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.15 CHANGE_MTDCAUXMOD_DATA

This API is obsolete. It has been replaced by the APIs CHANGE_MTDCAUXMOD_CON, CHANGE_MTDCAUXMOD_ICON and CHANGE_MTDCAUXMOD_CHRICN.

Use this API to change the value of an ICON and/or CON of an auxiliary signal model of a specified multi-terminal dc line.

Batch command syntax:

```
BAT_CHANGE_MTDCAUXMOD_DATA  IDC  ISGNDX  'NAME'  J  RDATA  I  IDATA
'CHDATA'
```

Python syntax:

```
ierr = change_mtdcauxmod_data(idc,isgndx,name,j,rdata,i,idata,chdata)
```

Fortran syntax:

```
CALL
CHANGE_MTDCAUXMOD_DATA(IDC,ISGNDX,NAME,J,RDATA,I,IDATA,CHDATA,IERR)
```

where:

Integer IDC	Is the number of the multi-terminal dc line receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Character*6 NAME	Is the model name (input; no default allowed).
Integer J	Is the CON number to change relative to the start of the block of CONs assigned to this model, or 0 for no CON change (1 for CON(J), 2 for CON(J+1), etc.) (input; 0 by default).
Real RDATA	Is the new CON value (input; unchanged by default).
Integer I	Is the ICON number to change relative to the start of the block of ICONs assigned to this model, or 0 for no ICON change (1 for ICON(I), 2 for ICON(I+1), etc.) (input; 0 by default).
Integer IDATA	Is the new integer ICON value (input; unchanged by default). When a character value is intended for an ICON value, the IDATA entry is ignored and the integer ICON is set to 0.
Character*2 CHDATA	Is the new character ICON value (input; no default allowed if the character ICON is to be changed). When an integer value is intended for an ICON value, the CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error.
IERR = 1	dc line not found.
IERR = 2	invalid ISGNDX value for a multi-terminal dc line.
IERR = 3	unknown model name.
IERR = 4	no auxiliary signal model at this index for this multi-terminal dc line.
IERR = 5	model at this index for this multi-terminal dc line is aaa not xxx.
IERR = 6	no CONs used in model; or
IERR = 6	no input ICONs used in model; or
IERR = 7	CON number outside of the CON range of this model; or
IERR = 7	ICON number outside of the ICON range of this model.
IERR = 8	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.16 CHANGE_MTDCAUXMOD_ICON

Use this API to change the value of an integer ICON of an auxiliary signal model of a specified multi-terminal dc line. The corresponding character CHRICN is set to '$$'.

Batch command syntax:

```
BAT_CHANGE_MTDCAUXMOD_ICON  DC  ISGNDX  'MODEL'  J  IDATA
```

Python syntax:

```
ierr = change_mtdcauxmod_icon(dc, isgndx, model, j, idata)
```

Fortran syntax:

```
CALL CHANGE_MTDCAUXMOD_ICON(NUM, ISGNDX, MODEL, J, IDATA, IERR)

CALL CHANGE_MTDCAUXMOD_ICON_I(NUM, ISGNDX, MODEL, J, IDATA, IERR)

CALL CHANGE_MTDCAUXMOD_ICON_C(NAME, ISGNDX, MODEL, J, IDATA, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the multi-terminal dc line receiving the output of the auxiliary signal model; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the multi-terminal dc line receiving the output of the auxiliary signal model (input; no default allowed).
Character*12 NAME	Is the name of the multi-terminal dc line receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change relative to the start of the block of ICONs assigned to this model (1 for ICON(I), 2 for ICON(I+1), etc.) (input; no default allowed).
Integer IDATA	Is the new integer ICON value (input; unchanged by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — dc line not found.
	IERR = 2 — invalid ISGNDX value for a multi-terminal dc line.
	IERR = 3 — unknown model name.
	IERR = 4 — no auxiliary signal model at this index for this multi-terminal dc line.
	IERR = 5 — model at this index for this multi-terminal dc line is aaa not xxx.
	IERR = 6 — no input ICONs used in model.
	IERR = 7 — ICON number outside of the ICON range of this model.
	IERR = 8 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.17 CHANGE_MTDCAUXMOD_VAR

Use this API to change the value of a VAR of an auxiliary signal model of a specified multi-terminal dc line.

Batch command syntax:

```
BAT_CHANGE_MTDCAUXMOD_VAR  DC  ISGNDX  'MODEL'  J  RDATA
```

Python syntax:

```
ierr = change_mtdcauxmod_var(dc, isgndx, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_MTDCAUXMOD_VAR(NUM, ISGNDX, MODEL, J, RDATA, IERR)

CALL CHANGE_MTDCAUXMOD_VAR_I(NUM, ISGNDX, MODEL, J, RDATA, IERR)

CALL CHANGE_MTDCAUXMOD_VAR_C(NAME, ISGNDX, MODEL, J, RDATA, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the multi-terminal dc line receiving the output of the auxiliary signal model; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the multi-terminal dc line receiving the output of the auxiliary signal model (input; no default allowed).
Character*12 NAME	Is the name of the multi-terminal dc line receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the VAR number to change relative to the start of the block of VARs assigned to this model (1 for VAR(J), 2 for VAR(J+1), etc.) (input; no default allowed).
Real RDATA	Is the new VAR value (input; unchanged by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — dc line not found.
	IERR = 2 — invalid ISGNDX value for a multi-terminal dc line.
	IERR = 3 — unknown model name.
	IERR = 4 — no auxiliary signal model at this index for this multi-terminal dc line.
	IERR = 5 — model at this index for this multi-terminal dc line is aaa not xxx.
	IERR = 6 — no VARs used in model.
	IERR = 7 — VAR number outside of the VAR range of this model.
	IERR = 8 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.18 CHANGE_TTDCAUXMOD_CHRICN

Use this API to change the value of a character ICON (i.e., an element of the CHRICN array) of an auxiliary signal model of a specified two-terminal dc line. The corresponding integer ICON is set to 0.

Batch command syntax:

```
BAT_CHANGE_TTDCAUXMOD_CHRICN  DC  ISGNDX  'MODEL'  J  'CHDATA'
```

Python syntax:

```
ierr = change_ttdcauxmod_chricn(dc, isgndx, model, j, chdata)
```

Fortran syntax:

```
CALL CHANGE_TTDCAUXMOD_CHRICN(NUM, ISGNDX, MODEL, J, CHDATA, IERR)

CALL CHANGE_TTDCAUXMOD_CHRICN_I(NUM,ISGNDX,MODEL,J,CHDATA,IERR)

CALL CHANGE_TTDCAUXMOD_CHRICN_C(NAME,ISGNDX,MODEL,J,CHDATA,IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the two-terminal dc line receiving the output of the auxiliary signal model; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the two-terminal dc line receiving the output of the auxiliary signal model (input; no default allowed).
Character*12 NAME	Is the name of the two-terminal dc line receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change relative to the start of the block of ICONs assigned to this model (1 for ICON(I), 2 for ICON(I+1), etc.) (input; no default allowed).
Character*2 CHDATA	Is the new character ICON value (input; blank by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — dc line not found.
	IERR = 2 — invalid ISGNDX value for a two-terminal dc line.
	IERR = 3 — unknown model name.
	IERR = 4 — no auxiliary signal model at this index for this two-terminal dc line.
	IERR = 5 — model at this index for this two-terminal dc line is aaa not xxx.
	IERR = 6 — no input ICONs used in model.
	IERR = 7 — ICON number outside of the ICON range of this model.
	IERR = 8 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.19 CHANGE_TTDCAUXMOD_CON

Use this API to change the value of a CON of an auxiliary signal model of a specified two-terminal dc line.

Batch command syntax:

```
BAT_CHANGE_TTDCAUXMOD_CON  DC  ISGNDX  'MODEL'  J  RDATA
```

Python syntax:

```
ierr = change_ttdcauxmod_con(dc, isgndx, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_TTDCAUXMOD_CON(NUM, ISGNDX, MODEL, J, RDATA, IERR)

CALL CHANGE_TTDCAUXMOD_CON_I(NUM, ISGNDX, MODEL, J, RDATA, IERR)

CALL CHANGE_TTDCAUXMOD_CON_C(NAME, ISGNDX, MODEL, J, RDATA, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the two-terminal dc line receiving the output of the auxiliary signal model; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the two-terminal dc line receiving the output of the auxiliary signal model (input; no default allowed).
Character*12 NAME	Is the name of the two-terminal dc line receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the CON number to change relative to the start of the block of CONs assigned to this model (1 for CON(J), 2 for CON(J+1), etc.) (input; no default allowed).
Real RDATA	Is the new CON value (input; unchanged by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — dc line not found.
	IERR = 2 — invalid ISGNDX value for a two-terminal dc line.
	IERR = 3 — unknown model name.
	IERR = 4 — no auxiliary signal model at this index for this two-terminal dc line.
	IERR = 5 — model at this index for this two-terminal dc line is aaa not xxx.
	IERR = 6 — no CONs used in model.
	IERR = 7 — CON number outside of the CON range of this model.
	IERR = 8 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.20 CHANGE_TTDCAUXMOD_DATA

This API is obsolete. It has been replaced by the APIs CHANGE_TMTDCAUXMOD_CON, CHANGE_TTDCAUXMOD_ICON and CHANGE_TTDCAUXMOD_CHRICN.

Use this API to change the value of an ICON and/or CON of an auxiliary signal model of a specified two-terminal dc line.

Batch command syntax:

```
BAT_CHANGE_TTDCAUXMOD_DATA  IDC  ISGNDX  'NAME'  J  RDATA  I  IDATA
'CHDATA'
```

Python syntax:

```
ierr = change_ttdcauxmod_data(idc,isgndx,name,j,rdata,i,idata,chdata)
```

Fortran syntax:

```
CALL
CHANGE_TTDCAUXMOD_DATA(IDC,ISGNDX,NAME,J,RDATA,I,IDATA,CHDATA,IERR)
```

where:

Integer IDC	Is the number of the two-terminal dc line receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Character*6 NAME	Is the model name (input; no default allowed).
Integer J	Is the CON number to change relative to the start of the block of CONs assigned to this model, or 0 for no CON change (1 for CON(J), 2 for CON(J+1), etc.) (input; 0 by default).
Real RDATA	Is the new CON value (input; unchanged by default).
Integer I	Is the ICON number to change relative to the start of the block of ICONs assigned to this model, or 0 for no ICON change (1 for ICON(I), 2 for ICON(I+1), etc.) (input; 0 by default).
Integer IDATA	Is the new integer ICON value (input; unchanged by default). When a character value is intended for an ICON value, the IDATA entry is ignored and the integer ICON is set to 0.
Character*2 CHDATA	Is the new character ICON value (input; no default allowed if the character ICON is to be changed). When an integer value is intended for an ICON value, the CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error.
IERR = 1	dc line not found.
IERR = 2	invalid ISGNDX value for a two-terminal dc line.
IERR = 3	unknown model name.
IERR = 4	no auxiliary signal model at this index for this two-terminal dc line.
IERR = 5	model at this index for this two-terminal dc line is aaa not xxx.
IERR = 6	no CONs used in model; or
IERR = 6	no input ICONs used in model; or
IERR = 7	CON number outside of the CON range of this model; or
IERR = 7	ICON number outside of the ICON range of this model.
IERR = 8	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.21 CHANGE_TTDCAUXMOD_ICON

Use this API to change the value of an integer ICON of an auxiliary signal model of a specified two-terminal dc line. The corresponding character CHRICN is set to '$$'.

Batch command syntax:

```
BAT_CHANGE_TTDCAUXMOD_ICON  DC  ISGNDX  'MODEL'  J  IDATA
```

Python syntax:

```
ierr = change_ttdcauxmod_icon(dc, isgndx, model, j, idata)
```

Fortran syntax:

```
CALL CHANGE_TTDCAUXMOD_ICON(NUM, ISGNDX, MODEL, J, IDATA, IERR)

CALL CHANGE_TTDCAUXMOD_ICON_I(NUM, ISGNDX, MODEL, J, IDATA, IERR)

CALL CHANGE_TTDCAUXMOD_ICON_C(NAME, ISGNDX, MODEL, J, IDATA, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the two-terminal dc line receiving the output of the auxiliary signal model; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the two-terminal dc line receiving the output of the auxiliary signal model (input; no default allowed).
Character*12 NAME	Is the name of the two-terminal dc line receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change relative to the start of the block of ICONs assigned to this model (1 for ICON(I), 2 for ICON(I+1), etc.) (input; no default allowed).
Integer IDATA	Is the new integer ICON value (input; unchanged by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — dc line not found.
	IERR = 2 — invalid ISGNDX value for a two-terminal dc line.
	IERR = 3 — unknown model name.
	IERR = 4 — no auxiliary signal model at this index for this two-terminal dc line.
	IERR = 5 — model at this index for this two-terminal dc line is aaa not xxx.
	IERR = 6 — no input ICONs used in model.
	IERR = 7 — ICON number outside of the ICON range of this model.
	IERR = 8 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.22 CHANGE_TTDCAUXMOD_VAR

Use this API to change the value of a VAR of an auxiliary signal model of a specified two-terminal dc line.

Batch command syntax:

```
BAT_CHANGE_TTDCAUXMOD_VAR  DC  ISGNDX  'MODEL'  J  RDATA
```

Python syntax:

```
ierr = change_ttdcauxmod_var(dc, isgndx, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_TTDCAUXMOD_VAR(NUM, ISGNDX, MODEL, J, RDATA, IERR)

CALL CHANGE_TTDCAUXMOD_VAR_I(NUM, ISGNDX, MODEL, J, RDATA, IERR)

CALL CHANGE_TTDCAUXMOD_VAR_C(NAME, ISGNDX, MODEL, J, RDATA, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the two-terminal dc line receiving the output of the auxiliary signal model; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the two-terminal dc line receiving the output of the auxiliary signal model (input; no default allowed).
Character*12 NAME	Is the name of the two-terminal dc line receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the VAR number to change relative to the start of the block of VARs assigned to this model (1 for VAR(J), 2 for VAR(J+1), etc.) (input; no default allowed).
Real RDATA	Is the new VAR value (input; unchanged by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — dc line not found.
	IERR = 2 — invalid ISGNDX value for a two-terminal dc line.
	IERR = 3 — unknown model name.
	IERR = 4 — no auxiliary signal model at this index for this two-terminal dc line.
	IERR = 5 — model at this index for this two-terminal dc line is aaa not xxx.
	IERR = 6 — no VARs used in model.
	IERR = 7 — VAR number outside of the VAR range of this model.
	IERR = 8 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.23 CHANGE_VSDCAUXMOD_CHRICN

Use this API to change the value of a character ICON (i.e., an element of the CHRICN array) of an auxiliary signal model of a specified VSC dc line. The corresponding integer ICON is set to 0.

Batch command syntax:

```
BAT_CHANGE_VSDCAUXMOD_CHRICN  'DCNAME'  ISGNDX  'MODEL'  J  'CHDATA'
```

Python syntax:

```
ierr = change_vsdcauxmod_chricn(dcname, isgndx, model, j, chdata)
```

Fortran syntax:

```
CALL CHANGE_VSDCAUXMOD_CHRICN(DCNAME,ISGNDX,MODEL,J,CHDATA,IERR)
```

where:

Character*12 DCNAME	Is the name of the VSC dc line receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change relative to the start of the block of ICONs assigned to this model (1 for ICON(I), 2 for ICON(I+1), etc.) (input; no default allowed).
Character*2 CHDATA	Is the new character ICON value (input; blank by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — dc line not found.
	IERR = 2 — invalid ISGNDX value for a VSC dc line.
	IERR = 3 — unknown model name.
	IERR = 4 — no auxiliary signal model at this index for this VSC dc line.
	IERR = 5 — model at this index for this VSC dc line is aaa not xxx.
	IERR = 6 — no input ICONs used in model.
	IERR = 7 — ICON number outside of the ICON range of this model.
	IERR = 8 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.24 CHANGE_VSDCAUXMOD_CON

Use this API to change the value of a CON of an auxiliary signal model of a specified VSC dc line.

Batch command syntax:

```
BAT_CHANGE_VSDCAUXMOD_CON  'DCNAME'  ISGNDX  'MODEL'  J  RDATA
```

Python syntax:

```
ierr = change_vsdcauxmod_con(dcname, isgndx, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_VSDCAUXMOD_CON(DCNAME, ISGNDX, MODEL, J, RDATA, IERR)
```

where:

Character*12 DCNAME	Is the name of the VSC dc line receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the CON number to change relative to the start of the block of CONs assigned to this model (1 for CON(J), 2 for CON(J+1), etc.) (input; no default allowed).
Real RDATA	Is the new CON value (input; unchanged by default).
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 dc line not found. IERR = 2 invalid ISGNDX value for a VSC dc line. IERR = 3 unknown model name. IERR = 4 no auxiliary signal model at this index for this VSC dc line. IERR = 5 model at this index for this VSC dc line is aaa not xxx. IERR = 6 no CONs used in model. IERR = 7 CON number outside of the CON range of this model. IERR = 8 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.25 CHANGE_VSDCAUXMOD_DATA

This API is obsolete. It has been replaced by the APIs CHANGE_VSDCAUXMOD_CON, CHANGE_VSDCAUXMOD_ICON and CHANGE_VSDCAUXMOD_CHRICN.

Batch command syntax:

```
BAT_CHANGE_VSDCAUXMOD_DATA  'DCNAME'  ISGNDX  'NAME'  J  RDATA  I
IDATA  'CHDATA'
```

Python syntax:

```
ierr =
change_vsdcauxmod_data(dcname,isgndx,name,j,rdata,i,idata,chdata)
```

Fortran syntax:

```
CALL CHANGE_VSDCAUXMOD_DATA(DCNAME, ISGNDX, NAME, J, RDATA, I, IDATA,
CHDATA, IERR)
```

where:

Character*12 DCNAME	Is the name of the VSC dc line receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Character*6 NAME	Is the model name (input; no default allowed).
Integer J	Is the CON number to change relative to the start of the block of CONs assigned to this model, or 0 for no CON change (1 for CON(J), 2 for CON(J+1), etc.) (input; 0 by default).
Real RDATA	Is the new CON value (input; unchanged by default).
Integer I	Is the ICON number to change relative to the start of the block of ICONs assigned to this model, or 0 for no ICON change (1 for ICON(I), 2 for ICON(I+1), etc.) (input; 0 by default).
Integer IDATA	Is the new integer ICON value (input; unchanged by default). When a character value is intended for an ICON value, the IDATA entry is ignored and the integer ICON is set to 0.
Character*2 CHDATA	Is the new character ICON value (input; no default allowed if the character ICON is to be changed). When an integer value is intended for an ICON value, the CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error.
IERR = 1	dc line not found.
IERR = 2	invalid ISGNDX value for a VSC dc line.
IERR = 3	unknown model name.
IERR = 4	no auxiliary signal model at this index for this VSC dc line.
IERR = 5	model at this index for this VSC dc line is aaa not xxx.
IERR = 6	no CONs used in model; or
IERR = 6	no input ICONs used in model; or
IERR = 7	CON number outside of the CON range of this model; or
IERR = 7	ICON number outside of the ICON range of this model.
IERR = 8	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.26 CHANGE_VSDCAUXMOD_ICON

Use this API to change the value of an integer ICON of an auxiliary signal model of a specified VSC dc line. The corresponding character CHRICN is set to '$$'.

Batch command syntax:

```
BAT_CHANGE_VSDCAUXMOD_ICON  'DCNAME'  ISGNDX  'MODEL'  J  IDATA
```

Python syntax:

```
ierr = change_vsdcauxmod_icon(dcname, isgndx, model, j, idata)
```

Fortran syntax:

```
CALL CHANGE_VSDCAUXMOD_ICON(DCNAME, ISGNDX, MODEL, J, IDATA, IERR)
```

where:

Character*12 DCNAME	Is the name of the VSC dc line receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change relative to the start of the block of ICONs assigned to this model (1 for ICON(I), 2 for ICON(I+1), etc.) (input; no default allowed).
Integer IDATA	Is the new integer ICON value (input; unchanged by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — dc line not found.
	IERR = 2 — invalid ISGNDX value for a VSC dc line.
	IERR = 3 — unknown model name.
	IERR = 4 — no auxiliary signal model at this index for this VSC dc line.
	IERR = 5 — model at this index for this VSC dc line is aaa not xxx.
	IERR = 6 — no input ICONs used in model.
	IERR = 7 — ICON number outside of the ICON range of this model.
	IERR = 8 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.27 CHANGE_VSDCAUXMOD_VAR

Use this API to change the value of a VAR of an auxiliary signal model of a specified VSC dc line.

Batch command syntax:

```
BAT_CHANGE_VSDCAUXMOD_VAR  'DCNAME'  ISGNDX  'MODEL'  J  RDATA
```

Python syntax:

```
ierr = change_vsdcauxmod_var(dcname, isgndx, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_VSDCAUXMOD_VAR(DCNAME, ISGNDX, MODEL, J, RDATA, IERR)
```

where:

Character*12 DCNAME	Is the name of the VSC dc line receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the VAR number to change relative to the start of the block of VARs assigned to this model (1 for VAR(J), 2 for VAR(J+1), etc.) (input; no default allowed).
Real RDATA	Is the new VAR value (input; unchanged by default).
Integer IERR	Is the error code (output).

IERR = 0	no error.
IERR = 1	dc line not found.
IERR = 2	invalid ISGNDX value for a VSC dc line.
IERR = 3	unknown model name.
IERR = 4	no auxiliary signal model at this index for this VSC dc line.
IERR = 5	model at this index for this VSC dc line is aaa not xxx.
IERR = 6	no VARs used in model.
IERR = 7	VAR number outside of the VAR range of this model.
IERR = 8	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.28 FCTSAUXMOD_REMOVE

Use this API to remove an auxiliary signal model from the specified injection point of a designated FACTS device.

Batch command syntax:

```
BAT_FCTSAUXMOD_REMOVE  FD  ISGNDX
```

Python syntax:

```
ierr = fctsauxmod_remove(fd, isgndx)
```

Fortran syntax:

```
CALL FCTSAUXMOD_REMOVE(NUM, ISGNDX, IERR)

CALL FCTSAUXMOD_REMOVE_I(NUM, ISGNDX, IERR)

CALL FCTSAUXMOD_REMOVE_C(NAME, ISGNDX, IERR)
```

where:

Character*12 or Integer FD	Is the quoted name of the FACTS device receiving the output of the auxiliary signal model; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the FACTS device receiving the output of the auxiliary signal model (input; no default allowed).
Character*12 NAME	Is the name of the FACTS device receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 FACTS device not found. IERR = 2 invalid ISGNDX value for a FACTS device. IERR = 3 no auxiliary signal model at this index for this FACTS device. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.29 FCTSAUXMOD_STATUS

Use this API to change the status of the auxiliary signal model at the specified injection point of a designated FACTS device.

Batch command syntax:

```
BAT_FCTSAUXMOD_STATUS  FD  ISGNDX  STATUS
```

Python syntax:

```
ierr = fctsauxmod_status(fd, isgndx, status)
```

Fortran syntax:

```
CALL FCTSAUXMOD_STATUS(NUM, ISGNDX, STATUS, IERR)

CALL FCTSAUXMOD_STATUS_I(NUM, ISGNDX, STATUS, IERR)

CALL FCTSAUXMOD_STATUS_C(NAME, ISGNDX, STATUS, IERR)
```

where:

Character*12 or Integer FD	Is the quoted name of the FACTS device receiving the output of the auxiliary signal model; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the FACTS device receiving the output of the auxiliary signal model (input; no default allowed).
Character*12 NAME	Is the name of the FACTS device receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Integer STATUS	Is the new model status (input; 1 by default). STATUS = 0 out-of-service. STATUS = 1 in-service.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 FACTS device not found. IERR = 2 invalid ISGNDX value for a FACTS device. IERR = 3 no auxiliary signal model at this index for this FACTS device. IERR = 4 invalid STATUS value. IERR = 5 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.30 MTDCAUXMOD_REMOVE

Use this API to remove an auxiliary signal model from the specified auxiliary signal index of the designated multi-terminal dc line.

Batch command syntax:

```
BAT_MTDCAUXMOD_REMOVE  DC  ISGNDX
```

Python syntax:

```
ierr = mtdcauxmod_remove(dc, isgndx)
```

Fortran syntax:

```
CALL MTDCAUXMOD_REMOVE(NUM, ISGNDX, IERR)

CALL MTDCAUXMOD_REMOVE_I(NUM, ISGNDX, IERR)

CALL MTDCAUXMOD_REMOVE_C(NAME, ISGNDX, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the multi-terminal dc line receiving the output of the auxiliary signal model; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the multi-terminal dc line receiving the output of the auxiliary signal model (input; no default allowed).
Character*12 NAME	Is the name of the multi-terminal dc line receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 dc line not found. IERR = 2 invalid ISGNDX value for a multi-terminal dc line. IERR = 3 no auxiliary signal model at this index for this multi-terminal dc line. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.31 MTDCAUXMOD_STATUS

Use this API to change the status of the auxiliary signal model at the specified auxiliary signal index of the designated multi-terminal dc line.

Batch command syntax:

```
BAT_MTDCAUXMOD_STATUS  DC  ISGNDX  STATUS
```

Python syntax:

```
ierr = mtdcauxmod_status(dc, isgndx, status)
```

Fortran syntax:

```
CALL MTDCAUXMOD_STATUS(NUM, ISGNDX, STATUS, IERR)

CALL MTDCAUXMOD_STATUS_I(NUM, ISGNDX, STATUS, IERR)

CALL MTDCAUXMOD_STATUS_C(NAME, ISGNDX, STATUS, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the multi-terminal dc line receiving the output of the auxiliary signal model; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the multi-terminal dc line receiving the output of the auxiliary signal model (input; no default allowed).
Character*12 NAME	Is the name of the multi-terminal dc line receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Integer STATUS	Is the new model status (input; 1 by default). STATUS = 0 out-of-service. STATUS = 1 in-service.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 dc line not found. IERR = 2 invalid ISGNDX value for a multi-terminal dc line. IERR = 3 no auxiliary signal model at this index for this multi-terminal dc line. IERR = 4 invalid STATUS value. IERR = 5 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.32 TTDCAUXMOD_REMOVE

Use this API to remove an auxiliary signal model from the specified injection point of the designated two-terminal dc line.

Batch command syntax:

```
BAT_TTDCAUXMOD_REMOVE  DC  ISGNDX
```

Python syntax:

```
ierr = ttdcauxmod_remove(dc, isgndx)
```

Fortran syntax:

```
CALL TTDCAUXMOD_REMOVE(NUM, ISGNDX, IERR)

CALL TTDCAUXMOD_REMOVE_I(NUM, ISGNDX, IERR)

CALL TTDCAUXMOD_REMOVE_C(NAME, ISGNDX, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the two-terminal dc line receiving the output of the auxiliary signal model; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the two-terminal dc line receiving the output of the auxiliary signal model (input; no default allowed).
Character*12 NAME	Is the name of the two-terminal dc line receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 dc line not found. IERR = 2 invalid ISGNDX value for a two-terminal dc line. IERR = 3 no auxiliary signal model at this index for this two-terminal dc line. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.33 TTDCAUXMOD_STATUS

Use this API to change the status of the auxiliary signal model at the specified injection point of the designated two-terminal dc line.

Batch command syntax:

```
BAT_TTDCAUXMOD_STATUS  DC  ISGNDX  STATUS
```

Python syntax:

```
ierr = ttdcauxmod_status(dc, isgndx, status)
```

Fortran syntax:

```
CALL TTDCAUXMOD_STATUS(NUM, ISGNDX, STATUS, IERR)

CALL TTDCAUXMOD_STATUS_I(NUM, ISGNDX, STATUS, IERR)

CALL TTDCAUXMOD_STATUS_C(NAME, ISGNDX, STATUS, IERR)
```

where:

Character*12 or Integer DC	Is the quoted name of the two-terminal dc line receiving the output of the auxiliary signal model; if an unquoted integer value is specified, the API converts it to a left-justified name (input; no default allowed).
Integer NUM	Is the number (which the API converts to a left-justified name) of the two-terminal dc line receiving the output of the auxiliary signal model (input; no default allowed).
Character*12 NAME	Is the name of the two-terminal dc line receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Integer STATUS	Is the new model status (input; 1 by default). STATUS = 0 out-of-service. STATUS = 1 in-service.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 2 dc line not found. IERR = 3 invalid ISGNDX value for a two-terminal dc line. IERR = 4 no auxiliary signal model at this index for this two-terminal dc line. IERR = 5 invalid STATUS value. IERR = 6 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.34 VSDCAUXMOD_REMOVE

Use this API to remove an auxiliary signal model from the specified injection point of the designated VSC dc line.

Batch command syntax:

```
BAT_VSDCAUXMOD_REMOVE  'DCNAME'  ISGNDX
```

Python syntax:

```
ierr = vsdcauxmod_remove(dcname, isgndx)
```

Fortran syntax:

```
CALL VSDCAUXMOD_REMOVE(DCNAME, ISGNDX, IERR)
```

where:

Character*12 DCNAME	Is the name of the VSC dc line receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 dc line not found. IERR = 2 invalid ISGNDX value for a VSC dc line. IERR = 3 no auxiliary signal model at this index for this VSC dc line. IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.10.35 VSDCAUXMOD_STATUS

Use this API to change the status of the auxiliary signal model at the specified injection point of the designated VSC dc line.

Batch command syntax:

```
BAT_VSDCAUXMOD_STATUS  'DCNAME'  ISGNDX  STATUS
```

Python syntax:

```
ierr = vsdcauxmod_status(dcname, isgndx, status)
```

Fortran syntax:

```
CALL VSDCAUXMOD_STATUS(DCNAME, ISGNDX, STATUS, IERR)
```

where:

Character*12 DCNAME	Is the name of the VSC dc line receiving the output of the auxiliary signal model (input; no default allowed).
Integer ISGNDX	Is the auxiliary signal injection point number, or the auxiliary signal index (input; 1 by default).
Integer STATUS	Is the new model status (input; 1 by default). STATUS = 0 out-of-service. STATUS = 1 in-service.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 VSC dc line not found. IERR = 2 invalid ISGNDX value for a VSC dc line. IERR = 3 no auxiliary signal model at this index for this VSC dc line. IERR = 4 invalid STATUS value. IERR = 5 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.11 Induction Machine Model APIs

4.11.1 ADD_INDMAC_MODEL

Use this API to add an induction machine model to a specified machine. If a model of the designated type already exists at the machine, it is replaced.

Batch command syntax:

```
BAT_ADD_INDMAC_MODEL  IBUS  'ID'  'NAME'  NICN  IDATA  CHDATA  NCON
RDATA
```

Python syntax:

```
ierr = add_indmac_model(ibus, id, name, nicn, idata, chdata, ncon,
rdata)
```

Fortran syntax:

```
CALL ADD_INDMAC_MODEL(IBUS, ID, NAME, NICN, IDATA, CHDATA, NCON,
RDATA, IERR)
```

where:

Integer IBUS	Is the machine bus number (input; no default allowed).
Character ID*2	Is the machine identifier (input; blank by default).
Character NAME*16	Is the model name (input; no default allowed).
Integer NICN	Is the number of ICONs required by the model (input; no default allowed).
Integer IDATA(NICN)	Are the integer ICON data (input; 0 by default). When a character value is intended for an ICON value, the corresponding IDATA entry is ignored and the integer ICON is set to 0.
Character CHDATA(NICN)*2	Are the character ICON data (input; no default allowed for an ICON for which a character value is to be specified). When an integer value is intended for an ICON value, the corresponding CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).
Integer NCON	Is the number of CONs required by the model (input; no default allowed).
Real RDATA(NCON)	Are the CON data (input; 0.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output).

IERR = 0	no error
IERR = 1	Induction machine model connection tables full--use pack induction machine model tables function
IERR = 2	Induction machine model tables are corrupt
IERR = 3	unknown model NAME
IERR = 4	model is not an induction machine model
IERR = 5	# CONs specified is not the number expected
IERR = 6	# ICONs specified is not the number expected
IERR = 7	maximum number of array elements exceeded
IERR = 8	bus not found
IERR = 9	machine not found
IERR = 10	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.11.2 CHANGE_IMMOD_CHRICN

Use this API to change the value of a character ICON (i.e., an element of the CHRICN array) of an induction machine model at a specified machine. The corresponding integer ICON is set to 0.

Batch command syntax:

```
BAT_CHANGE_IMMOD_CHRICN  IBUS  'ID'  'NAME'  J  'CHDATA'
```

Python syntax:

```
ierr = change_immod_chricn(ibus, id, name, j, chdata)
```

Fortran syntax:

```
CALL CHANGE_IMMOD_CHRICN(IBUS, ID, NAME, J, CHDATA, IERR)
```

where:

Integer IBUS	Is the induction machine bus number (input; no default allowed).
Character ID*2	Is the machine identifier (input; blank by default).
Character NAME*16	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change relative to the start of the block of ICONs assigned to this model (1 for ICON(I), 2 for ICON(I+1), etc.) (input; no default allowed).
Character CHDATA*2	Is the new character ICON value (input; blank by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — unknown model NAME.
	IERR = 2 — No model at this induction machine.
	IERR = 3 — model at this machine is aaa not NAME.
	IERR = 4 — no input ICONs used in model NAME.
	IERR = 5 — ICON number outside of the ICON range of this model.
	IERR = 6 — bus not found.
	IERR = 7 — machine not found.
	IERR = 8 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.11.3 CHANGE_IMMOD_CON

Use this API to change the value of a CON of a designated induction machine model at a specified machine.

Batch command syntax:

```
BAT_CHANGE_IMMOD_CON  IBUS  'ID'  'NAME'  J  RDATA
```

Python syntax:

```
ierr = change_immod_con(ibus, id, name, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_IMMOD_CON(IBUS, ID, NAME, J, RDATA, IERR)
```

where:

Integer IBUS	Is the induction machine bus number (input; no default allowed).
Character ID*2	Is the machine identifier (input; blank by default).
Character NAME*16	Is the model name (input; no default allowed).
Integer J	Is the CON number to change relative to the start of the block of CONs assigned to this model (1 for CON(J), 2 for CON(J+1), etc.) (input; no default allowed).
Real RDATA	Is the new CON value (input; unchanged by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — unknown model NAME.
	IERR = 2 — No model at this induction machine.
	IERR = 3 — model at this machine is aaa not NAME.
	IERR = 4 — no input ICONs used in model NAME.
	IERR = 5 — ICON number outside of the ICON range of this model.
	IERR = 6 — bus not found.
	IERR = 7 — machine not found.
	IERR = 8 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.11.4 CHANGE_IMMOD_ICON

Use this API to change the value of an integer ICON of a designated induction machine model at a specified machine. The corresponding character CHRICN is set to '$$'.

Batch command syntax:

```
BAT_CHANGE_IMMOD_ICON  IBUS  'ID'  'NAME'  J  IDATA
```

Python syntax:

```
ierr = change_immod_icon(ibus, id, name, j, idata)
```

Fortran syntax:

```
CALL CHANGE_IMMOD_ICON(IBUS, ID, NAME, J, IDATA, IERR)
```

where:

Integer IBUS	Is the induction machine bus number (input; no default allowed).
Character ID*2	Is the machine identifier (input; blank by default).
Character NAME*16	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change relative to the start of the block of ICONs assigned to this model (1 for ICON(I), 2 for ICON(I+1), etc.) (input; no default allowed).
Integer IDATA	Is the new integer ICON value (input; unchanged by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error.
	IERR = 1 — unknown model NAME.
	IERR = 2 — No model at this induction machine.
	IERR = 3 — model at this machine is aaa not NAME.
	IERR = 4 — no input CONs used in model NAME.
	IERR = 5 — CON number outside of the CON range of this model.
	IERR = 6 — bus not found.
	IERR = 7 — machine not found.
	IERR = 8 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.11.5 CHANGE_IMMOD_VAR

Use this API to change the value of a VAR of a designated induction machine model at a specified machine.

Batch command syntax:

```
BAT_CHANGE_IMMOD_VAR  IBUS  'ID'  'NAME'  J  RDATA
```

Python syntax:

```
ierr = change_immod_var(ibus, id, name, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_IMMOD_VAR(IBUS, ID, NAME, J, RDATA, IERR)
```

where:

Integer IBUS	Is the induction machine bus number (input; no default allowed).
Character ID*2	Is the machine identifier (input; blank by default).
Character NAME*16	Is the model name (input; no default allowed).
Integer J	Is the VAR number to change relative to the start of the block of VARs assigned to this model (1 for VAR(J), 2 for VAR(J+1), etc.) (input; no default allowed).
Real RDATA	Is the new VAR value (input; unchanged by default).
Integer IERR	Is the error code (output).
	IERR = 0 no error.
	IERR = 1 unknown model NAME.
	IERR = 2 No model at this induction machine.
	IERR = 3 model at this machine is aaa not NAME.
	IERR = 4 no input VARs used in model NAME.
	IERR = 5 VAR number outside of the VAR range of this model.
	IERR = 6 bus not found.
	IERR = 7 machine not found.
	IERR = 8 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.11.6 IMMOD_PACK

Use this API to remove entries that are marked as unused from the induction machine model con-tables. It packs the entries remaining in these tables at the front of the various arrays.
Batch command syntax:

```
BAT_IMMOD_PACK
```

Python syntax:

```
ierr = immod_pack()
```

Fortran syntax:

```
CALL IMMOD_PACK(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.11.7 IMMOD_REMOVE

Use this API to remove an induction machine model of a designated type from a specified machine.

Batch command syntax:

```
BAT_IMMOD_REMOVE  IBUS  'ID'
```

Python syntax:

```
ierr = immod_remove(ibus, id)
```

Fortran syntax:

```
CALL IMMOD_REMOVE(IBUS, ID, IERR)
```

where:

Integer IBUS	Is the induction machine bus number (input; no default allowed).
Character ID*2	Is the machine identifier (input; blank by default).
Integer IERR	Is the error code (output).
	IERR = 0 no error.
	IERR = 1 no induction machine model at this machine.
	IERR = 2 bus not found.
	IERR = 3 machine not found.
	IERR = 4 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.11.8 IMMOD_STATUS

Use this API to change the status of an induction machine model at a specified machine.
Batch command syntax:

```
BAT_IMMOD_STATUS  IBUS  'ID'  STATUS
```

Python syntax:

```
ierr = immod_status(ibus, id, status)
```

Fortran syntax:

```
CALL IMMOD_STATUS(IBUS, ID, STATUS, IERR)
```

where:

Integer IBUS	Is the induction machine bus number (input; no default allowed).
Character ID*2	Is the machine identifier (input; blank by default).
Integer STATUS	Is the new model status (input; 1 by default). STATUS = 0 out-of-service. STATUS = 1 in-service.
Integer IERR	Is the error code (output). IERR = 0 no error. IERR = 1 no induction machine model at this machine. IERR = 2 invalid STATUS value. IERR = 3 bus not found. IERR = 4 machine not found. IERR = 5 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.11.9 IMMOD_UNCONNECTED

Use this API to list or remove from dynamics working memory those induction machine models that are assigned to machines that are not present in the current power flow working case (unconnected).

Batch command syntax:

```
BAT_IMMOD_UNCONNECTED  OPT
```

Python syntax:

```
ierr = immod_unconnected(opt)
```

Fortran syntax:

```
CALL IMMOD_UNCONNECTED(OPT, IERR)
```

where:

Integer OPT	Is the function code (input; 0 by default):	
	OPT = 0	list unconnected induction machine models.
	OPT = 1	remove unconnected induction machine models.
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid OPT value.
	IERR = 2	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.11.10 IMMOD_USER

Use this API to list user-written induction machine model definitions or to remove user-written induction machine model definitions that are not assigned to any machines (unused) from the user model definition tables.

Batch command syntax:

```
BAT_IMMOD_USER  OPT
```

Python syntax:

```
ierr = immod_user(opt)
```

Fortran syntax:

```
CALL IMMOD_USER(OPT, IERR)
```

where:

Integer OPT	Is the function code (input; 0 by default):	
	OPT = 0	list user-written induction machine model definitions.
	OPT = 1	remove unused user-written induction machine model definitions.
Integer IERR	Is the error code (output).	
	IERR = 0	no error.
	IERR = 1	invalid OPT value.
	IERR = 2	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12 CCT Model APIs

4.12.1 ADD_CCTBRND_MODEL

Use this API to add a branch device model to the specified branch. If a model already exists at the branch, it is replaced.

Batch command syntax:

```
BAT_ADD_CCTBRND_MODEL  IBUS  JBUS  'ID'  'MODEL'  NICN  IDATA  CHDATA
NCON  RDATA
```

Python syntax:

```
ierr =
add_cctbrnd_model(ibus,jbus,id,model,nicn,idata,chdata,ncon,rdata)
```

Fortran syntax:

```
CALL ADD_CCTBRND_MODEL(IBUS, JBUS, ID, MODEL, NICN, IDATA, CHDATA,
NCON, RDATA, IERR)
```

where:

Integer IBUS	Is the from bus number (input; no default allowed).
Integer JBUS	Is the to bus number (input; no default allowed).
Character*2 ID	Is the circuit identifier (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer NICN	Is the number of ICONs required by the model (input; no default allowed).
Integer IDATA(NICN)	Are the integer ICON data (input; 0 by default). When a character value is intended for an ICON value, the corresponding IDATA entry is ignored and the integer ICON is set to 0.
Character*2 CHDATA(NICN)	Are the character ICON data (input; no default allowed for an ICON for which a character value is to be specified). When an integer value is intended for an ICON value, the corresponding CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).
Integer NCON	Is the number of CONs required by the model (input; no default allowed).
Real RDATA(NCON)	Are the CON data (input; 0.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error
	IERR = 1	Unknown model name
	IERR = 2	model is not a branch device model
	IERR = 3	# CONs specified is not the number expected
	IERR = 4	# ICONs specified is not the number expected
	IERR = 5	maximum number of array elements exceeded
	IERR = 6	CCT model connection tables full—use pack CCT model tables function
	IERR = 7	Branch is a transformer
	IERR = 8	Bus not found
	IERR = 9	Branch not found
	IERR = 10	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.2 ADD_CCT2WTD_MODEL

Use this API to add a 2-winding transformer device model to the specified 2-winding transformer. If a model already exists at the 2-winding transformer, it is replaced.

Batch command syntax:

```
BAT_ADD_CCT2WTD_MODEL  IBUS  JBUS  'ID'  'MODEL'  NICN  IDATA  CHDATA
NCON  RDATA
```

Python syntax:

```
ierr =
add_cct2wtd_model(ibus,jbus,id,model,nicn,idata,chdata,ncon,rdata)
```

Fortran syntax:

```
CALL ADD_CCT2WTD_MODEL(IBUS, JBUS, ID, MODEL, NICN, IDATA, CHDATA,
NCON, RDATA, IERR)
```

where:

Integer IBUS	Is the from bus number (input; no default allowed).
Integer JBUS	Is the to bus number (input; no default allowed).
Character*2 ID	Is the circuit identifier (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer NICN	Is the number of ICONs required by the model (input; no default allowed).
Integer IDATA(NICN)	Are the integer ICON data (input; 0 by default). When a character value is intended for an ICON value, the corresponding IDATA entry is ignored and the integer ICON is set to 0.
Character*2 CHDATA(NICN)	Are the character ICON data (input; no default allowed for an ICON for which a character value is to be specified). When an integer value is intended for an ICON value, the corresponding CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).
Integer NCON	Is the number of CONs required by the model (input; no default allowed).
Real RDATA(NCON)	Are the CON data (input; 0.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error
	IERR = 1	Unknown model name
	IERR = 2	model is not a 2-winding transformer device model
	IERR = 3	# CONs specified is not the number expected
	IERR = 4	# ICONs specified is not the number expected
	IERR = 5	maximum number of array elements exceeded
	IERR = 6	CCT model connection tables full—use pack CCT model tables function
	IERR = 7	Branch is not a transformer
	IERR = 8	Bus not found
	IERR = 9	2-winding transformer not found
	IERR = 10	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.3 ADD_CCT3WTD_MODEL

Use this API to add a 3-winding transformer device model to the specified 3-winding transformer. If a model already exists at the 3-winding transformer, it is replaced.

```
Batch command syntax:
```

Python syntax:

```
ierr = add_cct3wtd_model(ibus, jbus, kbus, id, model, nicn, idata,
chdata, ncon, rdata)
```

Fortran syntax:

```
CALL ADD_CCT3WTD_MODEL(IBUS, JBUS, KBUS, ID, MODEL, NICN, IDATA,
CHDATA, NCON, RDATA, IERR)
```

where:

Integer IBUS	Is the from bus number (input; no default allowed).
Integer JBUS	Is the to bus number (input; no default allowed).
Integer KBUS	Is the third bus number (input; no default allowed).
Character*2 ID	Is the circuit identifier (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer NICN	Is the number of ICONs required by the model (input; no default allowed).
Integer IDATA(NICN)	Are the integer ICON data (input; 0 by default). When a character value is intended for an ICON value, the corresponding IDATA entry is ignored and the integer ICON is set to 0.
Character*2 CHDATA(NICN)	Are the character ICON data (input; no default allowed for an ICON for which a character value is to be specified). When an integer value is intended for an ICON value, the corresponding CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).
Integer NCON	Is the number of CONs required by the model (input; no default allowed).
Real RDATA(NCON)	Are the CON data (input; 0.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error
	IERR = 1	Unknown model name
	IERR = 2	model is not a 3-winding transformer device model
	IERR = 3	# CONs specified is not the number expected
	IERR = 4	# ICONs specified is not the number expected
	IERR = 5	maximum number of array elements exceeded
	IERR = 6	CCT model connection tables full—use pack CCT model tables function
	IERR = 7	Branch is not a transformer
	IERR = 8	Bus not found
	IERR = 9	3-winding transformer not found
	IERR = 10	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.4 ADD_CCTMCNP_MODEL

Use this API to add a machine protection model to the specified machine. If another model exists, it is replaced.

Batch command syntax:

```
BAT_ADD_CCTMCNP_MODEL  IBUS  'ID'  'MODEL'  NICN  IDATA  CHDATA  NCON
RDATA
```

Python syntax:

```
ierr = add_cctmcnp_model(ibus, id, model, nicn, idata, chdata, ncon,
rdata)
```

Fortran syntax:

```
CALL ADD_CCTMCNP_MODEL(IBUS, ID, MODEL, NICN, IDATA, CHDATA, NCON,
RDATA, IERR)
```

where:

Integer IBUS	Is the bus number (input; no default allowed).
Character*2 ID	Is the machine identifier (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer NICN	Is the number of ICONs required by the model (input; no default allowed).
Integer IDATA(NICN)	Are the integer ICON data (input; 0 by default). When a character value is intended for an ICON value, the corresponding IDATA entry is ignored and the integer ICON is set to 0.
Character*2 CHDATA(NICN)	Are the character ICON data (input; no default allowed for an ICON for which a character value is to be specified). When an integer value is intended for an ICON value, the corresponding CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).
Integer NCON	Is the number of CONs required by the model (input; no default allowed).
Real RDATA(NCON)	Are the CON data (input; 0.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error
	IERR = 1	Unknown model name
	IERR = 2	model is not a machine protection model
	IERR = 3	# CONs specified is not the number expected
	IERR = 4	# ICONs specified is not the number expected
	IERR = 5	maximum number of array elements exceeded
	IERR = 6	CCT model connection tables full—use pack CCT model tables function
	IERR = 7	Bus not found
	IERR = 8	Machine not found
	IERR = 9	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.5 ADD_CCTBUSO_MODEL

Use this API to add an “other” type model attached to a bus. If another model exists, it is replaced.

Batch command syntax:

```
BAT_ADD_CCTBUSO_MODEL  IBUS  'MODEL'  NICN  IDATA  CHDATA  NCON  RDATA
```

Python syntax:

```
ierr = add_cctbuso_model(ibus, model, nicn, idata, chdata, ncon,
rdata)
```

Fortran syntax:

```
CALL ADD_CCTBUSO_MODEL(IBUS, MODEL, NICN, IDATA, CHDATA, NCON, RDATA,
IERR)
```

where:

Integer IBUS	Is the bus number (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer NICN	Is the number of ICONs required by the model (input; no default allowed).
Integer IDATA(NICN)	Are the integer ICON data (input; 0 by default). When a character value is intended for an ICON value, the corresponding IDATA entry is ignored and the integer ICON is set to 0.
Character*2 CHDATA(NICN)	Are the character ICON data (input; no default allowed for an ICON for which a character value is to be specified). When an integer value is intended for an ICON value, the corresponding CHDATA entry must be either defaulted or specified as either ’$$’ or ’ ’; the character ICON is set to ’$$’ (blank by default).
Integer NCON	Is the number of CONs required by the model (input; no default allowed).
Real RDATA(NCON)	Are the CON data (input; 0.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error
	IERR = 1	Unknown model name
	IERR = 2	model is not a bus “other” type model
	IERR = 3	# CONs specified is not the number expected
	IERR = 4	# ICONs specified is not the number expected
	IERR = 5	maximum number of array elements exceeded
	IERR = 6	CCT model connection tables full—use pack CCT model tables function
	IERR = 7	Bus not found
	IERR = 8	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.6 ADD_CCT2DCO_MODEL

Use this API to add an “other” type model attached to a 2-terminal dc line. If another model exists, it is replaced.

Batch command syntax:

```
BAT_ADD_CCT2DCO_MODEL  DC  'MODEL'  NICN  IDATA  CHDATA  NCON  RDATA
```

Python syntax:

```
ierr = add_cct2dco_model(dc, model, nicn, idata, chdata, ncon, rdata)
```

Fortran syntax:

```
CALL ADD_CCT2DCO_MODEL(DC, MODEL, NICN, IDATA, CHDATA, NCON, RDATA,
IERR)
```

where:

Character*12 DC	Is the quoted name of the two-terminal dc line for which model data is to be changed (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer NICN	Is the number of ICONs required by the model (input; no default allowed).
Integer IDATA(NICN)	Are the integer ICON data (input; 0 by default). When a character value is intended for an ICON value, the corresponding IDATA entry is ignored and the integer ICON is set to 0.
Character*2 CHDATA(NICN)	Are the character ICON data (input; no default allowed for an ICON for which a character value is to be specified). When an integer value is intended for an ICON value, the corresponding CHDATA entry must be either defaulted or specified as either ’$$’ or ’ ’; the character ICON is set to ’$$’ (blank by default).
Integer NCON	Is the number of CONs required by the model (input; no default allowed).
Real RDATA(NCON)	Are the CON data (input; 0.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error
	IERR = 1	Unknown model name
	IERR = 2	model is not a 2-terminal dc line “other” type model
	IERR = 3	# CONs specified is not the number expected
	IERR = 4	# ICONs specified is not the number expected
	IERR = 5	maximum number of array elements exceeded
	IERR = 6	CCT model connection tables full—use pack CCT model tables function
	IERR = 7	DC line not found
	IERR = 8	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.7 ADD_CCTMCNO_MODEL

Use this API to add a machine other model to the specified machine. If another model exists, it is replaced.

Batch command syntax:

```
BAT_ADD_CCTMCNO_MODEL  IBUS  'ID'  'MODEL'  NICN  IDATA  CHDATA  NCON
RDATA
```

Python syntax:

```
ierr = add_cctmcno_model(ibus, id, model, nicn, idata, chdata, ncon,
rdata)
```

Fortran syntax:

```
CALL ADD_CCTMCNO_MODEL(IBUS, ID, MODEL, NICN, IDATA, CHDATA, NCON,
RDATA, IERR)
```

where:

Integer IBUS	Is the bus number (input; no default allowed).
Character*2 ID	Is the machine identifier (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer NICN	Is the number of ICONs required by the model (input; no default allowed).
Integer IDATA(NICN)	Are the integer ICON data (input; 0 by default). When a character value is intended for an ICON value, the corresponding IDATA entry is ignored and the integer ICON is set to 0.
Character*2 CHDATA(NICN)	Are the character ICON data (input; no default allowed for an ICON for which a character value is to be specified). When an integer value is intended for an ICON value, the corresponding CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).
Integer NCON	Is the number of CONs required by the model (input; no default allowed).
Real RDATA(NCON)	Are the CON data (input; 0.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error
	IERR = 1	Unknown model name
	IERR = 2	model is not a machine protection model
	IERR = 3	# CONs specified is not the number expected
	IERR = 4	# ICONs specified is not the number expected
	IERR = 5	maximum number of array elements exceeded
	IERR = 6	CCT model connection tables full—use pack CCT model tables function
	IERR = 7	Bus not found
	IERR = 8	Machine not found
	IERR = 9	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.8 ADD_CCTMSCO_MODEL

Use this API to add a miscellaneous "other" type model with the specified model instance. If another model exists with the same model instance, it is replaced.

Batch command syntax:

```
BAT_ADD_CCTMSCO_MODEL  'MODEL'  MINS NICN  IDATA  CHDATA  NCON  RDATA
```

Python syntax:

```
ierr = add_cctmsco_model(model, mins,  nicn, idata, chdata, ncon,
rdata)
```

Fortran syntax:

```
CALL ADD_CCTMSCO_MODEL(MODEL, MINS, NICN, IDATA, CHDATA, NCON, RDATA,
IERR)
```

where:

Character*16 MODEL	Is the model name (input; no default allowed).
Integer MINS	Is the model instance (>0) (input; no default allowed).
Integer NICN	Is the number of ICONs required by the model (input; no default allowed).
Integer IDATA(NICN)	Are the integer ICON data (input; 0 by default). When a character value is intended for an ICON value, the corresponding IDATA entry is ignored and the integer ICON is set to 0.
Character*2 CHDATA(NICN)	Are the character ICON data (input; no default allowed for an ICON for which a character value is to be specified). When an integer value is intended for an ICON value, the corresponding CHDATA entry must be either defaulted or specified as either '$$' or ' '; the character ICON is set to '$$' (blank by default).
Integer NCON	Is the number of CONs required by the model (input; no default allowed).
Real RDATA(NCON)	Are the CON data (input; 0.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error
	IERR = 1	Unknown model name
	IERR = 2	model is not a bus “other” type model
	IERR = 3	# CONs specified is not the number expected
	IERR = 4	# ICONs specified is not the number expected
	IERR = 5	maximum number of array elements exceeded
	IERR = 6	CCT model connection tables full—use pack CCT model tables function
	IERR = 7	Invalid model instance
	IERR = 8	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.9 CHANGE_CCTBRNDMOD_CHRICN

Use this API to change the value of a character ICON (i.e., an element of the CHRICN array) of the branch device model of a specified branch. The corresponding integer ICON is set to 0.

Batch command syntax:

```
BAT_CHANGE_CCTBRNDMOD_CHRICN  IBUS  JBUS  'ID'  'MODEL'  J  CHDATA
```

Python syntax:

```
ierr = change_cctbrndmod_chricn(ibus, jbus, id, model, j, chdata)
```

Fortran syntax:

```
CALL CHANGE_CCTBRNDMOD_CHRICN(IBUS, JBUS, ID, MODEL, J, CHDATA,  IERR)
```

where:

Integer IBUS	Is the from bus number (input; no default allowed).
Integer JBUS	Is the to bus number (input; no default allowed).
Character*2 ID	Is the circuit identifier (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change (J is relative to the start of the block of ICONs assigned to this model, e,g., 1 for first ICON(M), 2 for ICON(M+1), etc.) (input; no default allowed).
Character*2 CHDATA	Is the new character ICON value (input; blank by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	Unknown model name
IERR = 2	No CCT table entry for the model
IERR = 3	Model names do not match – model is aaa not xxx
IERR = 4	no ICONs used in the model
IERR = 5	ICON number outside of the ICON range of this model
IERR = 6	Bus not found
IERR = 7	Branch not found
IERR = 8	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.10 CHANGE_CCTBRNDMOD_CON

Use this API to change the value of a real constant (CON) of the branch device model of a specified branch.
Batch command syntax:

```
BAT_CHANGE_CCTBRNDMOD_CON  IBUS  JBUS  'ID'  'MODEL'  J  RDATA
```

Python syntax:

```
ierr = change_cctbrndmod_con(ibus, jbus, id, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_CCTBRNDMOD_CON(IBUS, JBUS, ID, MODEL, J, RDATA,  IERR)
```

where:

Integer IBUS	Is the from bus number (input; no default allowed).
Integer JBUS	Is the to bus number (input; no default allowed).
Character*2 ID	Is the circuit identifier (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the CON number to change (J is relative to the start of the block of CONs assigned to this model, e,g., 1 for first CON(K), 2 for CON(K+1), etc.) (input; no default allowed).
Real RDATA	Is the new CON value (input; unchanged by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	Unknown model name
IERR = 2	No CCT table entry for the model
IERR = 3	Model names do not match – model is aaa not xxx
IERR = 4	no CONs used in the model
IERR = 5	CON number outside of the CON range of this model
IERR = 6	Bus not found
IERR = 7	Branch not found
IERR = 8	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.11 CHANGE_CCTBRNDMOD_ICON

Use this API to change the value of an integer ICON of the branch device model of a specified branch.

Batch command syntax:

```
BAT_CHANGE_CCTBRNDMOD_ICON  IBUS  JBUS  'ID'  'MODEL'  J  IDATA
```

Python syntax:

```
ierr = change_cctbrndmod_icon(ibus, jbus, id, model, j, idata)
```

Fortran syntax:

```
CALL CHANGE_CCTBRNDMOD_icon(IBUS, JBUS, ID, MODEL, J, IDATA,  IERR)
```

where:

Integer IBUS	Is the from bus number (input; no default allowed).
Integer JBUS	Is the to bus number (input; no default allowed).
Character*2 ID	Is the circuit identifier (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change (J is relative to the start of the block of ICONs assigned to this model, e,g., 1 for first ICON(M), 2 for ICON(M+1), etc.) (input; no default allowed).
Integer IDATA	Is the new ICON value (input; unchanged by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	Unknown model name
IERR = 2	No CCT table entry for the model
IERR = 3	Model names do not match – model is aaa not xxx
IERR = 4	no ICONs used in the model
IERR = 5	ICON number outside of the ICON range of this model
IERR = 6	Bus not found
IERR = 7	Branch not found
IERR = 8	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.12 CHANGE_CCTBRNDMOD_VAR

Use this API to change the value of VAR of the branch device model of a specified branch.
Batch command syntax:

```
BAT_CHANGE_CCTBRNDMOD_VAR  IBUS  JBUS  'ID'  'MODEL'  J  RDATA
```

Python syntax:

```
ierr = change_cctbrndmod_var(ibus, jbus, id, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_CCTBRNDMOD_VAR(IBUS, JBUS, ID, MODEL, J, RDATA,  IERR)
```

where:

Integer IBUS	Is the from bus number (input; no default allowed).
Integer JBUS	Is the to bus number (input; no default allowed).
Character*2 ID	Is the circuit identifier (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the VAR number to change (J is relative to the start of the block of VARs assigned to this model, e,g., 1 for first VAR(L), 2 for VAR(L+1), etc.) (input; no default allowed).
Real RDATA	Is the new VAR value (input; unchanged by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	Unknown model name
IERR = 2	No CCT table entry for the model
IERR = 3	Model names do not match – model is aaa not xxx
IERR = 4	no VARs used in the model
IERR = 5	VAR number outside of the VAR range of this model
IERR = 6	Bus not found
IERR = 7	Branch not found
IERR = 8	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.13 CHANGE_CCT2WTDMOD_CHRICN

Use this API to change the value of a character ICON (i.e., an element of the CHRICN array) of the 2-winding transformer device model of a specified branch. The corresponding integer ICON is set to 0.

Batch command syntax:

```
BAT_CHANGE_CCT2WTDMOD_CHRICN  IBUS  JBUS  'ID'  'MODEL'  J  CHDATA
```

Python syntax:

```
ierr = change_cct2wtdmod_chricn(ibus, jbus, id, model, j, chdata)
```

Fortran syntax:

```
CALL CHANGE_CCT2WTDMOD_CHRICN(IBUS, JBUS, ID, MODEL, J, CHDATA,  IERR)
```

where:

Integer IBUS	Is the from bus number (input; no default allowed).
Integer JBUS	Is the to bus number (input; no default allowed).
Character*2 ID	Is the circuit identifier (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change (J is relative to the start of the block of ICONs assigned to this model, e,g., 1 for first ICON(M), 2 for ICON(M+1), etc.) (input; no default allowed).
Character*2 CHDATA	Is the new character ICON value (input; blank by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	Unknown model name
IERR = 2	No CCT table entry for the model
IERR = 3	Model names do not match – model is aaa not xxx
IERR = 4	no ICONs used in the model
IERR = 5	ICON number outside of the ICON range of this model
IERR = 6	Bus not found
IERR = 7	Branch not found
IERR = 8	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.14 CHANGE_CCT2WTDMOD_CON

Use this API to change the value of a real constant (CON) of the 2-winding transformer device model of a specified branch.
Batch command syntax:

```
BAT_CHANGE_CCT2WTDMOD_CON  IBUS  JBUS  'ID'  'MODEL'  J  RDATA
```

Python syntax:

```
ierr = change_cct2wtdmod_con(ibus, jbus, id, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_CCT2WTDMOD_CON(IBUS, JBUS, ID, MODEL, J, RDATA,  IERR)
```

where:

Integer IBUS	Is the from bus number (input; no default allowed).
Integer JBUS	Is the to bus number (input; no default allowed).
Character*2 ID	Is the circuit identifier (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the CON number to change (J is relative to the start of the block of CONs assigned to this model, e,g., 1 for first CON(K), 2 for CON(K+1), etc.) (input; no default allowed).
Real RDATA	Is the new CON value (input; unchanged by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	Unknown model name
IERR = 2	No CCT table entry for the model
IERR = 3	Model names do not match – model is aaa not xxx
IERR = 4	no CONs used in the model
IERR = 5	CON number outside of the CON range of this model
IERR = 6	Bus not found
IERR = 7	Branch not found
IERR = 8	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.15 CHANGE_CCT2WTDMOD_ICON

Use this API to change the value of an integer ICON of the 2-winding transformer device model of a specified branch.
Batch command syntax:

```
BAT_CHANGE_CCT2WTDMOD_ICON  IBUS  JBUS  'ID'  'MODEL'  J  IDATA
```

Python syntax:

```
ierr = change_cct2wtdmod_icon(ibus, jbus, id, model, j, idata)
```

Fortran syntax:

```
CALL CHANGE_CCT2WTDMOD_icon(IBUS, JBUS, ID, MODEL, J, IDATA,  IERR)
```

where:

Integer IBUS	Is the from bus number (input; no default allowed).
Integer JBUS	Is the to bus number (input; no default allowed).
Character*2 ID	Is the circuit identifier (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change (J is relative to the start of the block of ICONs assigned to this model, e,g., 1 for first ICON(M), 2 for ICON(M+1), etc.) (input; no default allowed).
Integer IDATA	Is the new ICON value (input; unchanged by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	Unknown model name
IERR = 2	No CCT table entry for the model
IERR = 3	Model names do not match – model is aaa not xxx
IERR = 4	no ICONs used in the model
IERR = 5	ICON number outside of the ICON range of this model
IERR = 6	Bus not found
IERR = 7	Branch not found
IERR = 8	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.16 CHANGE_CCT2WTDMOD_VAR

Use this API to change the value of VAR of the 2-winding transformer device model of a specified branch.

Batch command syntax:

```
BAT_CHANGE_CCT2WTDMOD_VAR  IBUS  JBUS  'ID'  'MODEL'  J  RDATA
```

Python syntax:

```
ierr = change_cct2wtdmod_var(ibus, jbus, id, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_CCT2WTDMOD_VAR(IBUS, JBUS, ID, MODEL, J, RDATA,  IERR)
```

where:

Integer IBUS	Is the from bus number (input; no default allowed).
Integer JBUS	Is the to bus number (input; no default allowed).
Character*2 ID	Is the circuit identifier (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the VAR number to change (J is relative to the start of the block of VARs assigned to this model, e,g., 1 for first VAR(L), 2 for VAR(L+1), etc.) (input; no default allowed).
Real RDATA	Is the new VAR value (input; unchanged by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	Unknown model name
IERR = 2	No CCT table entry for the model
IERR = 3	Model names do not match – model is aaa not xxx
IERR = 4	no VARs used in the model
IERR = 5	VAR number outside of the VAR range of this model
IERR = 6	Bus not found
IERR = 7	Branch not found
IERR = 8	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.17 CHANGE_CCT3WTDMOD_CHRICN

Use this API to change the value of a character ICON (i.e., an element of the CHRICN array) of the 3-winding transformer device model of a specified branch. The corresponding integer ICON is set to 0.

Batch command syntax:

```
BAT_CHANGE_CCT3WTDMOD_CHRICN  IBUS  JBUS  KBUS 'ID'  'MODEL'  J
CHDATA
```

Python syntax:

```
ierr = change_cct3wtdmod_chricn(ibus, jbus, kbus, id, model, j,
chdata)
```

Fortran syntax:

```
CALL CHANGE_CCT3WTDMOD_CHRICN(IBUS, JBUS, KBUS, ID, MODEL, J, CHDATA,
IERR)
```

where:

Integer IBUS	Is the from bus number (input; no default allowed).
Integer JBUS	Is the to bus number (input; no default allowed).
Integer KBUS	Is the third bus number (input; no default allowed).
Character*2 ID	Is the circuit identifier (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change (J is relative to the start of the block of ICONs assigned to this model, e,g., 1 for first ICON(M), 2 for ICON(M+1), etc.) (input; no default allowed).
Character*2 CHDATA	Is the new character ICON value (input; blank by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error
	IERR = 1	Unknown model name
	IERR = 2	No CCT table entry for the model
	IERR = 3	Model names do not match – model is aaa not xxx
	IERR = 4	no ICONs used in the model
	IERR = 5	ICON number outside of the ICON range of this model
	IERR = 6	Bus not found
	IERR = 7	Branch not found
	IERR = 8	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.18 CHANGE_CCT3WTDMOD_CON

Use this API to change the value of a real constant (CON) of the 3-winding transformer device model of a specified branch.
Batch command syntax:

```
BAT_CHANGE_CCT3WTDMOD_CON  IBUS  JBUS  KBUS 'ID'  'MODEL'  J  RHDATA
```

Python syntax:

```
ierr = change_cct3wtdmod_con(ibus, jbus, kbus, id, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_CCT3WTDMOD_CON(IBUS, JBUS, KBUS, ID, MODEL, J, RDATA,
IERR)
```

where:

Integer IBUS	Is the from bus number (input; no default allowed).
Integer JBUS	Is the to bus number (input; no default allowed).
Integer KBUS	Is the third bus number (input; no default allowed).
Character*2 ID	Is the circuit identifier (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the CON number to change (J is relative to the start of the block of CONs assigned to this model, e,g., 1 for first CON(K), 2 for CON(K+1), etc.) (input; no default allowed).
Real RDATA	Is the new CON value (input; unchanged by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error
	IERR = 1	Unknown model name
	IERR = 2	No CCT table entry for the model
	IERR = 3	Model names do not match – model is aaa not xxx
	IERR = 4	no CONs used in the model
	IERR = 5	CON number outside of the CON range of this model
	IERR = 6	Bus not found
	IERR = 7	Branch not found
	IERR = 8	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.19 CHANGE_CCT3WTDMOD_ICON

Use this API to change the value of an integer ICON of the 3-winding transformer device model of a specified branch.
Batch command syntax:

```
BAT_CHANGE_CCT3WTDMOD_ICON  IBUS  JBUS  KBUS 'ID'  'MODEL'  J  IDATA
```

Python syntax:

```
ierr = change_cct3wtdmod_icon(ibus, jbus, kbus, id, model, j, idata)
```

Fortran syntax:

```
CALL CHANGE_CCT3WTDMOD_icon(IBUS, JBUS, KBUS, ID, MODEL, J, IDATA,
IERR)
```

where:

Integer IBUS	Is the from bus number (input; no default allowed).
Integer JBUS	Is the to bus number (input; no default allowed).
Integer KBUS	Is the third bus number (input; no default allowed).
Character*2 ID	Is the circuit identifier (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change (J is relative to the start of the block of ICONs assigned to this model, e,g., 1 for first ICON(M), 2 for ICON(M+1), etc.) (input; no default allowed).
Integer IDATA	Is the new ICON value (input; unchanged by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error
	IERR = 1	Unknown model name
	IERR = 2	No CCT table entry for the model
	IERR = 3	Model names do not match – model is aaa not xxx
	IERR = 4	no ICONs used in the model
	IERR = 5	ICON number outside of the ICON range of this model
	IERR = 6	Bus not found
	IERR = 7	Branch not found
	IERR = 8	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.20 CHANGE_CCT3WTDMOD_VAR

Use this API to change the value of VAR of the 3-winding transformer device model of a specified branch.

Batch command syntax:

```
BAT_CHANGE_CCT3WTDMOD_VAR  IBUS  JBUS  KBUS 'ID'  'MODEL'  J  RDATA
```

Python syntax:

```
ierr = change_cct3wtdmod_var(ibus, jbus, kbus, id, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_CCT3WTDMOD_VAR(IBUS, JBUS, KBUS, ID, MODEL, J, RDATA,
IERR)
```

where:

Integer IBUS	Is the from bus number (input; no default allowed).
Integer JBUS	Is the to bus number (input; no default allowed).
Integer KBUS	Is the third bus number (input; no default allowed).
Character*2 ID	Is the circuit identifier (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the VAR number to change (J is relative to the start of the block of VARs assigned to this model, e,g., 1 for first VAR(L), 2 for VAR(L+1), etc.) (input; no default allowed).
Real RDATA	Is the new VAR value (input; unchanged by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error
	IERR = 1	Unknown model name
	IERR = 2	No CCT table entry for the model
	IERR = 3	Model names do not match – model is aaa not xxx
	IERR = 4	no VARs used in the model
	IERR = 5	VAR number outside of the VAR range of this model
	IERR = 6	Bus not found
	IERR = 7	Branch not found
	IERR = 8	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.21 CHANGE_CCTMCNPMOD_CHRICN

Use this API to change the value of a character ICON (i.e., an element of the CHRICN array) of the machine protection model of a specified machine. The corresponding integer ICON is set to 0.
Batch command syntax:

```
BAT_CHANGE_CCTMCNPMOD_CHRICN  IBUS  'ID'  'MODEL' J  CHDATA
```

Python syntax:

```
ierr = change_cctmcnpmod_chricn(ibus, id, model, j, chdata)
```

Fortran syntax:

```
CALL CHANGE_CCTMCNPMOD_CHRICN(IBUS, ID, MODEL, J, CHDATA,  IERR)
```

where:

Integer IBUS	Is the bus number (input; no default allowed).
Character*2 ID	Is the circuit identifier (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change (J is relative to the start of the block of ICONs assigned to this model, e,g., 1 for first ICON(M), 2 for ICON(M+1), etc.) (input; no default allowed).
Character*2 CHDATA	Is the new character ICON value (input; blank by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	Unknown model name
IERR = 2	No CCT table entry for the model
IERR = 3	Model names do not match – model is aaa not xxx
IERR = 4	no ICONs used in the model
IERR = 5	ICON number outside of the ICON range of this model
IERR = 6	Bus not found
IERR = 7	Machine not found
IERR = 8	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.22 CHANGE_CCTMCNPMOD_CON

Use this API to change the value of a real constant (CON) of the machine protection model of a specified machine.
Batch command syntax:

```
BAT_CHANGE_CCTMCNPMOD_CON  IBUS  'ID'  'MODEL'  J  RDATA
```

Python syntax:

```
ierr = change_cctmcnpmod_con(ibus, id, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_CCTMCNPMOD_CON(IBUS, ID, MODEL, J, RDATA,  IERR)
```

where:

Integer IBUS	Is the bus number (input; no default allowed).
Character*2 ID	Is the circuit identifier (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the CON number to change (J is relative to the start of the block of CONs assigned to this model, e,g., 1 for first CON(K), 2 for CON(K+1), etc.) (input; no default allowed).
Real RDATA	Is the new CON value (input; unchanged by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	Unknown model name
IERR = 2	No CCT table entry for the model
IERR = 3	Model names do not match – model is aaa not xxx
IERR = 4	no CONs used in the model
IERR = 5	CON number outside of the CON range of this model
IERR = 6	Bus not found
IERR = 7	Machine not found
IERR = 8	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.23 CHANGE_CCTMCNPMOD_ICON

Use this API to change the value of an integer ICON of the machine protection model of a specified machine.

Batch command syntax:

```
BAT_CHANGE_CCTMCNPMOD_ICON  IBUS  'ID'  'MODEL'  J  IDATA
```

Python syntax:

```
ierr = change_cctmcnpmod_icon(ibus, id, model, j, idata)
```

Fortran syntax:

```
CALL CHANGE_CCTMCNPMOD_icon(IBUS, ID, MODEL, J, IDATA,  IERR)
```

where:

Integer IBUS	Is the bus number (input; no default allowed).
Character*2 ID	Is the circuit identifier (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change (J is relative to the start of the block of ICONs assigned to this model, e,g., 1 for first ICON(M), 2 for ICON(M+1), etc.) (input; no default allowed).
Integer IDATA	Is the new ICON value (input; unchanged by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	Unknown model name
IERR = 2	No CCT table entry for the model
IERR = 3	Model names do not match – model is aaa not xxx
IERR = 4	no ICONs used in the model
IERR = 5	ICON number outside of the ICON range of this model
IERR = 6	Bus not found
IERR = 7	Machine not found
IERR = 8	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.24 CHANGE_CCTMCNPMOD_VAR

Use this API to change the value of VAR of the machine protection model of a specified machine.
Batch command syntax:

```
BAT_CHANGE_CCTMCNPMOD_VAR  IBUS  'ID'  'MODEL'  J  RDATA
```

Python syntax:

```
ierr = change_cctmcnpmod_var(ibus, id, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_CCTMCNPMOD_VAR(IBUS, ID, MODEL, J, RDATA,  IERR)
```

where:

Integer IBUS	Is the bus number (input; no default allowed).
Character*2 ID	Is the circuit identifier (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the VAR number to change (J is relative to the start of the block of VARs assigned to this model, e,g., 1 for first VAR(L), 2 for VAR(L+1), etc.) (input; no default allowed).
Real RDATA	Is the new VAR value (input; unchanged by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	Unknown model name
IERR = 2	No CCT table entry for the model
IERR = 3	Model names do not match – model is aaa not xxx
IERR = 4	no VARs used in the model
IERR = 5	VAR number outside of the VAR range of this model
IERR = 6	Bus not found
IERR = 7	Machine not found
IERR = 8	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.25 CHANGE_CCTBUSOMOD_CHRICN

Use this API to change the value of a character ICON (i.e., an element of the CHRICN array) of the "other" type model attached to a specified bus. The corresponding integer ICON is set to 0.

Batch command syntax:

```
BAT_CHANGE_CCTBUSOMOD_CHRICN  IBUS  'MODEL' J  CHDATA
```

Python syntax:

```
ierr = change_cctbusomod_chricn(ibus, model, j, chdata)
```

Fortran syntax:

```
CALL CHANGE_CCTBUSOMOD_CHRICN(IBUS, MODEL, J, CHDATA,  IERR)
```

where:

Integer IBUS	Is the bus number (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change (J is relative to the start of the block of ICONs assigned to this model, e,g., 1 for first ICON(M), 2 for ICON(M+1), etc.) (input; no default allowed).
Character*2 CHDATA	Is the new character ICON value (input; blank by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	Unknown model name
IERR = 2	No CCT table entry for the model
IERR = 3	Model names do not match – model is aaa not xxx
IERR = 4	no ICONs used in the model
IERR = 5	ICON number outside of the ICON range of this model
IERR = 6	Bus not found
IERR = 7	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.26 CHANGE_CCTBUSOMOD_CON

Use this API to change the value of a real constant (CON) of the “other” type model attached to a specified bus.

Batch command syntax:

```
BAT_CHANGE_CCTBUSOMOD_CON  IBUS  'MODEL'  J  RDATA
```

Python syntax:

```
ierr = change_cctbusomod_con(ibus, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_CCTBUSOMOD_CON(IBUS, MODEL, J, RDATA,  IERR)
```

where:

Integer IBUS	Is the bus number (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the CON number to change (J is relative to the start of the block of CONs assigned to this model, e,g., 1 for first CON(K), 2 for CON(K+1), etc.) (input; no default allowed).
Real RDATA	Is the new CON value (input; unchanged by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	Unknown model name
IERR = 2	No CCT table entry for the model
IERR = 3	Model names do not match – model is aaa not xxx
IERR = 4	no CONs used in the model
IERR = 5	CON number outside of the CON range of this model
IERR = 6	Bus not found
IERR = 7	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.27 CHANGE_CCTBUSOMOD_ICON

Use this API to change the value of an integer ICON of the “other” type model attached to a specified bus.
Batch command syntax:

```
BAT_CHANGE_CCTBUSOMOD_ICON  IBUS  'MODEL'  J  IDATA
```

Python syntax:

```
ierr = change_cctbusomod_icon(ibus, model, j, idata)
```

Fortran syntax:

```
CALL CHANGE_CCTBUSOMOD_icon(IBUS, MODEL, J, IDATA,  IERR)
```

where:

Integer IBUS	Is the bus number (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change (J is relative to the start of the block of ICONs assigned to this model, e,g., 1 for first ICON(M), 2 for ICON(M+1), etc.) (input; no default allowed).
Integer IDATA	Is the new ICON value (input; unchanged by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	Unknown model name
IERR = 2	No CCT table entry for the model
IERR = 3	Model names do not match – model is aaa not xxx
IERR = 4	no ICONs used in the model
IERR = 5	ICON number outside of the ICON range of this model
IERR = 6	Bus not found
IERR = 7	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.28 CHANGE_CCTBUSOMOD_VAR

Use this API to change the value of VAR of the “other” type model attached to a specified bus.
Batch command syntax:

```
BAT_CHANGE_CCTBUSOMOD_VAR  IBUS  'MODEL'  J  RDATA
```

Python syntax:

```
ierr = change_cctbusomod_var(ibus, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_CCTBUSOMOD_VAR(IBUS, MODEL, J, RDATA,  IERR)
```

where:

Integer IBUS	Is the bus number (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the VAR number to change (J is relative to the start of the block of VARs assigned to this model, e,g., 1 for first VAR(L), 2 for VAR(L+1), etc.) (input; no default allowed).
Real RDATA	Is the new VAR value (input; unchanged by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	Unknown model name
IERR = 2	No CCT table entry for the model
IERR = 3	Model names do not match – model is aaa not xxx
IERR = 4	no VARs used in the model
IERR = 5	VAR number outside of the VAR range of this model
IERR = 6	Bus not found
IERR = 7	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.29 CHANGE_CCT2DCOMOD_CHRICN

Use this API to change the value of a character ICON (i.e., an element of the CHRICN array) of the "other" type model attached to a specified 2-terminal dc line. The corresponding integer ICON is set to 0.

Batch command syntax:

```
BAT_CHANGE_CCT2DCOMOD_CHRICN  DC  'MODEL' J  CHDATA
```

Python syntax:

```
ierr = change_cct2dcomod_chricn(dc, model, j, chdata)
```

Fortran syntax:

```
CALL CHANGE_CCT2DCOMOD_CHRICN(DC, MODEL, J, CHDATA,  IERR)
```

where:

Character*12 DC	Is the quoted name of the two-terminal dc line for which model data is to be changed (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change (J is relative to the start of the block of ICONs assigned to this model, e,g., 1 for first ICON(M), 2 for ICON(M+1), etc.) (input; no default allowed).
Character*2 CHDATA	Is the new character ICON value (input; blank by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	Unknown model name
IERR = 2	No CCT table entry for the model
IERR = 3	Model names do not match – model is aaa not xxx
IERR = 4	no ICONs used in the model
IERR = 5	ICON number outside of the ICON range of this model
IERR = 6	DC line not found
IERR = 7	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.30 CHANGE_CCT2DCOMOD_CON

Use this API to change the value of a real constant (CON) of the “other” type model attached to a specified 2-terminal dc line.
Batch command syntax:

```
BAT_CHANGE_CCT2DCOMOD_CON  DC  'MODEL'  J  RDATA
```

Python syntax:

```
ierr = change_cct2dcomod_con(dc, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_CCT2DCOMOD_CON(DC, MODEL, J, RDATA,  IERR)
```

where:

Character*12 DC	Is the quoted name of the two-terminal dc line for which model data is to be changed (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the CON number to change (J is relative to the start of the block of CONs assigned to this model, e,g., 1 for first CON(K), 2 for CON(K+1), etc.) (input; no default allowed).
Real RDATA	Is the new CON value (input; unchanged by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	Unknown model name
IERR = 2	No CCT table entry for the model
IERR = 3	Model names do not match – model is aaa not xxx
IERR = 4	no CONs used in the model
IERR = 5	CON number outside of the CON range of this model
IERR = 6	DC line not found
IERR = 6	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.31 CHANGE_CCT2DCOMOD_ICON

Use this API to change the value of an integer ICON of the “other” type model attached to a specified 2-terminal dc line.
Batch command syntax:

```
BAT_CHANGE_CCT2DCOMOD_ICON  DC  'MODEL'  J  IDATA
```

Python syntax:

```
ierr = change_cct2dcomod_icon(dc, model, j, idata)
```

Fortran syntax:

```
CALL CHANGE_CCT2DCOMOD_icon(DC, MODEL, J, IDATA,  IERR)
```

where:

Character*12 DC	Is the quoted name of the two-terminal dc line for which model data is to be changed (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change (J is relative to the start of the block of ICONs assigned to this model, e,g., 1 for first ICON(M), 2 for ICON(M+1), etc.) (input; no default allowed).
Integer IDATA	Is the new ICON value (input; unchanged by default).
Integer IERR	Is the error code (output). IERR = 0 no error IERR = 1 Unknown model name IERR = 2 No CCT table entry for the model IERR = 3 Model names do not match – model is aaa not xxx IERR = 4 no ICONs used in the model IERR = 5 ICON number outside of the ICON range of this model IERR = 6 DC line not found IERR = 7 Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.32 CHANGE_CCT2DCOMOD_VAR

Use this API to change the value of VAR of the “other” type model attached to a specified 2-terminal dc line.

Batch command syntax:

```
BAT_CHANGE_CCT2DCOMOD_VAR  DC  'MODEL'  J  RDATA
```

Python syntax:

```
ierr = change_cct2dcomod_var(dc, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_CCT2DCOMOD_VAR(DC, MODEL, J, RDATA,  IERR)
```

where:

Character*12 DC	Is the quoted name of the two-terminal dc line for which model data is to be changed (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the VAR number to change (J is relative to the start of the block of VARs assigned to this model, e,g., 1 for first VAR(L), 2 for VAR(L+1), etc.) (input; no default allowed).
Real RDATA	Is the new VAR value (input; unchanged by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	Unknown model name
IERR = 2	No CCT table entry for the model
IERR = 3	Model names do not match – model is aaa not xxx
IERR = 4	no VARs used in the model
IERR = 5	VAR number outside of the VAR range of this model
IERR = 6	DC line not found
IERR = 7	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.33 CHANGE_CCTMCNOMOD_CON

Use this API to change the value of a real constant (CON) of the machine “other” model for a specified machine.

Batch command syntax:

```
BAT_CHANGE_CCTMCNOMOD_CON  IBUS  'ID'  'MODEL'  J  RDATA
```

Python syntax:

```
ierr = change_cctmcnomod_con(ibus, id, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_CCTMCNOMOD_CON(IBUS, ID, MODEL, J, RDATA,  IERR)
```

where:

Integer IBUS	Is the bus number (input; no default allowed).
Character*2 ID	Is the circuit identifier (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the CON number to change (J is relative to the start of the block of CONs assigned to this model, e,g., 1 for first CON(K), 2 for CON(K+1), etc.) (input; no default allowed).
Real RDATA	Is the new CON value (input; unchanged by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	Unknown model name
IERR = 2	No CCT table entry for the model
IERR = 3	Model names do not match – model is aaa not xxx
IERR = 4	no CONs used in the model
IERR = 5	CON number outside of the CON range of this model
IERR = 6	Bus not found
IERR = 7	Machine not found
IERR = 8	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.34 CHANGE_CCTMCNOMOD_CHRICN

Use this API to change the value of a character ICON (i.e., an element of the CHRICN array) of the machine “other” model for a specified machine. The corresponding integer ICON is set to 0.
Batch command syntax:

```
BAT_CHANGE_CCTMCNOMOD_CHRICN  IBUS  'ID'  'MODEL' J  CHDATA
```

Python syntax:

```
ierr = change_cctmcnomod_chricn(ibus, id, model, j, chdata)
```

Fortran syntax:

```
CALL CHANGE_CCTMCNOMOD_CHRICN(IBUS, ID, MODEL, J, CHDATA,  IERR)
```

where:

Integer IBUS	Is the bus number (input; no default allowed).
Character*2 ID	Is the circuit identifier (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change (J is relative to the start of the block of ICONs assigned to this model, e,g., 1 for first ICON(M), 2 for ICON(M+1), etc.) (input; no default allowed).
Character*2 CHDATA	Is the new character ICON value (input; blank by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	Unknown model name
IERR = 2	No CCT table entry for the model
IERR = 3	Model names do not match – model is aaa not xxx
IERR = 4	no ICONs used in the model
IERR = 5	ICON number outside of the ICON range of this model
IERR = 6	Bus not found
IERR = 7	Machine not found
IERR = 8	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.35 CHANGE_CCTMCNOMOD_ICON

Use this API to change the value of an integer ICON of the machine “other” model for a specified machine.

Batch command syntax:

```
BAT_CHANGE_CCTMCNOMOD_ICON  IBUS  'ID'  'MODEL'  J  IDATA
```

Python syntax:

```
ierr = change_cctmcnomod_icon(ibus, id, model, j, idata)
```

Fortran syntax:

```
CALL CHANGE_CCTMCNOMOD_icon(IBUS, ID, MODEL, J, IDATA,  IERR)
```

where:

Integer IBUS	Is the bus number (input; no default allowed).
Character*2 ID	Is the circuit identifier (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the ICON number to change (J is relative to the start of the block of ICONs assigned to this model, e,g., 1 for first ICON(M), 2 for ICON(M+1), etc.) (input; no default allowed).
Integer IDATA	Is the new ICON value (input; unchanged by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	Unknown model name
IERR = 2	No CCT table entry for the model
IERR = 3	Model names do not match – model is aaa not xxx
IERR = 4	no ICONs used in the model
IERR = 5	ICON number outside of the ICON range of this model
IERR = 6	Bus not found
IERR = 7	Machine not found
IERR = 8	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.36 CHANGE_CCTMCNOMOD_VAR

Use this API to change the value of VAR of the machine “other” model for a specified machine.
Batch command syntax:

```
BAT_CHANGE_CCTMCNOMOD_VAR  IBUS  'ID'  'MODEL'  J  RDATA
```

Python syntax:

```
ierr = change_cctmcnomod_var(ibus, id, model, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_CCTMCNOMOD_VAR(IBUS, ID, MODEL, J, RDATA,  IERR)
```

where:

Integer IBUS	Is the bus number (input; no default allowed).
Character*2 ID	Is the circuit identifier (input; no default allowed).
Character*16 MODEL	Is the model name (input; no default allowed).
Integer J	Is the VAR number to change (J is relative to the start of the block of VARs assigned to this model, e,g., 1 for first VAR(L), 2 for VAR(L+1), etc.) (input; no default allowed).
Real RDATA	Is the new VAR value (input; unchanged by default).

Integer IERR — Is the error code (output).

IERR = 0	no error
IERR = 1	Unknown model name
IERR = 2	No CCT table entry for the model
IERR = 3	Model names do not match – model is aaa not xxx
IERR = 4	no VARs used in the model
IERR = 5	VAR number outside of the VAR range of this model
IERR = 6	Bus not found
IERR = 7	Machine not found
IERR = 8	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.37 CHANGE_CCTMSCOMOD_CHRICN

Use this API to change the value of a character ICON (i.e., an element of the CHRICN array) of the miscellaneous "other" type model with the specified model instance. The corresponding integer ICON is set to 0.

```
BAT_CHANGE_CCTMSCOMOD_CHRICN  'MODEL' MINS J  CHDATA
```

Python syntax:

```
ierr = change_cctmscomod_chricn(model, mins, j, chdata)
```

Fortran syntax:

```
CALL CHANGE_CCTMSCOMOD_CHRICN(MODEL, MINS, J, CHDATA,  IERR)
```

where:

Character*16 MODEL	Is the model name (input; no default allowed).
Integer MINS	Is the model instance (>0) (input; no default allowed).
Integer J	Is the ICON number to change (J is relative to the start of the block of ICONs assigned to this model, e,g., 1 for first ICON(M), 2 for ICON(M+1), etc.) (input; no default allowed).
Character*2 CHDATA	Is the new character ICON value (input; blank by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	Unknown model name
IERR = 2	No CCT table entry for the model aaa with model instance xxx
IERR = 3	Model names do not match – model is aaa not xxx
IERR = 4	no ICONs used in the model
IERR = 5	ICON number outside of the ICON range of this model
IERR = 6	Invalid model instance value
IERR = 7	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.38 CHANGE_CCTMSCOMOD_CON

Use this API to change the value of a real constant (CON) of the miscellaneous "other" type model with the specified model instance.
Batch command syntax:

```
BAT_CHANGE_CCTMSCOMOD_CON  'MODEL'  MINS J  RDATA
```

Python syntax:

```
ierr = change_cctmscomod_con(model, mins, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_CCTMSCOMOD_CON(MODEL, MINS, J, RDATA,  IERR)
```

where:

Character*16 MODEL	Is the model name (input; no default allowed).
Integer MINS	Is the model instance (>0) (input; no default allowed).
Integer J	Is the CON number to change (J is relative to the start of the block of CONs assigned to this model, e,g., 1 for first CON(K), 2 for CON(K+1), etc.) (input; no default allowed).
Real RDATA	Is the new CON value (input; unchanged by default).

Integer IERR	Is the error code (output).	
	IERR = 0	no error
	IERR = 1	Unknown model name
	IERR = 2	No CCT table entry for the model aaa with model instance xxx
	IERR = 3	Model names do not match – model is aaa not xxx
	IERR = 4	no CONs used in the model
	IERR = 5	CON number outside of the CON range of this model
	IERR = 6	Invalid model instance value
	IERR = 7	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.39 CHANGE_CCTMSCOMOD_ICON

Use this API to change the value of an integer ICON of the miscellaneous "other" type model with the specified model instance.

Batch command syntax:

```
BAT_CHANGE_CCTMSCOMOD_ICON  'MODEL'  MINS J  IDATA
```

Python syntax:

```
ierr = change_cctmscomod_icon(model, mins, j, idata)
```

Fortran syntax:

```
CALL CHANGE_CCTMSCOMOD_icon(MODEL, MINS, J, IDATA,  IERR)
```

where:

Character*16 MODEL	Is the model name (input; no default allowed).
Integer MINS	Is the model instance (>0) (input; no default allowed).
Integer J	Is the ICON number to change (J is relative to the start of the block of ICONs assigned to this model, e,g., 1 for first ICON(M), 2 for ICON(M+1), etc.) (input; no default allowed).
Integer IDATA	Is the new ICON value (input; unchanged by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	Unknown model name
IERR = 2	No CCT table entry for the model aaa with model instance xxx
IERR = 3	Model names do not match – model is aaa not xxx
IERR = 4	no ICONs used in the model
IERR = 5	ICON number outside of the ICON range of this model
IERR = 6	Invalid model instance value
IERR = 7	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.40 CHANGE_CCTMSCOMOD_VAR

Use this API to change the value of VAR of the miscellaneous “other” type model with the specified model instance.

Batch command syntax:

```
BAT_CHANGE_CCTMSCOMOD_VAR  'MODEL'  MINS J  RHDATA
```

Python syntax:

```
ierr = change_cctmscomod_var(model, mins, j, rdata)
```

Fortran syntax:

```
CALL CHANGE_CCTMSCOMOD_VAR(MODEL, MINS, J, RDATA,  IERR)
```

where:

Character*16 MODEL	Is the model name (input; no default allowed).
Integer MINS	Is the model instance (>0) (input; no default allowed).
Integer J	Is the VAR number to change (J is relative to the start of the block of VARs assigned to this model, e,g., 1 for first VAR(L), 2 for VAR(L+1), etc.) (input; no default allowed).
Real RDATA	Is the new VAR value (input; unchanged by default).
Integer IERR	Is the error code (output).

IERR = 0	no error
IERR = 1	Unknown model name
IERR = 2	No CCT table entry for the model aaa with model instance xxx
IERR = 3	Model names do not match – model is aaa not xxx
IERR = 4	no VARs used in the model
IERR = 5	VAR number outside of the VAR range of this model
IERR = 6	Invalid model instance value
IERR = 7	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.41 CCT2WTDMOD_REMOVE

Use this API to remove a 2-winding transformer device model attached to the specified 2-winding transformer.

Batch command syntax:

```
BAT_CCT2WTDMOD_REMOVE   IBUS  JBUS  'ID'
```

Python syntax:

```
ierr = cct2wtdmod_remove(ibus, jbus, id)
```

Fortran syntax:

```
CALL CCT2WTDMOD_REMOVE(IBUS, JBUS, ID, IERR)
```

where:

Integer IBUS	Is the from bus number (input; no default allowed).
Integer JBUS	Is the to bus number (input; no default allowed).
Character*2 ID	Is the circuit identifier (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error IERR = 1 No 2-winding transformer device model on this branch IERR = 2 Bus not found IERR = 3 2-winding transformer not found IERR = 4 Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.42 CCT3WTDMOD_REMOVE

Use this API to remove a 3-winding transformer device model attached to the specified 3-winding transformer.

Batch command syntax:

```
BAT_CCT3WTDMOD_REMOVE  IBUS  JBUS  KBUS 'ID'
```

Python syntax:

```
ierr = cct3wtdmod_remove(ibus, jbus, kbus, id)
```

Fortran syntax:

```
CALL CCT3WTDMOD_REMOVE (IBUS, JBUS, KBUS, ID, IERR)
```

where:

Integer IBUS	Is the from bus number (input; no default allowed).
Integer JBUS	Is the to bus number (input; no default allowed).
Integer KBUS	Is the third bus number (input; no default allowed).
Character*2 ID	Is the circuit identifier (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error IERR = 1 No 3-winding transformer device model on this branch IERR = 2 Bus not found IERR = 3 3-winding transformer not found IERR = 4 Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.43 CCTMCNPMOD_REMOVE

Use this API to remove a machine protection model attached to the specified machine.

Batch command syntax:

```
BAT_CCTMCNPMOD_REMOVE_  IBUS  'ID'  'MODEL'
```

Python syntax:

```
ierr = cctmcnpmod_remove(ibus, id, model)
```

Fortran syntax:

```
CALL CCTMCNPMOD_remove(IBUS, ID, MODEL, IERR)
```

where:

Integer IBUS	Is the bus number (input; no default allowed).
Character*2 ID	Is the machine identifier (input; no default allowed).
Character*16 MODEL	Is the name of he model to be removed (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 no error
	IERR = 1 No machine protection model at this machine
	IERR = 2 Bus not found
	IERR = 3 Machine not found
	IERR = 4 Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.44 CCTBUSOMOD_REMOVE

Use this API to remove an “other” type model attached to a bus.

Batch command syntax:

```
BAT_CCTBUSOMOD_REMOVE  IBUS  MODEL
```

Python syntax:

```
ierr = cctbusomod_remove(ibus, model)
```

Fortran syntax:

```
CALL CCTBUSOMOD_REMOVE(IBUS, MODEL, IERR)
```

where:

Integer IBUS	Is the bus number (input; no default allowed).
Character*16 MODEL	Is the name of the model to be removed (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error IERR = 1 No bus other model IERR = 2 Bus not found IERR = 3 Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.45 CCT2DCOMOD_REMOVE

Use this API to remove an “other” type model attached to a 2-terminal dc line.

Batch command syntax:

```
BAT_CCT2DCOMOD_REMOVE  DC  MODEL
```

Python syntax:

```
ierr = cct2dcomod_remove(dc, model)
```

Fortran syntax:

```
CALL CCT2DCOMOD_REMOVE(DC, MODEL, IERR)
```

where:

Character*12 DC	Is the quoted name of the two-terminal dc line for which model data is to be changed (input; no default allowed).
Character*16 MODEL	Is the name of the model to be removed (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error IERR = 1 No “other” model at this dc line IERR = 2 DC line not found IERR = 3 Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.46 CCTMCNOMOD_REMOVE

Use this API to remove a machine other model attached to the specified machine.

Batch command syntax:

```
BAT_CCTMCNOMOD_REMOVE_  IBUS  'ID'  'MODEL'
```

Python syntax:

```
ierr = cctmcnomod_remove(ibus, id, model)
```

Fortran syntax:

```
CALL CCTMCNOMOD_remove(IBUS, ID, MODEL, IERR)
```

where:

Integer IBUS	Is the bus number (input; no default allowed).
Character*2 ID	Is the machine identifier (input; no default allowed).
Character*16 MODEL	Is the name of the model to be removed (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 — no error
	IERR = 1 — No machine protection model at this machine
	IERR = 2 — Bus not found
	IERR = 3 — Machine not found
	IERR = 4 — Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.47 CCTMSCOMOD_REMOVE

Use this API to remove a miscellaneous “other” type model with the specified model instance.

Batch command syntax:

```
BAT_CCTMSCOMOD_REMOVE  MINS
```

Python syntax:

```
ierr = cctmscomod_remove(mins)
```

Fortran syntax:

```
CALL CCTMSCOMOD_REMOVE(MINS, IERR)
```

where:

Integer IBUS	Is the bus number (input; no default allowed).	
Integer MINS	Is the model instance (>0) (input; no default allowed).	
Integer IERR	Is the error code (output).	
	IERR = 0	no error
	IERR = 1	No bus other model with this model instance at this bus
	IERR = 2	Invalid model instance
	IERR = 3	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.48 CCTBRNDMOD_STATUS

Use this API to change the status of a branch device model attached to the specified branch.

Batch command syntax:

```
BAT_CCTBRNDMOD_STATUS  IBUS  JBUS  'ID'  STATUS
```

Python syntax:

```
ierr = cctbrndmod_status(ibus, jbus, id, status)
```

Fortran syntax:

```
CALL CCTBRNDMOD_STATUS(IBUS, JBUS, ID, STATUS, IERR)
```

where:

Integer IBUS	Is the from bus number (input; no default allowed).
Integer JBUS	Is the to bus number (input; no default allowed).
Character*2 ID	Is the circuit identifier (input; no default allowed).
Integer STATUS	Is the new model status (input; 1 by default) STATUS = 0 out-of-service STATUS = 1 in-service
Integer IERR	Is the error code (output). IERR = 0 no error IERR = 1 No branch device model on this branch IERR = 2 Invalid STATUS value IERR = 3 Bus not found IERR = 4 Branch not found IERR = 5 Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.49 CCT2WTDMOD_STATUS

Use this API to change the status of a 2-winding transformer device model attached to the specified 2-winding transformer.

Batch command syntax:

```
BAT_CCT2WTDMOD_STATUS   IBUS  JBUS  'ID'  STATUS
```

Python syntax:

```
ierr = cct2wtdmod_status(ibus, jbus, id, status)
```

Fortran syntax:

```
CALL CCT2WTDMOD_STATUS(IBUS, JBUS, ID, STATUS, IERR)
```

where:

Integer IBUS	Is the from bus number (input; no default allowed).
Integer JBUS	Is the to bus number (input; no default allowed).
Character*2 ID	Is the circuit identifier (input; no default allowed).
Integer STATUS	Is the new model status (input; 1 by default) STATUS = 0 out-of-service STATUS = 1 in-service
Integer IERR	Is the error code (output). IERR = 0 no error IERR = 1 No 2-winding transformer device model on this branch IERR = 2 Invalid STATUS value IERR = 3 Bus not found IERR = 4 2-winding transformer not found IERR = 5 Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.50 CCT3WTDMOD_STATUS

Use this API to change the status of a 3-winding transformer device model attached to the specified 3-winding transformer.

Batch command syntax:

```
BAT_CCT3WTDMOD_STATUS  IBUS  JBUS  KBUS 'ID'  STATUS
```

Python syntax:

```
ierr = cct3wtdmod_status(ibus, jbus, kbus, id, status)
```

Fortran syntax:

```
CALL CCT3WTDMOD_STATUS (IBUS, JBUS, KBUS, ID, STATUS, IERR)
```

where:

Integer IBUS	Is the from bus number (input; no default allowed).
Integer JBUS	Is the to bus number (input; no default allowed).
Integer KBUS	Is the third bus number (input; no default allowed).
Character*2 ID	Is the circuit identifier (input; no default allowed).
Integer STATUS	Is the new model status (input; 1 by default) STATUS = 0 out-of-service STATUS = 1 in-service
Integer IERR	Is the error code (output). IERR = 0 no error IERR = 1 No 3-winding transformer device model on this branch IERR = 2 Invalid STATUS value IERR = 3 Bus not found IERR = 4 3-winding transformer not found IERR = 5 Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.51 CCTMCNPMOD_STATUS

Use this API to change status of a machine protection model attached to the specified machine.
Batch command syntax:

```
BAT_CCTMCNPMOD_STATUS  IBUS  'ID'  MODEL  STATUS
```

Python syntax:

```
ierr = cctmcnpmod_status(ibus, id, model, status)
```

Fortran syntax:

```
CALL CCTMCNPMOD_STATUS(IBUS, ID, MODEL, STATUS, IERR)
```

where:

Integer IBUS	Is the bus number (input; no default allowed).
Character*2 ID	Is the machine identifier (input; no default allowed).
Character*16 MODEL	Is the name of the model whose status is to be changed (input; no default allowed).
Integer STATUS	Is the new model status (input; 1 by default) STATUS = 0 out-of-service STATUS = 1 in-service
Integer IERR	Is the error code (output). IERR = 0 no error IERR = 1 No machine protection model at this machine IERR = 2 Invalid STATUS value IERR = 3 Bus not found IERR = 4 Machine not found IERR = 5 Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.52 CCTBUSOMOD_STATUS

Use this API to change status of an “other” type model attached to a bus.
Batch command syntax:

```
BAT_CCTBUSOMOD_STATUS  IBUS  MODEL  STATUS
```

Python syntax:

```
ierr = cctbusomod_status(ibus, model, status)
```

Fortran syntax:

```
CALL CCTBUSOMOD_STATUS(IBUS, MODEL, STATUS, IERR)
```

where:

Integer IBUS	Is the bus number (input; no default allowed).
Character*16 MODEL	Is the name of the model whose status is to be changed (input; no default allowed).
Integer STATUS	Is the new model status (input; 1 by default) STATUS = 0 out-of-service STATUS = 1 in-service
Integer IERR	Is the error code (output). IERR = 0 no error IERR = 1 No bus other model at this bus IERR = 2 Invalid STATUS value IERR = 3 Bus not found IERR = 4 Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.53 CCT2DCOMOD_STATUS

Use this API to change status of an “other” type model attached to a 2-terminal dc line.
Batch command syntax:

```
BAT_CCT2DCOMOD_STATUS  DC  MODEL  STATUS
```

Python syntax:

```
ierr = cct2dcomod_status(dc, model, status)
```

Fortran syntax:

```
CALL CCT2DCOMOD_STATUS(DC, MODEL, STATUS, IERR)
```

where:

Character*12 DC	Is the quoted name of the two-terminal dc line for which model data is to be changed (input; no default allowed).
Character*16 MODEL	Is the name of the model whose status is to be changed (input; no default allowed).
Integer STATUS	Is the new model status (input; 1 by default) STATUS = 0 out-of-service STATUS = 1 in-service
Integer IERR	Is the error code (output). IERR = 0 no error IERR = 1 No “other” model at this dc line IERR = 2 Invalid STATUS value IERR = 3 DC line not found IERR = 4 Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.54 CCTMCNOMOD_STATUS

Use this API to change status of a machine other model attached to the specified machine.
Batch command syntax:

```
BAT_CCTMCNOMOD_STATUS  IBUS  'ID'  MODEL  STATUS
```

Python syntax:

```
ierr = cctmcnomod_status(ibus, id, model, status)
```

Fortran syntax:

```
CALL CCTMCNOMOD_STATUS(IBUS, ID, MODEL, STATUS, IERR)
```

where:

Integer IBUS	Is the bus number (input; no default allowed).
Character*2 ID	Is the machine identifier (input; no default allowed).
Character*16 MODEL	Is the name of the model whose status is to be changed (input; no default allowed).
Integer STATUS	Is the new model status (input; 1 by default) STATUS = 0 out-of-service STATUS = 1 in-service
Integer IERR	Is the error code (output). IERR = 0 no error IERR = 1 No machine other model at this machine IERR = 2 Invalid STATUS value IERR = 3 Bus not found IERR = 4 Machine not found IERR = 5 Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.55 CCTMSCOMOD_STATUS

Use this API to change status of a miscellaneous "other" type model with the specified model instance.

Batch command syntax:

```
BAT_CCTMSCOMOD_STATUS  MINS  STATUS
```

Python syntax:

```
ierr = cctmscomod_status(mins, status)
```

Fortran syntax:

```
CALL CCTMSCOMOD_STATUS(MINS, STATUS, IERR)
```

where:

Integer MINS	Is the model instance (>0) (input; no default allowed).
Integer STATUS	Is the new model status (input; 1 by default) STATUS = 0 out-of-service STATUS = 1 in-service
Integer IERR	Is the error code (output). IERR = 0 no error IERR = 1 No miscellaneous other model with this model instance IERR = 2 Invalid STATUS value IERR = 3 Invalid model instance IERR = 4 Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.56 CCTMOD_USER

Use this API to list user-written CCT model or to remove unused user-written model definitions.

Batch command syntax:

```
BAT_CCTMOD_USER   OPT
```

Python syntax:

```
ierr = cctmod_user(opt)
```

Fortran syntax:

```
CALL CCTMOD_USER(OPT, IERR)
```

where:

Integer OPT	Is the function code (input; 0 by default) OPT = 0 list user-written model definitions OPT = 1 remove unused user-written model definitions
Integer IERR	Is the error code (output). IERR = 0 no error IERR = 1 Invalid OPT value IERR = 2 Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.12.57 CCTMOD_PACK

Use this API to pack CCT model definition tables.

Batch command syntax:

```
BAT_CCTMOD_PACK
```

Python syntax:

```
ierr = cctmod_pack()
```

Fortran syntax:

```
CALL CCTMOD_PACK(IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error
	IERR = 1	Prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.13 Disturbance APIs

This section describes those APIs corresponding to functions available from the dynamics simulation ***Disturbance*** menu.

4.13.1 DIST_3WIND_CLOSE

Use this API to set a three-winding transformer to in-service during dynamic simulations.

Batch command syntax:

```
BAT_DIST_3WIND_CLOSE  IBUS  JBUS  KBUS  'ID'
```

Python syntax:

```
ierr = dist_3wind_close(ibus, jbus, kbus, id)
```

Fortran syntax:

```
CALL DIST_3WIND_CLOSEAPI(IBUS, JBUS, KBUS, ID, IERR)
```

where:

Integer IBUS	Is the bus number of one of the buses (input; no default allowed).
Integer JBUS	Is the bus number of one of the other buses (input; no default allowed).
Integer KBUS	Is the bus number of the third bus (input; no default allowed).
Character ID*2	Is the circuit identifier (input; ’1’ by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error occurred.
	IERR = 1 — STRT or MSTR has not been successfully executed.
	IERR = 2 — bus not found.
	IERR = 3 — three-winding transformer not found.
	IERR = 4 — three-winding transformer is already in-service.
	IERR = 5 — bus is out-of-service.
	IERR = 6 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.13.2 DIST_3WIND_FAULT

Use this API to apply a fault at the IBUS end of a three-winding transformer during dynamic simulations.

Batch command syntax:

```
BAT_DIST_3WIND_FAULT  IBUS  JBUS  KBUS  'ID'  UNITS,  BASEKV,
VALUES(1)  VALUES(2)
```

Python syntax:

```
ierr = dist_3wind_fault(ibus,jbus,kbus,id,units,basekv,values)
```

Fortran syntax:

```
CALL DIST_3WIND_FAULTAPI(IBUS,JBUS,KBUS,ID,UNITS,BASEKV,VALUES,IERR)
```

where:

Integer IBUS	Is the bus number of the bus at which the fault is to be placed (input; no default allowed).
Integer JBUS	Is the bus number of one of the other buses (input; no default allowed).
Integer KBUS	Is the bus number of the third bus (input; no default allowed).
Character ID*2	Is the circuit identifier (input; ’1’ by default).
Integer UNITS	Is the units in which fault admittance or impedance is specified in VALUES (input; 1 by default) UNITS = 1 admittance in MVA. UNITS = 2 admittance in mhos. UNITS = 3 impedance in ohms.
Real BASEKV	Is the base voltage in kV used to calculate the per unit fault admittance if UNITS is 2 or 3; ignored if UNITS is 1. If BASEKV is specified as 0.0, the base voltage of bus IBUS is used (input; 0.0 by default).
Real VALUES(2)	Is an array of two elements (input). VALUES(1) is the real component of the complex fault admittance or impedance according to the value specified for UNITS (0.0 by default). VALUES(2) is the reactive component of the complex fault admittance or impedance according to the value specified for UNITS (-2.0E11 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	STRT or MSTR has not been successfully executed.
	IERR = 2	bus not found.
	IERR = 3	three-winding transformer not found.
	IERR = 4	three-winding transformer winding is out-of-service.
	IERR = 5	invalid UNITS value.
	IERR = 6	invalid BASEKV value (<0.0).
	IERR = 7	both BASEKV and the base voltage of bus IBUS are 0.0.
	IERR = 8	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.13.3 DIST_3WIND_TRIP

Use this API to set a three-winding transformer to out-of-service during dynamic simulations.

Batch command syntax:

```
BAT_DIST_3WIND_TRIP  IBUS  JBUS  KBUS  'ID'
```

Python syntax:

```
ierr = dist_3wind_trip(ibus, jbus, kbus, id)
```

Fortran syntax:

```
CALL DIST_3WIND_TRIPAPI(IBUS, JBUS, KBUS, ID, IERR)
```

where:

Integer IBUS	Is the bus number of one of the buses (input; no default allowed).
Integer JBUS	Is the bus number of one of the other buses (input; no default allowed).
Integer KBUS	Is the bus number of the third bus (input; no default allowed).
Character ID*2	Is the circuit identifier (input; ’1’ by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error occurred.
	IERR = 1 — STRT or MSTR has not been successfully executed.
	IERR = 2 — bus not found.
	IERR = 3 — three-winding transformer not found.
	IERR = 4 — three-winding transformer is already out-of-service.
	IERR = 5 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.13.4 DIST_BRANCH_CLOSE

Use this API to set a non-transformer branch or a two-winding transformer to in-service during dynamic simulations.

Batch command syntax:

```
BAT_DIST_BRANCH_CLOSE  IBUS  JBUS  'ID'
```

Python syntax:

```
ierr = dist_branch_close(ibus, jbus, id)
```

Fortran syntax:

```
CALL DIST_BRANCH_CLOSEAPI(IBUS, JBUS, ID, IERR)
```

where:

Integer IBUS	Is the bus number of one of the buses (input; no default allowed).
Integer JBUS	Is the bus number of the other bus (input; no default allowed).
Character ID*2	Is the circuit identifier (input; ’1’ by default).
Integer IERR	Is the error code (output).
	IERR = 0 — no error occurred.
	IERR = 1 — STRT or MSTR has not been successfully executed.
	IERR = 2 — bus not found.
	IERR = 3 — branch not found.
	IERR = 4 — branch is already in-service.
	IERR = 5 — bus is out-of-service.
	IERR = 6 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.13.5 DIST_BRANCH_FAULT

Use this API to apply a fault at the IBUS end of a non-transformer branch or a two-winding transformer during dynamic simulations.

Batch command syntax:

```
BAT_DIST_BRANCH_FAULT  IBUS  JBUS  'ID'  UNITS,  BASEKV,  VALUES(1)
VALUES(2)
```

Python syntax:

```
ierr = dist_branch_fault(ibus, jbus, id, units, basekv, values)
```

Fortran syntax:

```
CALL DIST_BRANCH_FAULTAPI(IBUS,JBUS,ID,UNITS,BASEKV,VALUES,IERR)
```

where:

Integer IBUS	Is the bus number of the bus at which the fault is to be placed (input; no default allowed).
Integer JBUS	Is the bus number of the other bus (input; no default allowed).
Character ID*2	Is the circuit identifier (input; ’1’ by default).
Integer UNITS	Is the units in which fault admittance or impedance is specified in VALUES (input; 1 by default). UNITS = 1 admittance in MVA. UNITS = 2 admittance in mhos. UNITS = 3 impedance in ohms.
Real BASEKV	Is the base voltage in kV used to calculate the per unit fault admittance if UNITS is 2 or 3; ignored if UNITS is 1. If BASEKV is specified as 0.0, the base voltage of bus IBUS is used (input; 0.0 by default).
Real VALUES(2)	Is an array of two elements (input). VALUES(1) is the real component of the complex fault admittance or impedance according to the value specified for UNITS (0.0 by default). VALUES(2) is the reactive component of the complex fault admittance or impedance according to the value specified for UNITS (-2.0E11 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	STRT or MSTR has not been successfully executed.
	IERR = 2	bus not found.
	IERR = 3	branch not found.
	IERR = 4	branch is out-of-service.
	IERR = 5	invalid UNITS value.
	IERR = 6	invalid BASEKV value (<0.0).
	IERR = 7	both BASEKV and the base voltage of bus IBUS are 0.0.
	IERR = 8	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.13.6 DIST_BRANCH_TRIP

Use this API to set a non-transformer branch or a two-winding transformer to out-of-service during dynamic simulations.

Batch command syntax:

```
BAT_DIST_BRANCH_TRIP  IBUS  JBUS  'ID'
```

Python syntax:

```
ierr = dist_branch_trip(ibus, jbus, id)
```

Fortran syntax:

```
CALL DIST_BRANCH_TRIPAPI(IBUS, JBUS, ID, IERR)
```

where:

Integer IBUS	Is the bus number of one of the buses (input; no default allowed).
Integer JBUS	Is the bus number of the other bus (input; no default allowed).
Character ID*2	Is the circuit identifier (input; ’1’ by default).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 STRT or MSTR has not been successfully executed.
	IERR = 2 bus not found.
	IERR = 3 branch not found.
	IERR = 4 branch is already out-of-service.
	IERR = 5 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.13.7 DIST_BUS_FAULT

Use this API to apply a fault at a bus during dynamic simulations.

Batch command syntax:

```
BAT_DIST_BUS_FAULT  IBUS  UNITS  BASEKV  VALUES(1)  VALUES(2)
```

Python syntax:

```
ierr = dist_bus_fault(ibus, units, basekv, values)
```

Fortran syntax:

```
CALL DIST_BUS_FAULTAPI(IBUS, UNITS, BASEKV, VALUES, IERR)
```

where:

Integer IBUS	Is the bus number of the bus at which the fault is to be placed (input; no default allowed).
Integer UNITS	Is the units in which fault admittance or impedance is specified in VALUES (input; 1 by default). UNITS = 1 admittance in MVA. UNITS = 2 admittance in mhos. UNITS = 3 impedance in ohms.
Real BASEKV	Is the base voltage in kV used to calculate the per unit fault admittance if UNITS is 2 or 3; ignored if UNITS is 1. If BASEKV is specified as 0.0, the base voltage of bus IBUS is used (input; 0.0 by default).
Real VALUES(2)	Is an array of two elements (input). VALUES(1) is the real component of the complex fault admittance or impedance according to the value specified for UNITS (0.0 by default). VALUES(2) is the reactive component of the complex fault admittance or impedance according to the value specified for UNITS (-2.0E11 by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred IERR = 1 STRT or MSTR has not been successfully executed IERR = 2 bus not found IERR = 3 bus is out-of-service IERR = 4 invalid UNITS value IERR = 5 invalid BASEKV value (<0.0) IERR = 6 both BASEKV and the base voltage of bus IBUS are 0.0 IERR = 7 fixed bus shunt table is full IERR = 8 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.13.8 DIST_BUS_TRIP

Use this API to disconnect a bus during dynamic simulations.

Batch command syntax:

```
BAT_DIST_BUS_TRIP  IBUS
```

Python syntax:

```
ierr = dist_bus_trip(ibus)
```

Fortran syntax:

```
CALL DIST_BUS_TRIPAPI(IBUS, IERR)
```

where:

Integer IBUS	Is the bus number of the bus to be disconnected (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 — no error occurred.
	IERR = 1 — STRT or MSTR not successfully executed.
	IERR = 2 — bus not found.
	IERR = 3 — bus is already disconnected.
	IERR = 4 — prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.13.9 DIST_CLEAR_FAULT

Use this API to clear a fault during dynamic simulations. The fault must have previously been applied using one of the following APIs:

- DIST_3WIND_FAULT
- DIST_BRANCH_FAULT
- DIST_BUS_FAULT
- DIST_SCMU_FAULT
- DIST_SPCB_FAULT

Batch command syntax:

```
BAT_DIST_CLEAR_FAULT  IFAULT
```

Python syntax:

```
ierr = dist_clear_fault(ifault)
```

Fortran syntax:

```
CALL DIST_CLEAR_FAULTAPI(IFAULT, IERR)
```

where:

Integer IFAULT	Is the index in the fault memory tables of the fault to be cleared (input; 1 by default). Faults are stored in the fault memory tables in the order in which they are applied; the index assigned to a fault is the next available location in the tables. Each time a fault is cleared using this API, the fault memory tables are compressed. Thus, if there are three faults active, the most recently applied fault will have index number 3.
Integer IERR	Is the error code (output). IERR = 0 no error occurred IERR = 1 STRT or MSTR not successfully executed IERR = 2 bus not found IERR = 3 branch not found IERR = 4 three-winding transformer not found IERR = 5 fixed bus shunt not found IERR = 6 no faults in the fault memory tables IERR = 7 invalid IFAULT value IERR = 8 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.13.10 DIST_MACHINE_TRIP

Use this API to set a machine to out-of-service during dynamic simulations.

Batch command syntax:

```
BAT_DIST_MACHINE_TRIP  IBUS  'ID'
```

Python syntax:

```
ierr = dist_machine_trip(ibus,id)
```

Fortran syntax:

```
CALL DIST_MACHINE_TRIPAPI(IBUS, ID, IERR)
```

where:

Integer IBUS	Is the bus at which the machine to be tripped is connected (input; no default allowed).
Character ID*2	Is the identifier of the machine to be tripped (input; '1' by default).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR = 1 STRT or MSTR not successfully executed.
	IERR = 2 bus not found.
	IERR = 3 machine not found.
	IERR = 4 machine is already out-of-service.
	IERR = 5 prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.13.11 DIST_SCMU_FAULT

Use this API to calculate an unbalanced fault and apply the equivalent positive sequence fault admittance at a bus during dynamic simulations.

Batch command syntax:

```
BAT_DIST_SCMU_FAULT  OPTIONS(1)...OPTIONS(4)  VALUES(1)...VALUES(4)
```

Python syntax:

```
ierr = dist_scmu_fault(options, values)
```

Fortran syntax:

```
CALL DIST_SCMU_FAULTAPI(OPTIONS, VALUES, IERR)
```

where:

Integer OPTIONS(4)	Is an array of four elements specifying options and input values (input). The first two elements of OPTIONS define options to be used in the construction of the sequence admittance matrices; the remaining entries define the fault to be applied.
	OPTIONS(1) — dc line and FACTS device option (0 by default).
	OPTIONS(1) = 0 — block and ignore.
	OPTIONS(1) = 1 — convert to constant admittance load.
	OPTIONS(2) — transformer impedance correction option (1 by default).
	OPTIONS(2) = 0 — do not apply to zero sequence.
	OPTIONS(2) = 1 — apply to zero sequence.
	OPTIONS(3) — unbalanced type (1 by default).
	OPTIONS(3) = 1 — line-to-ground fault.
	OPTIONS(3) = 2 — line-to-line or line-to-line-to-ground fault.
	OPTIONS(4) — bus to be faulted (no default allowed).
Real VALUES(4)	Is an array of four elements specifying input values used to define the fault (input). They are dependent on the value specified for OPTIONS(3).
	VALUES(1) — resistance of the complex L-G (if OPTIONS(3) is 1) or L-L (if OPTIONS(3) is 2) fault impedance in pu (0.0 by default).
	VALUES(2) — reactance of the complex L-G (if OPTIONS(3) is 1) or L-L (if OPTIONS(3) is 2) fault impedance in pu (0.0 by default).
	VALUES(3) — resistance of the complex L-G fault impedance in pu (if OPTIONS(3) is 2) (0.0 by default).
	VALUES(4) — reactance of the complex L-G fault impedance in pu (if OPTIONS(3) is 2) (0.0 by default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred
	IERR = 1	STRT or MSTR not successfully executed
	IERR = 2	fixed bus shunt table is full
	IERR = 3	error encountered in executing SCMUAPI
	IERR = 4	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.13.12 DIST_SPCB_FAULT

This API routine is obsolete. It has been replaced by the API routine DIST_SPCB_FAULT_2, and is implemented by a call to the DIST_SPCB_FAULT_2 API routine.

Use this API to calculate a branch unbalance and apply the equivalent positive sequence pi model at a branch during dynamic simulations.

Batch command syntax:

```
BAT_DIST_SPCB_FAULT  IBUS  JBUS  'ID'  OPTIONS(1)...OPTIONS(6)
VALUES(1)...VALUES(5)
```

Python syntax:

```
ierr = dist_spcb_fault(ibus, jbus, id, options, values)
```

Fortran syntax:

```
CALL DIST_SPCB_FAULTAPI(IBUS, JBUS, ID, OPTIONS, VALUES, IERR)
```

where:

Integer IBUS	Is the from bus of branch to be subjected to unbalance (input; no default allowed).
Integer JBUS	Is the to bus of branch to be subjected to unbalance (input; no default allowed).
Character ID*2	Is the circuit identifier of branch to be subjected to unbalance (input; '1' by default).
Integer OPTIONS(6)	Is an array of six elements specifying unbalance options (input). The values are as follows:

OPTIONS(1)	unbalance code (1 by default).
OPTIONS(1) = 1	one phase open.
OPTIONS(1) = 2	two phases open.
OPTIONS(1) = 3	in-line fault.
OPTIONS(1) = 4	one breaker open.
OPTIONS(2)	path to ground code (used when OPTIONS(1) is 1 or 4) (0 by default).
OPTIONS(2) = 0	no path to ground.
OPTIONS(2) = 1	include a path to ground.
OPTIONS(3)	type of in-line fault code (used when OPTIONS(1) is 3) (1 by default).
OPTIONS(3) = 1	line-to-ground fault.
OPTIONS(3) = 2	line-to-line-to-ground fault.
OPTIONS(3) = 3	three phase fault.
OPTIONS(4)	open end code (used when OPTIONS(1) is 4) (1 by default).
OPTIONS(4) = 0	breaker at bus JBUS is open.
OPTIONS(4) = 1	breaker at bus IBUS is open.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTIONS(5)	dc line and FACTS device option (0 by default).
	OPTIONS(5) = 0	block and ignore.
	OPTIONS(5) = 1	convert to constant admittance load.
	OPTIONS(6)	transformer impedance correction option (1 by default).
	OPTIONS(6) = 0	do not apply to zero sequence.
	OPTIONS(6) = 1	apply to zero sequence.
Real VALUES(5)	Is an array of five elements specifying unbalance parameters (input).	
	VALUES(1)	fault location as fraction of line from bus IBUS; 0.0 < VALUES(1) < 1.0 (used when OPTIONS(1) is 1 or 4 and OPTIONS(2) is 1; or when OPTIONS(1) is 3) (0.5 by default).
	VALUES(2) & (3)	RI-g and XI-g (used when OPTIONS(1) is 1 or 4 and OPTIONS(2) is 1; or when OPTIONS(1) is 3 and OPTIONS(3) is 1 or 2) (0.0 by default).
	VALUES(4) & (5)	RI-l and XI-l (used when OPTIONS(1) is 3 and OPTIONS(3) is 2) (0.0 by default).
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	STRT or MSTR has not been successfully executed.
	IERR = 2	bus not found.
	IERR = 3	branch not found.
	IERR = 4	branch is out-of-service.
	IERR = 5	error encountered in executing SPCBAPI.
	IERR = 6	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

4.13.13 DIST_SPCB_FAULT_2

Use this API routine to run the second release of the dynamics branch unbalance function. Use this API routine to calculate a branch unbalance and apply the equivalent positive sequence pi model at a branch during dynamic simulations.

Batch command syntax:

```
BAT_DIST_SPCB_FAULT_2  IBUS  JBUS  'ID'  OPTIONS(1)...OPTIONS(7)
VALUES(1)...VALUES(5)
```

Python syntax:

```
ierr = dist_spcb_fault_2(ibus, jbus, id, options, values)
```

Fortran syntax:

```
CALL DIST_SPCB_FAULT_2(IBUS, JBUS, ID, OPTIONS, VALUES, IERR)
```

where:

Integer IBUS	Is the from bus of branch to be subjected to unbalance (input; no default allowed).
Integer JBUS	Is the to bus of branch to be subjected to unbalance (input; no default allowed).
Character ID*2	Is the circuit identifier of branch to be subjected to unbalance (input; '1' by default).
Integer OPTIONS(7)	Is an array of seven elements specifying unbalance options (input). The values are as follows:

OPTIONS(1)	unbalance code (1 by default).
OPTIONS(1) = 1	one phase open.
OPTIONS(1) = 2	two phases open.
OPTIONS(1) = 3	in-line fault.
OPTIONS(1) = 4	one breaker open.
OPTIONS(2)	path to ground code (used when OPTIONS(1) is 1 or 4) (0 by default).
OPTIONS(2) = 0	no path to ground.
OPTIONS(2) = 1	include a path to ground.
OPTIONS(3)	type of in-line fault code (used when OPTIONS(1) is 3) (1 by default).
OPTIONS(3) = 1	line-to-ground fault.
OPTIONS(3) = 2	line-to-line-to-ground fault.
OPTIONS(3) = 3	three phase fault.
OPTIONS(4)	open end code (used when OPTIONS(1) is 4) (1 by default).
OPTIONS(4) = 0	breaker at bus JBUS is open.
OPTIONS(4) = 1	breaker at bus IBUS is open.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	OPTIONS(5)	dc line and FACTS device option (0 by default).
	OPTIONS(5) = 0	block and ignore.
	OPTIONS(5) = 1	convert to constant admittance load.
	OPTIONS(6)	transformer impedance correction option (1 by default).
	OPTIONS(6) = 0	do not apply to zero sequence.
	OPTIONS(6) = 1	apply to zero sequence.
	OPTIONS(7)	fault analysis generator reactance option (0 by default).
	OPTIONS(7) = 0	use subtransient reactance.
	OPTIONS(7) = 1	use transient reactance.
	OPTIONS(7) = 2	use synchronous reactance.
Real VALUES(5)	Is an array of five elements specifying unbalance parameters (input).	
	VALUES(1)	fault location as fraction of line from bus IBUS; 0.0 < VALUES(1) < 1.0 (used when OPTIONS(1) is 1 or 4 and OPTIONS(2) is 1; or when OPTIONS(1) is 3) (0.5 by default).
	VALUES(2) & (3)	Rl-g and Xl-g (used when OPTIONS(1) is 1 or 4 and OPTIONS(2) is 1; or when OPTIONS(1) is 3 and OPTIONS(3) is 1 or 2) (0.0 by default).
	VALUES(4) & (5)	Rl-l and Xl-l (used when OPTIONS(1) is 3 and OPTIONS(3) is 2) (0.0 by default).
Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR = 1	STRT or MSTR has not been successfully executed.
	IERR = 2	bus not found.
	IERR = 3	branch not found.
	IERR = 4	branch is out-of-service.
	IERR = 5	error encountered in executing SPCBAPI.
	IERR = 6	prerequisite requirements for API are not met.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

This page intentionally left blank.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

SIEMENS

Chapter 5

Subsystem Definition

This chapter defines API routines that control subsystems. There are four kinds of subsystems in PSS®E: area subsystems (in Section 5.1 Area Subsystem Definition APIs), bus subsystems (in Section 5.2 Bus Subsystem Definition APIs), owner subsystems (in Section 5.3 Owner Subsystem Definition APIs), and zone subsystems (in Section 5.4 Zone Subsystem Definition APIs). Area subsystems are a set of areas, bus subsystems are a set of buses, etc.

Many PSS®E API routines accept a subsystem definition to limit the scope of operation of the routine. Most such use bus subsystems. Bus subsystems can be defined in terms of areas, zones, and/or owners; but a bus subsystem defined by areas, for example, is not an area subsystem, which is a set of areas, not a set of buses in an area.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

5.1 Area Subsystem Definition APIs

5.1.1 ASYS

Use this API to define an area subsystem.

Batch command syntax:

```
BAT_ASYS  SID  NUM  AREAS(1) ... AREAS(NUM)
```

Python syntax:

```
ierr = asys(sid, num, areas)
```

Fortran syntax:

```
CALL SBSASET(SID, NUM, AREAS, IERR)
```

where:

Integer SID	Is a valid area subsystem ID. Valid area subsystem IDs range from 0 to 11.
Integer NUM	Is the number of areas to set.
Integer AREAS(NUM)	Is an array of NUM elements that contains the areas to set.
Integer IERR	Is the error return. IERR = 0 no error occurred. IERR = 1 SID is not valid.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

5.1.2 ASYSDEF

Use this API to set the definition of an area subsystem. This can be used to toggle a particular area subsystem on or off.

Batch command syntax:

```
BAT_ASYSDEF  SID  DEFINED
```

Python syntax:

```
asysdef(sid, defined)
```

Fortran syntax:

```
CALL SBSASETDEF(SID, DEFINED)
```

where:

Integer SID	Is a valid area subsystem ID. Valid area subsystem IDs range from 0 to 11.	
Integer DEFINED	= 0	to mark the area subsystem as undefined.
	= 1	to mark the area subsystem as defined.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

5.1.3 ASYSINIT

Use this API to initialize or re-initialize an area subsystem.

Batch command syntax:

```
BAT_ASYSINIT  SID
```

Python syntax:

```
asysinit(sid)
```

Fortran syntax:

```
CALL SBSACREATE(SID, IERR)
```

where:

Integer SID	Is a valid area subsystem ID. Valid area subsystem IDs range from 0 to 11.
Integer IERR	Is the error return. IERR = 0 no error occurred. IERR = 1 SID is not valid.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

5.2 Bus Subsystem Definition APIs

5.2.1 BSYS

Use this API to define a bus subsystem.

Batch command syntax:

```
BAT_BSYS  SID  USEKV BASEKV(1)  BASEKV(2) NUMAREA
AREAS(1)...AREAS(NUMAREA)  NUMBUS  BUSNUM(1)...BUSNUM(NUMBUS)
NUMOWNER  OWNERS(1)...OWNERS(NUMOWNER)  NUMZONE
ZONES(1)...ZONES(NUMZONE)
```

or:

```
BAT_BSYS  SID  USEKV BASEKV(1)  BASEKV(2) NUMAREA
AREAS(1)...AREAS(NUMAREA)  NUMBUS  BUSNAME(1)...BUSNAME(NUMBUS)
NUMOWNER  OWNERS(1)...OWNERS(NUMOWNER)  NUMZONE
ZONES(1)...ZONES(NUMZONE)
```

Python syntax:

```
ierr = bsys(sid, usekv, basekv, numarea, areas, numbus, buses,
numowner, owners, numzone, zones)
```

Fortran syntax:

```
CALL SBSBSET(SID, USEKV, BASEKV, NUMAREA, AREAS, NUMBUS, BUSNUM,
NUMOWNER, OWNERS, NUMZONE, ZONES, IERR)
```

or:

```
CALL SBSBSETA(SID, USEKV, BASEKV, NUMAREA, AREAS, NUMBUS, BUSNAM,
NUMOWNER, OWNERS, NUMZONE, ZONES, IERR)
```

where:

Integer SID	Is a valid bus subsystem ID. Valid bus subsystem IDs range from 0 to 11.
Integer USEKV	USEKV = 0 do not use BASEKV values. USEKV = 1 use BASEKV values.
Real BASEKV(2)	Is an array of two elements. BASEKV(1) is the minimum basekV limit. BASEKV(2) is the maximum basekV limit.
Integer NUMAREA	Is the number of areas to set.
Integer AREAS(NUMAREA)	Is an array of NUMAREA elements that contains the areas to set.
Integer NUMBUS	Is the number of buses to set.
Integer BUSNUM(NUMBUS)	Is an array of NUMBUS elements that contains the buses to set.
Character BUSNAM(NUMBUS)*18	Is an array of NUMBUS elements that contains the extended bus names to set.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer BUSES(NUMBUS) or Character BUSNAM(NUMBUS)*18	Can be either BUSNUM or BUSNAM as defined above.
Integer BUSLEN	Is the length of a single element of the character array BUSNAM e.g., BUSNAM(1).
Integer NUMOWNER	Is the number of owners to set.
Integer OWNERS(NUMOWNER)	Is an array of NUMOWNER elements that contains the owners to set.
Integer NUMZONE	Is the number of zones to set.
Integer ZONES(NUMZONE)	Is an array of NUMZONE elements that contains the zones to set.
Integer IERR	Is the error return. IERR = 0 no error occurred. IERR = 1 SID is not valid.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

5.2.2 BSYSADD

Use this API to add elements to an existing bus subsystem.

Batch command syntax:

```
BAT_BSYSADD  SID  USEKV  BASEKV(1)  BASEKV(2)  NUMAREA
AREAS(1)...AREAS(NUMAREA)  NUMBUS  BUSES(1)...BUSES(NUMBUS)  NUMOWNER
OWNERS(1)...OWNERS(NUMOWNER)  NUMZONE  ZONES(1)...ZONES(NUMZONE)
```

Python syntax:

```
ierr = bsysadd(sid, usekv, basekv, numarea, areas, numbus, buses,
numowner, owners, numzone, zones)
```

Fortran syntax:

```
CALL SBSBADD(SID, USEKV, BASEKV, NUMAREA, AREAS, NUMBUS, BUSES,
NUMOWNER, OWNERS, NUMZONE, ZONES, IERR)
```

where:

Integer SID	Is a valid bus subsystem ID. Valid bus subsystem IDs range from 0 to 11.
Integer USEKV	USEKV = 0 do not use BASEKV values. USEKV = 1 use BASEKV values.
Real BASEKV(2)	Is an array of two elements. BASEKV(1) is the minimum basekV limit. BASEKV(2) is the maximum basekV limit.
Integer NUMAREA	Is the number of areas to set.
Integer AREAS(NUMAREA)	Is an array of NUMAREA elements that contains the areas to set.
Integer NUMBUS	Is the number of buses to set.
Integer BUSES(NUMBUS)	Is an array of NUMBUS elements that contains the buses to set.
Integer NUMOWNER	Is the number of owners to set.
Integer OWNERS(NUMOWNER)	Is an array of NUMOWNER elements that contains the owners to set.
Integer NUMZONE	Is the number of zones to set.
Integer ZONES(NUMZONE)	Is an array of NUMZONE elements that contains the zones to set.
Integer IERR	Is the error return. IERR = 0 no error occurred. IERR = 1 SID is not valid.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

5.2.3 BSYSDEF

Use this API to set the definition of a bus subsystem. This can be used to toggle a particular bus subsystem on or off.

Batch command syntax:

```
BAT_BSYSDEF  SID  DEFINED
```

Python syntax:

```
bsysdef(sid, defined)
```

Fortran syntax:

```
CALL SBSBSETDEF(SID, DEFINED)
```

where:

Integer SID	Is a valid bus subsystem ID. Valid bus subsystem IDs range from 0 to 11.	
Integer DEFINED	= 0	to mark the bus subsystem as undefined.
	= 1	to mark the bus subsystem as defined.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

5.2.4 BSYSINIT

Use this API to initialize or re-initialize a bus subsystem.

Batch command syntax:

```
BAT_BSYSINIT  SID
```

Python syntax:

```
bsysinit(sid)
```

Fortran syntax:

```
CALL SBSBCREATE(SID, IERR)
```

where:

Integer SID	Is a valid bus subsystem ID. Valid bus subsystem IDs range from 0 to 11.
Integer IERR	Is the error return. IERR = 0 no error occurred. IERR = 1 SID is not valid.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

5.2.5 BSYSMEM

Use this API to replicate the specified bus subsystem in a bus subsystem file. This bus subsystem file may then be recalled at a later time to restore the specified bus subsystem.

Batch command syntax:

```
BAT_BSYSMEM  SID  SFILE
```

Python syntax:

```
ierr = bsysmem(sid, sfile)
```

Fortran syntax:

```
CALL SBSBSMEM(SID, SFILE, IERR)
```

where:

Integer SID	Is a valid bus subsystem id. Valid bus subsystem ids range from 0 to 11.
Character SFILE*260	Is the bus subsystem filename.
Integer IERR	Is the error return. IERR = 0 no error occurred. IERR = 1 SID is not valid. IERR = 2 error opening SFILE.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

5.2.6 BSYSO

Use this API to build a bus subsystem one bus at a time.

Batch command syntax:

```
BAT_BSYSO  SID BUSNUM
```

Python syntax:

```
ierr = bsyso(sid, busnum)
```

Fortran syntax:

```
CALL SBSBSETO(SID, BUSNUM, IERR)
```

where:

Integer SID	Is a valid bus subsystem ID. Valid bus subsystem IDs range from 0 to 11.	
Integer BUSNUM	Is the bus number to add to subsystem.	
Integer IERR	Is the error return.	
	IERR = 0	no error occurred.
	IERR = 1	SID is not valid.
	IERR = 2	Bus subsystem contains other elements, not just buses.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

5.2.7 BSYSRCL

Use this API to recall a specified bus subsystem saved in a bus subsystem file.

Batch command syntax:

```
BAT_BSYSRCL  SID  SFILE
```

Python syntax:

```
ierr = bsysrcl(sid, sfile)
```

Fortran syntax:

```
CALL SBSBSRCL(SID, SFILE, IERR)
```

where:

Integer SID	Is a valid bus subsystem id. Valid bus subsystem ids range from 0 to 11.
Character SFILE*260	Is the bus subsystem filename.
Integer IERR	Is the error return. IERR = 0 no error occurred. IERR = 1 SID is not valid. IERR = 2 error opening SFILE. IERR = 3 not a valid bus subsystem file. IERR = 4 items in bus subsystem file not found in working case.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

5.3 Owner Subsystem Definition APIs

5.3.1 OSYS

Use this API to define an owner subsystem. The API is invoked by the use of the statement:

Batch command syntax:

```
BAT_OSYS  SID  NUM  OWNERS(1) ... OWNERS(NUM)
```

Python syntax:

```
ierr = osys(sid, num, owners)
```

Fortran syntax:

```
CALL SBSOSET(SID, NUM, OWNERS, IERR)
```

where:

Integer SID	Is a valid owner subsystem ID. Valid owner subsystem IDs range from 0 to 11.
Integer NUM	Is the number of owners to set.
Integer OWNERS(NUM)	Is an array of NUM elements that contains the owners to set.
Integer IERR	Is the error return. IERR = 0 no error occurred. IERR = 1 SID is not valid.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

5.3.2 OSYSDEF

Use this API to set the definition of an owner subsystem. This can be used to toggle a particular owner subsystem on or off.

Batch command syntax:

```
BAT_OSYSDEF  SID  DEFINED
```

Python syntax:

```
osysdef(sid, defined)
```

Fortran syntax:

```
CALL SBSOSETDEF(SID, DEFINED)
```

where:

Integer SID	Is a valid owner subsystem ID. Valid owner subsystem IDs range from 0 to 11.	
Integer DEFINED	= 0	to mark the owner subsystem as undefined.
	= 1	to mark the owner subsystem as defined.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

5.3.3 OSYSINIT

Use this API to initialize or re-initialize an owner subsystem.

Batch command syntax:

```
BAT_OSYSINIT  SID
```

Python syntax:

```
osysinit(sid)
```

Fortran syntax:

```
CALL SBSOCREATE(SID, IERR)
```

where:

Integer SID	Is a valid owner subsystem ID. Valid owner subsystem IDs range from 0 to 11.
Integer IERR	Is the error return. IERR = 0 no error occurred. IERR = 1 SID is not valid.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

5.4 Zone Subsystem Definition APIs

5.4.1 ZSYS

Use this API to define a zone subsystem.

The API is invoked by the use of the statement:

Batch command syntax:

```
BAT_ZSYS  SID  NUM  ZONES(1) ... ZONES(NUM)
```

Python syntax:

```
ierr = zsys(sid, num, zones)
```

Fortran syntax:

```
CALL SBSZSET(SID, NUM, ZONES, IERR)
```

where:

Integer SID	Is a valid zone subsystem ID. Valid zone subsystem IDs range from 0 to 11.
Integer NUM	Is the number of zones to set.
Integer ZONES(NUM)	Is an array of NUM elements that contains the zones to set.
Integer IERR	Is the error return. IERR = 0 no error occurred. IERR = 1 SID is not valid.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

5.4.2 ZSYSDEF

Use this API to set the definition of an zone subsystem. This can be used to toggle a particular zone subsystem on or off.

Batch command syntax:

```
BAT_ZSYSDEF  SID  DEFINED
```

Python syntax:

```
zsysdef(sid, defined)
```

Fortran syntax:

```
CALL SBSZSETDEF(SID, DEFINED)
```

where:

Integer SID	Is a valid zone subsystem ID. Valid zone subsystem IDs range from 0 to 11.	
Integer DEFINED	= 0	to mark the zone subsystem as undefined.
	= 1	to mark the zone subsystem as defined.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

5.4.3 ZSYSINIT

Use this API to initialize or re-initialize a zone subsystem.

Batch command syntax:

```
BAT_ZSYSINIT  SID
```

Python syntax:

```
zsysinit(sid)
```

Fortran syntax:

```
CALL SBSZCREATE(SID, IERR)
```

where:

Integer SID	Is a valid zone subsystem ID. Valid zone subsystem IDs range from 0 to 11.
Integer IERR	Is the error return. IERR = 0 no error occurred. IERR = 1 SID is not valid.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

SIEMENS

Chapter 6

Diagram View APIs

This chapter defines API routines that are used to manipulate Slider Diagrams in the interface. They will elicit no action if a GUI is not present when invoked.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.1 CHANGEDIAGCONTOUR

Use this API to change the contour parameters for the contour in the active Slider Diagram in the GUI. Requires a current network case.

Batch command syntax:

```
BAT_CHANGEDIAGCONTOUR ELEMENT QUANTITY METHOD COLOR ENABLE RES
INFLUENCE BOUND MAX MIN
```

Python syntax:

```
changediagcontour(element, quantity, method, color, enable, res,
influence, bound, max, min)
```

where:

Integer ELEMENT	USE SAME ENTIRE TABLE FROM CHANGEDIAGCONTOUR
Integer QUANTITY	Is the bus number of the bus to GROW (input, no default).
Integer METHOD	Is the bus number of the bus to GROW (input, no default).
Integer COLOR	Is the bus number of the bus to GROW (input, no default).
Integer ENABLE	Is the bus number of the bus to GROW (input, no default).
Real RES	X coordinate of the bus on the Slider Diagram.
Real INFLUENCE	Y coordinate of the bus on the Slider Diagram.
Real BOUND	X coordinate of the bus on the Slider Diagram.
Real MAX	X coordinate of the bus on the Slider Diagram.
Real MIN	X coordinate of the bus on the Slider Diagram.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.2 CHECKDIAGFILE

Use this API to check the active Slider Diagram in the GUI against the current network case.

Batch command syntax:

```
BAT_CHECKDIAGFILE
```

Python syntax:

```
checkdiagfile()
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.3 CLOSEDIAGFILE

Use this API to close the active Slider Diagram in the GUI without prompting to save the Diagram.

Batch command syntax:

```
BAT_CLOSEDIAGFILE
```

Python syntax:

```
closediagfile()
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.4 DELETEDIAGCONTOUR

Use this API to delete the contour from the active Slider Diagram in the GUI.

Batch command syntax:

```
BAT_DELETEDIAGCONTOUR
```

Python syntax:

```
deletediagcontour()
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.5 ENABLEDIAGCONTOUR

Use this API to specify the contour parameters and display a contour in the active Slider Diagram in the GUI. Requires a current network case.

Batch command syntax:

```
BAT_ENABLEDIAGCONTOUR ELEMENT QUANTITY METHOD COLOR ENABLE RES
INFLUENCE BOUND MAX MIN
```

Python syntax:

```
enablediagcontour(element, quantity, method, color, enable, res,
influence, bound, max, min)
```

where:

Integer ELEMENT	USE SAME ENTIRE TABLE FROM CHANGEDIAGCONTOUR
Integer QUANTITY	Is the bus number of the bus to GROW (input, no default).
Integer METHOD	Is the bus number of the bus to GROW (input, no default).
Integer COLOR	Is the bus number of the bus to GROW (input, no default).
Integer ENABLE	Is the bus number of the bus to GROW (input, no default).
Real RES	X coordinate of the bus on the Slider Diagram.
Real INFLUENCE	Y coordinate of the bus on the Slider Diagram.
Real BOUND	X coordinate of the bus on the Slider Diagram.
Real MAX	X coordinate of the bus on the Slider Diagram.
Real MIN	X coordinate of the bus on the Slider Diagram.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.6 EXPORTBUSLOCFILE

Use this API to export the bus location from the active Slider Diagram in the GUI to a file.

Batch command syntax:

```
BAT_EXPORTBUSLOCFILE BUSLOCFILE
```

Python syntax:

```
exportbuslocfile(buslocfile)
```

where:

Character BUSLOCFILE*260	Is the name of the Bus Location Data File to be written (input).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.7 EXPORTIMAGEFILE

Use this API to export an image of the active Slider Diagram in the GUI to a file.

Batch command syntax:

```
BAT_EXPORTIMAGEFILE TYPE IMAGEFILE QUALITY
```

Python syntax:

```
exportimagefile(type, imagefile, quality)
```

where:

Integer TYPE	Is the type of image to export (input) TYPE = 1 Bitmap TYPE = 2 JPEG TYPE = 3 PNG TYPE = 4 GIF TYPE = 5 TIF
Character IMAGEFILE*260	Is the name of the image file to be written (input).
Integer QUALITY	Is a value, between 1 and 100, used to control the quality of the exported image if TYPE = 2 (input)

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.8 GEXMBUS

Use this API to create a GEXM/GOUT display of a bus in an active Slider Diagram in the GUI. Requires a current network case.

Batch command syntax:

```
BAT_GEXMBUS BUS
```

Python syntax:

```
gexmbus(bus)
```

where:

Integer BUS	Is the bus number of the bus for which to create the GEXM display (input).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.9 GROWBUS

Use this API to GROW a bus in an active Slider Diagram in the GUI. Requires a current network case.

Batch command syntax:

```
BAT_GROWBUS BUSNUMBER XPOSITION YPOSITION
```

Python syntax:

```
growbus(busnumber, xposition, yposition)
```

where:

Integer BUSNUMBER	Is the bus number of the bus to GROW (input, no default).
Real XPOSITION	X coordinate of the bus on the Slider Diagram (input).
Real YPOSITION	Y coordinate of the bus on the Slider Diagram (input).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.10 GROWDIAGRAM

Use this API to GROW the specified components and all the connected buses up to the specified number of levels away in an active Slider Diagram in the GUI. Requires a current network case.

Batch command syntax:

```
BAT_ GROWDIAGRAM NUMLEVELS ITEMCOUNT ITEMLIST XPOSITION YPOSITION
```

Python syntax:

```
growdiagram(numlevels, itemcount, itemlist, xposition, yposition)
```

where:

Integer NUMLEVELS	Is the number of levels; include all buses up to this number of connections from each item in ITEMLIST in the GROW operation. A value of -1 can be used to grow all buses (subsystem and in-service options still apply), however attempting to lay out several hundred buses or more will take a very large amount of time to complete as PSSE attempts to calculate best positions for each bus. It is still recommended that the user only grow several levels at once (input, no default).
Integer ITEMCOUNT	Number of items in ITEMLIST. ITEMCOUNT can be 0 if NUMLEVELS is -1. Under this condition diagram growth will start at an arbitrary bus (intput, 0 by default).
Character ITEMLIST(ITEMCOUNT)*40	The list of items to grow. Each item is given by its generated map string (intput).
Real XPOSITION	Ignored (input).
Real YPOSITION	Ignored (input).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.11 GROWBUSLEVELS

Use this API to GROW a bus and all the connected buses up to the specified number of levels away in an active Slider Diagram in the GUI. Requires a current network case.

Batch command syntax:

```
BAT_GROWBUSLEVELS BUSNUMBER XPOSITION YPOSITION NUMLEVELS
```

Python syntax:

```
growbuslevels(busnumber, xposition, yposition, numlevels)
```

where:

Integer BUSNUMBER	Is the bus number of the bus to GROW (input, no default).
Real XPOSITION	X coordinate of the bus on the Slider Diagram (input).
Real YPOSITION	Y coordinate of the bus on the Slider Diagram (input).
Integer NUMLEVELS	Is the number of levels; include all buses up to this number of connections from BUSNUMBER in the GROW operation (input, no default).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.12 GRPG

Use this API to produce a GRPG report in the active Slider Diagram in the GUI.

Batch command syntax:

```
BAT_GRPG IMAGEFILE, DUMFILE
```

Python syntax:

```
grpg(imagefile, dumfile)
```

where:

Character IMAGEFILE*260	Is the name of the GRPG file to be used (input).
Character DUMFILE*260	Is a dummy filename; this should be initialized to " " (input)

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.13 IMPORTDRAWFILE

Use this API to import a Draw Data File into an active Slider Diagram in the GUI.

Batch command syntax:

```
BAT_IMPORTDRAWFILE DRAWFILE
```

Python syntax:

```
importdrawfile(drawfile)
```

where:

Character DRAWFILE*260	Is the name of an existing Draw Data File, *.drw (input).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.14 IMPORTIMAGEFILE

Use this API to import an image into the background layer of the active Slider Diagram in the GUI.

Batch command syntax:

```
BAT_IMPORTIMAGEFILE IMAGEFILE
```

Python syntax:

```
importimagefile(imagefile)
```

where:

Character IMAGEFILE*260	Is the name of the image file to be imported (input).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.15 NEWDIAGFILE

Use this API to open a new Slider Diagram in the GUI.

Batch command syntax:

```
BAT_NEWDIAGFILE
```

Python syntax:

```
newdiagfile()
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.16 OPENACCFILE

This API is used to open an existing ACC results file for displaying results on Diagrams.

Batch command syntax:

```
BAT_OPENACCFILE BASERATE CNTGRATE PERRATE ACCFILE
```

Python syntax:

```
ierr = openaccfile(baserate, cntgrate, perrate, accfile)
```

Fortran syntax

```
CALL OPENACCFILE(BASERATE, CNTGRATE, PERRATE, ACCFILE)
```

where:

Integer BASERATE	Is the base case rating to use for results display(input) BASERATE = 1 RATEA BASERATE = 2 RATEB BASERATE = 3 RATEC
Integer CNTGRATE	Is the contingency case rating to use for results display (input) CNTGRATE = 1 RATEA CNTGRATE = 2 RATEB CNTGRATE = 3 RATEC
Real PERRATE	Is the loading percentage to use for results display (input).
Character ACCFILE*260	Is the ACC file to open (input).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.17 OPENBUSLOCFILE

Use this API to open an existing bus location file in the GUI. The bus location file can be used to control the placement of buses on a diagram during auto-draw and grow operations

Batch command syntax:

```
BAT_OPENBUSLOCFILE BUSLOCFILE
```

Python syntax:

```
openbuslocfile(buslocfile)
```

where:

Character BUSLOCFILE*260	Is the name of an existing Bus Location Data File (input).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.18 OPENDIAGFILE

Use this API to open an existing Slider Diagram in the GUI.

Batch command syntax:

```
BAT_OPENDIAGFILE SFILE
```

Python syntax:

```
ierr = opendiagfile(sfile)
```

where:

Character SFILE*260	Is the name of an existing Slider Diagram file (input).
Integer IERR	Is the error code (output).
	IERR = 0 — No error occured.
	IERR = 1 — Error opening the diagram file.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.19 PRINTDIAGFILE

Use this API to print the active Slider Diagram in the GUI.

Batch command syntax:

```
BAT_PRINTDIAGFILE SDEV COPIES ORIENT
```

Python syntax:

```
printdiagfile(sdev, copies, orient)
```

where:

Character SDEV*260	Is the name of a valid attached Windows printer (input).
Integer COPIES	Is the number of copies to print (input).
Integer ORIENT	Output orientation (input). = 1 PORTRAIT = 2 LANDSCAPE

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.20 REFRESHDIAGCONTOUR

Use this API to refresh the contour in the active Slider Diagram in the GUI. Requires a current network case. This API would be used in an automation file to force a contour refresh that wouldn't normally take place until the automation file completed.

Batch command syntax:

```
BAT_REFRESHDIAGCONTOUR
```

Python syntax:

```
refreshdiagcontour()
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.21 REFRESHDIAGFILE

Use this API to refresh the results and styles in the active Slider Diagram in the GUI. It requires a current network case. This API would be used in an automation file to force a results refresh that wouldn't normally take place until the automation file completed.

Batch command syntax:

```
BAT_REFRESHDIAGFILE
```

Python syntax:

```
refreshdiagfile()
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.22 RENUMDIAGFILE

Use this API to renumber buses in the active Slider Diagram in the GUI.

Batch command syntax:

```
BAT_RENUMDIAGFILE TRNFILE
```

Python syntax:

```
renumdiagfile(trnfile)
```

where:

Character TRNFILE*260	Is the name of the Bus Number Translation File (input).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.23 SAVEDIAGFILE

Use this API to save the active Slider Diagram in the GUI.

Batch command syntax:

```
BAT_SAVEDIAGFILE SFILE
```

Python syntax:

```
ierr = savediagfile(sfile)
```

where:

Character SFILE*260	Is the name of the file to save the Slider Diagram in (input).	
Integer IERR	Is the error code (output).	
	IERR = 0	No error occured.
	IERR = 1	Error saving the diagram file.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.24 SETDIAGAUTOFILE

Use this API to specify an automation file that is run each time results are updated on the active Slider Diagram.

Batch command syntax:

```
BAT_SETDIAGAUTOFILE AUTONAME
```

Python syntax:

```
setdiagautofile(autoname)
```

where:

Character AUTONAME*260	Is the name of the automation file (input).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.25 SETDIAGFONT

Use this API to specify the default text font in the active Slider Diagram.

Batch command syntax:

```
BAT_SETDIAGFONT FNTSIZ FNTWGT FNTNAM
```

Python syntax:

```
setdiagfont(fntsiz, fntwgt, fntnam)
```

where:

Integer FNTSIZ	Is the font point size (input).
Integer FNTWGT	Is the font weight (input).
Character FNTNAM*260	Is the name of the font family (input).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.26 SETDIAGPREC

Use this API to specify the precision of results displayed in Slider Diagrams.

Batch command syntax:

```
BAT_SETDIAGPREC DIAGPREC
```

Python syntax:

```
setdiagprec(diagprec)
```

where:

Integer DIAGPREC(8)	Is an array of eight elements specifying the number of digits after the decimal point for various result types. Currently, only DIAGPREC(1) is used for power flow results; valid values are 1 through 4 (input).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.27 SETDIAGPROP

Use this API to specify basic properties in the active Slider Diagram.

Batch command syntax:

```
BAT_SETDIAGPROP UNBNDCLR(1)…UNBNDCLR(3) BNDCLR(1)…BNDCLR(3)
BKGCLR(1)…BKGCLR(3) PRTCLR(1)…PRTCLR(3) GRDCLR(1)…GRDCLR(3)
UPARWCLR(1)…UPARWCLR(3) DWNARWCLR(1)…DWNARWCLR(3) LABCLR(1)…LABCLR(3)
GRDSTYL GRDPTRN FLTVAL(1) FLTVAL(2) DBLVAL(1)…DBLVAL(10)
LOGVAL(1)…LOGVAL(6) BARCHRT LABLHIDE PRNTMODE IMGDIR
```

Python syntax:

```
setdiagprop(unbndclr, bndclr, bkgclr, prtclr, grdclr, uparwclr,
dwnarwclr, labclr, grdstyl, grdptrn, fltval, dblval, logval, barchrt,
lablhide, prntmode, imgdir)
```

where:

Integer UNBNDCLR(3)	Is an array of three elements specifying the unbound item color (input). The values are as follows:	
	UNBNDCLR(1) = 0->255	Red component
	UNBNDCLR(2) = 0->255	Blue component
	UNBNDCLR(3) = 0->255	Green component
Integer BNDCLR(3)	Is an array of three elements specifying the bound item color (input). The values are as follows:	
	BNDCLR(1) = 0->255	Red component
	BNDCLR(2) = 0->255	Blue component
	BNDCLR(3) = 0->255	Green component
Integer BKGCLR(3)	Is an array of three elements specifying the background color (input). The values are as follows:	
	BNDCLR(1) = 0->255	Red component
	BNDCLR(2) = 0->255	Blue component
	BNDCLR(3) = 0->255	Green component
Integer PRTCLR(3)	Is an array of three elements specifying the item connection port color (input). The values are as follows:	
	PRTCLR(1) = 0->255	Red component
	PRTCLR(2) = 0->255	Blue component
	PRTCLR(3) = 0->255	Green component
Integer GRDCLR(3)	Is an array of three elements specifying the on-diagram layout grid color (input). The values are as follows:	
	GRDCLR(1) = 0->255	Red component
	GRDCLR(2) = 0->255	Blue component
	GRDCLR(3) = 0->255	Green component

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer UPARWCLR(3)	Is an array of three elements specifying the active power animation arrow color (input). The values are as follows: UPARWCLR(1) = 0->255 Red component UPARWCLR(2) = 0->255 Blue component UPARWCLR(3) = 0->255 Green component
Integer DWNARWCLR(3)	Is an array of three elements specifying the reactive power animation arrow color (input). The values are as follows: DWNARWCLR(1) = 0->255 Red component DWNARWCLR(2) = 0->255 Blue component DWNARWCLR(3) = 0->255 Green component
Integer LABCLR(3)	Is an array of three elements specifying the label color (input). The values are as follows: LABCLR(1) = 0->255 Red component LABCLR(2) = 0->255 Blue component LABCLR(3) = 0->255 Green component
Integer GRDSTYL	Grid style (input). The values are as follows: GRDSTYLE = 0 Dotted GRDSTYLE = 1 Lined
Integer GRDPTRN	Grid pattern (input).
Real FLTVAL(2)	Is an array of two elements (input). The values are as follows: FLTVAL(1) Grid point size FLTVAL(2) > = 1 Default line width in pixels.
Real DBLVAL(10)	Is an array of ten elements (input). The values are as follows: DBLVAL(1) Maximum zoom factor x 100, i.e. 10=1000% zoom DBLVAL(2) Minimum zoom factor DBLVAL(3) Zoom increment DBLVAL(4) Zoom at which to hide selection handles DBLVAL(5) Zoom at which to hide text DBLVAL(6) Zoom at which to hide grid DBLVAL(7) Multi-page print scaling DBLVAL(8) Pan distance DBLVAL(9) Grid spacing DBLVAL(10) Grid snap distance
Integer LOGVAL(6)	Is an array of six elements (input). Each element has only two possible values. 0 = FALSE and 1 = TRUE. The elements are as follows: LOGVAL(1) No upside down text LOGVAL(2) High quality printing LOGVAL(3) Display tooltips LOGVAL(4) Stipple grid LOGVAL(5) Hide unbound network items LOGVAL(6) Display filled busbars

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer BARCHRT	Two-winding transformer bar chart display side (input). The values are as follows: BARCHRT = 0 High current loading side BARCHRT = 1 Low current loading side BARCHRT = 2 Both sides
Integer LABLHIDE	Label hide mode (input). The values are as follows: LABLHIDE = 0 Show all labels and results LABLHIDE = 1 Hide results LABLHIDE = 2 Hide all labels and results
Integer PRNTMODE	Printing mode (input). The values are as follows: PRNTHIDE = 0 As seen on the screen, WYSIWYG PRNTHIDE = 1 Fit the whole diagram on the page PRNTHIDE = 2 Multi-page
Character IMGDIR*260	Is the name of the directory where image files for this Diagram are stored (input).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.28 SETDIAGPROP_2

Use this API to specify basic properties in the active Slider Diagram.

Batch command syntax:

```
BAT_SETDIAGPROP_2 BKGCLR(1)…BKGCLR(3) PRTCLR(1)…PRTCLR(3)
GRDCLR(1)…GRDCLR(3) UPARWCLR(1)…UPARWCLR(3) DWNARWCLR(1)…DWNARWCLR(3)
LABCLR(1)…LABCLR(3) GRDSTYL GRDPTRN FLTVAL(1) FLTVAL(2)
DBLVAL(1)…DBLVAL(10) LOGVAL(1)…LOGVAL(6) LABLHIDE PRNTMODE IMGDIR
```

Python syntax:

```
setdiagprop_2(bkgclr, prtclr, grdclr, uparwclr, dwnarwclr, labclr,
grdstyl, grdptrn, fltval, dblval, logval, lablhide, prntmode, imgdir)
```

where:

Integer BKGCLR(3)	Is an array of three elements specifying the background color (input). The values are as follows: BKGCLR(1) = 0->255 Red component BKGCLR(2) = 0->255 Blue component BKGCLR(3) = 0->255 Green component
Integer PRTCLR(3)	Is an array of three elements specifying the item connection port color (input). The values are as follows: PRTCLR(1) = 0->255 Red component PRTCLR(2) = 0->255 Blue component PRTCLR(3) = 0->255 Green component
Integer GRDCLR(3)	Is an array of three elements specifying the on-diagram layout grid color (input). The values are as follows: GRDCLR(1) = 0->255 Red component GRDCLR(2) = 0->255 Blue component GRDCLR(3) = 0->255 Green component
Integer UPARWCLR(3)	Is an array of three elements specifying the active power animation arrow color (input). The values are as follows: UPARWCLR(1) = 0->255 Red component UPARWCLR(2) = 0->255 Blue component UPARWCLR(3) = 0->255 Green component
Integer DWNARWCLR(3)	Is an array of three elements specifying the reactive power anima-tion arrow color (input). The values are as follows: DWNARWCLR(1) = 0->255 Red component DWNARWCLR(2) = 0->255 Blue component DWNARWCLR(3) = 0->255 Green component
Integer LABCLR(3)	Is an array of three elements specifying the label color (input). The values are as follows: LABCLR(1) = 0->255 Red component LABCLR(2) = 0->255 Blue component LABCLR(3) = 0->255 Green component

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer GRDSTYL	Grid style (input). The values are as follows: GRDSTYLE = 0 Dotted GRDSTYLE = 1 Lined
Integer GRDPTRN	Grid pattern (input).
Real FLTVAL(2)	Is an array of two elements (input). The values are as follows: FLTVAL(1) Grid point size FLTVAL(2) > = 1 Default line width in pixels.
Real DBLVAL(10)	Is an array of ten elements (input). The values are as follows: DBLVAL(1) Maximum zoom factor x 100, i.e. 10=1000% zoom DBLVAL(2) Minimum zoom factor DBLVAL(3) Zoom increment DBLVAL(4) Zoom at which to hide selection handles DBLVAL(5) Zoom at which to hide text DBLVAL(6) Zoom at which to hide grid DBLVAL(7) Multi-page print scaling DBLVAL(8) Pan distance DBLVAL(9) Grid spacing DBLVAL(10) Grid snap distance
Integer LOGVAL(6)	Is an array of six elements (input). Each element has only two possible values. 0 = FALSE and 1 = TRUE. The elements are as follows: LOGVAL(1) No upside down text LOGVAL(2) High quality printing LOGVAL(3) Display tooltips LOGVAL(4) Stipple grid LOGVAL(5) Hide unbound network items LOGVAL(6) Display filled busbars
Integer LABLHIDE	Label hide mode (input). The values are as follows: LABLHIDE = 0 Show all labels and results LABLHIDE = 1 Hide results LABLHIDE = 2 Hide all labels and results
Integer PRNTMODE	Printing mode (input). The values are as follows: PRNTHIDE = 0 As seen on the screen, WYSIWYG PRNTHIDE = 1 Fit the whole diagram on the page PRNTHIDE = 2 Multi-page
Character IMGDIR*260	Is the name of the directory where image files for this Diagram are stored (input).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.29 SETDIAGPROP_3

Use this API to specify basic properties in the active Slider Diagram.

Batch command syntax:

```
BAT_SETDIAGPROP_2 BKGCLR(1)…BKGCLR(3) PRTCLR(1)…PRTCLR(3)
GRDCLR(1)…GRDCLR(3) UPARWCLR(1)…UPARWCLR(3) DWNARWCLR(1)…DWNARWCLR(3)
LABCLR(1)…LABCLR(3) GRDSTYL GRDPTRN FLTVAL(1) FLTVAL(2)
DBLVAL(1)…DBLVAL(10) LOGVAL(1)…LOGVAL(6) LABLHIDE PRNTMODE IMGDIR
```

Python syntax:

```
setdiagprop_2(bkgclr, prtclr, grdclr, uparwclr, dwnarwclr, labclr,
grdstyl, grdptrn, fltval, dblval, logval, lablhide, prntmode, imgdir)
```

where:

Integer BKGCLR(3)	Is an array of three elements specifying the background color (input). The values are as follows: BKGCLR(1) = 0->255 Red component BKGCLR(2) = 0->255 Blue component BKGCLR(3) = 0->255 Green component
Integer PRTCLR(3)	Is an array of three elements specifying the item connection port color (input). The values are as follows: PRTCLR(1) = 0->255 Red component PRTCLR(2) = 0->255 Blue component PRTCLR(3) = 0->255 Green component
Integer GRDCLR(3)	Is an array of three elements specifying the on-diagram layout grid color (input). The values are as follows: GRDCLR(1) = 0->255 Red component GRDCLR(2) = 0->255 Blue component GRDCLR(3) = 0->255 Green component
Integer UPARWCLR(3)	Is an array of three elements specifying the active power animation arrow color (input). The values are as follows: UPARWCLR(1) = 0->255 Red component UPARWCLR(2) = 0->255 Blue component UPARWCLR(3) = 0->255 Green component
Integer DWNARWCLR(3)	Is an array of three elements specifying the reactive power anima-tion arrow color (input). The values are as follows: DWNARWCLR(1) = 0->255 Red component DWNARWCLR(2) = 0->255 Blue component DWNARWCLR(3) = 0->255 Green component
Integer LABCLR(3)	Is an array of three elements specifying the label color (input). The values are as follows: LABCLR(1) = 0->255 Red component LABCLR(2) = 0->255 Blue component LABCLR(3) = 0->255 Green component

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer GRDSTYL	Grid style (input). The values are as follows: GRDSTYLE = 0 Dotted GRDSTYLE = 1 Lined
Integer GRDPTRN	Grid pattern (input).
Real FLTVAL(2)	Is an array of two elements (input). The values are as follows: FLTVAL(1) Grid point size FLTVAL(2) > = 1 Default line width in pixels.
Real DBLVAL(10)	Is an array of ten elements (input). The values are as follows: DBLVAL(1) Maximum zoom factor x 100, i.e. 10=1000% zoom DBLVAL(2) Minimum zoom factor DBLVAL(3) Zoom increment DBLVAL(4) Zoom at which to hide selection handles DBLVAL(5) Zoom at which to hide text DBLVAL(6) Zoom at which to hide grid DBLVAL(7) Multi-page print scaling DBLVAL(8) Pan distance DBLVAL(9) Grid spacing DBLVAL(10) Grid snap distance
Integer LOGVAL(7)	LOGVAL(7) Make branch connections perpendicular to branches.
Integer LABLHIDE	Label hide mode (input). The values are as follows: LABLHIDE = 0 Show all labels and results LABLHIDE = 1 Hide results LABLHIDE = 2 Hide all labels and results
Integer PRNTMODE	Printing mode (input). The values are as follows: PRNTHIDE = 0 As seen on the screen, WYSIWYG PRNTHIDE = 1 Fit the whole diagram on the page PRNTHIDE = 2 Multi-page
Character IMGDIR*260	Is the name of the directory where image files for this Diagram are stored (input).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.30 SETDIAGRESACCC

This API is used to set the ACCC analysis annotation options in the active Slider Diagram.

This API is obsolete and has been replaced with SETDIAGRESACCC_2

Batch command syntax:

```
BAT_SETDIAGRESACCC  RATE CNTGOPT  CONTOUR  CNTROPT  LOADPER FILE
```

Python syntax:

```
setdiagresaccc(rate, cntgopt, contour, cntropt, loadper, file)
```

where:

Integer RATE	Branch rating set (input) RATE = 1 Rate A RATE = 2 Rate B RATE = 3 Rate C
Integer CNTGOPT	Contingencies to view (input). CNTGOPT = 1 All contingencies CNTGOPT = 2 Contingencies causing overload violations CNTGOPT = 3 Contingencies causing any violations
Integer CONTOUR	Contour selection (input). CONTOUR = 0 Display data values as text CONTOUR = 1 Display data values as a contour
Integer CNTROPT	Contour value selection (input). The values are as follows. CNTROPT = 0 Contour bus voltages CNTROPT = 1 Contour % branch flows
Real LOADPER	Loading percentage (input).
Character FILE*260	Is the ACCC Contingency Solution data file (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.31 SETDIAGRESACCC_2

This API is used to set the ACCC analysis annotation options in the active Slider Diagram.

Batch command syntax:

```
BAT_SETDIAGRESACCC_2 CNTGOPT  SELCNTG, CONTOUR  CNTROPT
```

Python syntax:

```
setdiagresaccc_2(cntgopt, selecntg, contour, cntropt)
```

where:

Integer CNTGOPT	Contingencies to view (input). CNTGOPT = 1 All contingencies CNTGOPT = 2 Contingencies causing overload violations CNTGOPT = 3 Contingencies causing any violations
Integer SELCNTG	Is the selected contingency (input) SELCNTG = 0 None selected SELCNTG ≠0 Selected contingency from list determined by CNTGOPT
Integer CONTOUR	Contour selection (input). CONTOUR = 0 Display data values as text CONTOUR = 1 Display data values as a contour
Integer CNTROPT	Contour value selection (input). The values are as follows. CNTROPT = 0 Contour bus voltages CNTROPT = 1 Contour % branch flows
Real LOADPER	Loading percentage (input).
Character FILE*260	Is the ACCC Contingency Solution data file (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.32 SETDIAGRESASCC

This API is used to set the ASCC short circuit analysis annotation options in the active Slider Diagram.

Batch command syntax:

```
BAT_SETDIAGRESASCC ANNOT, TYPE, BUS, BOE, CTRIB, FRM, TO, LAST, ID,
STYLE, WIDTH, COLOR(1) .. COLOR(3)
nTnum, nLnum, sId, nStyle, nWidth, nClr
```

Python syntax:

```
setdiagresascc (annot, type, bus, boe, ctrib, frm, to, last, id,
style, width, color)
```

where:

Integer ANNOT	Specifies the fault to display:	
	ANNOT = 0	Fault currents (only bus faults)
	ANNOT = 1	Three phase
	ANNOT = 2	Line to Ground (LG)
	ANNOT = 3	Line Line to Ground (LLG)
	ANNOT = 4	Line to Line (LL)
Integer TYPE	Specifies the fault display quantity:	
	TYPE = 1	Ia1
	TYPE = 2	Ia2
	TYPE = 3	Ia0
	TYPE = 4	IA
	TYPE = 5	IB
	TYPE = 6	IC
	TYPE = 7	3Ia0
Integer BUS	Specifies the faulted bus:	
	BUS = 0	No bus faulted
	BUS > 0	Bus to be faulted
Integer BOE	Specifies the fault location:	
	BOE = 1	Fault located at the faulted bus
	BOE = 2	Fault located at the faulted bus with lines outaged
	BOE = 3	Fault located at the end of lines
Integer CTRIB	Specifies contributions at home bus for bus faults:	
	CTRIB = 0	Contributions for selected fault
	CTRIB = 1	Contributions at home bus for all faulted buses for selected fault
Integer FRM	Specifies the from bus if a line fault is applied:	
	FRM = 0	No line fault applied
	FRM > 0	From bus

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer TO	Specifies the to bus if a line fault is applied: TO = 0 No line fault applied TO > 0 To bus
Integer LAST	Specifies the third bus if a line fault is applied to a 3-winding transformer: TO = 0 No line fault applied TO > 0 To bus
Character ID*2	Is the branch/transformer identifier.
Integer STYLE	Is the line style applied to the faulted bus diagram item.
Integer WIDTH	Is the line width applied to the faulted bus diagram item, ≥ 1.
Integer COLOR(3)	Is an array of three elements specifying the color for diagram items that are faulted. The values are as follows: COLOR(1) = 0->255Red component COLOR(2) = 0->255Blue component COLOR(3) = 0->255Green component

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.33 SETDIAGRESDATA

Use this API to specify the impedance data voltage and range checking annotation options in the active Slider Diagram.

Batch command syntax:

```
BAT_SETDIAGRESDATA USEVLT VLNWDT(1)…VLNWDT(8) VLNSTY(1)…VLNSTY(8)
VLTVAL(1)…VLTVAL(7) VLNCLRR(1)…VLNCLRR(8) VLNCLRG(1)…VLNCLRG(8)
VLNCLRB(1)…VLNCLRB(8) OUTWDT OUTSTY OUTCLR(1)…OUTCLR(3) BUSANN,
BRNANN, EQPANN, BUSVLT
```

Python syntax:

```
setdiagresdata(usevlt, vlnwdt, vlnsty, vltval, vlnclrr, vlnclrg,
vlnclrb, outwdt, outsty, outclr, busann, brnann, eqpann, busvlt)
```

where:

Integer USEVLT	Is the option to apply voltage level annotation to the Diagram elements (input). The values are as follows: USEVLT = 0 Don't apply voltage level annotation USEVLT = 1 Apply voltage level annotation
Integer VLNWDT(8)	Is an array of 8 elements used to set the width of Diagram elements that fall in the appropriate voltage range (input). VLNWDT(1) base voltage < VLTVAL(1) VLNWDT(2) VLTVAL(1) >= base voltage < VLTVAL(2) VLNWDT(3) VLTVAL(2) >= base voltage < VLTVAL(3) VLNWDT(4) VLTVAL(3) >= base voltage < VLTVAL(4) VLNWDT(5) VLTVAL(4) >= base voltage < VLTVAL(5) VLNWDT(6) VLTVAL(5) >= base voltage < VLTVAL(6) VLNWDT(7) VLTVAL(6) >= base voltage < VLTVAL(7) VLNWDT(8) base voltage >= VLTVAL(7)
Integer VLNSTY(8)	Is an array of 8 elements used to set the line style of Diagram elements that fall in the appropriate voltage range (input). VLNSTY(1) base voltage < VLTVAL(1) VLNSTY(2) VLTVAL(1) >= base voltage < VLTVAL(2) VLNSTY(3) VLTVAL(2) >= base voltage < VLTVAL(3) VLNSTY(4) VLTVAL(3) >= base voltage < VLTVAL(4) VLNSTY(5) VLTVAL(4) >= base voltage < VLTVAL(5) VLNSTY(6) VLTVAL(5) >= base voltage < VLTVAL(6) VLNSTY(7) VLTVAL(6) >= base voltage < VLTVAL(7) VLNSTY(8) base voltage >= VLTVAL(7)
Real VLTVAL(7)	Is an array of 7 elements used to set the voltage ranges used to assign annotation Diagram. Base voltage values should be in ascending order (input).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer VLNCLRR(8) Is an array of 8 elements used to set the Red color component of Diagram elements that fall in the appropriate voltage range, 0->255 (input).

VLNCLRR(1)	base voltage < VLTVAL(1)
VLNCLRR(2)	VLTVAL(1) >= base voltage < VLTVAL(2)
VLNCLRR(3)	VLTVAL(2) >= base voltage < VLTVAL(3)
VLNCLRR(4)	VLTVAL(3) >= base voltage < VLTVAL(4)
VLNCLRR(5)	VLTVAL(4) >= base voltage < VLTVAL(5)
VLNCLRR(6)	VLTVAL(5) >= base voltage < VLTVAL(6)
VLNCLRR(7)	VLTVAL(6) >= base voltage < VLTVAL(7)
VLNCLRR(8)	base voltage >= VLTVAL(7)

Integer VLNCLRG(8) Is an array of 8 elements used to set Green color component of Diagram elements that fall in the appropriate voltage range, 0->255 (input).

VLNCLRG(1)	base voltage < VLTVAL(1)
VLNCLRG(2)	VLTVAL(1) >= base voltage < VLTVAL(2)
VLNCLRG(3)	VLTVAL(2) >= base voltage < VLTVAL(3)
VLNCLRG(4)	VLTVAL(3) >= base voltage < VLTVAL(4)
VLNCLRG(5)	VLTVAL(4) >= base voltage < VLTVAL(5)
VLNCLRG(6)	VLTVAL(5) >= base voltage < VLTVAL(6)
VLNCLRG(7)	VLTVAL(6) >= base voltage < VLTVAL(7)
VLNCLRG(8)	base voltage >= VLTVAL(7)

Integer VLNCLRB(8) Is an array of 8 elements used to set Blue color component of Diagram elements that fall in the appropriate voltage range, 0->255 (input).

VLNCLRB(1)	base voltage < VLTVAL(1)
VLNCLRB(2)	VLTVAL(1) >= base voltage < VLTVAL(2)
VLNCLRB(3)	VLTVAL(2) >= base voltage < VLTVAL(3)
VLNCLRB(4)	VLTVAL(3) >= base voltage < VLTVAL(4)
VLNCLRB(5)	VLTVAL(4) >= base voltage < VLTVAL(5)
VLNCLRB(6)	VLTVAL(5) >= base voltage < VLTVAL(6)
VLNCLRB(7)	VLTVAL(6) >= base voltage < VLTVAL(7)
VLNCLRB(8)	base voltage >= VLTVAL(7)

Integer OUTWDT Is the line width applied to out of service Diagram items, >= 1 (input).

Integer OUTSTY Is the line style applied to out of service Diagram items (input).

Integer OUTCLR(3) Is an array of three elements specifying the color for Diagram items that are out of service (input). The values are as follows:

OUTCLR(1) = 0->255	Red component
OUTCLR(2) = 0->255	Blue component
OUTCLR(3) = 0->255	Green component

Integer BUSANN Bus ID annotation (input).

BUSANN = 0	Number, no name
BUSANN = 1	Name, no number
BUSANN = 2	Number and name
BUSANN = 3	No number, no name

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer BRNANN	Branch annotation (input).	
	BRNANN = 0	MW/MVAR flows
	BRNANN = 1	MVA
	BRNANN = 2	MVA/% rating
	BRNANN = 3	Amps/pu current
	BRNANN = 4	MW/%rating
	BRNANN = 5	Amps/% rating
	BRNANN = 6	None
Integer EQPANN	Equipment annotation (input).	
	EQPANN = 0	MW/MVAR
	EQPANN = 2	None
Integer BUSVLT	Bus voltage annotation (input).	
	BUSVLT = 0	magnitude (pu)
	BUSVLT = 1	magnitude (kV)
	BUSVLT = 2	angle (degrees)
	BUSVLT = 3	magnitude (pu) and angle (degrees)
	BUSVLT = 4	magnitude (pu) and magnitude (kV)
	BUSVLT = 5	magnitude (kV) and angle (degrees)
	BUSVLT = 7	magnitude (pu), magnitude (kV) and angle (degrees)
	BUSVLT = 8	none

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.34 SETDIAGRESDATA_2

Use this API to specify the basic Power Flow annotation options in the active Slider Diagram.

Batch command syntax:

```
BAT_SETDIAGRESDATA_2 BRNANN BUSANN
```

Python syntax:

```
setdiagresdata_2(brnann, busann)
```

where:

Integer BRNANN	Branch annotation (input). BRNANN = 6 display impedances BRNANN = 7 display ratings BRNANN = 8 display impedances & ratings BRNANN = 9 display nothing
Integer BUSANN	Bus voltage annotation (input). BUSANN = 0 don't display base kV BUSANN = 1 display base kV

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.35 SETDIAGRESGDIF

Use this API to specify the case comparison voltage and range checking annotation options in the active Slider Diagram.

Batch command syntax:

```
BAT_SETDIAGRESGDIF USEVLT VLNWDT(1)…VLNWDT(8) VLNSTY(1)…VLNSTY(8)
VLTVAL(1)…VLTVAL(7) VLNCLRR(1)…VLNCLRR(8) VLNCLRG(1)…VLNCLRG(8)
VLNCLRB(1)…VLNCLRB(8) OUTWDT OUTSTY OUTCLR(1)…OUTCLR(3) BUSANN
```

Python syntax:

```
setdiagresgdif(usevlt, vlnwdt, vlnsty, vltval, vlnclrr, vlnclrg,
vlnclrb, outwdt, outsty, outclr, busann)
```

where:

Integer USEVLT	Is the option to apply voltage level annotation to the Diagram elements (input). The values are as follows: USEVLT = 0 Don't apply voltage level annotation USEVLT = 1 Apply voltage level annotation
Integer VLNWDT(8)	Is an array of 8 elements used to set the width of Diagram elements that fall in the appropriate voltage range (input). VLNWDT(1) base voltage < VLTVAL(1) VLNWDT(2) VLTVAL(1) >= base voltage < VLTVAL(2) VLNWDT(3) VLTVAL(2) >= base voltage < VLTVAL(3) VLNWDT(4) VLTVAL(3) >= base voltage < VLTVAL(4) VLNWDT(5) VLTVAL(4) >= base voltage < VLTVAL(5) VLNWDT(6) VLTVAL(5) >= base voltage < VLTVAL(6) VLNWDT(7) VLTVAL(6) >= base voltage < VLTVAL(7) VLNWDT(8) base voltage >= VLTVAL(7)
Integer VLNSTY(8)	Is an array of 8 elements used to set the line style of Diagram elements that fall in the appropriate voltage range (input). VLNSTY(1) base voltage < VLTVAL(1) VLNSTY(2) VLTVAL(1) >= base voltage < VLTVAL(2) VLNSTY(3) VLTVAL(2) >= base voltage < VLTVAL(3) VLNSTY(4) VLTVAL(3) >= base voltage < VLTVAL(4) VLNSTY(5) VLTVAL(4) >= base voltage < VLTVAL(5) VLNSTY(6) VLTVAL(5) >= base voltage < VLTVAL(6) VLNSTY(7) VLTVAL(6) >= base voltage < VLTVAL(7) VLNSTY(8) base voltage >= VLTVAL(7)
Real VLTVAL(7)	Is an array of 7 elements used to set the voltage ranges used to assign annotation Diagram. Base voltage values should be in ascending order (input).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer VLNCLRR(8) Is an array of 8 elements used to set the Red color component of Diagram elements that fall in the appropriate voltage range, 0->255 (input).

VLNCLRR(1)	base voltage < VLTVAL(1)
VLNCLRR(2)	VLTVAL(1) >= base voltage < VLTVAL(2)
VLNCLRR(3)	VLTVAL(2) >= base voltage < VLTVAL(3)
VLNCLRR(4)	VLTVAL(3) >= base voltage < VLTVAL(4)
VLNCLRR(5)	VLTVAL(4) >= base voltage < VLTVAL(5)
VLNCLRR(6)	VLTVAL(5) >= base voltage < VLTVAL(6)
VLNCLRR(7)	VLTVAL(6) >= base voltage < VLTVAL(7)
VLNCLRR(8)	base voltage >= VLTVAL(7)

Integer VLNCLRG(8) Is an array of 8 elements used to set Green color component of Diagram elements that fall in the appropriate voltage range, 0->255 (input).

VLNCLRG(1)	base voltage < VLTVAL(1)
VLNCLRG(2)	VLTVAL(1) >= base voltage < VLTVAL(2)
VLNCLRG(3)	VLTVAL(2) >= base voltage < VLTVAL(3)
VLNCLRG(4)	VLTVAL(3) >= base voltage < VLTVAL(4)
VLNCLRG(5)	VLTVAL(4) >= base voltage < VLTVAL(5)
VLNCLRG(6)	VLTVAL(5) >= base voltage < VLTVAL(6)
VLNCLRG(7)	VLTVAL(6) >= base voltage < VLTVAL(7)
VLNCLRG(8)	base voltage >= VLTVAL(7)

Integer VLNCLRB(8) Is an array of 8 elements used to set Blue color component of Diagram elements that fall in the appropriate voltage range, 0->255 (input).

VLNCLRB(1)	base voltage < VLTVAL(1)
VLNCLRB(2)	VLTVAL(1) >= base voltage < VLTVAL(2)
VLNCLRB(3)	VLTVAL(2) >= base voltage < VLTVAL(3)
VLNCLRB(4)	VLTVAL(3) >= base voltage < VLTVAL(4)
VLNCLRB(5)	VLTVAL(4) >= base voltage < VLTVAL(5)
VLNCLRB(6)	VLTVAL(5) >= base voltage < VLTVAL(6)
VLNCLRB(7)	VLTVAL(6) >= base voltage < VLTVAL(7)
VLNCLRB(8)	base voltage >= VLTVAL(7)

Integer OUTWDT Is the line width applied to out of service Diagram items, >= 1 (input).

Integer OUTSTY Is the line style applied to out of service Diagram items (input).

Integer OUTCLR(3) Is an array of three elements specifying the color for Diagram items that are out of service (input). The values are as follows:

OUTCLR(1) = 0->255	Red component
OUTCLR(2) = 0->255	Blue component
OUTCLR(3) = 0->255	Green component

Integer BUSANN Bus ID annotation (input).

BUSANN = 0	Number, no name
BUSANN = 1	Name, no number
BUSANN = 2	Number and name
BUSANN = 3	No number, no name

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.36 SETDIAGRESDYN

This API is used to set the dynamic simulation annotation options in the active Slider Diagram.

Batch command syntax:

```
BAT_SETDIAGRESDYN MODE, CONTOUR
```

Python syntax:

```
setdiagresdyn (mode, contour)
```

where:

Integer MODE	Specifies the simulation result/value to display:	
	MODE = 0	Voltage violations as specified by SET_VOLTAGE_REC_CHECK and SET_VOLTAGE_DIP_CHECK
	MODE = 1	Contour simulation quantities
Integer CONTOUR	Specifies the dynamic quantity to contour:	
	CONTOUR = 0	Bus voltage -kV

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.37 SETDIAGRESIEC

Use this API to specify the IEC annotation options in the active Slider Diagram.

Batch command syntax:

```
BAT_SETDIAGRESIEC ANNOT, TYPE, BUS, BOE, CTRIB, FRM, TO, LAST, ID,
STYLE, WIDTH, COLOR(1) .. COLOR(3)
nTnum, nLnum, sId, nStyle, nWidth, nClr
```

Python syntax:

```
setdiagresiec (annot, type, bus, boe, ctrib, frm, to, last, id,
style, width, color)
```

where:

IntegerANNOT	Specifies the fault to display:	
	ANNOT = 0	Fault currents (only bus faults)
	ANNOT = 1	Three phase
	ANNOT = 2	Line to Ground (LG)
	ANNOT = 3	Line Line to Ground (LLG)
	ANNOT = 4	Line to Line (LL)
Integer TYPE	Specifies the fault display quantity:	
	TYPE = 1	Ia1
	TYPE = 2	Ia2
	TYPE = 3	Ia0
	TYPE = 4	IA
	TYPE = 5	IB
	TYPE = 6	IC
	TYPE = 7	3Ia0
Integer BUS	Specifies the faulted bus:	
	BUS = 0	No bus faulted
	BUS > 0	Bus to be faulted
Integer BOE	Specifies the fault location:	
	BOE = 1	Fault located at the faulted bus
	BOE = 2	Fault located at the faulted bus with lines outaged
	BOE = 3	Fault located at the end of lines
Integer CTRIB	Specifies contributions at home bus for bus faults:	
	CTRIB = 0	Contributions for selected fault
	CTRIB = 1	Contributions at home bus for all faulted buses for selected fault
Integer FRM	Specifies the from bus if a line fault is applied:	
	FRM = 0	No line fault applied
	FRM > 0	From bus

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer TO	Specifies the to bus if a line fault is applied: TO = 0 No line fault applied TO > 0 To bus
Integer LAST	Specifies the third bus if a line fault is applied to a 3-winding transformer: TO = 0 No line fault applied TO > 0 To bus
Character ID*2	Is the branch/transformer identifier.
Integer STYLE	Is the line style applied to the faulted bus Diagram item.
Integer WIDTH	Is the line width applied to the faulted bus Diagram item, ≥ 1.
Integer COLOR(3)	Is an array of three elements specifying the color for Diagram items that are faulted. The values are as follows: COLOR(1) = 0->255Red component COLOR(2) = 0->255Blue component COLOR(3) = 0->255Green component

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.38 SETDIAGRESPFLOWCHECK

Use this API to specify the Power Flow voltage and range checking annotation options in the active Slider Diagram. This API is obsolete and has been replaced with SETDIAGRESPFLOWCHECK_2.

Batch command syntax:

```
BAT_SETDIAGRESPFLOWCHECK LOGARR(1)…LOGARR(4) VLNWDT(1)…VLNWDT(8)
VLNSTY(1)…VLNSTY(8) VLTVAL(1)…VLTVAL(7) VLNCLRR(1)…VLNCLRR(8)
VLNCLRG(1)…VLNCLRG(8) VLNCLRB(1)…VLNCLRB(8) LINLOD LLODARR(1)
LLODARR(2) LLODCLR(1)…LLODCLR(3) BUSVLT(1) BUSVLT(2) BUSVLTWDT(1)
BUSVLTWDT(2) BUSCLRR(1) BUSCLRR(2) BUSCLRG(1) BUSCLRG(2) BUSCLRB(1)
BUSCLRB(2) OUTARR(1) OUTARR(2) OUTCLR(1)…OUTCLR(3) BRNRAT BARPERCENT
BARCLRR(1) BARCLRR(2) BARCLRR(3) BARCLRR(4) BARCLRG(1) BARCLRG(2)
BARCLRG(3) BARCLRG(4) BARCLRB(1) BARCLRB(2) BARCLRB(3) BARCLRB(4)
```

Python syntax:

```
setdiagrespflowcheck(logarr, vlnwdt, vlnsty, vltval, vlnclrr,
vlnclrg, vlnclrb, linlod, llodarr, llodclr, busvlt, busvltwdt,
busclrr, busclrg, busclrb, outarr, outclr, brnrat, barpercent,
barclrr, barclrg, barclrb)
```

where:

Integer LOGARR(4) Is an array of 4 elements used to determine what annotation is applied (input). Each element has only two possible values. 0 = FALSE and 1 = TRUE. The values are as follows:

LOGARR(1)	Use Voltage level thresholds
LOGARR(2)	Use Line ratings
LOGARR(3)	Use Bus voltage
LOGARR(4)	Use out of service

Integer VLNWDT(8) Is an array of 8 elements used to set the width of Diagram elements that fall in the appropriate voltage range (input).

VLNWDT(1)	base voltage < VLTVAL(1)
VLNWDT(2)	VLTVAL(1) >= base voltage < VLTVAL(2)
VLNWDT(3)	VLTVAL(2) >= base voltage < VLTVAL(3)
VLNWDT(4)	VLTVAL(3) >= base voltage < VLTVAL(4)
VLNWDT(5)	VLTVAL(4) >= base voltage < VLTVAL(5)
VLNWDT(6)	VLTVAL(5) >= base voltage < VLTVAL(6)
VLNWDT(7)	VLTVAL(6) >= base voltage < VLTVAL(7)
VLNWDT(8)	base voltage >= VLTVAL(7)

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer VLNSTY(8) Is an array of 8 elements used to set the line style of Diagram elements that fall in the appropriate voltage range (input).

VLNSTY(1)	base voltage < VLTVAL(1)
VLNSTY(2)	VLTVAL(1) >= base voltage < VLTVAL(2)
VLNSTY(3)	VLTVAL(2) >= base voltage < VLTVAL(3)
VLNSTY(4)	VLTVAL(3) >= base voltage < VLTVAL(4)
VLNSTY(5)	VLTVAL(4) >= base voltage < VLTVAL(5)
VLNSTY(6)	VLTVAL(5) >= base voltage < VLTVAL(6)
VLNSTY(7)	VLTVAL(6) >= base voltage < VLTVAL(7)
VLNSTY(8)	base voltage >= VLTVAL(7)

Real VLTVAL(7) Is an array of 7 elements used to set the voltage ranges used to assign annotation Diagram. Base voltage values should be in ascending order (input).

Integer VLNCLRR(8) Is an array of 8 elements used to set the Red color component of Diagram elements that fall in the appropriate voltage range, 0->255 (input).

VLNCLRR(1)	base voltage < VLTVAL(1)
VLNCLRR(2)	VLTVAL(1) >= base voltage < VLTVAL(2)
VLNCLRR(3)	VLTVAL(2) >= base voltage < VLTVAL(3)
VLNCLRR(4)	VLTVAL(3) >= base voltage < VLTVAL(4)
VLNCLRR(5)	VLTVAL(4) >= base voltage < VLTVAL(5)
VLNCLRR(6)	VLTVAL(5) >= base voltage < VLTVAL(6)
VLNCLRR(7)	VLTVAL(6) >= base voltage < VLTVAL(7)
VLNCLRR(8)	base voltage >= VLTVAL(7)

Integer VLNCLRG(8) Is an array of 8 elements used to set Green color component of Diagram elements that fall in the appropriate voltage range, 0->255 (input).

VLNCLRG(1)	base voltage < VLTVAL(1)
VLNCLRG(2)	VLTVAL(1) >= base voltage < VLTVAL(2)
VLNCLRG(3)	VLTVAL(2) >= base voltage < VLTVAL(3)
VLNCLRG(4)	VLTVAL(3) >= base voltage < VLTVAL(4)
VLNCLRG(5)	VLTVAL(4) >= base voltage < VLTVAL(5)
VLNCLRG(6)	VLTVAL(5) >= base voltage < VLTVAL(6)
VLNCLRG(7)	VLTVAL(6) >= base voltage < VLTVAL(7)
VLNCLRG(8)	base voltage >= VLTVAL(7)

Integer VLNCLRB(8) Is an array of 8 elements used to set Blue color component of Diagram elements that fall in the appropriate voltage range, 0->255 (input).

VLNCLRB(1)	base voltage < VLTVAL(1)
VLNCLRB(2)	VLTVAL(1) >= base voltage < VLTVAL(2)
VLNCLRB(3)	VLTVAL(2) >= base voltage < VLTVAL(3)
VLNCLRB(4)	VLTVAL(3) >= base voltage < VLTVAL(4)
VLNCLRB(5)	VLTVAL(4) >= base voltage < VLTVAL(5)
VLNCLRB(6)	VLTVAL(5) >= base voltage < VLTVAL(6)
VLNCLRB(7)	VLTVAL(6) >= base voltage < VLTVAL(7)
VLNCLRB(8)	base voltage >= VLTVAL(7)

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real LINLOD	% loading for line rating (input).
Integer LLODARR(2)	Is an array of two elements used set the width and style of Diagram items that exceed the % of line loading (input). LLODARR(1) Line width >= 1 LLODARR(2) Line style
Integer LLODCLR(3)	Is an array of three elements specifying the color for lines that exceed the % loading rating, LINLOD (input). The values are as follows: LLODCLR(1) = 0->255 Red component LLODCLR(2) = 0->255 Blue component LLODCLR(3) = 0->255 Green component
Real BUSVLT(2)	Is an array of two elements used in the bus voltage check (input). BUSVLT(1) Min. bus voltage BUSVLT(2) Max. bus voltage
Integer BUSVLTWDT(2)	Is an array of two elements used in the bus voltage check to set the width of Diagram items at the min and max (input). BUSVLTWDT(1) Min. bus voltage width >= 1 BUSVLTWDT(2) Max. bus voltage width >= 1
Integer BUSCLRR(2)	Is an array of two elements used to set the Red color component of Diagram elements that fall in the appropriate bus voltage range, 0->255 (input). BUSCLRR(1) base voltage < BUSVLTWDT(1) BUSCLRR(2) BUSVLTWDT(1) >= base voltage < BUSVLTWDT (2)
Integer BUSCLRG(2)	Is an array of two elements used to set Green color component of Diagram elements that fall in the appropriate bus voltage range, 0->255 (input). BUSCLRG(1) base voltage < BUSVLTWDT (1) BUSCLRG(2) BUSVLTWDT (1) >= base voltage < BUSVLTWDT (2)
Integer BUSCLRB(2)	Is an array of two elements used to set Blue color component of Diagram elements that fall in the appropriate bus voltage range, 0->255 (input). BUSCLRB(1) base voltage < BUSVLTWDT (1) BUSCLRB(2) BUSVLTWDT(1) >= base voltage < BUSVLTWDT (2)
Integer OUTARR(2)	Is an array of two elements used to set the width and style of Diagram items that are out of service (input). OUTARR(1) Line width >= 1 OUTARR(2) Line style
Integer OUTCLR(3)	Is an array of three elements specifying the color for Diagram items that are out of service (input). The values are as follows: OUTCLR(1) = 0->255 Red component OUTCLR(2) = 0->255 Blue component OUTCLR(3) = 0->255 Green component

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer BRNRAT	Branch rating set (input). BRNRAT = 0 Rate A BRNRAT = 1 Rate B BRNRAT = 2 Rate C
Integer BARPERCENT(4)	Is an array of four elements specifying the percent overloads at which to apply different color schemes for the line loading charts on the Diagram. The values are as follows: BARPERCENT(1) First overload voltage point BARPERCENT(2) Second overload voltage point BARPERCENT(3) Third overload voltage point BARPERCENT(4) Fourth overload voltage point
Integer BARCLRR(4)	Is an array of four elements used to set the Red color component for the line loading charts on the Diagram, 0->255 (input). BARCLRR(1) First overload voltage point BARCLRR(2) Second overload voltage point BARCLRR(3) Third overload voltage point BARCLRR(4) Fourth overload voltage point
Integer BARCLRG(4)	Is an array of four elements used to set Green color component for the line loading charts on the Diagram, 0->255 (input). BARCLRG(1) First overload voltage point BARCLRG(2) Second overload voltage point BARCLRG(3) Third overload voltage point BARCLRG(4) Fourth overload voltage point
Integer BARCLRB(4)	Is an array of four elements used to set Blue color component for the line loading charts on the Diagram, 0->255 (input). BARCLRB(1) First overload voltage point BARCLRB(2) Second overload voltage point BARCLRB(3) Third overload voltage point BARCLRB(4) Fourth overload voltage point

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.39 SETDIAGRESPFLOWCHECK_2

Use this API to specify the Power Flow voltage and range checking annotation options in the active Slider Diagram.

Batch command syntax:

```
BAT_SETDIAGRESPFLOWCHECK_2 LOGARR(1) LOGARR(2) LINLOD LLODARR(1)
LLODARR(2) LLODCLR(1)…LLODCLR(3) BUSVLT(1) BUSVLT(2) BUSVLTWDT(1)
BUSVLTWDT(2) BUSCLRR(1) BUSCLRR(2) BUSCLRG(1) BUSCLRG(2) BUSCLRB(1)
BUSCLRB(2) BRNRAT BARPERCENT BARCLRR(1) BARCLRR(2) BARCLRR(3)
BARCLRR(4) BARCLRG(1) BARCLRG(2) BARCLRG(3) BARCLRG(4) BARCLRB(1)
BARCLRB(2) BARCLRB(3) BARCLRB(4)
```

Python syntax:

```
setdiagrespflowcheck_2(logarr, linlod, llodarr, llodclr, busvlt,
busvltwdt, busclrr, busclrg, busclrb, outarr, outclr, brnrat,
barpercent, barclrr, barclrg, barclrb)
```

where:

Integer LOGARR(2)	Is an array of 2 elements used to determine what annotation is applied (input). Each element has only two possible values. 0 = FALSE and 1 = TRUE. The values are as follows: LOGARR(1) Use Line ratings LOGARR(2) Use Bus voltage
Real LINLOD	% loading for line rating (input).
Integer LLODARR(2)	Is an array of two elements used set the width and style of Diagram items that exceed the % of line loading (input). LLODARR(1) Line width >= 1 LLODARR(2) Line style
Integer LLODCLR(3)	Is an array of three elements specifying the color for lines that exceed the % loading rating, LINLOD (input). The values are as follows: LLODCLR(1) = 0->255 Red component LLODCLR(2) = 0->255 Blue component LLODCLR(3) = 0->255 Green component
Real BUSVLT(2)	Is an array of two elements used in the bus voltage check (input). BUSVLT(1) Min. bus voltage BUSVLT(2) Max. bus voltage
Integer BUSVLTWDT(2)	Is an array of two elements used in the bus voltage check to set the width of Diagram items at the min and max (input). BUSVLTWDT(1) Min. bus voltage width >= 1 BUSVLTWDT(2) Max. bus voltage width >= 1

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer BUSCLRR(2) Is an array of two elements used to set the Red color component of Diagram elements that fall in the appropriate bus voltage range, 0->255 (input).

BUSCLRR(1)	base voltage < BUSVLTWDT(1)
BUSCLRR(2)	BUSVLTWDT(1) >= base voltage < BUSVLTWDT (2)

Integer BUSCLRG(2) Is an array of two elements used to set Green color component of Diagram elements that fall in the appropriate bus voltage range, 0->255 (input).

BUSCLRG(1)	base voltage < BUSVLTWDT (1)
BUSCLRG(2)	BUSVLTWDT (1) >= base voltage < BUSVLTWDT (2)

Integer BUSCLRB(2) Is an array of two elements used to set Blue color component of Diagram elements that fall in the appropriate bus voltage range, 0->255 (input).

BUSCLRB(1)	base voltage < BUSVLTWDT (1)
BUSCLRB(2)	BUSVLTWDT(1) >= base voltage < BUSVLTWDT (2)

Integer OUTARR(2) Is an array of two elements used to set the width and style of Diagram items that are out of service (input).

OUTARR(1)	Line width >= 1
OUTARR(2)	Line style

Integer OUTCLR(3) Is an array of three elements specifying the color for Diagram items that are out of service (input). The values are as follows:

OUTCLR(1) = 0->255	Red component
OUTCLR(2) = 0->255	Blue component
OUTCLR(3) = 0->255	Green component

Integer BRNRAT Branch rating set (input).

BRNRAT = 0	Rate A
BRNRAT = 1	Rate B
BRNRAT = 2	Rate C

Integer BARPERCENT(4) Is an array of four elements specifying the percent overloads at which to apply different color schemes for the line loading charts on the Diagram. The values are as follows:

BARPERCENT(1)	First overload voltage point
BARPERCENT(2)	Second overload voltage point
BARPERCENT(3)	Third overload voltage point
BARPERCENT(4)	Fourth overload voltage point

Integer BARCLRR(4) Is an array of four elements used to set the Red color component for the line loading charts on the Diagram, 0->255 (input).

BARCLRR(1) First overload voltage point

BARCLRR(2) Second overload voltage point

BARCLRR(3) Third overload voltage point

BARCLRR(4) Fourth overload voltage point

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer BARCLRG(4)	Is an array of four elements used to set Green color component for the line loading charts on the Diagram, 0->255 (input). BARCLRG(1) First overload voltage point BARCLRG(2) Second overload voltage point BARCLRG(3) Third overload voltage point BARCLRG(4) Fourth overload voltage point
Integer BARCLRB(4)	Is an array of four elements used to set Blue color component for the line loading charts on the Diagram, 0->255 (input). BARCLRB(1) First overload voltage point BARCLRB(2) Second overload voltage point BARCLRB(3) Third overload voltage point BARCLRB(4) Fourth overload voltage point

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.40 SETDIAGRESPFLOWOPTIONS

Use this API to specify the basic Power Flow annotation options in the active Slider Diagram. This API is obsolete and has been replaced with SETDIAGRESPFLOWOPTIONS_2.

Batch command syntax:

```
BAT_SETDIAGRESPFLOWOPTIONS BRNANN FLWOPT BUSANN BUSVLT EQPANN DIAGTTL
PMVALINE PMVATRAN USEPGMSET
```

Python syntax:

```
setdiagrespflowoptions(brnann, flwopt, busann, busvlt, eqpann,
diagttl, pmvaline, pmvatran, usepgmset)
```

where:

Integer BRNANN	Branch annotation (input).	
	BRNANN = 0	MW/MVAR flows
	BRNANN = 1	MVA
	BRNANN = 2	MVA/% rating
	BRNANN = 3	Amps/pu current
	BRNANN = 4	MW/%rating
	BRNANN = 5	Amps/% rating
	BRNANN = 6	None
Integer FLWOPT	Flow direction (input).	
	FLWOPT = 0	Arrows
	FLWOPT = 1	Signs
Integer BUSANN	Bus ID annotation (input).	
	BUSANN = 0	Number, no name
	BUSANN = 1	Name, no number
	BUSANN = 2	Number and name
	BUSANN = 3	No number, no name
Integer BUSVLT	Bus voltage annotation (input).	
	BUSVLT = 0	magnitude (pu)
	BUSVLT = 1	magnitude (kV)
	BUSVLT = 2	angle (degrees)
	BUSVLT = 3	magnitude (pu) and angle (degrees)
	BUSVLT = 4	magnitude (pu) and magnitude (kV)
	BUSVLT = 5	magnitude (kV) and angle (degrees)
	BUSVLT = 7	magnitude (pu), magnitude (kV) and angle (degrees)
	BUSVLT = 8	none
Integer EQPANN	Equipment annotation (input).	
	EQPANN = 0	MW/MVAR
	EQPANN = 1	MVA
	EQPANN = 2	None

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character DIAGTTL*260	Diagram title (input).
Integer PMVALINE	Loading basis for % loading on non-transformer branches (input). PMVALINE = 0 current loading PMVALINE = 1 MVA loading
Integer PMVATRAN	Loading basis for % loading on transformer branches (input). PMVATRAN = 0 current loading PMVATRAN = 1 MVA loading
Integer USEPGMSET	Use system or Diagram settings for displaying MVATrans and MVA-Line values (input). USEPGMSET = 0 Use Diagram settings USEPGMSET = 1 Use System settings

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.41 SETDIAGRESPFLOWOPTIONS_2

Use this API to specify the basic Power Flow annotation options in the active Slider Diagram.

Batch command syntax:

```
BAT_SETDIAGRESPFLOWOPTIONS_2 BRNANN FLWOPT BUSVLT PMVALINE PMVATRAN
USEPGMSET DIAGTTL
```

Python syntax:

```
setdiagrespflowoptions_2(brnann, flwopt, busvlt, pmvaline, pmvatran,
usepgmset, diagttl)
```

where:

Integer BRNANN	Branch annotation (input).	
	BRNANN = 0	MW/MVAR flows
	BRNANN = 1	MVA
	BRNANN = 2	MVA/% rating
	BRNANN = 3	Amps/pu current
	BRNANN = 4	MW/%rating
	BRNANN = 5	Amps/% rating
	BRNANN = 6	None
Integer FLWOPT	Flow direction (input).	
	FLWOPT = 0	Arrows
	FLWOPT = 1	Signs
Integer BUSVLT	Bus voltage annotation (input).	
	BUSVLT = 0	magnitude (pu)
	BUSVLT = 1	magnitude (kV)
	BUSVLT = 2	angle (degrees)
	BUSVLT = 3	magnitude (pu) and angle (degrees)
	BUSVLT = 4	magnitude (pu) and magnitude (kV)
	BUSVLT = 5	magnitude (kV) and angle (degrees)
	BUSVLT = 7	magnitude (pu), magnitude (kV) and angle (degrees)
	BUSVLT = 8	none
Integer PMVALINE	Loading basis for % loading on non-transformer branches (input).	
	PMVALINE = 0	current loading
	PMVALINE = 1	MVA loading
Integer PMVATRAN	Loading basis for % loading on transformer branches (input).	
	PMVATRAN = 0	current loading
	PMVATRAN = 1	MVA loading
Integer USEPGMSET	Use system or Diagram settings for displaying MVATrans and MVA-Line values (input).	
	USEPGMSET = 0	Use Diagram settings
	USEPGMSET = 1	Use System settings

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character DIAGTTL*260	Diagram title (input).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.42 SETDIAGRESREL

Use this API to specify the reliability annotation options in the active Slider Diagram.

This API is obsolete and has been replaced with SETDIAGRESREL_2

Batch command syntax:

```
BAT_SETDIAGRESREL OUTMODE MODE BUSMODE CNTBUS CNTBRN DETBUS DETBRN
PRBBUS PRBBRN VOLTREC CNTLABEL CNTLIST
```

Python syntax:

```
setdiagresrel(outmode, mode, busmode, cntbus, cntbrn, detbus, detbrn,
prbbus, prbbrn, voltrec, cntlabel, cntlist)
```

where:

Integer OUTMODE	Is the option to specify the output mode of results displayed (input). The values are as follows: OUTMODE = 1 Post contingency stage OUTMODE = 2 Post tripping stage OUTMODE = 3 Post corrective actions stage
Integer MODE	Is the option to specify the type of reliability results displayed (input). MODE = 0 Contingency analysis results MODE = 1 Deterministic reliability results MODE = 2 Probabilistic reliability results
Integer BUSMODE	Is the option to specify the mode of bus reporting (input). BUSMODE = 0 Bus voltage violations BUSMODE = 1 Bus load curtailments
Integer CNTBUS	If MODE = 0, is the option to specify whether Bus voltage is displayed (input). CNTBUS = 1 Display Bus voltage CNTBUS = 0 Don’t display Bus voltage
Integer CNTBRN	If MODE = 0, is the option to specify whether Branch flow loading is displayed (input). CNTBRN = 0 Don’t display Branch flow loading CNTBRN = 1 Display MVA loading CNTBRN = 2 Display Percent (%) CNTBRN = 3 Display both MVA and Percent (%)

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer DETBUS	If MODE = 1, is the option to specify which bus voltage violations are displayed (input).	
	DETBUS = 0	Don't display bus voltage violations
	DETBUS = DETBUS + 1	# of contingencies resulting in violations
	DETBUS = DETBUS + 2	Contingency with maximum violation
	DETBUS = DETBUS + 4	Average violations (AV)
	DETBUS = DETBUS + 8	Maximum violation (MV)
Integer DETBRN	If MODE = 1, is the option to specify which branch flow overloading violations are displayed (input).	
	DETBRN = 0	Don't display bus voltage violations
	DETBRN = DETBRN + 1	# of contingencies resulting in violations
	DETBRN = DETBRN + 2	Contingency with maximum loading
	DETBRN = DETBRN + 4	Available capacity (AC)
	DETBRN = DETBRN + 8	Average violations (AV)
	DETBRN = DETBRN + 16	Maximum loading % (ML)
Integer PRBBUS	If MODE = 2, is the option to specify which bus voltage violations are displayed (input).	
	PRBBUS = 0	Don't display bus voltage violations
	PRBBUS = PRBBUS + 1	Frequency (OC/Y)
	PRBBUS = PRBBUS + 2	Impact (VI)
	PRBBUS = PRBBUS + 4	Probability (H/Y)
	PRBBUS = PRBBUS + 8	Expected energy curtailed
Integer PRBBRN	If MODE = 2, is the option to specify which branch flow overloading violations are displayed (input).	
	PRBBRN = 0	Don't display bus voltage violations
	PRBBRN = PRBBRN + 1	Frequency (OC/Y)s
	PRBBRN = PRBBRN + 2	Impact (FI)
	PRBBRN = PRBBRN + 4	Probability (H/Y)
Integer VOLTREC	Is an index into the voltage limit record array for the current DFAX file (input).	
Character CNTLABEL*12	Is the contingency to display results for (input).	
Integer CNTLIST	Is the option to specify what contingencies are listed for selection (input).	
	CNTLIST = 1	All contingencies are listed
	CNTLIST = 2	Only those contingencies that cause overloading violations are listed
	CNTLIST = 3	Only those contingencies that cause violations are listed

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.43 SETDIAGRESREL_2

Use this API to specify the reliability annotation options in the active Slider Diagram.

Batch command syntax:

```
BAT_SETDIAGRESREL_2 OUTMODE MODE BUSMODE DETBUS DETBRN PRBBUS PRBBRN
VOLTREC
```

Python syntax:

```
setdiagresrel_2(outmode, mode, busmode, detbus, detbrn, prbbus,
prbbrn, voltrec)
```

where:

Integer OUTMODE	Is the option to specify the output mode of results displayed (input). The values are as follows:	
	OUTMODE = 1	Post contingency stage
	OUTMODE = 2	Post tripping stage
	OUTMODE = 3	Post corrective actions stage
Integer MODE	Is the option to specify the type of reliability results displayed (input).	
	MODE = 0	Deterministic reliability results
	MODE = 1	Probabilistic reliability results
Integer BUSMODE	Is the option to specify the mode of bus reporting (input).	
	BUSMODE = 0	Bus voltage violations
	BUSMODE = 1	Bus load curtailments
Integer DETBUS	If MODE = 1, is the option to specify which bus voltage violations are displayed (input).	
	DETBUS = 0	Don't display bus voltage violations
	DETBUS = DETBUS + 1	# of contingencies resulting in violations
	DETBUS = DETBUS + 2	Contingency with maximum violation
	DETBUS = DETBUS + 4	Average violations (AV)
	DETBUS = DETBUS + 8	Maximum violation (MV)
Integer DETBRN	If MODE = 1, is the option to specify which branch flow overloading violations are displayed (input).	
	DETBRN = 0	Don't display bus voltage violations
	DETBRN = DETBRN + 1	# of contingencies resulting in violations
	DETBRN = DETBRN + 2	Contingency with maximum loading
	DETBRN = DETBRN + 4	Available capacity (AC)
	DETBRN = DETBRN + 8	Average violations (AV)
	DETBRN = DETBRN + 16	Maximum loading % (ML)

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer PRBBUS	If MODE = 2, is the option to specify which bus voltage violations are displayed (input).	
	PRBBUS = 0	Don't display bus voltage violations
	PRBBUS = PRBBUS + 1	Frequency (OC/Y)
	PRBBUS = PRBBUS + 2	Impact (VI)
	PRBBUS = PRBBUS + 4	Probability (H/Y)
	PRBBUS = PRBBUS + 8	Expected energy curtailed
Integer PRBBRN	If MODE = 2, is the option to specify which branch flow overloading violations are displayed (input).	
	PRBBRN = 0	Don't display bus voltage violations
	PRBBRN = PRBBRN + 1	Frequency (OC/Y)s
	PRBBRN = PRBBRN + 2	Impact (FI)
	PRBBRN = PRBBRN + 4	Probability (H/Y)
Integer VOLTREC	Is an index into the voltage limit record array for the current DFAX file (input).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.44 SETDIAGRESSCGR

This API routine is obsolete. It has been replaced by the API routine SETDIAGRESDYN, and is implemented by a call to SETDIAGRESDYN.

Use this API to specify the short circuit analysis voltage and range checking annotation options in the active Slider Diagram.

Batch command syntax:

```
BAT_SETDIAGRESSCGR USEVLT VLNWDT(1)…VLNWDT(8) VLNSTY(1)…VLNSTY(8)
VLTVAL(1)…VLTVAL(7) VLNCLRR(1)…VLNCLRR(8) VLNCLRG(1)…VLNCLRG(8)
VLNCLRB(1)…VLNCLRB(8) OUTWDT OUTSTY OUTCLR(1)…OUTCLR(3) FLTWDT FLTSTY
FLTCLR(1)…FLTCLR(3) BUSANN
```

Python syntax:

```
setdiagresscgr(usevlt, vlnwdt, vlnsty, vltval, vlnclrr, vlnclrg,
vlnclrb, outwdt, outsty, outclr, fltwdt, fltsty, fltclr, busann)
```

where:

Integer USEVLT	Is the option to apply voltage level annotation to the Diagram elements (input). The values are as follows: USEVLT = 0 Don't apply voltage level annotation USEVLT = 1 Apply voltage level annotation
Integer VLNWDT(8)	Is an array of 8 elements used to set the width of Diagram elements that fall in the appropriate voltage range (input). VLNWDT(1) base voltage < VLTVAL(1) VLNWDT(2) VLTVAL(1) >= base voltage < VLTVAL(2) VLNWDT(3) VLTVAL(2) >= base voltage < VLTVAL(3) VLNWDT(4) VLTVAL(3) >= base voltage < VLTVAL(4) VLNWDT(5) VLTVAL(4) >= base voltage < VLTVAL(5) VLNWDT(6) VLTVAL(5) >= base voltage < VLTVAL(6) VLNWDT(7) VLTVAL(6) >= base voltage < VLTVAL(7) VLNWDT(8) base voltage >= VLTVAL(7)

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer VLNSTY(8) Is an array of 8 elements used to set the line style of Diagram elements that fall in the appropriate voltage range (input).

VLNSTY(1)	base voltage < VLTVAL(1)
VLNSTY(2)	VLTVAL(1) >= base voltage < VLTVAL(2)
VLNSTY(3)	VLTVAL(2) >= base voltage < VLTVAL(3)
VLNSTY(4)	VLTVAL(3) >= base voltage < VLTVAL(4)
VLNSTY(5)	VLTVAL(4) >= base voltage < VLTVAL(5)
VLNSTY(6)	VLTVAL(5) >= base voltage < VLTVAL(6)
VLNSTY(7)	VLTVAL(6) >= base voltage < VLTVAL(7)
VLNSTY(8)	base voltage >= VLTVAL(7)

Real VLTVAL(7) Is an array of 7 elements used to set the voltage ranges used to assign annotation Diagram. Base voltage values should be in ascending order (input).

Integer VLNCLRR(8) Is an array of 8 elements used to set the Red color component of Diagram elements that fall in the appropriate voltage range, 0->255 (input).

VLNCLRR(1)	base voltage < VLTVAL(1)
VLNCLRR(2)	VLTVAL(1) >= base voltage < VLTVAL(2)
VLNCLRR(3)	VLTVAL(2) >= base voltage < VLTVAL(3)
VLNCLRR(4)	VLTVAL(3) >= base voltage < VLTVAL(4)
VLNCLRR(5)	VLTVAL(4) >= base voltage < VLTVAL(5)
VLNCLRR(6)	VLTVAL(5) >= base voltage < VLTVAL(6)
VLNCLRR(7)	VLTVAL(6) >= base voltage < VLTVAL(7)
VLNCLRR(8)	base voltage >= VLTVAL(7)

Integer VLNCLRG(8) Is an array of 8 elements used to set Green color component of Diagram elements that fall in the appropriate voltage range, 0->255 (input).

VLNCLRG(1)	base voltage < VLTVAL(1)
VLNCLRG(2)	VLTVAL(1) >= base voltage < VLTVAL(2)
VLNCLRG(3)	VLTVAL(2) >= base voltage < VLTVAL(3)
VLNCLRG(4)	VLTVAL(3) >= base voltage < VLTVAL(4)
VLNCLRG(5)	VLTVAL(4) >= base voltage < VLTVAL(5)
VLNCLRG(6)	VLTVAL(5) >= base voltage < VLTVAL(6)
VLNCLRG(7)	VLTVAL(6) >= base voltage < VLTVAL(7)
VLNCLRG(8)	base voltage >= VLTVAL(7)

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer VLNCLRB(8) Is an array of 8 elements used to set Blue color component of Diagram elements that fall in the appropriate voltage range, 0->255 (input).

VLNCLRB(1)	base voltage < VLTVAL(1)
VLNCLRB(2)	VLTVAL(1) >= base voltage < VLTVAL(2)
VLNCLRB(3)	VLTVAL(2) >= base voltage < VLTVAL(3)
VLNCLRB(4)	VLTVAL(3) >= base voltage < VLTVAL(4)
VLNCLRB(5)	VLTVAL(4) >= base voltage < VLTVAL(5)
VLNCLRB(6)	VLTVAL(5) >= base voltage < VLTVAL(6)
VLNCLRB(7)	VLTVAL(6) >= base voltage < VLTVAL(7)
VLNCLRB(8)	base voltage >= VLTVAL(7)

Integer OUTWDT Is the line width applied to out of service Diagram items, >= 1 (input).

Integer OUTSTY Is the line style applied to out of service Diagram items (input).

Integer OUTCLR(3) Is an array of three elements specifying the color for Diagram items that are out of service (input). The values are as follows:

OUTCLR(1) = 0->255	Red component
OUTCLR(2) = 0->255	Blue component
OUTCLR(3) = 0->255	Green component

Integer FLTWDT Is the line width applied to the faulted bus Diagram item, >= 1 (input).

Integer FLTSTY Is the line style applied to the faulted bus Diagram item (input).

Integer FLTCLR(3) Is an array of three elements specifying the color for Diagram items that are out of service (input). The values are as follows:

OUTCLR(1) = 0->255	Red component
OUTCLR(2) = 0->255	Blue component
OUTCLR(3) = 0->255	Green component

Integer BUSANN Bus ID annotation (input).

BUSANN = 0	Number, no name
BUSANN = 1	Name, no number
BUSANN = 2	Number and name
BUSANN = 3	No number, no name

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.45 SETDIAGRESVRCS

Use this API to specify the system diagram defaults for Voltage coloring and style, Out of Service coloring and style etc.

Batch command syntax:

```
BAT_SETDIAGRESVCRS BUSANN EQPANN USEVCLR VLNWDT(1)…VLNWDT(8)
VLNSTY(1)…VLNSTY(8) VLTVAL(1)…VLTVAL(7) VLNCLRR(1)…VLNCLRR(8)
VLNCLRG(1)…VLNCLRG(8) VLNCLRB(1)…VLNCLRB(8) USEOOSCLR OOSWDT OOSSTY
OUTCLR(1)…OUTCLR(3) BNDCLR(1)…BARCLR(3) UNBNDCLR(1)…UNBNDCLR(3)
TRANBARCHRT
```

Python syntax:

```
setdiagresvrcs(busann, eqpann, usevclr, vlnwdt, vlnsty, vltval,
vlnclrr, vlnclrg, vlnclrb, useoosclr, ooswdt, oossty, outclr, bndclr,
unbndclr, tranbarchrt)
```

where:

Integer BUSANN	Bus ID annotation (input). BUSANN = 0 Number, no name BUSANN = 1 Name, no number BUSANN = 2 Number and name BUSANN = 3 No number, no name
Integer EQPANN	Equipment annotation (input). EQPANN = 0 MW/MVAR EQPANN = 1 MVA EQPANN = 2 None
Integer USEVCLR	Use voltage level coloring (input). The element has only two possible values. 0 = FALSE and 1 = TRUE.
Integer VLNWDT(8)	Is an array of 8 elements used to set the width of Diagram elements that fall in the appropriate voltage range (input). VLNWDT(1) base voltage < VLTVAL(1) VLNWDT(2) VLTVAL(1) >= base voltage < VLTVAL(2) VLNWDT(3) VLTVAL(2) >= base voltage < VLTVAL(3) VLNWDT(4) VLTVAL(3) >= base voltage < VLTVAL(4) VLNWDT(5) VLTVAL(4) >= base voltage < VLTVAL(5) VLNWDT(6) VLTVAL(5) >= base voltage < VLTVAL(6) VLNWDT(7) VLTVAL(6) >= base voltage < VLTVAL(7) VLNWDT(8) base voltage >= VLTVAL(7)

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer VLNSTY(8)	Is an array of 8 elements used to set the line style of Diagram elements that fall in the appropriate voltage range (input). VLNSTY(1) base voltage < VLTVAL(1) VLNSTY(2) VLTVAL(1) >= base voltage < VLTVAL(2) VLNSTY(3) VLTVAL(2) >= base voltage < VLTVAL(3) VLNSTY(4) VLTVAL(3) >= base voltage < VLTVAL(4) VLNSTY(5) VLTVAL(4) >= base voltage < VLTVAL(5) VLNSTY(6) VLTVAL(5) >= base voltage < VLTVAL(6) VLNSTY(7) VLTVAL(6) >= base voltage < VLTVAL(7) VLNSTY(8) base voltage >= VLTVAL(7)
Real VLTVAL(7)	Is an array of 7 elements used to set the voltage ranges used to assign annotation Diagram. Base voltage values should be in ascending order (input).
Integer VLNCLRR(8)	Is an array of 8 elements used to set the Red color component of Diagram elements that fall in the appropriate voltage range, 0->255 (input). VLNCLRR(1) base voltage < VLTVAL(1) VLNCLRR(2) VLTVAL(1) >= base voltage < VLTVAL(2) VLNCLRR(3) VLTVAL(2) >= base voltage < VLTVAL(3) VLNCLRR(4) VLTVAL(3) >= base voltage < VLTVAL(4) VLNCLRR(5) VLTVAL(4) >= base voltage < VLTVAL(5) VLNCLRR(6) VLTVAL(5) >= base voltage < VLTVAL(6) VLNCLRR(7) VLTVAL(6) >= base voltage < VLTVAL(7) VLNCLRR(8) base voltage >= VLTVAL(7)
Integer VLNCLRG(8)	Is an array of 8 elements used to set Green color component of Diagram elements that fall in the appropriate voltage range, 0->255 (input). VLNCLRG(1) base voltage < VLTVAL(1) VLNCLRG(2) VLTVAL(1) >= base voltage < VLTVAL(2) VLNCLRG(3) VLTVAL(2) >= base voltage < VLTVAL(3) VLNCLRG(4) VLTVAL(3) >= base voltage < VLTVAL(4) VLNCLRG(5) VLTVAL(4) >= base voltage < VLTVAL(5) VLNCLRG(6) VLTVAL(5) >= base voltage < VLTVAL(6) VLNCLRG(7) VLTVAL(6) >= base voltage < VLTVAL(7) VLNCLRG(8) base voltage >= VLTVAL(7)

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer VLNCLRB(8)	Is an array of 8 elements used to set Blue color component of Diagram elements that fall in the appropriate voltage range, 0->255 (input). VLNCLRB(1) base voltage < VLTVAL(1) VLNCLRB(2) VLTVAL(1) >= base voltage < VLTVAL(2) VLNCLRB(3) VLTVAL(2) >= base voltage < VLTVAL(3) VLNCLRB(4) VLTVAL(3) >= base voltage < VLTVAL(4) VLNCLRB(5) VLTVAL(4) >= base voltage < VLTVAL(5) VLNCLRB(6) VLTVAL(5) >= base voltage < VLTVAL(6) VLNCLRB(7) VLTVAL(6) >= base voltage < VLTVAL(7) VLNCLRB(8) base voltage >= VLTVAL(7)
Integer USEOOSCLR	Use Out-of-Service coloring (input). The element has only two possible values. 0 = FALSE and 1 = TRUE.
Integer OOSWDT	Is used to set the width of Diagram items that are out of service (input). OOSWDT Line width >= 1
Integer OOSSTY	Is used set the style of Diagram items that are out of service (input). OOSSTY Line style
Integer OUTCLR(3)	Is an array of three elements specifying the color for Diagram items that are out of service (input). The values are as follows: OUTCLR(1) = 0->255 Red component OUTCLR(2) = 0->255 Blue component OUTCLR(3) = 0->255 Green component
Integer BNDCLR(3)	Is an array of three elements specifying the bound item color (input). The values are as follows: BNDCLR(1) = 0->255 Red component BNDCLR(2) = 0->255 Blue component BNDCLR(3) = 0->255 Green component
Integer UNBNDCLR(3)	Is an array of three elements specifying the unbound item color (input). The values are as follows: UNBNDCLR(1) = 0->255 Red component UNBNDCLR(2) = 0->255 Blue component UNBNDCLR(3) = 0->255 Green component
Integer TRANBARCHRT	Two-winding transformer bar chart display side (input). The values are as follows: TRANBARCHRT = 0 High current loading side TRANBARCHRT = 1 Low current loading side TRANBARCHRT = 2 Both sides

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.46 SETDIAGRESTYPEASCC

Use this API to set the type of results displayed in the active Slider Diagram to ASCC fault calculation results. Requires that current network elements are being displayed in the diagram view.

Batch command syntax:

```
BAT_SETDIAGRESTYPEASCC
```

Python syntax:

```
setdiagrestypeascc()
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.47 SETDIAGRESTYPEDATA

Use this API to set the type of results displayed in the active Slider Diagram to impedance data values. Requires that current network elements are being displayed in the Diagram View.

Batch command syntax:

```
BAT_SETDIAGRESTYPEDATA
```

Python syntax:

```
setdiagrestypedata()
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.48 SETDIAGRESTYPEDYN

Use this API to set the type of results displayed in the active Slider Diagram to Dynamics data results. Requires that current network elements are being displayed in the Diagram View.

Batch command syntax:

```
BAT_SETDIAGRESTYPEDYN
```

Python syntax:

```
setdiagrestypedyn()
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.49 SETDIAGRESTYPEGDIF

Use this API to set the type of results displayed in the active Slider Diagram to the differences in values between the current network case and a supplied comparison case.

Batch command syntax:

```
BAT_SETDIAGRESTYPEGDIF NAMEARG SFILE
```

Python syntax:

```
setdiagrestypegdif(namearg, sfile)
```

where:

Integer NAMEARG	Flag indicating how to match up buses in the two cases (input). NAMEARG = 0 by bus number only NAMEARG = 1 by extended bus name only NAMEARG = 2 by both number and name.
Character SFILE*260	Is the name of an existing Saved case to compare the current case against (input).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.50 SETDIAGRESTYPEIEC

Use this API to set the type of results displayed in the active Slider Diagram to IEC Fault calculation results. Requires that current network elements are being displayed in the Diagram View.

Batch command syntax:

```
BAT_SETDIAGRESTYPEIEC
```

Python syntax:

```
setdiagrestypeiec()
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.51 SETDIAGRESTYPEPFLOW

Use this API to set the type of results displayed in the active Slider Diagram to Power Flow results. Requires that current network elements are being displayed in the Diagram View.

Batch command syntax:

```
BAT_SETDIAGRESTYPEPFLOW
```

Python syntax:

```
setdiagrestypepflow()
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.52 SETDIAGRESTYPEREL

Use this API to set the type of results displayed in the active Slider Diagram to reliability analysis results. Requires that current network elements are being displayed in the Diagram View.

Batch command syntax:

```
BAT_SETDIAGRESTYPEREL
```

Python syntax:

```
setdiagrestyperel()
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.53 SETDIAGRESTYPESCGR

Use this API to set the type of results displayed in the active Slider Diagram to the results of a short circuit analysis.

Batch command syntax:

```
BAT_SETDIAGRESTYPESCGR OPTIONS(1) … OPTIONS(8)
```

Python syntax:

```
setdiagrestypescgr (options)
```

where:

Integer OPTIONS(8)	Is an array of eight elements specifying analysis options (input). The values are as follows:

OPTIONS(1)	Faults
OPTIONS(1) = 0	3 Phase
OPTIONS(1) = 1	3 Phase and L-G
OPTIONS(2)	L-G fault output
OPTIONS(2) = 0	A phase
OPTIONS(2) = 1	Zero sequence
OPTIONS(3)	Fault location
OPTIONS(3) = 0	All
OPTIONS(3) > 0	Faulted bus number
OPTIONS(4)	Flat conditions
OPTIONS(4) = 0	No
OPTIONS(4) = 1	Yes
OPTIONS(5)	Double precision
OPTIONS(5) = 0	No
OPTIONS(5) = 1	Yes (ignored)
OPTIONS(6)	dc line/FACTS device option
OPTIONS(6) = 0	Block
OPTIONS(6) = 1	Model as load
OPTIONS(7)	Zero sequence transformer impedance correction option
OPTIONS(7) = 0	Ignore
OPTIONS(7) = 1	Apply
OPTIONS(8)	Branch quantity option
OPTIONS(8) = 0	Currents
OPTIONS(8) = 1	Apparent impedances.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.54 UPDATEBUSLOCDIAGFILE

Use this API to update bus locations in the active Slider Diagram in the GUI using the positions found in the current bus location file.

Batch command syntax:

```
BAT_UPDATEBUSLOCDIAGFILE
```

Python syntax:

```
updatebuslocdiagfile()
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.55 NEXTDIAGFILEDATASET

This API is used to display the next element in the active data set on the active Slider Diagram. Currently, this API is only used with ACCC results.

Batch command syntax:

```
BAT_NEXTDIAGFILEDATASET
```

Python syntax:

```
nextdiagfiledataset()
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.56 PREVDIAGFILEDATASET

This API is used to display the previous element in the active data set on the active Slider Diagram. Currently, this API is only used with ACCC results.

Batch command syntax:

```
BAT_PREVDIAGFILEDATASET
```

Python syntax:

```
prevdiagfiledataset()
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

6.57 SWITCHDIAGFILEDATASET

This API is used to rotate through values for the current element in the active data set on the active Slider Diagram. Currently, this API is only used with ACCC results and is used for displaying the post contin-gency, post tripping and post corrective actions values.

Batch command syntax:

```
BAT_SWITCHDIAGFILEDATASET
```

Python syntax:

```
switchdiagfiledataset()
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

This page intentionally left blank.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

SIEMENS

Chapter 7

Single Element Data Retrieval

This chapter defines API routines that are used to retrieve data for a single element (e.g., bus, branch, area, etc.) from the PSS®E engine and test on various program states.

These routines do not record themselves when called with recording enabled.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.1 ARDAT

Use this API to return area totals.

Python syntax:

```
ierr, cmpval  = ardat(iar, string)
```

IPLAN syntax:

```
CALL ARDAT(IAR, STRING, P, Q, IERR)
```

Fortran syntax:

```
CALL ARDAT(IAR, STRING, CMPVAL, IERR)
```

where:

Integer IAR	Area number (input).	
Character STRING	String indicating the area total desired (input):	
	’LOAD’	Total area load by bus area assignment.
	’LOADLD’	Total area load by load area assignment.
	’GEN’	Total area generation.
	’LOSS’	Total area losses.
	’INT’	Net area interchange.
	’INDMAC’	Total area induction machine powers by bus area assignment.
	’INDMACMC’	Total area induction machine powers by machine area assignment.
	’INDGEN’	Total area induction generator powers by bus area assignment.
	’INDGENMC’	Total area induction generator powers by machine area assignment.
	’INDMOT’	Total area induction motor powers by bus area assignment.
	’INDMOTMC’	Total area induction motor powers by machine area assignment.
Real P	Real component of the desired complex power (output).	
Real Q	Reactive component of the desired complex power (output).	
Complex CMPVAL	Desired complex power (output).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’P’ and ’Q’ or ’CMPVAL’ returned.
	IERR = 1	Area number ≤ 0 or > largest allowable area number; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
	IERR = 2	No in-service buses with in-service loads (for ’LOAD’), no in-service loads (for ’LOADLD’), no type 2 or type 3 buses (for ’GEN’), no in-service branches (for ’LOSS’), no in-service ties (for ’INT’), no in-service buses with in-service induction machines of the appropriate type (for ’INDMAC’, ’INDGEN’ and ’INDMOT’), or no in-service induction machines of the appropriate type (for ’INDMACMC’, ’INDGENMC’ and ’INDMOTMC’) in area; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
	IERR = 3	Area not found; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
	IERR = 4	Bad ’STRING’ value; ’P’ and ’Q’ or ’CMPVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.2 AREDAT

Use this API to return real area data items.

Python syntax:

```
ierr, rval  = aredat(iar, string)
```

IPLAN syntax:

```
CALL AREDAT(IAR, STRING, RVAL, IERR)
```

Fortran syntax:

```
CALL AREDAT(IAR, STRING, RVAL, IERR)
```

where:

Integer IAR	Area number (input).	
Character STRING	String indicating the area quantity desired (input):	
	’PDES’	Desired net interchange leaving area, in MW.
	’PTOL’	Interchange tolerance bandwidth, in MW.
	O_PDES	Same as ’PDES’, in units determined by the power output option setting.
	O_PTOL	Same as ’PTOL’, in units determined by the power output option setting.
Real RVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’RVAL’ returned.
	IERR = 1	Area number ≤ 0 or > largest allowable area number; ’RVAL’ unchanged.
	IERR = 2	Area not found; ’RVAL’ unchanged.
	IERR = 3	Bad ’STRING’ value; ’RVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.3 AREINT

Use this API to return integer area data items.

Python syntax:

```
ierr, ival  = areint(iar, string)
```

IPLAN syntax:

```
CALL AREINT(IAR, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL AREINT(IAR, STRING, IVAL, IERR)
```

where:

Integer IAR	Area number (input).
Character STRING	String indicating the area quantity desired (input): 'SWING' Area slack bus number.
Integer IVAL	Value indicated by STRING (output).
Integer IERR	Is the error code (output): IERR = 0 No error; 'IVAL' returned. IERR = 1 Area number ≤ 0 or > largest allowable area number; 'IVAL' unchanged. IERR = 2 Area not found; 'IVAL' unchanged. IERR = 3 Bad 'STRING' value; 'IVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.4 ARENAM

Use this API to return the area name.

Python syntax:

```
ierr, cval  = arenam(iar)
```

IPLAN syntax:

```
CALL ARENAM(IAR, CVAL, IERR)
```

Fortran syntax:

```
CALL ARENAM(IAR, CVAL, IERR)
```

where:

Integer IAR	Area number (input).
Character CVAL	Area name (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; 'CVAL' returned.
	IERR = 1 — Area number ≤ 0 or > largest allowable area number; 'CVAL' unchanged.
	IERR = 2 — Area not found; 'CVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.5 ARENUM

Use this API to return the area number.

Python syntax:

```
ierr, iar  = arenum(cval)
```

IPLAN syntax:

```
CALL ARENUM(IAR, CVAL, IERR)
```

Fortran syntax:

```
CALL ARENUM(IAR, CVAL, IERR)
```

where:

Integer IAR	Area number (output).	
Character CVAL	Area name (input).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; 'IAR' returned.
	IERR = 1	Area with name 'CVAL' not found; 'IAR' unchanged.
	IERR > 1	Multiple areas with the name 'CVAL' found, 'IAR' returned as lowest number of these.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.6 AREUSE

Use this API to indicate if an area is in use.

Python syntax:

```
ival  = areuse(iar)
```

IPLAN syntax:

```
CALL AREUSE(IAR, IVAL)
```

Fortran syntax:

```
CALL AREUSE(IAR, IVAL)
```

where:

Integer IAR	Area number (input).	
Integer IVAL	Area status (output):	
	IVAL = 1	If any equipment is assigned to area IAR.
	IVAL = 0	If no equipment is assigned to area IAR.
	IVAL = -1	Area number ≤ 0 or > largest allowable area number.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.7 ARITOJ

Use this API to return the interchange between two areas.

Python syntax:

```
ierr, cmpval  = aritoj(iar, jar)
```

IPLAN syntax:

```
CALL ARITOJ(IAR, JAR, P, Q, IERR)
```

Fortran syntax:

```
CALL ARITOJ(IAR, JAR, CMPVAL, IERR)
```

where:

Integer IAR	From area number (input).
Integer JAR	To area number (input).
Real P	Real component of the complex interchange (output).
Real Q	Reactive component of the complex interchange (output).
Complex CMPVAL	Complex interchange (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; ’P’ and ’Q’ or ’CMPVAL’ returned.
	IERR = 1 — Area number ≤ 0 or > largest allowable area number; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
	IERR = 2 — No in-service branches between areas; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
	IERR = 3 — Area not found; ’P’ and ’Q’ or ’CMPVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.8 BRNCUR

This API is obsolete. It has been replaced by the API BRNMSC with STRING='AMPS'.

Python syntax:

```
ierr, rval = brncur(ibus,jbus,ickt)
```

IPLAN syntax:

```
CALL BRNCUR(IBUS, JBUS, ICKT, RVAL, IERR)
```

Fortran syntax:

```
CALL BRNCUR(IBUS, JBUS, ICKT, RVAL, IERR)
```

where:

Integer IBUS	From bus number (input).
Integer JBUS	To bus number (input).
Character ICKT	Circuit or multisection line identifier (input).
Real RVAL	Flow value (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; 'RVAL' returned.
	IERR = 1 — Bus not found; 'RVAL' unchanged.
	IERR = 2 — Branch not found; 'RVAL' unchanged.
	IERR = 3 — Branch out-of-service; 'RVAL' returned set to zero.
	IERR = 4 — No base kV at IBUS; 'RVAL' returned = per unit current.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.9 BRNDAT

Use this API to return real branch parameters.

Python syntax:

```
ierr, rval = brndat(ibus,jbus,ickt,string)
```

IPLAN syntax:

```
CALL BRNDAT(IBUS, JBUS, ICKT, STRING, RVAL, IERR)
```

Fortran syntax:

```
CALL BRNDAT(IBUS, JBUS, ICKT, STRING, RVAL, IERR)
```

where:

Integer IBUS	From bus number (input).
Integer JBUS	To bus number (input).
Character ICKT	Circuit identifier (input).
Character STRING	String indicating the branch quantity desired (input):
	'RATEA' First rating.
	'RATEB' Second rating.
	'RATEC' Third rating.
	'RATE' Default rating.
	'LENGTH' Line length.
	'CHARG' Total branch charging.
	'CHARGZ' Zero sequence charging.
	'FRACT1' Fraction of total ownership assigned to the first owner.
	'FRACT2' Fraction of total ownership assigned to the second owner.
	'FRACT3' Fraction of total ownership assigned to the third owner.
	'FRACT4' Fraction of total ownership assigned to the fourth owner.
Real RVAL	Value indicated by STRING (output).
Integer IERR	Is the error code (output):
	IERR = 0 No error; 'RVAL' returned.
	IERR = 1 Bus not found; 'RVAL' unchanged.
	IERR = 2 Branch not found; 'RVAL' unchanged.
	IERR = 3 Invalid value of 'STRING'; 'RVAL' unchanged.
	IERR = 4 'BRNDAT' invalid with multisection line identifier; 'RVAL' unchanged.
	IERR = 5 Sequence data not in case; 'RVAL' unchanged.
	IERR = 6 For STRING = 'FRACTn', n > number of owners; 'RVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.10 BRNDT2

Use this API to return complex branch parameters.

Python syntax:

```
ierr, cmpval = brndt2(ibus,jbus,ickt,string)
```

IPLAN syntax:

```
CALL BRNDT2(IBUS, JBUS, ICKT, STRING, P, Q, IERR)
```

Fortran syntax:

```
CALL BRNDT2(IBUS, JBUS, ICKT, STRING, CMPVAL, IERR)
```

where:

Integer IBUS	From bus number (input).
Integer JBUS	To bus number (input).
Character ICKT	Circuit identifier (input).
Character STRING	String indicating the branch quantity desired (input): 'RX' — Branch resistance and reactance in per unit. 'ISHNT' — Admittance of line shunt at IBUS end of the branch. 'JSHNT' — Admittance of line shunt at JBUS end of the branch. 'RXZ' — Zero sequence branch resistance and reactance. 'ISHNTZ' — Zero sequence admittance of line shunt at IBUS end of the branch. 'JSHNTZ' — Zero sequence admittance of line shunt at JBUS end of the branch. 'LOSSES' — Branch losses in MW and Mvar. 'O_LOSSES' — Branch losses in units determined by the power output option setting.
Real P	Real component of the complex value indicated by STRING (output).
Real Q	Reactive component of the complex value indicated by STRING (output).
Complex CMPVAL	Complex value indicated by STRING (output).
Integer IERR	Is the error code (output): IERR = 0 — No error; 'P' and 'Q' or 'CMPVAL' returned. IERR = 1 — Bus not found; 'P' and 'Q' or 'CMPVAL' unchanged. IERR = 2 — Branch not found; 'P' and 'Q' or 'CMPVAL' unchanged. IERR = 3 — Invalid value of 'STRING'; 'P' and 'Q' or 'CMPVAL' unchanged. IERR = 4 — 'BRNDT2' invalid with multisection line identifier; 'P' and 'Q' or 'CMPVAL' unchanged. IERR = 5 — Sequence data not in case; 'P' and 'Q' or 'CMPVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.11 BRNFLO

Use this API to return the complex branch flow (P+jQ) as calculated at IBUS.

Python syntax:

```
ierr, cmpval = brnflo(ibus,jbus,ickt)
```

IPLAN syntax:

```
CALL BRNFLO(IBUS, JBUS, ICKT, P, Q, IERR)
```

Fortran syntax:

```
CALL BRNFLO(IBUS, JBUS, ICKT, CMPVAL, IERR)
```

where:

Integer IBUS	From bus number (input).
Integer JBUS	To bus number (input).
Character ICKT	Circuit or multisection line identifier (input).
Real P	Real component of the complex branch flow (output).
Real Q	Reactive component of the complex branch flow (output).
Complex CMPVAL	Complex branch flow (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; ’P’ and ’Q’ or ’CMPVAL’ returned.
	IERR = 1 — Bus not found; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
	IERR = 2 — Branch not found; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
	IERR = 3 — Branch is out-of-service; ’P’ and ’Q’ of 0.0 or ’CMPVAL’ of (0.0,0.0) returned.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.12 BRNINT

Use this API to return integer branch parameters.

Python syntax:

```
ierr, ival = brnint(ibus,jbus,ickt,string)
```

IPLAN syntax:

```
CALL BRNINT(IBUS, JBUS, ICKT, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL BRNINT(IBUS, JBUS, ICKT, STRING, IVAL, IERR)
```

where:

Integer IBUS	From bus number (input).
Integer JBUS	To bus number (input).
Character ICKT	Circuit or multisection line identifier (input).
Character STRING	String indicating the branch quantity desired (input):
	'STATUS' Branch status.
	'METER' Bus number of metered end.
	'NMETR' Bus number of non-metered end.
	'OWNERS' Number of owners for this branch.
	'OWN1' Owner number of the first owner.
	'OWN2' Owner number of the second owner.
	'OWN3' Owner number of the third owner.
	'OWN4' Owner number of the fourth owner.
Integer IVAL	Value indicated by STRING (integer output).
Integer IERR	Is the error code (integer output):
	IERR = 0 No error; 'IVAL' returned.
	IERR = 1 Bus not found; 'IVAL' unchanged.
	IERR = 2 Branch not found; 'IVAL' unchanged.
	IERR = 3 Invalid value of 'STRING'; 'IVAL' unchanged.
	IERR = 6 For 'STRING' = 'OWNn', n > number of owners; 'IVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.13 BRNMSC

Use this API to return real branch flow values.

Python syntax:

```
ierr, rval = brnmsc(ibus,jbus,ickt,string)
```

IPLAN syntax:

```
CALL BRNMSC(IBUS, JBUS, ICKT, STRING, RVAL, IERR)
```

Fortran syntax:

```
CALL BRNMSC(IBUS, JBUS, ICKT, STRING, RVAL, IERR)
```

where:

Integer IBUS	From bus number (input).
Integer JBUS	To bus number (input).
Character ICKT	Circuit or multisection line identifier (input).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING	String indicating the branch quantity desired (input):	
	'MVA'	Branch MVA flow.
	'O_MVA'	Same as 'MVA', in units determined by the power output option setting.
	'AMPS'	Branch current in amps.
	'PUCUR'	Branch current in pu.
	'CURANG'	Angle of branch current in degrees.
	'PCTRTA'	Percent current loading of RATEA.
	'PCTRTB'	Percent current loading of RATEB.
	'PCTRTC'	Percent current loading of RATEC.
	'PCTMVA'	Percent MVA loading of RATEA.
	'PCTMVB'	Percent MVA loading of RATEB.
	'PCTMVC'	Percent MVA loading of RATEC.
	'PCTCPA'	Percent current or MVA loading (according to the appropriate percent loading units program option setting) of RATEA.
	'PCTCPB'	Percent current or MVA loading (according to the appropriate percent loading units program option setting) of RATEB.
	'PCTCPC'	Percent current or MVA loading (according to the appropriate percent loading units program option setting) of RATEC.
	'P'	Branch MW flow at the IBUS end of the branch.
	'O_P'	Same as 'P', in units determined by the power output option setting.
	'Q'	Branch Mvar flow at the IBUS end of the branch.
	'O_Q'	Same as 'Q', in units determined by the power output option setting.
	'PLOS'	Branch active power losses in MW.
	'O_PLOS'	Branch active power losses in units determined by the power output option setting.
	'QLOS'	Branch reactive power losses in Mvar.
	'O_QLOS'	Branch reactive power losses in units determined by the power output option setting.
Real RVAL	Flow value (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; 'RVAL' returned.
	IERR = 1	Bus not found; 'RVAL' unchanged.
	IERR = 2	Branch not found; 'RVAL' unchanged.
	IERR = 3	Branch out-of-service; 'RVAL' returned set to zero.
	IERR = 4	No base kV at IBUS; 'RVAL' returned = per unit current.
	IERR = 5	Invalid 'STRING'; 'RVAL' unchanged.
	IERR = 6	Rating is zero; 'RVAL' returned set to zero.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.14 BRNMVA

This API is obsolete. It has been replaced by the API BRNMSC with STRING='MVA'.

Python syntax:

```
ierr, rval = brnmva(ibus,jbus,ickt)
```

IPLAN syntax:

```
CALL BRNMVA(IBUS, JBUS, ICKT, RVAL, IERR)
```

Fortran syntax:

```
CALL BRNMVA(IBUS, JBUS, ICKT, RVAL, IERR)
```

where:

Integer IBUS	From bus number (input).
Integer JBUS	To bus number (input).
Character ICKT	Circuit or multisection line identifier (input).
Real RVAL	Flow value (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; 'RVAL' returned.
	IERR = 1 — Bus not found; 'RVAL' unchanged.
	IERR = 2 — Branch not found; 'RVAL' unchanged.
	IERR = 3 — Branch out-of-service; 'RVAL' returned set to zero.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.15 BRNSTT

This API is obsolete. It has been replaced by the API BRNINT with STRING='STATUS'.

Python syntax:

```
ierr, ival = brnstt(ibus,jbus,ickt)
```

IPLAN syntax:

```
CALL BRNSTT(IBUS, JBUS, ICKT, IVAL, IERR)
```

Fortran syntax:

```
CALL BRNSTT(IBUS, JBUS, ICKT, IVAL, IERR)
```

where:

Integer IBUS	From bus number (input).
Integer JBUS	To bus number (input).
Character ICKT	Circuit or multisection line identifier (input).
Integer IVAL	Branch status (output). 0 for out of service 1 for in-service
Integer IERR	Is the error code (output): IERR = 0 No error; 'IVAL' returned. IERR = 1 Bus not found; 'IVAL' unchanged. IERR = 2 Branch not found; 'IVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.16 BSYSISDEF

Use this API to check whether a bus subsystem has been defined for a given subsystem ID. A return value of FALSE means that the given subsystem ID is operating in ALL buses mode, e.g., all buses are operated on.

Python syntax:

```
exist = bsysisdef(sid)
```

IPLAN syntax:

```
CALL BSYSISDEF(SID,EXIST)
```

Fortran syntax:

```
CALL SBSBDEF(SID,EXIST)
```

where:

Integer SID	Is a valid bus subsystem ID. Valid bus subsystem IDs range from 0 to 11.	
Logical EXIST	Is the function return.	
	EXIST = False	no subsystem defined, operating in ALL buses mode.
	EXIST = True	subsystem is defined.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.17 BUSDAT

Use this API to return real bus values.

Python syntax:

```
ierr, rval = busdat(ibus ,string)
```

IPLAN syntax:

```
CALL BUSDAT(IBUS, STRING, RVAL, IERR)
```

Fortran syntax:

```
CALL BUSDAT(IBUS, STRING, RVAL, IERR)
```

where:

Integer IBUS	Bus number (input).	
Character STRING	String indicating the bus quantity desired (input):	
	’BASE’	Bus base voltage, in kV.
	’PU’	Actual bus voltage magnitude, in pu.
	’KV’	Actual bus voltage magnitude, in kV.
	’ANGLE’	Bus voltage angle, in radians.
	’ANGLED’	Bus voltage angle, in degrees.
	’NVLMHI’	Normal high voltage limit, in pu.
	’NVLMLO’	Normal low voltage limit, in pu.
	’EVLMHI’	Emergency high voltage limit, in pu.
	’EVLMLO’	Emergency low voltage limit, in pu.
Real RVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’RVAL’ returned.
	IERR = 1	Bus not found; ’RVAL’ unchanged.
	IERR = 2	Bad value of ’STRING’; ’RVAL’ unchanged.
	IERR = 3	Bus type code 4 or greater; ’RVAL’ returned.
	IERR = 4	No base voltage and ’STRING’ = ’KV’; ’RVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.18 BUSDT1

Use this API to return complex bus parameters as MVA. Values may be either nominal or actual.

Python syntax:

```
ierr, rval = busdt1(ibus, string1, string2)
```

IPLAN syntax:

```
CALL BUSDT1(IBUS, STRING1, STRING2, RVAL, IERR)
```

Fortran syntax:

```
CALL BUSDT1(IBUS, STRING1, STRING2, RVAL, IERR)
```

where:

Integer IBUS	Bus number (input).	
Character STRING1	String indicating the bus quantity desired (input):	
	'MVA'	Total in-service constant MVA load.
	'IL'	Total in-service constant current load.
	'YL'	Total in-service constant admittance load.
	'TOTAL'	Total in-service load.
	'SC_MVA'	Total in-service scalable constant MVA load.
	'SC_IL'	Total in-service scalable constant current load.
	'SC_YL'	Total in-service scalable constant admittance load.
	'SC_TOTAL'	Total in-service scalable load.
	'FX_MVA'	Total in-service fixed constant MVA load.
	'FX_IL'	Total in-service fixed constant current load.
	'FX_YL'	Total in-service fixed constant admittance load.
	'FX_TOTAL'	Total in-service fixed load.
	'YS'	Total in-service fixed bus shunt.
	'YSZERO'	Total nominal in-service zero sequence fixed bus shunt.
	'YSZ'	Total nominal in-service zero sequence fixed bus shunt in per unit.
	'YSW'	In-service switched shunt.
	'YSWZ'	In-service zero sequence switched shunt.
	'SHUNTN'	Total in-service negative sequence load (pu, nominal).
	'SHUNTZ'	Total in-service zero sequence load (pu, nominal).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING2 — String indicating the units to be used (input; ignored if STRING1 = 'YSZ' or 'YSWZ'):

- 'ACT' — Actual load or shunt (nominal shunt for STRING1 = 'YSZERO' or 'YSWZERO'). 'RVAL' returned in MVA.
- 'O_ACT' — Actual load or shunt (nominal shunt for STRING1 = 'YSZERO' or 'YSWZERO'). 'RVAL' in units determined by the power output option setting.
- 'NOM' — Nominal load or shunt (at 1.0 pu voltage). 'RVAL' returned in MVA.
- 'O_NOM' — Nominal load or shunt (at 1.0 pu voltage). 'RVAL' in units determined by the power output option setting.

Real RVAL — Magnitude of the complex value indicated by STRING1 in the units indicated by STRING2 (output).

Integer IERR — Is the error code (output):

- IERR = 0 — No error; 'RVAL' returned.
- IERR = 1 — Bus not found; 'RVAL' unchanged.
- IERR = 2 — Bad value of 'STRING1' or 'STRING2'; 'RVAL' unchanged.
- IERR = 3 — Bus type code 4 or greater; 'RVAL' returned.
- IERR = 4 — No loads at bus for 'STRING1' = 'MVA', 'IL', 'YL', or 'TOTAL'; 'RVAL' returned set to zero.
- IERR = 5 — No switched shunt at bus for 'STRING1' = 'YSW'; 'YSWZERO' or 'YSWZ'; 'RVAL' unchanged.
- IERR = 6 — Sequence data not in case for 'STRING1' = 'YSZERO', 'YSZ', 'YSWZERO', 'YSWZ', 'SHUNTN' or 'SHUNTZ'; 'RVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.19 BUSDT2

Use this API to return complex bus parameters. Values may be either nominal or actual. Nominal values return negative values for reactors. Actual values show flow into reactor.

Python syntax:

```
ierr, cmpval = busdt2(ibus, string1, string2)
```

IPLAN syntax:

```
CALL BUSDT2(IBUS, STRING1, STRING2, P, Q, IERR)
```

Fortran syntax:

```
CALL BUSDT2(IBUS, STRING1, STRING2, CMPVAL, IERR)
```

where:

Integer IBUS	Bus number (input).	
Character STRING1	String indicating the bus quantity desired (input):	
	’MVA’	Total in-service constant MVA load.
	’IL’	Total in-service constant current load.
	’YL’	Total in-service constant admittance load.
	’TOTAL’	Total in-service load.
	’SC_MVA’	Total in-service scalable constant MVA load.
	’SC_IL’	Total in-service scalable constant current load.
	’SC_YL’	Total in-service scalable constant admittance load.
	’SC_TOTAL’	Total in-service scalable load.
	’FX_MVA’	Total in-service fixed constant MVA load.
	’FX_IL’	Total in-service fixed constant current load.
	’FX_YL’	Total in-service fixed constant admittance load.
	’FX_TOTAL’	Total in-service fixed load.
	’YS’	Total in-service fixed bus shunt.
	’YSZERO’	Total nominal in-service zero sequence fixed bus shunt.
	’YSZ’	Total nominal in-service zero sequence fixed bus shunt in per unit.
	’YSW’	In-service switched shunt.
	’YSWZERO’	Nominal in-service zero sequence switched shunt.
	’YSWZ’	Nominal in-service zero sequence switched shunt in per unit.
	’SHUNTN’	Total in-service negative sequence load (pu, nominal).
	’SHUNTZ’	Total in-service zero sequence load (pu, nominal).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING2	String indicating the units to be used (input; ignored if STRING1 = 'YSZ' or 'YSWZ'):	
	'ACT'	Actual load or shunt (nominal shunt for STRING1 = 'YSZERO' or 'YSWZERO'). 'P', 'Q' returned in MW, Mvar.
	'O_ACT'	Actual load or shunt (nominal shunt for STRING1 = 'YSZERO' or 'YSWZERO'). 'P', 'Q' in units determined by the power output option setting.
	'NOM'	Nominal load (at 1.0 pu voltage). 'P', 'Q' returned in MW, Mvar.
	'O_NOM'	Nominal load (at 1.0 pu voltage). 'P', 'Q' in units determined by the power output option setting.
Real P	Real component of the complex value indicated by STRING1 in the units indicated by STRING2 (output).	
Real Q	Reactive component of the complex value indicated by STRING1 in the units indicated by STRING2 (output).	
Complex CMPVAL	Complex value indicated by STRING1 in the units indicated by STRING2 (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; 'P' and 'Q' or 'CMPVAL' returned.
	IERR = 1	Bus not found; 'P' and 'Q' or 'CMPVAL' unchanged.
	IERR = 2	Bad value of 'STRING1' or 'STRING2'; 'P' and 'Q' or 'CMPVAL' unchanged.
	IERR = 3	Bus type code 4 or greater; 'P' and 'Q' or 'CMPVAL' returned.
	IERR = 4	No loads at bus for 'STRING1' = 'MVA', 'IL', 'YL', or 'TOTAL'; 'P' and 'Q' or 'CMPVAL' returned set to zero.
	IERR = 5	No switched shunt at bus for 'STRING1' = 'YSW', 'YSWZERO' or 'YSWZ'; 'P' and 'Q' or 'CMPVAL' unchanged.
	IERR = 6	Sequence data not in case for 'STRING1' = 'YSZERO', 'YSZ', 'YSWZERO', 'YSWZ', 'SHUNTN' or 'SHUNTZ'; 'CMPVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.20 BUSEXS

Use this API to check for the existence of a specified bus.

Python syntax:

```
ierr = busexs(ibus)
```

IPLAN syntax:

```
CALL BUSEXS(IBUS, IERR)
```

Fortran syntax:

```
CALL BUSEXS(IBUS, IERR)
```

where:

Integer IBUS	Bus number (input).	
Integer IERR	Is the error code (output):	
	IERR = 0	Bus found.
	IERR = 1	Bus not found.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.21 BUSINT

Use this API to return integer bus parameters.

Python syntax:

```
ierr, ival = busint(ibus ,string)
```

IPLAN syntax:

```
CALL BUSINT(IBUS, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL BUSINX(IBUS, STRING, IVAL, IERR)
```

where:

Integer IBUS	Bus number (input).	
Character STRING	String indicating the bus quantity desired (input):	
	’TYPE’	Bus type code.
	’AREA’	Bus area number.
	’ZONE’	Bus zone number.
	’OWNER’	Bus owner number.
	’DUMMY’	Returns 1 (one) if bus is a dummy bus of a multisection line, else 0 (zero).
Integer IVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’IVAL’ returned.
	IERR = 1	Bus not found; ’IVAL’ unchanged.
	IERR = 2	Bad value of ’STRING’; ’IVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.22 BUSMSM

Use this API to return complex bus mismatch.

Python syntax:

```
ierr, cmpval = busmsm(ibus)
```

IPLAN syntax:

```
CALL BUSMSM(IBUS, P, Q, IERR)
```

Fortran syntax:

```
CALL BUSMSM(IBUS, CMPVAL, IERR)
```

where:

Integer IBUS	Bus number (input).
Real P	Real component (MW) of the bus mismatch (output).
Real Q	Reactive component (Mvar) of the bus mismatch (output).
Complex CMPVAL	Complex bus mismatch (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; ’P’ and ’Q’ or ’CMPVAL’ returned.
	IERR = 1 — Bus not found; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
	IERR = 2 — Bus type code greater than 4; ’P’ and ’Q’ or ’CMPVAL’ returned set to zero.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.23 CHKTRE

Use this API to check for Type 4 (or greater) buses with in-service branches connected to them. If none are found, it then counts number of ac islands not containing a Type 3 (swing) bus. It may be inappropriate to call this subroutine with generators converted; in this case, there are no Type 3 buses, so there is normally one or more swingless islands.

Python syntax:

```
ival = chktre()
```

IPLAN syntax:

```
CALL CHKTRE(IVAL)
```

Fortran syntax:

```
CALL CHKTRE(IVAL)
```

where:

Integer IVAL	Return code (output):	
	IVAL = 0	None of the following found.
	IVAL = -1	No in-service buses remaining.
	IVAL = -2	There are Type 4 buses with in-service branches connected to them.
	IVAL > 0	Number of islands without a Type 3 bus.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.24 CHNVAL

Use this API to return the present value of the simulation variable assigned to a specified output channel.

Python syntax:

```
ierr, rval = chnval(n)
```

IPLAN syntax:

```
CALL CHNVAL(N, RVAL, IERR)
```

Fortran syntax:

```
CALL CHNVAL(N, RVAL, IERR)
```

where:

Integer N	Channel number (input).	
Real RVAL	Channel value (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; 'RVAL' returned.
	IERR = 1	Invalid channel index; 'RVAL' unchanged.
	IERR = 2	Invalid channel address; 'RVAL' unchanged.
	IERR = 9	Dynamics data not present in working memory; 'RVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.25 DC2AUXMIND

Use this API to return the starting array indices and status of auxiliary signal model associated with 2-terminal dc line for the specified signal index.

Python syntax:

```
ierr, ival = dc2auxmind(name, isgndx, string)
```

IPLAN syntax:

```
CALL DC2AUXMIND(NAME, ISGNDX, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL DC2AUXMIND(NAME, ISGNDX, STRING, IVAL, IERR)
```

where:

Character NAME	Two-terminal dc line name (input).	
Integer ISGNDX	Auxiliary signal index (input).	
Character STRING	String indicating the 2-terminal dc line auxiliary signal model quantity desired (input):	
	’CON’	Starting CON index.
	’STATE’	Starting STATE index.
	’VAR’	Starting VAR index.
	’ICON’	Starting ICON index.
	’NCON’	Number of CON values used by the model.
	’NSTATE’	Number of STATE values used by the model.
	’NVAR’	Number of VAR values used by the model.
	’NICON’	Number of ICON values used by the model.
	’STATUS’	Bypassed flag: 0 = bypassed; 1 = not bypassed. For a model to be called during simulations, the model’s bypassed flag must be *not bypassed*.
Integer IVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’IVAL’ returned.
	IERR = 1	dc line not in network but auxiliary signal model in model tables; ’IVAL’ returned.
	IERR = 2	dc line not in network and auxiliary signal model not in model tables; ’IVAL’ unchanged.
	IERR = 4	No model in model tables for the given auxiliary signal index; ’IVAL’ unchanged
	IERR = 5	Invalid value of ’STRING’; ’IVAL’ unchanged.
	IERR = 6	Model does not use any of this data type; ’IVAL’ = 0 returned.
	IERR = 9	Dynamics data not present in working memory; ’IVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.26 DC2AUXMNAM

Use this API to return the auxiliary signal model name associated with a 2-terminal dc line for the specified auxiliary signal index.

Python syntax:

```
ierr, cval = dc2auxmnam(name, isgndx)
```

IPLAN syntax:

```
CALL DC2AUXMNAM(NAME, ISGNDX, CVAL, IERR)
```

Fortran syntax:

```
CALL DC2AUXMNAM(NAME, ISGNDX, CVAL, IERR)
```

where:

Character NAME	Two-terminal dc line name (input).
Integer ISGNDX	Auxiliary signal index (input).
Character CVAL	Model name (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; ’CVAL’ returned.
	IERR = 1 — dc line not in network but auxiliary signal model in model tables; ’CVAL’ returned.
	IERR = 2 — dc line not in network and auxiliary signal model not in model tables; ’CVAL’ unchanged.
	IERR = 4 — No model in model tables for the given auxiliary signal index; ’CVAL’ unchanged
	IERR = 9 — Dynamics data not present in working memory; ’CVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.27 DC2DAT

This API is obsolete. It has been replaced by the API DC2DAT_2, and is implemented by a call to the DC2DAT_2 API with the dc line number converted to a left-justified name.

Use this API to return real two-terminal dc line quantities.

Python syntax:

```
ierr, rval = dc2dat(n, ibus, string)
```

IPLAN syntax:

```
CALL DC2DAT(N, IBUS, STRING, RVAL, IERR)
```

Fortran syntax:

```
CALL DC2DAT(N, IBUS, STRING, RVAL, IERR)
```

where:

Integer N	Two-terminal dc line number (input).
Integer IBUS	Converter bus number (input).
Character STRING	String indicating the dc line quantity desired (input):

Line Quantities:

’DCCUR’	dc line current, in amps.
’RDC’	dc line resistance, in ohms.
’SETVAL’	Current (amps) or power (MW) demand.
’VSCHD’	Scheduled compounded dc voltage, in kV.
’VCMOD’	Mode switch dc voltage, in kV.
’RCOMP’	Compounding resistance, in ohms.
’DELTI’	Margin in pu of desired dc power or current.
’DCVMIN’	Minimum compounded dc voltage, in kV.

Converter Quantities:

’PAC’	Apparent active power flow into the dc line, in MW.
’O_PAC’	Same as ’PAC’, in units determined by the power output option setting.
’QAC’	Apparent reactive power flow into the dc line, in Mvar.
’O_QAC’	Same as ’QAC’, in units determined by the power output option setting.
’MVA’	Apparent ac flow into the dc line, in MVA.
’O_MVA’	Same as ’MVA’, in units determined by the power output option setting.
’ACAMPS’	ac current flow into the dc line, in amps.
’PUCUR’	ac current flow into the dc line, in pu.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	’KVDC’	dc voltage, in kV.
	’ANGLE’	Converter angle, in degrees.
	’ANGMIN’	Minimum steady-state converter angle, in degrees.
	’ANGMAX’	Nominal maximum converter angle, in degrees.
	’RC’	Commutating transformer resistance per bridge, in ohms.
	’XC’	Commutating transformer reactance per bridge, in ohms.
	’EBASE’	Primary base ac voltage, in kV.
	’TR’	Transformer ratio, in pu.
	’TAP’	Tap setting, in pu.
	’TPMX’	Maximum tap setting, in pu.
	’TPMN’	Minimum tap setting, in pu.
	’STP’	Tap step, in pu.
	’XCAP’	Commutating capacitor reactance magnitude per bridge, in ohms.
Real RVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’RVAL’ returned.
	IERR = 1	dc line not found; ’RVAL’ unchanged.
	IERR = 2	Bus not a converter bus of dc line ’N’; ’RVAL’ unchanged.
	IERR = 3	Invalid value of ’STRING’; ’RVAL’ unchanged.
	IERR = 4	dc line blocked; ’RVAL’ of 0.0 returned.
	IERR = 5	No base kV at bus; ’RVAL’ returned as the per unit current.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.28 DC2DAT_2

This API is the second release of the API that returns real two-terminal dc line quantities.

Python syntax:

```
ierr, rval = dc2dat_2(name, ibus, string)
```

IPLAN syntax:

```
CALL DC2DAT_2(NAME, IBUS, STRING, RVAL, IERR)
```

Fortran syntax:

```
CALL DC2DAT_2(NAME, IBUS, STRING, RVAL, IERR)
```

where:

Character NAME	Two-terminal dc line name (input).
Integer IBUS	Converter bus number (input).
Character STRING	String indicating the dc line quantity desired (input):

Line Quantities:

’DCCUR’	dc line current, in amps.
’RDC’	dc line resistance, in ohms.
’SETVAL’	Current (amps) or power (MW) demand.
’VSCHD’	Scheduled compounded dc voltage, in kV.
’VCMOD’	Mode switch dc voltage, in kV.
’RCOMP’	Compounding resistance, in ohms.
’DELTI’	Margin in pu of desired dc power or current.
’DCVMIN’	Minimum compounded dc voltage, in kV.

Converter Quantities:

’PAC’	Apparent active power flow into the dc line, in MW.
’O_PAC’	Same as ’PAC’, in units determined by the power output option setting.
’QAC’	Apparent reactive power flow into the dc line, in Mvar.
’O_QAC’	Same as ’QAC’, in units determined by the power output option setting.
’MVA’	Apparent ac flow into the dc line, in MVA.
’O_MVA’	Same as ’MVA’, in units determined by the power output option setting.
’ACAMPS’	ac current flow into the dc line, in amps.
’PUCUR’	ac current flow into the dc line, in pu.
’KVDC’	dc voltage, in kV.
’ANGLE’	Converter angle, in degrees.
’ANGMIN’	Minimum steady-state converter angle, in degrees.
’ANGMAX’	Nominal maximum converter angle, in degrees.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	’RC’	Commutating transformer resistance per bridge, in ohms.
	’XC’	Commutating transformer reactance per bridge, in ohms.
	’EBASE’	Primary base ac voltage, in kV.
	’TR’	Transformer ratio, in pu.
	’TAP’	Tap setting, in pu.
	’TPMX’	Maximum tap setting, in pu.
	’TPMN’	Minimum tap setting, in pu.
	’STP’	Tap step, in pu.
	’XCAP’	Commutating capacitor reactance magnitude per bridge, in ohms.
Real RVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’RVAL’ returned.
	IERR = 1	dc line not found; ’RVAL’ unchanged.
	IERR = 2	Bus not a converter bus of dc line ’NAME’; ’RVAL’ unchanged.
	IERR = 3	Invalid value of ’STRING’; ’RVAL’ unchanged.
	IERR = 4	dc line blocked; ’RVAL’ of 0.0 returned.
	IERR = 5	No base kV at bus; ’RVAL’ returned as the per unit current.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.29 DC2INT

This API is obsolete. It has been replaced by the API DC2INT_2, and is implemented by a call to the DC2INT_2 API with the dc line number converted to a left-justified name.

Use this API to return integer two-terminal dc line quantities.

Python syntax:

```
ierr, ival = dc2int(n, string)
```

IPLAN syntax:

```
CALL DC2INT(N, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL DC2INT(N, STRING, IVAL, IERR)
```

where:

Integer N	Two-terminal dc line number (input).	
Character STRING	String indicating the dc line quantity desired (input):	
	’MDC’	dc line control mode.
	’RECT’	Bus number of rectifier end of dc line.
	’INV’	Bus number of inverter end of dc line.
	’METER’	Bus number of metered end of dc line.
	’NBR’	Number of bridges in series, rectifier end.
	’NBI’	Number of bridges in series, inverter end.
	’ICR’	Rectifier firing angle measuring bus.
	’ICI’	Inverter firing angle measuring bus.
	’ANSTTR’	Firing angle status, rectifier end: (-1) low; (0) regulating; (1) high.
	’ANSTTI’	Firing angle status, inverter end: (-1) low; (0) regulating (1) high.
	’TPSTTR’	Tap status, rectifier end: (-1) low; (0) regulating; (1) high; (-2) locked.
	’TPSTTI’	Tap status, inverter end: (-1) low; (0) regulating; (1) high; (-2) locked.
Integer IVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’IVAL’ returned.
	IERR = 2	dc line not found; ’IVAL’ unchanged.
	IERR = 3	Invalid value of ’STRING’; ’IVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.30 DC2INT_2

This API is the second release of the API that returns integer two-terminal dc line quantities.

Python syntax:

```
ierr, ival = dc2int_2(name, string)
```

IPLAN syntax:

```
CALL DC2INT_2(NAME, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL DC2INT_2(NAME, STRING, IVAL, IERR)
```

where:

Character NAME	Two-terminal dc line name (input).	
Character STRING	String indicating the dc line quantity desired (input):	
	'MDC'	dc line control mode.
	'RECT'	Bus number of rectifier end of dc line.
	'INV'	Bus number of inverter end of dc line.
	'METER'	Bus number of metered end of dc line.
	'NBR'	Number of bridges in series, rectifier end.
	'NBI'	Number of bridges in series, inverter end.
	'ICR'	Rectifier firing angle measuring bus.
	'ICI'	Inverter firing angle measuring bus.
	'ANSTTR'	Firing angle status, rectifier end: (-1) low; (0) regulating; (1) high.
	'ANSTTI'	Firing angle status, inverter end: (-1) low; (0) regulating (1) high.
	'TPSTTR'	Tap status, rectifier end: (-1) low; (0) regulating; (1) high; (-2) locked.
	'TPSTTI'	Tap status, inverter end: (-1) low; (0) regulating; (1) high; (-2) locked.
Integer IVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; 'IVAL' returned.
	IERR = 2	dc line not found; 'IVAL' unchanged.
	IERR = 3	Invalid value of 'STRING'; 'IVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.31 DC2MIND

Use this API to return 2-terminal dc line model starting array indices and status.

Python syntax:

```
ierr, ival = dc2mind(name, string)
```

IPLAN syntax:

```
CALL DC2MIND(NAME, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL DC2MIND(NAME, STRING, IVAL, IERR)
```

where:

Character NAME	Two-terminal dc line name (input).	
Character STRING	String indicating the 2-terminal dc line model quantity desired (input):	
	’CON’	Starting CON index.
	’STATE’	Starting STATE index.
	’VAR’	Starting VAR index.
	’ICON’	Starting ICON index.
	’NCON’	Number of CON values used by the model.
	’NSTATE’	Number of STATE values used by the model.
	’NVAR’	Number of VAR values used by the model.
	’NICON’	Number of ICON values used by the model.
	’STATUS’	Bypassed flag: 0 = bypassed; 1 = not bypassed. Does not reflect dc line status. For a model to be called during simulations, the dc line must be in service, the model’s bypassed flag must be *not bypassed*.
Integer IVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’IVAL’ returned.
	IERR = 1	dc line not in network but in model tables; ’IVAL’ returned.
	IERR = 2	dc line not in network or in model tables; ’IVAL’ unchanged.
	IERR = 4	No model in model tables; ’IVAL’ unchanged
	IERR = 5	Invalid value of ’STRING’; ’IVAL’ unchanged.
	IERR = 6	Model does not use any of this data type; ’IVAL’ = 0 returned.
	IERR = 9	Dynamics data not present in working memory; ’IVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.32 DC2MNAM

Use this API to return 2-terminal dc line model name.

Python syntax:

```
ierr, cval = dc2mnam(name)
```

IPLAN syntax:

```
CALL DC2MNAM(NAME, CVAL, IERR)
```

Fortran syntax:

```
CALL DC2MNAM(NAME, CVAL, IERR)
```

where:

Character NAME	Two-terminal dc line name (input).
Character CVAL	Model name (output).
Integer IERR	Is the error code (output):

IERR = 0	No error; ’CVAL’ returned.
IERR = 1	dc line not in network but in model tables; ’CVAL’ returned.
IERR = 2	dc line not in network or in model tables; ’CVAL’ unchanged.
IERR = 4	No model in model tables; ’CVAL’ unchanged
IERR = 9	Dynamics data not present in working memory; ’CVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.33 DCNAUXMIND

Use this API to return the starting array indices and status of auxiliary signal model associated with N-terminal dc line for the specified signal index.

Python syntax:

```
ierr, ival = dcnauxmind(name, isgndx, string)
```

IPLAN syntax:

```
CALL DCNAUXMIND(NAME, ISGNDX, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL DCNAUXMIND(NAME, ISGNDX, STRING, IVAL, IERR)
```

where:

Character NAME	N-terminal dc line name (input).	
Integer ISGNDX	Auxiliary signal index (input).	
Character STRING	String indicating the N-terminal dc line auxiliary signal model quantity desired (input):	
	’CON’	Starting CON index.
	’STATE’	Starting STATE index.
	’VAR’	Starting VAR index.
	’ICON’	Starting ICON index.
	’NCON’	Number of CON values used by the model.
	’NSTATE’	Number of STATE values used by the model.
	’NVAR’	Number of VAR values used by the model.
	’NICON’	Number of ICON values used by the model.
	’STATUS’	Bypassed flag: 0 = bypassed; 1 = not bypassed. For a model to be called during simulations, the model’s bypassed flag must be *not bypassed*.
Integer IVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’IVAL’ returned.
	IERR = 1	dc line not in network but auxiliary signal model in model tables; ’IVAL’ returned.
	IERR = 2	dc line not in network and auxiliary signal model not in model tables; ’IVAL’ unchanged.
	IERR = 4	No model in model tables for the given auxiliary signal index; ’IVAL’ unchanged
	IERR = 5	Invalid value of ’STRING’; ’IVAL’ unchanged.
	IERR = 6	Model does not use any of this data type; ’IVAL’ = 0 returned.
	IERR = 9	Dynamics data not present in working memory; ’IVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.34 DCNAUXMNAM

Use this API to return the auxiliary signal model name associated with a N-terminal dc line for the specified auxiliary signal index.

Python syntax:

```
ierr, cval = dcnauxmnam(name, isgndx)
```

IPLAN syntax:

```
CALL DCNAUXMNAM(NAME, ISGNDX, CVAL, IERR)
```

Fortran syntax:

```
CALL DCNAUXMNAM(NAME, ISGNDX, CVAL, IERR)
```

where:

Character NAME	N-terminal dc line name (input).
Integer ISGNDX	Auxiliary signal index (input).
Character CVAL	Model name (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; ’CVAL’ returned.
	IERR = 1 — dc line not in network but auxiliary signal model in model tables; ’CVAL’ returned.
	IERR = 2 — dc line not in network and auxiliary signal model not in model tables; ’CVAL’ unchanged.
	IERR = 4 — No model in model tables for the given auxiliary signal index; ’CVAL’ unchanged
	IERR = 9 — Dynamics data not present in working memory; ’CVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.35 DCNCIN

This API is obsolete. It has been replaced by the API DCNCIN_2, and is implemented by a call to the DCNCIN_2 API with the dc line number converted to a left-justified name.

Use this API to return integer multi-terminal dc line converter quantities.

Python syntax:

```
ierr, ival = dcncin(n, ibus, string)
```

IPLAN syntax:

```
CALL DCNCIN(N, IBUS, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL DCNCIN(N, IBUS, STRING, IVAL, IERR)
```

where:

Integer N	Multi-terminal dc line number (input).	
Integer IBUS	Converter bus number (input).	
Character STRING	String indicating the dc line quantity desired (input).	
	'ANSTT'	Firing angle status: (-1) low; (0) regulating; (1) high.
	'TPSTT'	Tap status: (-1) low; (0) regulating; (1) high; (-2) locked.
Integer IVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; 'IVAL' returned.
	IERR = 1	dc line not found; 'IVAL' unchanged.
	IERR = 2	Bus not a converter bus of dc line 'N'; 'IVAL' unchanged.
	IERR = 3	Invalid 'STRING'; 'IVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.36 DCNCIN_2

This API is the second release of the API that returns integer multi-terminal dc line converter quantities.

Python syntax:

```
ierr, ival = dcncin_2(name, ibus, string)
```

IPLAN syntax:

```
CALL DCNCIN_2(NAME, IBUS, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL DCNCIN_2(NAME, IBUS, STRING, IVAL, IERR)
```

where:

Character NAME	Multi-terminal dc line name (input).	
Integer IBUS	Converter bus number (input).	
Character STRING	String indicating the dc line quantity desired (input):	
	'ANSTT'	Firing angle status: (-1) low; (0) regulating; (1) high.
	'TPSTT'	Tap status: (-1) low; (0) regulating; (1) high; (-2) locked.
Integer IVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; 'IVAL' returned.
	IERR = 1	dc line not found; 'IVAL' unchanged.
	IERR = 2	Bus not a converter bus of dc line 'NAME'; 'IVAL' unchanged.
	IERR = 3	Invalid 'STRING'; 'IVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.37 DCNDAT

This API is obsolete. It has been replaced by the API DCNDAT_2, and is implemented by a call to the DCNDAT_2 API with the dc line number converted to a left-justified name.

Use this API to return real multi-terminal dc line quantities.

Python syntax:

```
ierr, rval = dcndat(n, ibus, string)
```

IPLAN syntax:

```
CALL DCNDAT(N, IBUS, STRING, RVAL, IERR)
```

Fortran syntax:

```
CALL DCNDAT(N, IBUS, STRING, RVAL, IERR)
```

where:

Integer N	Multi-terminal dc line number (input).	
Integer IBUS	Converter bus number (input).	
Character STRING	String indicating the dc line quantity desired (input):	
	Line Quantities:	
	'VCMOD'	Mode switch dc voltage in kV.
	Converter Quantities:	
	'PAC'	Active power into dc line at converter bus, in MW.
	'O_PAC'	Same as 'PAC', in units determined by the power output option setting.
	'QAC'	Reactive power into dc line at converter bus, in Mvar.
	'O_QAC'	Same as 'QAC', in units determined by the power output option setting.
	'KVDC'	dc voltage at converter bus end.
	'DCCUR'	dc current.
	'SETVAL'	Converter setpoint.
	'ANGLE'	Converter angle.
	'TAP'	Tap setting.
	'ANGMIN'	Minimum steady-state ALPHA or GAMMA angle, in degrees.
	'ANGMAX'	Nominal maximum ALPHA or GAMMA angle, in degrees.
	'RC'	Commutating resistance per bridge, in ohms.
	'XC'	Commutating reactance per bridge, in ohms.
	'MVA'	Apparent ac flow into the dc line, in MVA.
	'O_MVA'	Same as 'MVA', in units determined by the power output option setting.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	’ACAMPS’	ac current flow into the dc line, in amps.
	’PUCUR’	ac current flow into the dc line, in pu.
	'EBASE'	Primary base ac voltage, in kV.
	'TR'	Transformer ratio, in pu
	'TPMX'	Maximum tap setting, in pu.
	'TPMN'	Minimum tap setting, in pu.
	'STP'	Tap step, in pu.
	'DELTI'	Rectifier margin in pu of desired dc power or current.
Real RVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’RVAL’ returned.
	IERR = 1	dc line not found; ’RVAL’ unchanged.
	IERR = 2	Bus not a converter bus of dc line ’N’; ’RVAL’ unchanged.
	IERR = 3	Invalid value of ’STRING’; ’RVAL’ unchanged.
	IERR = 4	dc line blocked; ’RVAL’ of 0.0 returned.
	IERR = 5	No base kV at bus; ’RVAL’ returned as the per unit current.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.38 DCNDAT_2

This API is the second release of the API that returns real multi-terminal dc line quantities.

Python syntax:

```
ierr, rval = dcndat_2(name, ibus, string)
```

IPLAN syntax:

```
CALL DCNDAT_2(NAME, IBUS, STRING, RVAL, IERR)
```

Fortran syntax:

```
CALL DCNDAT_2(NAME, IBUS, STRING, RVAL, IERR)
```

where:

Character NAME	Multi-terminal dc line name (input).	
Integer IBUS	Converter bus number (input).	
Character STRING	String indicating the dc line quantity desired (input):	
	Line Quantities:	
	’VCMOD’	Mode switch dc voltage in kV.
	Converter Quantities:	
	’PAC’	Active power into dc line at converter bus, in MW.
	’O_PAC’	Same as ’PAC’, in units determined by the power output option setting.
	’QAC’	Reactive power into dc line at converter bus, in Mvar.
	’O_QAC’	Same as ’QAC’, in units determined by the power output option setting.
	’KVDC’	dc voltage at converter bus end.
	’DCCUR’	dc current.
	’SETVAL’	Converter setpoint.
	’ANGLE’	Converter angle.
	’TAP’	Tap setting.
	’ANGMIN’	Minimum steady-state ALPHA or GAMMA angle, in degrees.
	’ANGMAX’	Nominal maximum ALPHA or GAMMA angle, in degrees.
	’RC’	Commutating resistance per bridge, in ohms.
	’XC’	Commutating reactance per bridge, in ohms.
	’MVA’	Apparent ac flow into the dc line, in MVA.
	’O_MVA’	Same as ’MVA’, in units determined by the power output option setting.
	’ACAMPS’	ac current flow into the dc line, in amps.
	’PUCUR’	ac current flow into the dc line, in pu.
	'EBASE'	Primary base ac voltage, in kV.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	'TR'	Transformer ratio, in pu.
	'TPMX'	Maximum tap setting, in pu.
	'TPMN'	Minimum tap setting, in pu.
	'STP'	Tap step, in pu.
	'DELTI'	Rectifier margin in pu of desired dc power or current.
Real RVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’RVAL’ returned.
	IERR = 1	dc line not found; ’RVAL’ unchanged.
	IERR = 2	Bus not a converter bus of dc line ’NAME’; ’RVAL’ unchanged.
	IERR = 3	Invalid value of ’STRING’; ’RVAL’ unchanged.
	IERR = 4	dc line blocked; ’RVAL’ of 0.0 returned.
	IERR = 5	No base kV at bus; ’RVAL’ returned as the per unit current.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.39 DCNINT

This API is obsolete. It has been replaced by the API DCNINT_2, and is implemented by a call to the DCNINT_2 API with the dc line number converted to a left-justified name.

Use this API to return integer multi-terminal dc line quantities.

Python syntax:

```
ierr, ival = dcnint(n, string)
```

IPLAN syntax:

```
CALL DCNINT(N, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL DCNINT(N, STRING, IVAL, IERR)
```

where:

Integer N	Multi-terminal dc line number (input).
Character STRING	String indicating the dc line quantity desired (input). 'MDC' Control mode.
Integer IVAL	Value indicated by STRING (output).
Integer IERR	Is the error code (output): IERR = 0 No error; 'IVAL' returned. IERR = 2 dc line not found; 'IVAL' unchanged. IERR = 3 Invalid value of 'STRING'; 'IVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.40 DCNINT_2

This API is the second release of the API that returns integer multi-terminal dc line quantities.

Python syntax:

```
ierr, ival = dcnint_2(name, string)
```

IPLAN syntax:

```
CALL DCNINT_2(NAME, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL DCNINT_2(NAME, STRING, IVAL, IERR)
```

where:

Character NAME	Multi-terminal dc line name (input).
Character STRING	String indicating the dc line quantity desired (input): 'MDC' Control mode.
Integer IVAL	Value indicated by STRING (output).
Integer IERR	Is the error code (output): IERR = 0 No error; 'IVAL' returned. IERR = 2 dc line not found; 'IVAL' unchanged. IERR = 3 Invalid value of 'STRING'; 'IVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.41 DCNMIND

Use this API to return N-terminal dc line model starting array indices and status.

Python syntax:

```
ierr, ival = dcnmind(name, string)
```

IPLAN syntax:

```
CALL DCNMIND(NAME, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL DCNMIND(NAME, STRING, IVAL, IERR)
```

where:

Character NAME	N-terminal dc line name (input).	
Character STRING	String indicating the N-terminal dc line model quantity desired (input):	
	’CON’	Starting CON index.
	’STATE’	Starting STATE index.
	’VAR’	Starting VAR index.
	’ICON’	Starting ICON index.
	’NCON’	Number of CON values used by the model.
	’NSTATE’	Number of STATE values used by the model.
	’NVAR’	Number of VAR values used by the model.
	’NICON’	Number of ICON values used by the model.
	’STATUS’	Bypassed flag: 0 = bypassed; 1 = not bypassed. Does not reflect dc line status. For a model to be called during simulations, the dc line must be in service, the model’s bypassed flag must be *not bypassed*.
Integer IVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’IVAL’ returned.
	IERR = 1	dc line not in network but in model tables; ’IVAL’ returned.
	IERR = 2	dc line not in network or in model tables; ’IVAL’ unchanged.
	IERR = 4	No model in model tables; ’IVAL’ unchanged
	IERR = 5	Invalid value of ’STRING’; ’IVAL’ unchanged.
	IERR = 6	Model does not use any of this data type; ’IVAL’ = 0 returned.
	IERR = 9	Dynamics data not present in working memory; ’IVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.42 DCNMNAM

Use this API to return N-terminal dc line model name.

Python syntax:

```
ierr, cval = dcnmnam(name)
```

IPLAN syntax:

```
CALL DCNMNAM(NAME, CVAL, IERR)
```

Fortran syntax:

```
CALL DCNMNAM(NAME, CVAL, IERR)
```

where:

Character NAME	N-terminal dc line name (input).
Character CVAL	Model name (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; ’CVAL’ returned.
	IERR = 1 — dc line not in network but in model tables; ’CVAL’ returned.
	IERR = 2 — dc line not in network or in model tables; ’CVAL’ unchanged.
	IERR = 4 — No model in model tables; ’CVAL’ unchanged.
	IERR = 9 — Dynamics data not present in working memory; ’CVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.43 DSCVAL

Use this API to return dynamics character array values.

Python syntax:

```
ierr, cval = dscval(string, indx)
```

IPLAN syntax:

```
CALL DSCVAL(STRING, INDX, CVAL, IERR)
```

Fortran syntax:

```
CALL DSCVAL(STRING, INDX, CVAL, IERR)
```

where:

Character STRING	String indicating the dynamics quantity desired (input) 'ICON' Character model parameters.
Integer INDX	Index to the dynamics character array (input).
Character CVAL	Character dynamics array value (output).
Integer IERR	Is the error code (output): IERR = 0 No error; 'CVAL' returned. IERR = 1 Invalid value of 'STRING'; 'CVAL' unchanged. IERR = 2 'INDX' invalid; 'CVAL' unchanged. IERR = 3 'INDX' points to integer icon; 'CVAL' unchanged. IERR = 9 Dynamics data not present in working memory; 'CVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.44 DSIVAL

Use this API to return dynamics integer array values.

Python syntax:

```
ierr, ival = dsival(string, indx)
```

IPLAN syntax:

```
CALL DSIVAL(STRING, INDX, IVAL, IERR)
```

Fortran syntax:

```
CALL DSIVAL(STRING, INDX, IVAL, IERR)
```

where:

Character STRING	String indicating the dynamics quantity desired (input): 'ICON' Integer model parameters.
Integer INDX	Index to the dynamics integer array (input).
Integer IVAL	Integer dynamics array value (output).
Integer IERR	Is the error code (output): IERR = 0 No error; 'IVAL' returned. IERR = 1 Invalid value of 'STRING'; 'IVAL' unchanged. IERR = 2 'INDX' invalid; 'IVAL' unchanged. IERR = 3 'INDX' points to character icon; 'IVAL' unchanged. IERR = 9 Dynamics data not present in working memory; 'IVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.45 DSRVAL

Use this API to return dynamics real values.

Python syntax:

```
ierr, rval = dsrval(string, indx)
```

IPLAN syntax:

```
CALL DSRVAL(STRING, INDX, RVAL, IERR)
```

Fortran syntax:

```
CALL DSRVAL(STRING, INDX, RVAL, IERR)
```

where:

Character STRING	String indicating the dynamics quantity desired (input):	
	’TIME’	Simulation time.
	’DELT’	Simulation time step.
	’STATE’	State variable values.
	’CON’	Real model parameters.
	’VAR’	Real model variable values.
Integer INDX	Index to dynamics array (input). Ignored for ’TIME’ and ’DELT’.	
Real RVAL	Real dynamics quantity or array value (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’RVAL’ returned.
	IERR = 1	Invalid value of ’STRING’; ’RVAL’ unchanged.
	IERR = 2	’INDX’ invalid; ’RVAL’ unchanged.
	IERR = 9	Dynamics data not present in working memory; ’RVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.46 FCDAUXMIND

Use this API to return the starting array indices and status of auxiliary signal model associated with FACTS device for the specified signal index.

Python syntax:

```
ierr, ival = fcdauxmind(name, isgndx, string)
```

IPLAN syntax:

```
CALL FCDAUXMIND(NAME, ISGNDX, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL FCDAUXMIND(NAME, ISGNDX, STRING, IVAL, IERR)
```

where:

Character NAME	FACTS device name (input).	
Integer ISGNDX	Auxiliary signal index (input).	
Character STRING	String indicating the FACTS device auxiliary signal model quantity desired (input):	
	’CON’	Starting CON index.
	’STATE’	Starting STATE index.
	’VAR’	Starting VAR index.
	’ICON’	Starting ICON index.
	’NCON’	Number of CON values used by the model.
	’NSTATE’	Number of STATE values used by the model.
	’NVAR’	Number of VAR values used by the model.
	’NICON’	Number of ICON values used by the model.
	’STATUS’	Bypassed flag: 0 = bypassed; 1 = not bypassed. For a model to be called during simulations, the model’s bypassed flag must be *not bypassed*.
Integer IVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’IVAL’ returned.
	IERR = 1	FACTS device not in network but auxiliary signal model in model tables; ’IVAL’ returned.
	IERR = 2	FACTS device not in network and auxiliary signal model not in model tables; ’IVAL’ unchanged.
	IERR = 4	No model in model tables for the given auxiliary signal index; ’IVAL’ unchanged
	IERR = 5	Invalid value of ’STRING’; ’IVAL’ unchanged.
	IERR = 6	Model does not use any of this data type; ’IVAL’ = 0 returned.
	IERR = 9	Dynamics data not present in working memory; ’IVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.47 FCDAUXMNAM

Use this API to return the auxiliary signal model name associated with a FACTS device for the specified auxiliary signal index.

Python syntax:

```
ierr, cval = fcdauxmnam(name, isgndx)
```

IPLAN syntax:

```
CALL FCDAUXMNAM(NAME, ISGNDX, CVAL, IERR)
```

Fortran syntax:

```
CALL FCDAUXMNAM(NAME, ISGNDX, CVAL, IERR)
```

where:

Character NAME	FACTS device name (input).	
Integer ISGNDX	Auxiliary signal index (input).	
Character CVAL	Model name (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’CVAL’ returned.
	IERR = 1	FACTS device not in network but auxiliary signal model in model tables; ’CVAL’ returned.
	IERR = 2	FACTS device not in network and auxiliary signal model not in model tables; ’CVAL’ unchanged.
	IERR = 4	No model in model tables for the given auxiliary signal index; ’CVAL’ unchanged
	IERR = 9	Dynamics data not present in working memory; ’CVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.48 FCDDAT

This API is obsolete. It has been replaced by the API FCDDAT_2, and is implemented by a call to the FCDDAT_2 API with the FACTS device number converted to a left-justified name.

Use this API to return FACTS device real quantities.

Python syntax:

```
ierr, rval = fcddat(n, string)
```

IPLAN syntax:

```
CALL FCDDAT(N, STRING, RVAL, IERR)
```

Fortran syntax:

```
CALL FCDDAT(N, STRING, RVAL, IERR)
```

where:

Integer N	FACTS device number (input).	
Character STRING	String indicating the FACTS device quantity desired (input):	
	’PDES’	Desired active power flow at terminal end bus, in MW.
	’O_PDES’	Same as ’PDES’, in units determined by the power output option setting.
	’QDES’	Desired reactive power flow at terminal end bus, in Mvar.
	’O_QDES’	Same as ’QDES’, in units determined by the power output option setting.
	’VSET’	Voltage setpoint at sending end bus.
	’SHMX’	Maximum shunt current at sending end bus, in MVA at unity voltage.
	’O_SHMX’	Same as ’SHMX’, in units determined by the power output option setting.
	’TRMX’	Maximum bridge real power transfer, in MW.
	’O_TRMX’	Same as ’TRMX’, in units determined by the power output option setting.
	’VTMN’	Minimum voltage at terminal end bus, in pu.
	’VTMX’	Maximum voltage at terminal end bus, in pu.
	’VSMX’	Maximum series voltage, in pu.
	’IMX’	Maximum series current, in MVA at unity voltage.
	’O_IMX’	Same as ’IMX’, in units determined by the power output option setting.
	’LINX’	Reactance of the dummy series element, in pu.
	’PSEND’	Active power into series element at sending end bus, in MW.
	’O_PSEND’	Same as ’PSEND’, in units determined by the power output option setting.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

'QSEND'	Reactive power into series element at sending end bus, in Mvar.
'O_QSEND'	Same as 'QSEND', in units determined by the power output option setting.
'PTERM'	Active power into series element at terminal end bus, in MW.
'O_PTERM'	Same as 'PTERM', in units determined by the power output option setting.
'QTERM'	Reactive power into series element at terminal end bus, in Mvar.
'O_QTERM'	Same as 'QTERM', in units determined by the power output option setting.
'PBRDG'	Active bridge power exchange, in MW.
'O_PBRDG'	Same as 'PBRDG', in units determined by the power output option setting.
'QSHNT'	Reactive power into shunt element, in Mvar.
'O_QSHNT'	Same as 'QSHNT', in units determined by the power output option setting.
'VSERES'	Series voltage magnitude.
'MVASND'	Flow into series element at sending bus end, in MVA.
'O_MVASND'	Same as 'MVASND', in units determined by the power output option setting.
'MVATRM'	Flow into series element at terminal bus end, in MVA.
'O_MVATRM'	Same as 'MVATRM', in units determined by the power output option setting.
'MVASHN'	Flow into shunt element, in MVA.
'O_MVASHN'	Same as 'MVASHN', in units determined by the power output option setting.
'IPUSND'	Current flow into series element at sending bus end, in pu.
'IPUTRM'	Current flow into series element at terminal bus end, in pu.
'IPUSHN'	Current flow into shunt element, in pu.
'AMPSND'	Current flow into series element at sending bus end, in amps.
'AMPTRM'	Current flow into series element at terminal bus end, in amps.
'AMPSHN'	Current flow into shunt element, in amps.
'PCTSND'	Percent series current loading at sending bus end.
'PCTTRM'	Percent series current loading at terminal bus end.
'PCTSHN'	Percent shunt current loading.
'PCTPBR'	Percent bridge real power loading.
'RMPCT'	Percent of contributed vars.

Real RVAL Value indicated by STRING (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output):	
	IERR = 0	No error; 'RVAL' returned.
	IERR = 2	FACTS device 'N' not found; 'RVAL' unchanged.
	IERR = 3	Invalid value of 'STRING'; 'RVAL' unchanged.
	IERR = 4	No base kV at bus; 'RVAL' returned = per unit current.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.49 FCDDAT_2

This API is the second release of the API that returns FACTS device real quantities.

Python syntax:

```
ierr, rval = fcddat_2(name, string)
```

IPLAN syntax:

```
CALL FCDDAT_2(NAME, STRING, RVAL, IERR)
```

Fortran syntax:

```
CALL FCDDAT_2(NAME, STRING, RVAL, IERR)
```

where:

Character NAME	FACTS device name (input).	
Character STRING	String indicating the FACTS device quantity desired (input):	
	'PDES'	Desired active power flow at terminal end bus, in MW.
	'O_PDES'	Same as 'PDES', in units determined by the power output option setting.
	'QDES'	Desired reactive power flow at terminal end bus, in Mvar.
	'O_QDES'	Same as 'QDES', in units determined by the power output option setting.
	'VSET'	Voltage setpoint at sending end bus.
	'SHMX'	Maximum shunt current at sending end bus, in MVA at unity voltage.
	'O_SHMX'	Same as 'SHMX', in units determined by the power output option setting.
	'TRMX'	Maximum bridge real power transfer, in MW.
	'O_TRMX'	Same as 'TRMX', in units determined by the power output option setting.
	'VTMN'	Minimum voltage at terminal end bus, in pu.
	'VTMX'	Maximum voltage at terminal end bus, in pu.
	'VSMX'	Maximum series voltage, in pu.
	'IMX'	Maximum series current, in MVA at unity voltage.
	'O_IMX'	Same as 'IMX', in units determined by the power output option setting.
	'LINX'	Reactance of the dummy series element, in pu.
	'PSEND'	Active power into series element at sending end bus, in MW.
	'O_PSEND'	Same as 'PSEND', in units determined by the power output option setting.
	'QSEND'	Reactive power into series element at sending end bus, in Mvar.
	'O_QSEND'	Same as 'QSEND', in units determined by the power output option setting.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

'PTERM'	Active power into series element at terminal end bus, in MW.
'O_PTERM'	Same as 'PTERM', in units determined by the power output option setting.
'QTERM'	Reactive power into series element at terminal end bus, in Mvar.
'O_QTERM'	Same as 'QTERM', in units determined by the power output option setting.
'PBRDG'	Active bridge power exchange, in MW.
'O_PBRDG'	Same as 'PBRDG', in units determined by the power output option setting.
'QSHNT'	Reactive power into shunt element, in Mvar.
'O_QSHNT'	Same as 'QSHNT', in units determined by the power output option setting.
'VSERES'	Series voltage magnitude.
'MVASND'	Flow into series element at sending bus end, in MVA.
'O_MVASND'	Same as 'MVASND', in units determined by the power output option setting.
'MVATRM'	Flow into series element at terminal bus end, in MVA.
'O_MVATRM'	Same as 'MVATRM', in units determined by the power output option setting.
'MVASHN'	Flow into shunt element, in MVA.
'O_MVASHN'	Same as 'MVASHN', in units determined by the power output option setting.
'IPUSND'	Current flow into series element at sending bus end, in pu.
'IPUTRM'	Current flow into series element at terminal bus end, in pu.
'IPUSHN'	Current flow into shunt element, in pu.
'AMPSND'	Current flow into series element at sending bus end, in amps.
'AMPTRM'	Current flow into series element at terminal bus end, in amps.
'AMPSHN'	Current flow into shunt element, in amps.
'PCTSND'	Percent series current loading at sending bus end.
'PCTTRM'	Percent series current loading at terminal bus end.
'PCTSHN'	Percent shunt current loading.
'PCTPBR'	Percent bridge real power loading.
'RMPCT'	Percent of contributed vars.

Real RVAL Value indicated by STRING (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’RVAL’ returned.
	IERR = 2	FACTS device ’NAME’ not found; ’RVAL’ unchanged.
	IERR = 3	Invalid value of ’STRING’; ’RVAL’ unchanged.
	IERR = 4	No base kV at bus; ’RVAL’ returned = per unit current.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.50 FCDINT

This API is obsolete. It has been replaced by the API FCDINT_2, and is implemented by a call to the FCDINT_2 API with the FACTS device number converted to a left-justified name.

Use this API to return FACTS device integer quantities.

Python syntax:

```
ierr, ival = fcdint(n, string)
```

IPLAN syntax:

```
CALL FCDINT(N, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL FCDINT(N, STRING, IVAL, IERR)
```

where:

Integer N	FACTS device number (input).	
Character STRING	String indicating the FACTS device quantity desired (input):	
	’SEND’	Sending end bus number.
	’TERM’	Terminal end bus number.
	’REMOTE’	Bus number of remote bus or zero for local control.
	’OWNER’	Owner number.
	’MODE’	Control mode (negative for blocked during solution).
	’STATVT’	Terminal end voltage condition code returns:
		0 Terminal end voltage not constrained.
		1 Holding terminal end voltage high limit.
		-1 Holding terminal end voltage low limit.
	’STATVS’	Series voltage limit code returns.
		0 Series voltage not constrained.
		1 Holding series voltage limit.
	’STATBR’	Bridge active power transfer limit code returns.
		0 Bridge real power transfer not constrained.
		1 Holding bridge real power transfer limit.
Integer IVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’IVAL’ returned.
	IERR = 2	FACTS device ’N’ not found; ’IVAL’ unchanged.
	IERR = 3	Invalid value of ’STRING’; ’IVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.51 FCDINT_2

This API is the second release of the API that returns FACTS device integer quantities.

Python syntax:

```
ierr, ival = fcdint_2(name, string)
```

IPLAN syntax:

```
CALL FCDINT_2(NAME, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL FCDINT_2(NAME, STRING, IVAL, IERR)
```

where:

Character NAME	FACTS device name (input).	
Character STRING	String indicating the FACTS device quantity desired (input):	
	’SEND’	Sending end bus number.
	’TERM’	Terminal end bus number.
	’REMOTE’	Bus number of remote bus or zero for local control.
	’OWNER’	Owner number.
	’MODE’	Control mode (negative for blocked during solution).
	’STATVT’	Terminal end voltage condition code returns:
		0 Terminal end voltage not constrained.
		1 Holding terminal end voltage high limit.
		-1 Holding terminal end voltage low limit.
	’STATVS’	Series voltage limit code returns:
		0 Series voltage not constrained.
		1 Holding series voltage limit.
	’STATBR’	Bridge active power transfer limit code returns:
		0 Bridge real power transfer not constrained.
		1 Holding bridge real power transfer limit.
Integer IVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’IVAL’ returned.
	IERR = 2	FACTS device ’NAME’ not found; ’IVAL’ unchanged.
	IERR = 3	Invalid value of ’STRING’; ’IVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.52 FCDMIND

Use this API to return FACTS device model starting array indices and status.

Python syntax:

```
ierr, ival = fcdmind(name, string)
```

IPLAN syntax:

```
CALL FCDMIND(NAME, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL FCDMIND(NAME, STRING, IVAL, IERR)
```

where:

Character NAME	FACTS device name (input).	
Character STRING	String indicating the FACTS device model quantity desired (input):	
	'CON'	Starting CON index.
	'STATE'	Starting STATE index.
	'VAR'	Starting VAR index.
	'ICON'	Starting ICON index.
	'NCON'	Number of CON values used by the model.
	'NSTATE'	Number of STATE values used by the model.
	'NVAR'	Number of VAR values used by the model.
	'NICON'	Number of ICON values used by the model.
	'STATUS'	Bypassed flag: 0 = bypassed; 1 = not bypassed. Does not reflect status of FACTS device. For a model to be called during simulations, the FACTS device must be in service, the model's bypassed flag must be *not bypassed*.
Integer IVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; 'IVAL' returned.
	IERR = 1	FACTS device not in network but in model tables; 'IVAL' returned.
	IERR = 2	FACTS device not in network or in model tables; 'IVAL' unchanged.
	IERR = 4	No model in model tables; 'IVAL' unchanged
	IERR = 5	Invalid value of 'STRING'; 'IVAL' unchanged.
	IERR = 6	Model does not use any of this data type; 'IVAL' = 0 returned.
	IERR = 9	Dynamics data not present in working memory; 'IVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.53 FCDMNAM

Use this API to return FACTS device model name.

Python syntax:

```
ierr, cval = fcdmnam(name)
```

IPLAN syntax:

```
CALL FCDMNAM(NAME, CVAL, IERR)
```

Fortran syntax:

```
CALL FCDMNAM(NAME, CVAL, IERR)
```

where:

Character NAME	FACTS device name (input).	
Character CVAL	Model name (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’CVAL’ returned.
	IERR = 1	FACTS device not in network but in model tables; ’CVAL’ returned.
	IERR = 2	FACTS device not in network or in model tables; ’CVAL’ unchanged.
	IERR = 4	No model in model tables; ’CVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.54 FXSDT1

Use this API to return the magnitude of a specified fixed bus shunt.

Python syntax:

```
ierr, rval = fxsdt1(ibus, id ,string)
```

IPLAN syntax:

```
CALL FXSDT1(IBUS, ID, STRING, RVAL, IERR)
```

Fortran syntax:

```
CALL FXSDT1(IBUS, ID, STRING, RVAL, IERR)
```

where:

Integer IBUS	Bus number (input).
Character ID	Shunt identifier (input).
Character STRING	String indicating the units to be used (input):
	'ACT' — Actual shunt. 'RVAL' returned in MVA.
	'O_ACT' — Actual shunt. 'RVAL' in units determined by the power output option setting.
	'NOM' — Nominal shunt (at 1.0 pu voltage). 'RVAL' returned in MVA.
	'O_NOM' — Nominal shunt (at 1.0 pu voltage). 'RVAL' in units determined by the power output option setting.
	'PQZERO' — Nominal zero sequence shunt (at 1.0 pu voltage). 'RVAL' returned in per unit.
	'PQZ' — Nominal zero sequence shunt (at 1.0 pu voltage). 'RVAL' returned in MVA.
	'O_PQZ' — Nominal zero sequence shunt (at 1.0 pu voltage). 'RVAL' returned in units determined by the power output option setting.
Real RVAL	Magnitude of the complex shunt in the units indicated by STRING (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; 'RVAL' returned.
	IERR = 1 — Bus not found; 'RVAL' unchanged.
	IERR = 2 — Shunt not found; 'RVAL' unchanged.
	IERR = 3 — Bus type code is not 1, 2 or 3; 'RVAL' returned.
	IERR = 4 — Shunt out-of-service; 'RVAL' returned.
	IERR = 5 — Invalid value of 'STRING'; 'RVAL' unchanged.
	IERR = 6 — Sequence data not in case for 'STRING1' = 'PQZERO', 'PQZ' or 'O_PQZ'; 'RVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.55 FXSDT2

Use this API to return complex fixed bus shunt.

Python syntax:

```
ierr, cmpval = fxsdt2(ibus, id ,string)
```

IPLAN syntax:

```
CALL FXSDT2(IBUS, ID, STRING, P, Q, IERR)
```

Fortran syntax:

```
CALL FXSDT2(IBUS, ID, STRING, CMPVAL, IERR)
```

where:

Integer IBUS	Bus number (input).	
Character ID	Shunt identifier (input).	
Character STRING	String indicating the units to be used (input):	
	'ACT'	Actual shunt. 'CMPVAL' returned in MW, Mvar.
	'O_ACT'	Actual shunt. 'CMPVAL' in units determined by the power output option setting.
	'NOM'	Nominal shunt (at 1.0 pu voltage). 'CMPVAL' returned in MW, Mvar.
	'O_NOM'	Nominal shunt (at 1.0 pu voltage). 'CMPVAL' in units determined by the power output option setting.
	'PQZERO'	Nominal zero sequence shunt (at 1.0 pu voltage). 'CMPVAL' returned in per unit.
	'PQZ'	Nominal zero sequence shunt (at 1.0 pu voltage). 'CMPVAL' returned in MW, Mvar.
	'O_PQZ'	Nominal zero sequence shunt (at 1.0 pu voltage). 'CMPVAL' returned in units determined by the power output option setting.
Real P	Active power component of the complex shunt in the units indicated by STRING (output).	
Real Q	Reactive power component of the complex shunt in the units indicated by STRING (output).	
Complex CMPVAL	Complex shunt in the units indicated by STRING (output).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output):

IERR = 0	No error; ’P’ and ’Q’ or ’CMPVAL’ returned.
IERR = 1	Bus not found; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
IERR = 2	Shunt not found; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
IERR = 3	Bus type code is not 1, 2 or 3; ’P’ and ’Q’ or ’CMPVAL’ returned.
IERR = 4	Shunt out-of-service; ’P’ and ’Q’ or ’CMPVAL’ returned.
IERR = 5	Invalid value of ’STRING’; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
IERR = 6	Sequence data not in case for ’STRING1’ = ’PQZERO’, ’PQZ’ or ’O_PQZ’; ’P’ and ’Q’ or ’CMPVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.56 FXSINT

Use this API to return integer quantities of the specified fixed bus shunt.

Python syntax:

```
ierr, ival = fxsint(ibus, id, string)
```

IPLAN syntax:

```
CALL FXSINT(IBUS, ID, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL FXSINT(IBUS, ID, STRING, IVAL, IERR)
```

where:

Integer IBUS	Bus number (input).
Character ID	Shunt identifier (input).
Character STRING	String indicating the shunt quantity desired (input): 'STATUS' Shunt status; 1 (one) for in-service, 0 (zero) for out-of-service.
Integer IVAL	Value indicated by STRING (output).
Integer IERR	Is the error code (output): IERR = 0 No error; 'IVAL' returned. IERR = 1 Bus not found; 'IVAL' unchanged. IERR = 2 Shunt not found; 'IVAL' unchanged. IERR = 3 Bus type code is not 1, 2 or 3; 'IVAL' returned. IERR = 4 Shunt out-of-service; 'IVAL' returned. IERR = 5 Invalid value of 'STRING'; 'IVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.57 GENCNV

Use this API to return a flag indicating whether generators are converted.

Python syntax:

```
icode = gencnv()
```

IPLAN syntax:

```
CALL GENCNV(ICODE)
```

Fortran syntax:

```
CALL GENCNV(ICODE)
```

where:

Integer ICODE	0 if generators are converted, otherwise 1 (output). Stand-alone simulator returns a 1.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.58 GENDAT

Use this API to return plant total power output.

Python syntax:

```
ierr, cmpval = gendat(ibus)
```

IPLAN syntax:

```
CALL GENDAT(IBUS, P, Q, IERR)
```

Fortran syntax:

```
CALL GENDAT(IBUS, CMPVAL, IERR)
```

where:

Integer IBUS	Bus number (input).
Real P	Real component of the complex generation (output).
Real Q	Reactive component of the complex generation (output).
Complex CMPVAL	Complex generation (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; ’P’ and ’Q’ or ’CMPVAL’ returned.
	IERR = 1 — Bus not found; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
	IERR = 2 — Bus type code is not 2 or 3; ’P’ and ’Q’ or ’CMPVAL’ returned.
	IERR = 3 — No machines at bus; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
	IERR = 4 — No online machines; ’P’ and ’Q’ of 0.0 or ’CMPVAL’ of (0.0,0.0) returned.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.59 GENDT1

Returns total plant power output in MVA.

Python syntax:

```
ierr, rval = gendt1(ibus)
```

IPLAN syntax:

```
CALL GENDT1(IBUS, RVAL, IERR)
```

Fortran syntax:

```
CALL GENDT1(IBUS, RVAL, IERR)
```

where:

Integer IBUS	Bus number (input).
Real RVAL	Magnitude of total plant generation (output).
Integer IERR	Is the error code (output):
	IERR = 0 No error; 'RVAL' returned.
	IERR = 1 Bus not found; 'RVAL' unchanged.
	IERR = 2 Bus type code is not 2 or 3; 'RVAL' returned.
	IERR = 3 No machines at bus; 'RVAL' unchanged.
	IERR = 4 No online machines; 'RVAL' of 0.0 returned.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.60 GET_MSTATE

Use this API to return the MSTATE value. MSTATE is meaningful in MODEs one through four and indicates the type of simulation being run by activities STRT/RUN, MSTR/MRUN, ESTR/ERUN, GSTR/GRUN or ASTR.

Python syntax:

```
n = get_mstate()
```

IPLAN syntax:

```
CALL GET_MSTATE(N)
```

Fortran syntax:

```
CALL GET_MSTATE(N)
```

where:

Integer N	MSTATE value (output):	
	N = -1	No dynamics data is present.
	N = 0	Standard state-space dynamic simulation via activities STRT and RUN, or activity ASTR.
	N = 1	Excitation system response ratio test via activities ESTR and ERUN. Models other than generator and excitation system models are not called.
	N = 2	Excitation system open circuit step response test via activities ESTR and ERUN. Models other than generator and excitation system models are not called.
	N = 3	Governor response test via activities GSTR and GRUN. Models other than generator and turbine governor models are not called.
	N = 4	Extended term dynamic simulation via activities MSTR and MRUN.
	N = 5	Dynamics data is present but no initialization activity has been successfully executed.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.61 GNECHR

Use this API routine to return character quantities of a specified GNE device.

Python syntax:

```
ierr, cval = gnechr(name, string)
```

IPLAN syntax:

```
CALL GNECHR(NAME, STRING, CVAL, IERR)
```

Fortran syntax:

```
CALL GNECHR(NAME, STRING, CVAL, IERR)
```

where:

Character NAME	GNE device name (input).	
Character STRING	String indicating the GNE device quantity desired (input):	
	'MODEL'	Name of the BOSL model used by this GNE device.
	'CHARn'	n^{th} two-character identifier required by this model; "n" must be between 1 and the number of character identifiers required by the model to calculate its inputs.
Character*16 CVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; 'CVAL' returned.
	IERR = 1	GNE device not found; 'CVAL' unchanged.
	IERR = 2	Invalid value of 'STRING'; 'CVAL' unchanged.
	IERR = 3	For 'STRING' value 'CHARn', n is not valid for this model; 'CVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.62 GNEDAT

Use this API routine to return real quantities of a specified GNE device.

Python syntax:

```
ierr, rval = gnedat(name, string)
```

IPLAN syntax:

```
CALL GNEDAT(NAME, STRING, RVAL, IERR)
```

Fortran syntax:

```
CALL GNEDAT(NAME, STRING, RVAL, IERR)
```

where:

Character NAME	GNE device name (input).	
Character STRING	String indicating the GNE device quantity desired (input):	
	'REALn'	n^{th} real data item required by this model; "n" must be between 1 and the number of real data items used by the model.
	'Pn'	active power in MW flowing into the GNE device at terminal n; "n" must be between 1 and the number of terminals used by the model.
	'Qn'	reactive power in MVAR flowing into the GNE device at terminal n; "n" must be between 1 and the number of terminals used by the model.
Real RVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; 'RVAL' returned.
	IERR = 1	GNE device not found; 'RVAL' unchanged.
	IERR = 2	Invalid value of 'STRING'; 'RVAL' unchanged.
	IERR = 3	For 'STRING' value 'REALn', n is not valid for this model; 'RVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.63 GNEINT

Use this API routine to return integer quantities of a specified GNE device.

Python syntax:

```
ierr, ival = gneint(name, string)
```

IPLAN syntax:

```
CALL GNEINT(NAME, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL GNEINT(NAME, STRING, IVAL, IERR)
```

where:

Character NAME	GNE device name (input).	
Character STRING	String indicating the GNE device quantity desired (input):	
	'NTERM'	Number of terminals used by this model (1 or 2).
	'BUS1'	Number of the first (or only) bus to which this model is connected.
	'BUS2'	Number of the second bus to which this model is connected; 0 is returned for a 1 terminal model.
	'STATUS'	Device status.
	'OWNER'	Owner to which the device is assigned.
	'NMETER'	Number of the non-metered end bus; used for GNE devices with more than one terminal.
	'NREAL'	Number of floating point data items required by this model.
	'NINTG'	Number of buses required by the model to calculate its inputs.
	'NCHAR'	Number of number of two-character identifiers required by the model to calculate its inputs.
	'INTGn'	n^{th} bus number required by this model; "n" must be between 1 and the number of buses required by the model to calculate its inputs.
Integer IVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; 'IVAL' returned.
	IERR = 1	GNE device not found; 'IVAL' unchanged.
	IERR = 2	Invalid value of 'STRING'; 'IVAL' unchanged.
	IERR = 3	For 'STRING' value 'INTGn', n is not valid for this model; 'IVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.64 INI2DC

Initializes the two-terminal dc line fetching routine 'NXT2DC' for retrieving two-terminal dc lines in dc line name alphabetical order.

Python syntax:

```
ierr = ini2dc()
```

IPLAN syntax:

```
CALL INI2DC(IERR)
```

Fortran syntax:

```
CALL INI2DC(IERR)
```

where:

Integer IERR	Is the error code (output):	
	IERR = 0	No error, ’NXT2DC’ is properly initialized.
	IERR = 1	No two-terminal dc lines in case; ’NXT2DC’ may not be used.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.65 INDDT1

Use this API routine to return real induction machine quantities.

Python syntax:

```
ierr, rval = inddt1(ibus, id, string)
```

IPLAN syntax:

```
CALL INDDT1(IBUS, ID, STRING, RVAL, IERR)
```

Fortran syntax:

```
CALL INDDT1(IBUS, ID, STRING, RVAL, IERR)
```

where:

Integer IBUS	Bus number (input).
Character ID	Induction machine identifier (input).
Character STRING	String indicating the induction machine quantity desired (input):

’MBASE’	Machine power base.
’RATEKV’	Machine rated voltage.
’PSET’	Power setpoint (positive for a motor).
’RA’	Armature resistance.
’XA’	Armature leakage reactance.
’R1’	First cage resistance.
’X1’	First cage reactance.
’R2’	Second cage resistance.
’X2’	Second cage reactance.
’X3’	Third reactance.
’E1’	Main saturation parameter.
’SE1’	Main saturation parameter.
’E2’	Main saturation parameter.
’SE2’	Main saturation parameter.
’IA1’	Leakage saturation parameter.
’IA2’	Leakage saturation parameter.
’XAMULT’	Leakage saturation parameter.
’TRQA’	Load torque variation constant.
’TRQB’	Load torque variation constant.
’TRQD’	Load torque variation constant.
’TRQE’	Load torque variation constant.
’H’	Machine inertia constant.
’IRATIO’	Ratio of locked rotor current to rated current.
’ROVERX’	Ratio of Rpos to Xpos.
’RZERO’	Machine zero sequence fault analysis resistance.
’XZERO’	Machine zero sequence fault analysis reactance.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	'RGRND'	Machine zero sequence grounding resistance.
	'XGRND'	Machine zero sequence grounding reactance.
	'P'	Active power flowing from the bus to the machine, in MW.
	'O_P'	Same as 'P', in units determined by the power output option setting.
	'Q'	Reactive power flowing from the bus to the machine, in Mvar.
	'O_Q'	Same as 'Q', in units determined by the power output option setting.
	'MVA'	Complex power flowing from the bus to the machine, in MVA.
	'O_MVA'	Same as 'MVA', in units determined by the power output option setting.
	'SLIP'	Machine slip.
Real RVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; 'RVAL' returned.
	IERR = 1	Bus not found; 'RVAL' unchanged.
	IERR = 2	Induction machine not found; 'RVAL' unchanged.
	IERR = 3	Bus type code is not 2 or 3; 'RVAL' returned.
	IERR = 4	Induction machine is out-of-service; 'RVAL' returned.
	IERR = 5	Invalid value of 'STRING'; 'RVAL' unchanged.
	IERR = 6	Sequence data not in case (when STRING = 'IRATIO', 'ROVERX', 'RZERO', 'XZERO', 'RGRND' or 'XGRND').

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.66 INDDT2

Use this API routine to return complex induction machine quantities.

Python syntax:

```
ierr, cmpval = inddt2(ibus, id, string)
```

IPLAN syntax:

```
CALL INDDT2(IBUS, ID, STRING, P, Q, IERR)
```

Fortran syntax:

```
CALL INDDT2(IBUS, ID, STRING, CMPVAL, IERR)
```

where:

Integer IBUS	Bus number (input).
Character ID	Induction machine identifier (input).
Character STRING	String indicating the induction machine quantity desired (input): 'ZA' Armature impedance (Ra + j Xa). 'Z1' First cage impedance (R1 + j X1). 'Z2' Second cage impedance (R2 + j X2). 'ZZERO' Machine zero sequence fault analysis impedance. 'ZGRND' Machine zero sequence grounding impedance. 'PQ' Actual power flowing from the bus to the machine, in MW, Mvar. 'O_PQ' Same as 'PQ', in units determined by the power output option setting.
Real P	Real component of the complex value indicated by STRING (output).
Real Q	Reactive component of the complex value indicated by STRING (output).
Complex CMPVAL	Complex value indicated by STRING (output).
Integer IERR	Is the error code (output): IERR = 0 No error; 'P' and 'Q' or 'CMPVAL' returned. IERR = 1 Bus not found; 'P' and 'Q' or 'CMPVAL' unchanged. IERR = 2 Induction machine not found; 'P' and 'Q' or 'CMPVAL' unchanged. IERR = 3 Bus type code is not 2 or 3; 'P' and 'Q' or 'CMPVAL' returned. IERR = 4 Induction machine is out-of-service; 'P' and 'Q' or 'CMPVAL' returned. IERR = 5 Invalid value of 'STRING'; 'P' and 'Q' or 'CMPVAL' unchanged. IERR = 6 Sequence data not in case (when STRING = 'ZZERO' or ZGRND').

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.67 INDINT

Use this API routine to return integer induction machine quantities.

Python syntax:

```
ierr, ival = indint(ibus, id, string)
```

IPLAN syntax:

```
CALL INDINT(IBUS, ID, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL INDINT(IBUS, ID, STRING, IVAL, IERR)
```

where:

Integer IBUS	Bus number (input).	
Character ID	Induction machine identifier (input).	
Character STRING	String indicating the induction machine quantity desired (input):	
	’STATUS’	Machine status; 1 (one) for in-service, 0 (zero) for out-of-service.
	’SCODE’	Machine standard type; 1 for NEMA, 2 for IEC).
	’DCODE’	Machine design code; 0 for custom, 1 through 5 for NEMA designs A through E.
	’AREA’	Area number of its area.
	’ZONE’	Zone number of its zone.
	’OWNER’	Owner number of its owner.
	’TCODE’	Type of mechanical torque variation; 1 for a simple power law, 2 for the WECC model).
	’BCODE’	Base power code; 1 for mechanical power (MW) output of the machine, 2 for apparent electrical power (MVA) drawn by the machine.
	’PCODE’	Scheduled power code; 1 for mechanical power (MW) output of the machine, 2 for electrical active power (MW) drawn by the machine.
	’CZG’	Grounding impedance data input/outout (I/O) code (1 for per unit, 2 for ohms).
	’CGR’	Grounding code (0 for an ungrounded machine, 1 for a grounded machine).
Integer IVAL	Value indicated by STRING (output).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output):

IERR = 0	No error; ’IVAL’ returned.
IERR = 1	Bus not found; ’IVAL’ unchanged.
IERR = 2	Induction machine not found; ’IVAL’ unchanged.
IERR = 3	Bus type code is not 2 or 3; ’IVAL’ returned.
IERR = 4	Induction machine is out-of-service, ’IVAL’ returned.
IERR = 5	Invalid value of ’STRING’; ’IVAL’ unchanged.
IERR = 6	Sequence data not in case (when STRING = ’CZG’ or ’CGR’).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.68 INIBRN

Initializes the branch fetching routine 'NXTBRN' or 'NXTBRN3' for returning branches connected to IBUS. For multisection lines, 'NXTBRN' or 'NXTBRN3' returns the adjacent dummy bus as the to bus.

Python syntax:

```
ierr = inibrn(ibus, single)
```

IPLAN syntax:

```
CALL INIBRN(IBUS, SINGLE, IERR)
```

Fortran syntax:

```
CALL INIBRN(IBUS, SINGLE, IERR)
```

where:

Integer IBUS	From bus number (input).
Integer SINGLE	If 1, ’NXTBRN’ or ’NXTBRN3’ will return only branches for which the from bus (’IBUS’) number (or bus name) is lower than the to bus (or buses); if 2, then all branches connected to ’IBUS’ are returned by ’NXTBRN’ or ’NXTBRN3’ (input).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; ’NXTBRN/NXTBRN3’ is properly initialized.
	IERR = 1 — Bus not found; ’NXTBRN/NXTBRN3’ may not be used.
	IERR = 2 — Invalid ’SINGLE’ value; ’NXTBRN/NXTBRN3’ may not be used.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.69 INIBRX

Initializes the branch fetching routine, 'NXTBRN' or 'NXTBRN3' for returning branches connected to IBUS. For multisection lines, 'NXTBRN' or 'NXTBRN3' returns the other endpoint bus as the to bus.

Python syntax:

```
ierr = inibrx(ibus, single)
```

IPLAN syntax:

```
CALL INIBRX(IBUS, SINGLE, IERR)
```

Fortran syntax:

```
CALL INIBRX(IBUS, SINGLE, IERR)
```

where:

Integer IBUS	From bus (input).
Integer SINGLE	If 1, ’NXTBRN’ or ’NXTBRN3’ will return only branches for which the from bus (’IBUS’) number (or bus name) is lower than the to bus (or buses); if 2, then all branches connected to ’IBUS’ are returned by ’NXTBRN’ or ’NXTBRN3’ (input).
Integer IERR	Is the error code (output):
	IERR = 0 — No error, ’NXTBRN/NXTBRN3’ properly initialized.
	IERR = 1 — Bus not found; ’NXTBRN/NXTBRN3’ may not be used.
	IERR = 2 — Invalid ’SINGLE’ value; ’NXTBRN/NXTBRN3’ may not be used.
	IERR = 3 — ’IBUS’ is a multi-section line dummy bus; ’NXTBRN/NXTBRN3’ may not be used.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.70 INIBUS

Initializes the bus fetching routine 'NXTBUS' for retrieving buses in ascending numerical order. Dummy buses for multisection lines are included. Hidden star point buses of three-winding transformers are excluded.

Python syntax:

```
ierr = inibus(ibus)
```

IPLAN syntax:

```
CALL INIBUS(IBUS, IERR)
```

Fortran syntax:

```
CALL INIBUS(IBUS, IERR)
```

where:

Integer IBUS	Starting bus number, 0 for first bus in ordered list (input).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; ’NXTBUS’ is properly initialized.
	IERR = 1 — Bus not found; ’NXTBUS’ may not be used.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.71 INIBUX

Initializes the bus fetching routine 'NXTBUS' for retrieving buses in ascending numerical order. Dummy buses of multisection lines are excluded. Hidden star point buses of three-winding transformers are excluded.

Python syntax:

```
ierr = inibux(ibus)
```

IPLAN syntax:

```
CALL INIBUX(IBUS, IERR)
```

Fortran syntax:

```
CALL INIBUX(IBUS, IERR)
```

where:

Integer IBUS	Starting bus number, 0 for first bus in ordered list (input).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error, ’NXTBUS’ properly initialized.
	IERR = 1	Bus not found; ’NXTBUS’ may not be used.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.72 INIFAX

Initializes the FACTS device fetching routine 'NXTFAX' for retrieving FACTS devices in FACTS device name alphabetical order.

Python syntax:

```
ierr = inifax()
```

IPLAN syntax:

```
CALL INIFAX(IERR)
```

Fortran syntax:

```
CALL INIFAX(IERR)
```

where:

Integer IERR	Is the error code (output):	
	IERR = 0	No error, 'NXTFAX' is properly initialized.
	IERR = 1	No FACTS devices in case; 'NXTFAX' may not be used.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.73 INIFXS

Initializes the fixed bus shunt fetching routine 'NXTFXS' for returning fixed shunts attached to bus IBUS.

Python syntax:

```
ierr = inifxs(ibus)
```

IPLAN syntax:

```
CALL INIFXS(IBUS, IERR)
```

Fortran syntax:

```
CALL INIFXS(IBUS, IERR)
```

where:

Integer IBUS	Bus number (input).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’NXTFXS’ is properly initialized.
	IERR = 1	Bus not found; ’NXTFXS’ may not be used.
	IERR = 2	Bus has no fixed shunts; ’NXTFXS’ may not be used.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.74 INIIND

Initializes induction machine fetching routine 'NXTIND' for returning induction machines attached to IBUS.

Python syntax:

```
ierr = iniind(ibus)
```

IPLAN syntax:

```
CALL INIIND(IBUS, IERR)
```

Fortran syntax:

```
CALL INIIND(IBUS, IERR)
```

where:

Integer IBUS	Bus number (input).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’NXTIND’ is properly initialized.
	IERR = 1	Bus not found; ’NXTIND’ may not be used.
	IERR = 2	Bus has no induction machines connected to it; ’NXTIND’ may not be used.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.75 INILOD

Initializes load fetching routine 'NXTLOD' for returning loads attached to IBUS.

Python syntax:

```
ierr = inilod(ibus)
```

IPLAN syntax:

```
CALL INILOD(IBUS, IERR)
```

Fortran syntax:

```
CALL INILOD(IBUS, IERR)
```

where:

Integer IBUS	Bus number (input).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’NXTLOD’ is properly initialized.
	IERR = 1	Bus not found; ’NXTLOD’ may not be used.
	IERR = 2	Bus has no load entries; ’NXTLOD’ may not be used.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.76 INIMAC

Initializes machine fetching routine 'NXTMAC' for returning machines attached to IBUS.

Python syntax:

```
ierr = inimac(ibus)
```

IPLAN syntax:

```
CALL INIMAC(IBUS, IERR)
```

Fortran syntax:

```
CALL INIMAC(IBUS, IERR)
```

where:

Integer IBUS	Bus number (input).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’NXTMAC’ is properly initialized.
	IERR = 1	Bus not found; ’NXTMAC’ may not be used.
	IERR = 2	Bus not a generator bus; ’NXTMAC’ may not be used.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.77 INIMDC

Initializes the multi-terminal dc line fetching routine 'NXTMDC' for retrieving multi-terminal dc lines in dc line name alphabetical order.

Python syntax:

```
ierr = inimdc()
```

IPLAN syntax:

```
CALL INIMDC(IERR)
```

Fortran syntax:

```
CALL INIMDC(IERR)
```

where:

Integer IERR	Is the error code (output):	
	IERR = 0	No error, ’NXTMDC’ is properly initialized.
	IERR = 1	No multi-terminal dc lines in case; ’NXTMDC’ may not be used.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.78 INIMSL

Initializes the multi-section line member fetching routine 'NXTMSL' for retrieving lines that are sections of the specified multi-section line from IBUS to JBUS.

Python syntax:

```
ierr = inimsl(ibus, jbus, ickt)
```

IPLAN syntax:

```
CALL INIMSL(IBUS, JBUS, ICKT, IERR)
```

Fortran syntax:

```
CALL INIMSL(IBUS, JBUS, ICKT, IERR)
```

where:

Integer IBUS	From bus number (input).
Integer JBUS	To bus number (input).
Character ICKT	Multi-section line identifier (must begin with &) (input).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; NXTMSL properly initialized.
	IERR = 1 — Bus not found; NXTMSL may not be used.
	IERR = 2 — & not the first character of 'ICKT'; NXTMSL may not be used.
	IERR = 3 — Multi-section line not found; NXTMSL may not be used.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.79 ININAM

Initializes the bus fetching routine 'NXTBUS' for retrieving buses in ascending alphabetical order. Dummy buses for multisection lines are included. Hidden star point buses of three winding transformers are excluded.

Python syntax:

```
ierr = ininam(string)
```

IPLAN syntax:

```
CALL ININAM(STRING, IERR)
```

Fortran syntax:

```
CALL ININAM(STRING, IERR)
```

where:

Character STRING	Extended bus name, blank for first bus in ordered list (input).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error, ’NXTBUS’ is properly initialized.
	IERR = 1	Bus not found; ’NXTBUS’ may not be used.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.80 ININAX

Initializes the bus fetching routine 'NXTBUS' for retrieving buses in ascending alphabetical order. Dummy buses of multisection lines are excluded. Hidden star point buses of three winding transformers are excluded.

Python syntax:

```
ierr = ininax(string)
```

IPLAN syntax:

```
CALL ININAX(STRING, IERR)
```

Fortran syntax:

```
CALL ININAX(STRING, IERR)
```

where:

Character STRING	Extended bus name, blank for first bus in ordered list (input).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error, ’NXTBUS’ properly initialized.
	IERR = 1	Bus not found; ’NXTBUS’ may not be used.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.81 INITIE

Initializes the tie branch fetching routine 'NXTTIE' or 'NXTTIE3' for returning tie branches from area IAR. For multisection lines connected to a from area bus, the adjacent dummy bus is treated as the to bus. Dummy buses for multisection lines are included.

Python syntax:

```
ierr = initie(iar)
```

IPLAN syntax:

```
CALL INITIE(IAR, IERR)
```

Fortran syntax:

```
CALL INITIE(IAR, IERR)
```

where:

Integer IAR	From area number (input).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error, ’NXTTIE’ or ’NXTTIE3’ is properly initialized.
	IERR = 1	Area number ≤ 0 or > largest allowable area number; ’NXTTIE’ or ’NXTTIE3’ may not be used.
	IERR = 2	No buses in area ’IAR’; ’NXTTIE’ or ’NXTTIE3’ may not be used.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.82 INITIX

Initializes the tie branch fetching routine 'NXTTIE' or 'NXTTIE3' for returning tie branches from area IAR. For multisection lines connected to a from area bus, the other endpoint bus is treated as the to bus. Dummy buses of multisection lines are excluded.

Python syntax:

```
ierr = initix(iar)
```

IPLAN syntax:

```
CALL INITIX(IAR, IERR)
```

Fortran syntax:

```
CALL INITIX(IAR, IERR)
```

where:

Integer IAR	From area number (input).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error, ’NXTTIE’ or ’NXTTIE3’ is properly initialized.
	IERR = 1	Area number ≤ 0 or > largest allowable area number; ’NXTTIE’ or ’NXTTIE3’ may not be used.
	IERR = 2	No buses in area ’IAR’; ’NXTTIE’ or ’NXTTIE3’ may not be used.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.83 INIVSC

Initializes the VSC dc line fetching routine 'NXTVSC' for retrieving VSC dc lines in dc line name alphabetical order.

Python syntax:

```
ierr = inivsc()
```

IPLAN syntax:

```
CALL INIVSC(IERR)
```

Fortran syntax:

```
CALL INIVSC(IERR)
```

where:

Integer IERR	Is the error code (output):	
	IERR = 0	No error, 'NXTVSC' is properly initialized.
	IERR = 1	No VSC dc lines in case; 'NXTVSC' may not be used.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.84 ISLAND

Trips in-service branches connected to Type 4 (or greater) buses, and disconnects islands not containing a swing bus.

Note: Calling ISLAND with generators converted will disconnect all buses.

Python syntax:

```
ierr = island()
```

IPLAN syntax:

```
CALL ISLAND(ICODE)
```

Fortran syntax:

```
CALL ISLAND(ICODE)
```

where:

Integer ICODE	Return code (output):	
	ICODE = 0	No errors.
	ICODE = -1	No in-service buses remaining.
	ICODE > 0	Number of islands tripped.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.85 ITERAT

Use this API to return the number of iterations used in the last solution attempt.

Python syntax:

```
ival = iterat()
```

IPLAN syntax:

```
CALL ITERAT(IVAL)
```

Fortran syntax:

```
CALL ITERAT(IVAL)
```

where:

Integer IVAL	Number of iterations used in the last solution attempt.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.86 LMODIND

Returns load-related model starting array indices and status.

Python syntax:

```
ierr, ival = lmodind(ibus, id ,string1, string2)
```

IPLAN syntax:

```
CALL LMODIND(IBUS, ID, STRING1, STRING2, IVAL, IERR)
```

Fortran syntax:

```
CALL LMODIND(IBUS, ID, STRING1, STRING2, IVAL, IERR)
```

where:

Integer IBUS	Bus number (input).	
Character ID	Load identifier (input).	
Character STRING1	String indicating the type of load model desired (input):	
	’CHARAC’	Load characteristic model.
	’RELAY’	Load relay model.
Character STRING2	String indicating the load model quantity desired (input):	
	’CON’	Starting CON index.
	’STATE’	Starting STATE index.
	’VAR’	Starting VAR index.
	’ICON’	Starting ICON index.
	’RESICON’	Starting reserved ICON index.
	’NCON’	Number of CON values used by the model.
	’NSTATE’	Number of STATE values used by the model.
	’NVAR’	Number of VAR values used by the model.
	’NICON’	Number of ICON values used by the model.
	’NRESICON’	Number of reserved ICON values used by the model.
	’STATUS’	Bypassed flag: 0 = bypassed; 1 = not bypassed. Does not reflect load status. For a model to be called during simulations, the load must be in-service, bus type code must be 1 or 2, and the model's bypassed flag must be *not bypassed*.
Integer IVAL	Load model array index or status (output).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output):

IERR = 0	No error; 'IVAL' returned.
IERR = 1	Load not in network but in model tables; 'IVAL' returned.
IERR = 2	Load not in network or in model tables; 'IVAL' unchanged.
IERR = 3	Invalid value of 'STRING1'; 'IVAL' unchanged.
IERR = 4	No model of type 'STRING1' for this load; 'IVAL' unchanged.
IERR = 5	Invalid value of 'STRING2'; 'IVAL' unchanged.
IERR = 6	Model does not use any of this data type; 'IVAL' = 0 returned.
IERR = 9	Dynamics data not present in working memory; 'IVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.87 LMODNAM

Returns load-related model name.

Python syntax:

```
ierr, cval = lmodnam(ibus, id ,string)
```

IPLAN syntax:

```
CALL LMODNAM(IBUS, ID, STRING, CVAL, IERR)
```

Fortran syntax:

```
CALL LMODNAM(IBUS, ID, STRING, CVAL, IERR)
```

where:

Integer IBUS	Bus number (input).
Character ID	Load identifier (input).
Character STRING	String indicating the type of load model desired (input). 'CHARAC' Load characteristic model. 'RELAY' Load relay model.
Character CVAL	Model name (output).
Integer IERR	Is the error code (output): IERR = 0 No error; 'CVAL' returned. IERR = 1 Load not in network but in model tables; 'CVAL' returned. IERR = 2 Load not in network or in model tables; 'CVAL' unchanged. IERR = 3 Invalid value of 'STRING'; 'CVAL' unchanged IERR = 4 No model of type 'STRING' for this load; 'CVAL' unchanged IERR = 9 Dynamics data not present in working memory; 'CVAL' unchanged

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.88 LODCNV

This obsolete API always returns a 0.

Python syntax:

```
icode = lodcnv()
```

IPLAN syntax:

```
CALL LODCNV(ICODE)
```

Fortran syntax:

```
CALL LODCNV(ICODE)
```

where:

Integer ICODE	Returns a 0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.89 LODDT1

Returns load quantities in MVA.

Python syntax:

```
ierr, rval = loddt1(ibus, id ,string1, string2)
```

IPLAN syntax:

```
CALL LODDT1(IBUS, ID, STRING1, STRING2, RVAL, IERR)
```

Fortran syntax:

```
CALL LODDT1(IBUS, ID, STRING1, STRING2, RVAL, IERR)
```

where:

Integer IBUS	Bus number (input).	
Character ID	Load identifier (input).	
Character STRING1	String indicating the load characteristic desired (input):	
	'MVA'	Constant MVA load.
	'IL'	Constant current load.
	'YL'	Constant admittance load.
	'TOTAL'	Total load.
	'YNEG'	Exceptional nominal negative sequence load.
	'YZERO'	Exceptional nominal zero sequence load.
Character STRING2	String indicating the units to be used (input):	
	'ACT'	Actual load (nominal load for STRING1 = 'YNEG' or 'YZERO'). 'P', 'Q' returned in MW, Mvar.
	'O_ACT'	Actual load (nominal load for STRING1 = 'YNEG' or 'YZERO'). 'P', 'Q' in units determined by the power output option setting.
	'NOM'	Nominal load (at 1.0 pu voltage). 'RVAL' returned in MVA.
	'O_NOM'	Nominal load (at 1.0 pu voltage). 'RVAL' in units determined by the power output option setting.
Real RVAL	Magnitude of the complex value indicated by STRING1 in the units indicated by STRING2 (output).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output):

IERR = 0	No error; 'RVAL' returned.
IERR = 1	Bus not found; 'RVAL' unchanged.
IERR = 2	Load not found; 'RVAL' unchanged.
IERR = 3	Bus type code is not 1, 2 or 3; 'RVAL' returned.
IERR = 4	Load out-of-service; 'RVAL' returned.
IERR = 5	Invalid value of 'STRING1' or 'STRING2'; 'RVAL' unchanged.
IERR = 6	Sequence data not in case for 'STRING1' = 'YNEG' or 'YZERO'; 'RVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.90 LODDT2

Use this API to return complex load quantities.

Python syntax:

```
ierr, cmpval = loddt2(ibus, id ,string1, string2)
```

IPLAN syntax:

```
CALL LODDT2(IBUS, ID, STRING1, STRING2, P, Q, IERR)
```

Fortran syntax:

```
CALL LODDT2(IBUS, ID, STRING1, STRING2, CMPVAL, IERR)
```

where:

Integer IBUS	Bus number (input).	
Character ID	Load identifier (input).	
Character STRING1	String indicating the load characteristic desired (input):	
	’MVA’	Constant MVA load.
	’IL’	Constant current load.
	’YL’	Constant admittance load.
	’TOTAL’	Total load.
	’YNEG’	Exceptional nominal negative sequence load.
	’YZERO’	Exceptional nominal zero sequence load.
Character STRING2	String indicating the units to be used (input):	
	’ACT’	Actual load (nominal load for STRING1 = ’YNEG’ or ’YZERO’). ’P’, ’Q’ returned in MW, Mvar.
	’O_ACT’	Actual load (nominal load for STRING1 = ’YNEG’ or ’YZERO’). ’P’, ’Q’ in units determined by the power output option setting.
	’NOM’	Nominal load (at 1.0 pu voltage). ’P’, ’Q’ returned in MW, Mvar.
	’O_NOM’	Nominal load (at 1.0 pu voltage). ’P’, ’Q’ in units determined by the power output option setting.
Real P	Real component of complex value indicated by STRING1 in the units indicated by STRING2 (output).	
Real Q	Reactive component of complex value indicated by STRING1 in the units indicated by STRING2 (output).	
Complex CMPVAL	Complex value indicated by STRING1 in the units indicated by STRING2 (output).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output):

IERR = 0	No error; ’P’ and ’Q’ or ’CMPVAL’ returned.
IERR = 1	Bus not found; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
IERR = 2	Load not found; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
IERR = 3	Bus type code is not 1, 2 or 3; ’P’ and ’Q’ or ’CMPVAL’ returned.
IERR = 4	Load out-of-service; ’P’ and ’Q’ or ’CMPVAL’ returned.
IERR = 5	Invalid value of ’STRING1’ or ’STRING2’; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
IERR = 6	Sequence data not in case for ’STRING1’ = ’YNEG’ or ’YZERO’; ’CMPVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.91 LODIND

Use this API to return the load array index.

Python syntax:

```
ierr, ival = lodind(ibus, id)
```

IPLAN syntax:

```
CALL LODIND(IBUS, ID, IVAL, IERR)
```

Fortran syntax:

```
CALL LODIND(IBUS, ID, IVAL, IERR)
```

where:

Integer IBUS	Bus number (input).
Character ID	Load identifier (input).
Integer IVAL	Load array index (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; 'IVAL' returned.
	IERR = 1 — Bus not found; 'IVAL' unchanged.
	IERR = 2 — Load not found; 'IVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.92 LODINT

Use this API to return integer load quantities.

Python syntax:

```
ierr, ival = lodint(ibus, id, string)
```

IPLAN syntax:

```
CALL LODINT(IBUS, ID, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL LODINT(IBUS, ID, STRING, IVAL, IERR)
```

where:

Integer IBUS	Bus number (input).	
Character ID	Load identifier (input).	
Character STRING	String indicating the load quantity desired (input):	
	’STATUS’	Load status; 1 (one) for in-service, else 0 (zero).
	’AREA’	Load area number.
	’ZONE’	Load zone number.
	’OWNER’	Load owner number.
	’SCALE’	Scalable load flag.
	’CGR’	Load grounding code (0 for ungrounded, 1 for grounded).
Integer IVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’IVAL’ returned.
	IERR = 1	Bus not found; ’IVAL’ unchanged.
	IERR = 2	Load not found; ’IVAL’ unchanged.
	IERR = 3	Bus type code is not 1, 2 or 3; ’IVAL’ returned.
	IERR = 4	Load out-of-service; ’IVAL’ returned.
	IERR = 5	Invalid value of ’STRING’; ’IVAL’ unchanged.
	IERR = 6	Sequence data not in case for ’STRING1’ = ’CGR’; ’IVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.93 MACDAT

Use this API to return real machine quantities.

Python syntax:

```
ierr, rval = macdat(ibus, id, string)
```

IPLAN syntax:

```
CALL MACDAT(IBUS, ID, STRING, RVAL, IERR)
```

Fortran syntax:

```
CALL MACDAT(IBUS, ID, STRING, RVAL, IERR)
```

where:

Integer IBUS	Bus number (input).
Character ID	Machine identifier (input).
Character STRING	String indicating the machine quantity desired (input):

’QMAX’	Maximum generator reactive output, in Mvar.
’O_QMAX’	Same as ’QMAX’, in units determined by the power output option setting.
’QMIN’	Minimum generator reactive output, in Mvar.
’O_QMIN’	Same as ’QMIN’, in units determined by the power output option setting.
’PMAX’	Maximum generator active output, in MW.
’O_PMAX’	Same as ’PMAX’, in units determined by the power output option setting.
’PMIN’	Minimum generator real active, in MW.
’O_PMIN’	Same as ’PMIN’, in units determined by the power output option setting.
’MBASE’	Total MVA base.
’MVA’	Machine loading, in MVA.
’O_MVA’	Same as ’MVA’, in units determined by the power output option setting.
’P’	Machine loading, in MW.
’O_P’	Same as ’P’, in units determined by the power output option setting.
’Q’	Machine loading, in Mvar.
’O_Q’	Same as ’Q’, in units determined by the power output option setting.
’PERCENT’	MVA loading of the machines as a percentage of machine base.
’GENTAP’	Step-up transformer off-nominal turns ratio.
’VSCHED’	Regulated voltage setpoint.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	’WPF	Power factor used in setting reactive power limits for this wind machine when WMOD is 2 or 3.
	’FRACT1’	Fraction of total ownership assigned to the first owner.
	’FRACT2’	Fraction of total ownership assigned to the second owner.
	’FRACT3’	Fraction of total ownership assigned to the third owner.
	’FRACT4’	Fraction of total ownership assigned to the fourth owner.
	’RMPCT’	Percentage of total MVAR required to regulate remote bus voltage.
	’RPOS’	Positive sequence fault analysis machine resistance.
	’XSUBTR’	Positive sequence fault analysis machine subtransient reactance.
	’XTRANS’	Positive sequence fault analysis machine transient reactance.
	’XSYNCH’	Positive sequence fault analysis machine synchronous reactance.
Real RVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’RVAL’ returned.
	IERR = 1	Bus not found; ’RVAL’ unchanged.
	IERR = 2	Machine not found; ’RVAL’ unchanged.
	IERR = 3	Bus type code is not 2 or 3; ’RVAL’ returned.
	IERR = 4	Machine off-line; ’RVAL’ returned.
	IERR = 5	Invalid value of ’STRING’; ’RVAL’ unchanged.
	IERR = 6	For STRING = ’FRACTn’, n > number of owners; ’RVAL’ unchanged.
	IERR = 7	For STRING = ’WPF’, this machine is not a wind machine; ’RVAL’ of 1.0 returned.
	IERR = 8	Sequence data not in case for STRING = ’RPOS’, ’XSUBTR’, ’XTRANS’, or ’XSYNCH’; ’RVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.94 MACDT2

Use this API to return complex machine quantities.

Python syntax:

```
ierr, cmpval = macdt2(ibus, id, string)
```

IPLAN syntax:

```
CALL MACDT2(IBUS, ID, STRING, P, Q, IERR)
```

Fortran syntax:

```
CALL MACDT2(IBUS, ID, STRING, CMPVAL, IERR)
```

where:

Integer IBUS	Bus number (input).	
Character ID	Machine identifier (input).	
Character STRING	String indicating the machine quantity desired (input):	
	'PQ'	Actual generator power output, in MW/Mvar.
	'O_PQ'	Same as 'PQ', in units determined by the power output option setting.
	'ZSORCE'	Machine impedance.
	'XTRAN'	Step-up transformer impedance.
	'ZPOS'	Positive sequence fault analysis machine impedance (Rpos + j X").
	'ZNEG''	Negative sequence machine impedance.
	'ZZERO'	Zero sequence machine impedance.
	'ZGRND'	Zero sequence grounding impedance, in per unit or ohms, according to the machine's grounding impedance data input/output code.
	'ZGRNDPU'	Zero sequence grounding impedance in per unit.
Real P	Real component of the complex value indicated by STRING (output).	
Real Q	Reactive component of the complex value indicated by STRING (output).	
Complex CMPVAL	Complex value indicated by STRING (output).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output):

IERR = 0	No error; ’P’ and ’Q’ or ’CMPVAL’ returned.
IERR = 1	Bus not found; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
IERR = 2	Machine not found; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
IERR = 3	Bus type code is not 2 or 3; ’P’ and ’Q’ or ’CMPVAL’ returned.
IERR = 4	Machine off-line; ’P’ and ’Q’ or ’CMPVAL’ returned.
IERR = 5	Invalid value of ’STRING’; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
IERR = 6	Sequence data not in case for STRING = ’ZPOS’, ’ZNEG’, or ’ZZERO’, ’ZGRND’, or ’ZGRNDPU’; ’P’ and ’Q’ or ’CMPVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.95 MACIND

Use this API to return the machine array index.

Python syntax:

```
ierr, ival = macind(ibus, id)
```

IPLAN syntax:

```
CALL MACIND(IBUS, ID, IVAL, IERR)
```

Fortran syntax:

```
CALL MACIND(IBUS, ID, IVAL, IERR)
```

where:

Integer IBUS	Bus number (input).
Character ID	Machine identifier (input).
Integer IVAL	Machine array index (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; 'IVAL' returned.
	IERR = 1 — Bus not found; 'IVAL' unchanged.
	IERR = 2 — Machine not found; 'IVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.96 MACINT

Use this API to return integer machine quantities.

Python syntax:

```
ierr, ival = macint(ibus, id, string)
```

IPLAN syntax:

```
CALL MACINT(IBUS, ID, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL MACINT(IBUS, ID, STRING, IVAL, IERR)
```

where:

Integer IBUS	Bus number (input).	
Character ID	Machine identifier (input).	
Character STRING	String indicating the machine quantity desired (input):	
	’STATUS’	Machine status; 1 (one) for in-service, else 0 (zero).
	’IREG’	Bus number of remote regulated bus; zero for none.
	’OWNERS’	Number of owners.
	’WMOD’	Wind machine reactive power limits mode; 0 if this machine is not a wind machine.
	’OWN1’	Owner number of the first owner.
	’OWN2’	Owner number of the second owner.
	’OWN3’	Owner number of the third owner.
	’OWN4’	Owner number of the fourth owner.
	’PERCENT’	MVA loading of the machines as a percentage of machine base.
	’CZG’	Grounding impedance data input/output code.
Integer IVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’IVAL’ returned.
	IERR = 1	Bus not found; ’IVAL’ unchanged.
	IERR = 2	Machine not found; ’IVAL’ unchanged.
	IERR = 3	Bus type code is not 2 or 3; ’IVAL’ returned.
	IERR = 4	Machine off-line, ’IVAL’ returned.
	IERR = 5	Invalid value of ’STRING’; ’IVAL’ unchanged.
	IERR = 6	For STRING = ’OWNn’, n> number of owners; ’IVAL’ unchanged.
	IERR = 7	Sequence data not in case for STRING = ’CZG’; ’IVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.97 MACSTT

Use this API to return the machine status value.

Python syntax:

```
ierr, ival = macstt(ibus, id)
```

IPLAN syntax:

```
CALL MACSTT(IBUS, ID, IVAL, IERR)
```

Fortran syntax:

```
CALL MACSTT(IBUS, ID, IVAL, IERR)
```

where:

Integer IBUS	Bus number (input).
Character ID	Machine identifier (input).
Integer IVAL	Machine status value; 0 for out of service, 1 for in service (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; 'IVAL' returned.
	IERR = 1 — Bus not found; 'IVAL' unchanged.
	IERR = 2 — Machine not found; 'IVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.98 MAXMSM

Use this API to return the complex bus mismatch at the bus with the largest MVA mismatch.

Python syntax:

```
ierr, ibus, cmpval = maxmsm()
```

IPLAN syntax:

```
CALL MAXMSM(IBUS, P, Q, IERR)
```

Fortran syntax:

```
CALL MAXMSM(IBUS, CMPVAL, IERR)
```

where:

Integer IBUS	Bus number of the bus with the largest MVA mismatch (output).
Real P	Real component (MW) of the bus mismatch (output).
Real Q	Reactive component (Mvar) of the bus mismatch (output).
Complex CMPVAL	Complex bus mismatch (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; 'IBUS', and 'P' and 'Q' or 'CMPVAL' returned.
	IERR = 1 — No in-service buses found; 'IBUS' of 0, and 'P' and 'Q' of 0.0 or 'CMPVAL' of (0.0,0.0) returned.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.99 MDLFLDCNT

This API is obsolete and performs no function.

Python syntax:

```
ival = mdlfldcnt()
```

IPLAN syntax:

```
CALL MDLFLDCNT(IVAL)
```

Fortran syntax:

```
CALL GETMODELFOLDERSCOUNT(IVAL)
```

where:

Integer IVAL	The number of folders in the list (input).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.100 MDLFOLDER

This API is obsolete and performs no function.

Python syntax:

```
ierr,cval = mdlfolder(n)
```

IPLAN syntax:

```
CALL MDLFOLDER(N,CVAL,IERR)
```

Fortran syntax:

```
CALL GETMODELFOLDER(N,CVAL,IERR)
```

where:

Character N	The index (ordinal position, starting with 1) in the list of the folder for which the name should be retrieved (input).
Integer CVAL*260	The name of the folder (output).
Integer IERR	Is the error code (output): IERR = 0 No error; 'CVAL' returned. IERR = 1 Index value (N) is invalid.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.101 MDLIND

Use this API to return plant-related model starting array indices and status.

Python syntax:

```
ierr, ival = mdlind(ibus, id, string1, string2)
```

IPLAN syntax:

```
CALL MDLIND(IBUS, ID, STRING1, STRING2, IVAL, IERR)
```

Fortran syntax:

```
CALL MDLIND(IBUS, ID, STRING1, STRING2, IVAL, IERR)
```

where:

Integer IBUS	Bus number (input).	
Character ID	Machine identifier (input).	
Character STRING1	String indicating the type of plant-related model (input).	
	’GEN’	Generator model.
	’COMP’	Compensator model.
	’STAB’	Stabilizer model.
	’EXC’	Excitation system model.
	’GOV’	Turbine governor model.
	’TLC’	Turbine Load Controller model.
	’MINXL’	Minimum excitation limiter model.
	’MAXXL’	Maximum excitation limiter model.
Character STRING2	String indicating the plant-related model quantity desired (input):	
	’CON’	Starting CON index.
	’STATE’	Starting STATE index.
	’VAR’	Starting VAR index.
	’ICON’	Starting ICON index.
	’NCON’	Number of CON values used by the model.
	’NSTATE’	Number of STATE values used by the model.
	’NVAR’	Number of VAR values used by the model.
	’NICON’	Number of ICON values used by the model.
	’STATUS’	Bypassed flag: 0 = bypassed; 1 = not bypassed. Does not reflect machine status. For a model to be called during simulations, the machine must be in service, the type code of the bus must be 2, and the model’s bypassed flag must be *not bypassed*.
Integer IVAL	Value indicated by STRING (output).Integer	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IERR Is the error code (output):

IERR = 0	No error; ’IVAL’ returned.
IERR = 1	Machine not in network but in model tables; ’IVAL’ returned.
IERR = 2	Machine not in network or in model tables; ’IVAL’ unchanged.
IERR = 3	Invalid value of ’STRING1’, ’IVAL’ unchanged.
IERR = 4	No model of type ’STRING1’ for this machine; ’IVAL’ unchanged.
IERR = 5	Invalid value of ’STRING2’; ’IVAL’ unchanged.
IERR = 6	Model does not use any of this data type; ’IVAL’ = 0 returned.
IERR = 9	Dynamics data not present in working memory; ’IVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.102 MDLLIBCNT

Use this API to return the number of libraries in the list to be searched for dynamics library models.

Python syntax:

```
ival = mdllibcnt()
```

IPLAN syntax:

```
CALL MDLLIBCNT(IVAL)
```

Fortran syntax:

```
CALL GETMODELLIBRARYCOUNT(IVAL)
```

where:

Integer IVAL	The number of libraries in the list (input).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.103 MDLPGMCNT

Use this API to returns the number of model programs loaded from dynamics model libraries.

Python syntax:

```
ival = mdlpgmcnt()
```

IPLAN syntax:

```
CALL MDLPGMCNT(IVAL)
```

Fortran syntax:

```
CALL GETMODELPROGRAMCOUNT(IVAL)
```

where:

Integer IVAL	The number of programs loaded (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.104 MDLNAM

Use this API to return plant-related model name.

Python syntax:

```
ierr, cval = mdlnam(ibus, id, string)
```

IPLAN syntax:

```
CALL MDLNAM(IBUS, ID, STRING, CVAL, IERR)
```

Fortran syntax:

```
CALL MDLNAM(IBUS, ID, STRING, CVAL, IERR)
```

where:

Integer IBUS	Bus number (input).	
Character ID	Machine identifier (input).	
Character STRING	String indicating the type of plant-related model (input).	
	’GEN’	Generator model.
	’COMP’	Compensator model.
	’STAB’	Stabilizer model.
	’EXC’	Excitation system model.
	’GOV’	Turbine governor model.
	’TLC’	Turbine Load Controller model.
	’MINXL’	Minimum excitation limiter model.
	’MAXXL’	Maximum excitation limiter model.
Character CVAL	Model name (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’CVAL’ returned.
	IERR = 1	Machine not in network but in model tables; ’CVAL’ returned.
	IERR = 2	Machine not in network or in model tables; ’CVAL’ unchanged.
	IERR = 3	Invalid value of ’STRING’, ’CVAL’ unchanged.
	IERR = 4	No model of type ’STRING’ for this machine; ’CVAL’ unchanged.
	IERR = 9	Dynamics data not present in working memory; ’CVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.105 NATONO

Use this API to return the bus number for a specified 18-character extended bus name.

Python syntax:

```
ierr, ibus = natono(string)
```

IPLAN syntax:

```
CALL NATONO(STRING, IBUS, IERR)
```

Fortran syntax:

```
CALL NATONO(STRING, IBUS, IERR)
```

where:

Character STRING	Extended bus name (input).
Integer IBUS	Bus number (output).
Integer IERR	Is the error code (output): IERR = 0 No error; 'IBUS' returned. IERR = 1 Bus not found; 'IBUS' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.106 NOTONA

Use this API to return the bus 18-character extended bus name for a specified bus number.

Python syntax:

```
ierr, cval = notona(ibus)
```

IPLAN syntax:

```
CALL NOTONA(IBUS, CVAL, IERR)
```

Fortran syntax:

```
CALL NOTONA(IBUS, CVAL, IERR)
```

where:

Integer IBUS	Bus number (input).
Character CVAL	Extended bus name (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; ’CVAL’ returned.
	IERR = 1 — Bus not found; ’CVAL’ unchanged.
	IERR = 2 — ’CVAL’ not long enough to contain entire extended bus name; ’CVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.107 NXT2DC

Returns the next two-terminal dc line in dc line name alphabetical order. The routine INI2DC must be called first.

Python syntax:

```
ierr, string = nxt2dc()
```

IPLAN syntax:

```
CALL NXT2DC(STRING, IERR)
```

Fortran syntax:

```
CALL NXT2DC(STRING, IERR)
```

where:

Character STRING	Two-terminal dc line name (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’STRING’ returned.
	IERR = 1	No more two-terminal dc lines; ’STRING’ unchanged.
	IERR = 2	’INI2DC’ was not called; ’STRING’ unchanged.
	IERR = 3	’STRING’ not long enough to contain entire dc line name; ’STRING’ returned but contains a truncated dc line name.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.108 NXTBRN

Returns the next branch connected to a bus, excluding three-winding transformers. Each time 'NXTBRN' is called it returns the next branch in either numeric or alphabetic ascending order, depending on the PSS®E bus output option setting. 'INIBRN' or 'INIBRX' must first be called to establish the from bus number.

Python syntax:

```
ierr, jbus, ickt = nxtbrn(ibus)
```

IPLAN syntax:

```
CALL NXTBRN(IBUS, JBUS, ICKT, IERR)
```

Fortran syntax:

```
CALL NXTBRN(IBUS, JBUS, ICKT, IERR)
```

where:

Integer IBUS	From bus number (input).
Integer JBUS	To bus number (output).
Character ICKT	Circuit identifier (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; ’JBUS’ and ’ICKT’ returned.
	IERR = 1 — No more branches from ’IBUS’; ’JBUS’ is set to 0, and ’ICKT’ is unchanged.
	IERR = 2 — ’INIBRN’ or ’INIBRX’ was not called for bus ’IBUS’; ’JBUS’, and ’ICKT’ are unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.109 NXTBRN3

Returns the next branch connected to a bus, including three-winding transformers. Each time 'NXTBRN3' is called it returns the next branch in either numeric or alphabetic ascending order, depending on the PSS®E bus output option setting. 'INIBRN' or 'INIBRX' must first be called to establish the from bus number.

Python syntax:

```
ierr, jbus, kbus, ickt = nxtbrn3(ibus)
```

IPLAN syntax:

```
CALL NXTBRN3(IBUS, JBUS, KBUS, ICKT, IERR)
```

Fortran syntax:

```
CALL NXTBRN3(IBUS, JBUS, KBUS, ICKT, IERR)
```

where:

Integer IBUS	From bus number (input).
Integer JBUS	To bus number (output).
Integer KBUS	Third bus number (output).
Character ICKT	Circuit identifier (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; ’JBUS’ and ’ICKT’ returned.
	IERR = 1 — No more branches from ’IBUS’; ’JBUS’ & ’KBUS’ are set to 0, and ’ICKT’ is unchanged.
	IERR = 2 — ’INIBRN’ or ’INIBRX’ was not called for bus ’IBUS’; ’JBUS’, ’KBUS’, and ’ICKT’ are unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.110 NXTBUS

Returns the next bus in ordered sequence. One of the routines INIBUS, INIBUX, ININAM or ININAX must be called first. The sequence can also be affected by STAREA, STZONE, and STBSKV (more than one may be used; conditions are applied in an 'AND' fashion).

Python syntax:

```
ierr, ibus, cval = nxtbus()
```

IPLAN syntax:

```
CALL NXTBUS(IBUS, CVAL, IERR)
```

Fortran syntax:

```
CALL NXTBUS(IBUS, CVAL, IERR)
```

where:

Integer IBUS	Bus number (output).	
Character CVAL	Extended bus name (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’IBUS’ and ’CVAL’ returned.
	IERR = 1	No more buses; ’IBUS’ and ’CVAL’ unchanged.
	IERR = 2	’INIBUS’, ’INIBUX’,’ININAM’ or ’ININAX’ was not called; ’IBUS’ and ’CVAL’ unchanged.
	IERR = 3	’STRING’ not long enough to contain entire extended bus name; ’IBUS’ and ’CVAL’ returned but ’CVAL’ does not contain any part of the base voltage portion of extended bus name.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.111 NXTFAX

Returns the next FACTS device in FACTS device name alphabetical order. The routine INIFAX must be called first.

Python syntax:

```
ierr, string = nxtfax()
```

IPLAN syntax:

```
CALL NXTFAX(STRING, IERR)
```

Fortran syntax:

```
CALL NXTFAX(STRING, IERR)
```

where:

Character STRING	FACTS device name (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; 'STRING' returned.
	IERR = 1	No more FACTS devices; 'STRING' unchanged.
	IERR = 2	'INIFAX' was not called; 'STRING' unchanged.
	IERR = 3	'STRING' not long enough to contain entire FACTS device name; 'STRING' returned but contains a truncated FACTS device name.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.112 NXTFXS

Use this API to return the identifier of the next fixed shunt connected to a bus. Each time 'NXTFXS' is called it returns the next shunt identifier in ascending order. 'INIFXS' must first be called to establish the shunt bus number.

Python syntax:

```
ierr, id = nxtfxs(ibus)
```

IPLAN syntax:

```
CALL NXTFXS(IBUS, ID, IERR)
```

Fortran syntax:

```
CALL NXTFXS(IBUS, ID, IERR)
```

where:

Integer IBUS	Bus number (input).	
Character ID	Shunt identifier (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’ID’ returned.
	IERR = 1	No more fixed shunts at ’IBUS’; ’ID’ unchanged.
	IERR = 2	’INIFXS’ was not called for bus ’IBUS’; ’ID’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.113 NXTIND

Use this API routine to return the identifier of the next induction machine connected to a bus. Each time 'NXTIND' is called it returns the next induction machine identifier in ascending order. 'INIIND' must first be called to establish the induction machine bus number.

Python syntax:

```
ierr, id = nxtind(ibus)
```

IPLAN syntax:

```
CALL NXTIND(IBUS, ID, IERR)
```

Fortran syntax:

```
CALL NXTIND(IBUS, ID, IERR)
```

where:

Integer IBUS	Bus number (input).
Character ID	Induction machine identifier (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; ’ID’ returned.
	IERR = 1 — No more induction machines at ’IBUS’; ’ID’ unchanged.
	IERR = 2 — ’INIIND’ was not called for bus ’IBUS’; ’ID’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.114 NXTLOD

Use this API to return the identifier of the next load connected to a bus. Each time 'NXTLOD' is called it returns the next load identifier in ascending order. 'INILOD' must first be called to establish the load bus number.

Python syntax:

```
ierr, id = nxtlod(ibus)
```

IPLAN syntax:

```
CALL NXTLOD(IBUS, ID, IERR)
```

Fortran syntax:

```
CALL NXTLOD(IBUS, ID, IERR)
```

where:

Integer IBUS	Bus number (input).	
Character ID	Load identifier (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’ID’ returned.
	IERR = 1	No more loads at ’IBUS’; ’ID’ unchanged.
	IERR = 2	’INILOD’ was not called for bus ’IBUS’; ’ID’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.115 NXTMAC

Use this API to return the identifier of the next machine connected to a bus. Each time 'NXTMAC' is called it returns the next machine identifier in ascending order. 'INIMAC' must first be called to establish the plant bus number.

Python syntax:

```
ierr, id = nxtmac(ibus)
```

IPLAN syntax:

```
CALL NXTMAC(IBUS, ID, IERR)
```

Fortran syntax:

```
CALL NXTMAC(IBUS, ID, IERR)
```

where:

Integer IBUS	Bus number (input).
Character ID	Machine identifier (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; ’ID’ returned.
	IERR = 1 — No more machines at ’IBUS’; ’ID’ unchanged.
	IERR = 2 — ’INIMAC’ was not called for bus ’IBUS’; ’ID’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.116 NXTMDC

Returns the next multi-terminal dc line in dc line name alphabetical order. The routine INIMDC must be called first.

Python syntax:

```
ierr, string = nxtmdc()
```

IPLAN syntax:

```
CALL NXTMDC(STRING, IERR)
```

Fortran syntax:

```
CALL NXTMDC(STRING, IERR)
```

where:

Character STRING	Multi-terminal dc line name (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’STRING’ returned.
	IERR = 1	No more multi-terminal dc lines; ’STRING’ unchanged.
	IERR = 2	’INIMDC’ was not called; ’STRING’ unchanged.
	IERR = 3	’STRING’ not long enough to contain entire dc line name; ’STRING’ returned but contains a truncated dc line name.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.117 NXTMSL

Use this API to return the next multi-section line member branch.

Python syntax:

```
ierr, ibus, jbus, ickt = nxtmsl()
```

IPLAN syntax:

```
CALL NXTMSL(IBUS, JBUS, ICKT, IERR)
```

Fortran syntax:

```
CALL NXTMSL(IBUS, JBUS, ICKT, IERR)
```

where:

Integer IBUS	From bus number (output).
Integer JBUS	To bus number (output).
Character ICKT	Multi-section line identifier (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; ’IBUS’, ’JBUS’, and ’ICKT’ returned.
	IERR = 1 — No more branches in multi-section line; ’IBUS’, ’JBUS’, and ’ICKT’ unchanged.
	IERR = 2 — & not the first character of ’ICKT’; NXTMSL may not be used.
	IERR = 3 — INIMSL was not called; ’IBUS’, ’JBUS’, and ’ICKT’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.118 NXTTIE

Use this API to return the next tie branch from an area, excluding three-winding transformers. IBUS will be in the area IAR specified in the call to 'INITIE' or 'INITIX', which must be called first; JBUS will be in the other area.

Python syntax:

```
ierr, ibus, jbus, ickt = nxttie()
```

IPLAN syntax:

```
CALL NXTTIE(IBUS, JBUS, ICKT, IERR)
```

Fortran syntax:

```
CALL NXTTIE(IBUS, JBUS, ICKT, IERR)
```

where:

Integer IBUS	From bus number (output).
Integer JBUS	To bus number (output).
Character ICKT	Circuit identifier (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error, ’IBUS’, ’JBUS’, and ’ICKT’ returned.
	IERR = 1 — No more tie branches from area ’IAR’; ’IBUS’ set to 0, and ’JBUS’ and ’ICKT’ unchanged.
	IERR = 2 — ’INITIE’ or ’INITIX’ was not called; ’IBUS’, ’JBUS’ and ’ICKT’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.119 NXTTIE3

Use this API to return the next tie branch from an area, including three-winding transformers. IBUS will be in the area IAR specified in the call to 'INITIE' or 'INITIX', which must be called first; JBUS will be in the other area.

Python syntax:

```
ierr, ibus, jbus, kbus, ickt = nxttie3()
```

IPLAN syntax:

```
CALL NXTTIE3(IBUS, JBUS, KBUS, ICKT, IERR)
```

Fortran syntax:

```
CALL NXTTIE3(IBUS, JBUS, KBUS, ICKT, IERR)
```

where:

Integer IBUS	From bus number (output).
Integer JBUS	To bus number (output).
Integer KBUS	Third bus number (output).
Character ICKT	Circuit identifier (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error, ’IBUS’, ’JBUS’, ’KBUS’, and ’ICKT’ returned.
	IERR = 1 — No more tie branches from area ’IAR’; ’IBUS’ set to 0, and ’JBUS’, ’KBUS’, and ’ICKT’ unchanged.
	IERR = 2 — ’INITIE’ or ’INITIX’ was not called; ’IBUS’, ’JBUS’, ’KBUS’, and ’ICKT’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.120 NXTVSC

Returns the next VSC dc line in dc line name alphabetical order. The routine INIVSC must be called first.

Python syntax:

```
ierr, string = nxtvsc()
```

IPLAN syntax:

```
CALL NXTVSC(STRING, IERR)
```

Fortran syntax:

```
CALL NXTVSC(STRING, IERR)
```

where:

Character STRING	VSC dc line name (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’STRING’ returned.
	IERR = 1	No more VSC dc lines; ’STRING’ unchanged.
	IERR = 2	’INIVSC’ was not called; ’STRING’ unchanged.
	IERR = 3	’STRING’ not long enough to contain entire dc line name; ’STRING’ returned but contains a truncated dc line name.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.121 OKSTRT

Routine which may be called after activity STRT or MSTR to indicate the presence of INITIAL CONDITIONS SUSPECT errors (for activity STRT) or at least one MODEL NOT AVAILABLE IN MSTR/MRUN message (for MSTR).

Python syntax:

```
ierr = okstrt()
```

IPLAN syntax:

```
CALL OKSTRT(IERR)
```

Fortran syntax:

```
CALL OKSTRT(IERR)
```

where:

Integer IERR	Is the error code (output):	
	IERR = 0	No error.
	IERR = 1	Simulation initialization errors were detected.
	IERR = 9	Dynamics data not present in working memory.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.122 OPENUN

Connects an IPLAN unit number to the Fortran unit used by activity OPEN and returns that IPLAN unit number.

IPLAN syntax:

```
CALL OPENUN(N)
```

Fortran syntax:

```
CALL OPENUN(N)
```

where:

Integer N	IPLAN unit number; zero if no unit is currently open by activity OPEN (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.123 ORDBUS

Returns the bus number corresponding to the specified position in the ordered list (numeric or alphabetic) of buses. Dummy buses of multisection lines are included; hidden star point buses of three-winding transformers are not included.

Python syntax:

```
ierr, ibus = ordbus(iord)
```

IPLAN syntax:

```
CALL ORDBUS(IORD, IBUS, IERR)
```

Fortran syntax:

```
CALL ORDBUS(IORD, IBUS, IERR)
```

where:

Integer IORD	Ordinal position; 1 through the number of non-hidden buses in the working case (input).
Integer IBUS	Bus number (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; 'IBUS' returned.
	IERR = 1 — 'IORD' outside of range 1 through the number of non-hidden buses in the working case; 'IBUS' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.124 OWNDAT

Returns the data associated with an owner ('LOAD' gets loads at buses of owner 'IAR', 'LOADLD' gets loads assigned to owner 'IAR').

Python syntax:

```
ierr, cmpval = owndat(iar, string)
```

IPLAN syntax:

```
CALL OWNDAT(IAR, STRING, P, Q, IERR)
```

Fortran syntax:

```
CALL OWDAT(IAR, STRING, CMPVAL, IERR)
```

where:

Integer IAR	Owner number (input).	
Character STRING	String indicating the owner total desired (input):	
	’LOAD’	Total load by bus owner assignment.
	’LOADLD’	Total load by load owner assignment.
	’GEN’	Total generation by bus owner assignment.
	’GENMC’	Total generation by machine owner assignment.
	’LOSS’	Total losses by branch owner assignment.
	’INDMAC’	Total owner induction machine powers by bus owner assignment.
	’INDMACMC’	Total owner induction machine powers by machine owner assignment.
	’INDGEN’	Total owner induction generator powers by bus owner assignment.
	’INDGENMC’	Total owner induction generator powers by machine owner assignment.
	’INDMOT’	Total owner induction motor powers by bus owner assignment.
	’INDMOTMC’	Total owner induction motor powers by machine owner assignment.
Real P	Real component of complex power (output).	
Real Q	Reactive component of complex power (output).	
Complex CMPVAL	Complex power (output).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output):

IERR = 0	No error; 'P' and 'Q' or 'CMPVAL' returned.
IERR = 1	Owner number ≤ 0 or > largest allowable owner number; 'P' and 'Q' or 'CMPVAL' unchanged.
IERR = 2	No in-service buses with in-service loads (for 'LOAD'), no in-service loads (for 'LOADLD'), no type 2 or type 3 buses (for 'GEN'), no in-service machines (for 'GENMC'), no in-service branches (for 'LOSS'), no in-service buses with in-service induction machines of the appropriate type (for 'INDMAC', 'INDGEN' and 'INDMOT'), or no in-service induction machines of the appropriate type (for 'INDMACMC', 'INDGENMC' and 'INDMOTMC') in owner; 'P' and 'Q' or 'CMPVAL' unchanged.
IERR = 3	Owner not found; 'P' and 'Q' or 'CMPVAL' unchanged.
IERR = 4	Bad 'STRING' value; 'P' and 'Q' or 'CMPVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.125 OWNNAM

Returns the owner name for a specified owner number.

Python syntax:

```
ierr, cval = ownnam(iar)
```

IPLAN syntax:

```
CALL OWNNAM(IAR, CVAL, IERR)
```

Fortran syntax:

```
CALL OWNNAM(IAR, CVAL, IERR)
```

where:

Integer IAR	Owner number (input).
Character CVAL	Owner name (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; ’CVAL’ returned.
	IERR = 1 — Owner number ≤ 0 or > largest allowable owner number; ’CVAL’ unchanged.
	IERR = 2 — Owner not found; ’CVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.126 OWNNUM

Returns the owner number for a specified owner name.

Python syntax:

```
ierr, iar = ownnum(string)
```

IPLAN syntax:

```
CALL OWNNUM(IAR, STRING, IERR)
```

Fortran syntax:

```
CALL OWNNUM(IAR, STRING, IERR)
```

where:

Integer IAR	Owner number (output).	
Character STRING	Owner name (input).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; 'IAR' returned.
	IERR = 1	Owner with name 'STRING' not found; 'IAR' unchanged.
	IERR > 1	Multiple owners with the name 'STRING' found; 'IAR' returned as the lowest numbered of these.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.127 OWNUSE

Use this API to indicate whether an owner is in use.

Python syntax:

```
ival  = ownuse(iar)
```

IPLAN syntax:

```
CALL OWNUSE(IAR, IVAL)
```

Fortran syntax:

```
CALL OWNUSE(IAR, IVAL)
```

where:

Integer IAR	Owner number (input).	
Integer IVAL	Owner status (output):	
	IVAL = 1	If any equipment is assigned to owner IAR.
	IVAL = 0	If no equipment is assigned to owner IAR.
	IVAL = -1	Owner number ≤ 0 or > largest allowable owner number.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.128 PRMDAT

Use this API routine to return real-valued solution parameters.

Python syntax:

```
ierr, rval = prmdat(string)
```

IPLAN syntax:

```
CALL PRMDAT(STRING, RVAL, IERR)
```

Fortran syntax:

```
CALL PRMDAT(STRING, RVAL, IERR)
```

where:

Character STRING	String indicating the solution parameter desired (input):	
	'ACCP'	Gauss-Seidel real component acceleration factor.
	'ACCQ'	Gauss-Seidel imaginary component acceleration factor.
	'ACCM'	special MSLV acceleration factor.
	'TOL'	Gauss-Seidel convergence tolerance.
	'ACCN'	Newton-Raphson acceleration factor.
	'TOLN'	Newton_Raphson mismatch convergence tolerance.
	'ACCTY	TYSL acceleration factor.
	'TOLTY'	TYSL convergence tolerance.
	'BLOWUP'	blown-up voltage change threshold.
	'PQBRAK'	constant power load characteristic breakpoint.
	'THRSHZ'	zero impedance line threshold.
	'ADJTHR'	automatic adjustment threshold tolerance.
	'ACCTAP'	tap movement deceleration factor.
	'TAPLIM'	maximum tap change per adjustment cycle.
	'DVLIM'	largest | (delta V) / |V| | for Newton-Raphson solutions.
	'NDVFCT'	non-divergent required improvement factor.
	'VCTOLQ'	Newton-Raphson controlled bus reactive power mismatch convergence tolerance.
	'VCTOLV'	Newton-Raphson controlled bus voltage error convergence tolerance.
	'SWVBND'	percent of voltage controlling band mode switched shunts to be adjusted.
Real RVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; 'RVAL' returned.
	IERR = 1	Invalid value of 'STRING'; 'RVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.129 PRMINT

Use this API routine to return integer-valued solution parameters.

Python syntax:

```
ierr, ival = prmint(string)
```

IPLAN syntax:

```
CALL PRMINT(STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL PRMINT(STRING, IVAL, IERR)
```

where:

Character STRING	String indicating the solution parameter desired (input):	
	'ITMX'	Gauss-Seidel maximum number of iterations.
	'ITMXN'	Newton-Raphson maximum number of iterations.
	'ITMXTY'	TYSL maximum number of iterations.
	'MXTPSS'	maximum number of tap and/or switched shunt adjustment cycles that change data between iterations.
	'MXSWIM'	maximum number of induction machine state switchings during a power flow solution.
Integer IVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; 'IVAL' returned.
	IERR = 1	Invalid value of 'STRING'; 'IVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.130 RMODIND

Returns the branch relay model starting array indices and status.

Python syntax:

```
ierr, ival = rmodind(ibus, jbus, ickt, slot, string)
```

IPLAN syntax:

```
CALL RMODIND(IBUS, JBUS, ICKT, SLOT, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL RMODIND(IBUS, JBUS, ICKT, SLOT, STRING, IVAL, IERR)
```

where:

Integer IBUS	From bus number (input).
Integer JBUS	To bus number (input).
Character ICKT	Circuit identifier (input).
Integer SLOT	Relay slot (1 or 2) (input).
Character STRING	String indicating the relay model quantity desired (input):
	’CON’ — Starting CON index.
	’STATE’ — Starting STATE index.
	’VAR’ — Starting VAR index.
	’ICON’ — Starting ICON index.
	’NCON’ — Number of CON values used by the model.
	’NSTATE’ — Number of STATE values used by the model.
	’NVAR’ — Number of VAR values used by the model.
	’NICON’ — Number of ICON values used by the model.
	’STATUS’ — Bypassed flag: 0 = bypassed; 1 = not bypassed. Does not reflect branch status. For a model to be called during simulations, the branch must be in-service, and the model's bypassed flag must be *not bypassed*.
Integer IVAL	Branch model index or status (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output):

IERR = 0	No error; 'IVAL' returned.
IERR = 1	Branch not in network but in model tables; 'IVAL' returned.
IERR = 2	Branch not in network or in model tables; 'IVAL' unchanged.
IERR = 3	Invalid value of 'SLOT'; 'IVAL' unchanged.
IERR = 4	No SLOT 'SLOT' model for this branch; 'IVAL' unchanged.
IERR = 5	Invalid value of 'STRING'; 'IVAL' unchanged.
IERR = 6	Model does not use any of this data type; 'IVAL' = 0 returned.
IERR = 9	Dynamics data not present in working memory; 'IVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.131 RMODNAM

Returns the branch relay model name.

Python syntax:

```
ierr, cval = rmodnam(ibus, jbus, ickt, slot)
```

IPLAN syntax:

```
CALL RMODNAM(IBUS, JBUS, ICKT, SLOT, CVAL, IERR)
```

Fortran syntax:

```
CALL RMODNAM(IBUS, JBUS, ICKT, SLOT, CVAL, IERR)
```

where:

Integer IBUS	From bus number (input).
Integer JBUS	To bus number (input).
Character ICKT	Circuit identifier (input).
Integer SLOT	Relay slot (1 or 2) (input).
Character CVAL	Model name (output).
Integer IERR	Is the error code (output):

IERR = 0	No error; ’CVAL’ returned.
IERR = 1	Branch not in network but in model tables; ’CVAL’ returned.
IERR = 2	Branch not in network or in model tables; ’CVAL’ unchanged.
IERR = 3	Invalid value of ’SLOT’; ’CVAL’ unchanged
IERR = 4	No SLOT ’SLOT’ model for this branch; ’CVAL’ unchanged
IERR = 9	Dynamics data not present in working memory; ’CVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.132 SC3WND

Returns the three-winding transformer short circuit currents following activity SCMU.

Python syntax:

```
ierr, cmpval = sc3wnd(ibus, jbus, kbus, ickt, string)
```

IPLAN syntax:

```
CALL SC3WND(IBUS, JBUS, KBUS, ICKT, STRING, P, Q, IERR)
```

Fortran syntax:

```
CALL SC3WND(IBUS, JBUS, KBUS, ICKT, STRING, CMPVAL, IERR)
```

where:

Integer IBUS	From bus number (input).
Integer JBUS	To bus number (input).
Integer KBUS	Third bus number (input).
Character ICKT	Circuit identifier (string input).
Character STRING	String indicating the short circuit current desired (input): 'IPOS' Positive sequence. 'INEG' Negative sequence. 'IZERO' Zero sequence. 'IA' A-phase. 'IB' B-phase. 'IC' C-phase.
Real P	Real component of the complex current indicated by STRING (output).
Real Q	Reactive component of the complex current indicated by STRING (output).
Complex CMPVAL	Complex current indicated by STRING (output).
Integer IERR	Is the error code (output): IERR = 0 No error; 'P' and 'Q' or 'CMPVAL' returned. IERR = 1 'SCINIT' not successfully called; 'P' and 'Q' or 'CMPVAL' unchanged. IERR = 2 Bus not found; 'P' and 'Q' or 'CMPVAL' unchanged. IERR = 3 Branch value not found; 'P' and 'Q' or 'CMPVAL' unchanged. IERR = 4 Invalid 'STRING'; 'P' and 'Q' or 'CMPVAL' unchanged. IERR = 5 Branch out of service; 'P' and 'Q' of 0.0 or 'CMPVAL' of (0.0,0.0) returned.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.133 SCBRN2

Returns the branch short circuit currents following activity SCMU.

Python syntax:

```
ierr, cmpval = scbrn2(ibus, jbus, ickt, string)
```

IPLAN syntax:

```
CALL SCBRN2(IBUS, JBUS, ICKT, STRING, P, Q, IERR)
```

Fortran syntax:

```
CALL SCBRN2(IBUS, JBUS, ICKT, STRING, CMPVAL, IERR)
```

where:

Integer IBUS	From bus number (input).
Integer JBUS	To bus number (input).
Character ICKT	Circuit identifier (input).
Character STRING	String indicating the short circuit current desired (input): 'IPOS' Positive sequence. 'INEG' Negative sequence. 'IZERO' Zero sequence. 'IA' A-phase. 'IB' B-phase. 'IC' C-phase.
Real P	Real component of the complex current indicated by STRING (output).
Real Q	Reactive component of the complex current indicated by STRING (output).
Complex CMPVAL	Complex current indicated by STRING (output).
Integer IERR	Is the error code (output): IERR = 0 No error; 'P' and 'Q' or 'CMPVAL' returned. IERR = 1 'SCINIT' not successfully called; 'P' and 'Q' or 'CMPVAL' unchanged. IERR = 2 Bus not found; 'P' and 'Q' or 'CMPVAL' unchanged. IERR = 3 Branch not found; 'P' and 'Q' or 'CMPVAL' unchanged. IERR = 4 Bad value of 'STRING1'; 'P' and 'Q' or 'CMPVAL' unchanged. IERR = 5 Branch out of service; 'P' and 'Q' of 0.0 or 'CMPVAL' of (0.0,0.0) returned.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.134 SCBUS2

Returns complex bus voltages and currents following activity SCMU.

Python syntax:

```
ierr, cmpval = scbus2(ibus, string)
```

IPLAN syntax:

```
CALL SCBUS2(IBUS, STRING, P, Q, IERR)
```

Fortran syntax:

```
CALL SCBUS2(IBUS, STRING, CMPVAL, IERR)
```

where:

Integer IBUS	Bus number (input).	
Character STRING	String indicating the quantity desired (input):	
	’VPOS’	Positive sequence voltage.
	’VNEG’	Negative sequence voltage.
	’VZERO’	Zero sequence voltage.
	’VA’	A-phase voltage.
	’VB’	B-phase voltage.
	’VC’	C-phase voltage.
	’SHUNTP’	Positive sequence shunt current.
	’SHUNTN’	Negative sequence shunt current.
	’SHUNTZ’	Zero sequence shunt current.
	’SHUNTA’	A-phase shunt current.
	’SHUNTB’	B-phase shunt current.
	’SHUNTC’	C-phase shunt current.
	’FAULTP’	Positive sequence fault current.
	’FAULTN’	Negative sequence fault current.
	’FAULTZ’	Zero sequence fault current.
	’FAULTA’	A-phase fault current.
	’FAULTB’	B-phase fault current.
	’FAULTC’	C-phase fault current.
Real P	Real component of the complex value indicated by STRING (output).	
Real Q	Reactive component of the complex value indicated by STRING (output).	
Complex CMPVAL	Complex value indicated by STRING (output).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output):

IERR = 0	No error; ’P’ and ’Q’ or ’CMPVAL’ returned.
IERR = 1	’SCINIT’ not successfully called; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
IERR = 2	Bus not found; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
IERR = 3	Bad value of ’STRING’; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
IERR = 4	Bus disconnected; ’P’ and ’Q’ of 0.0 or ’CMPVAL’ of (0.0,0.0) returned.
IERR = 5	Bus not faulted; ’P’ and ’Q’ of 0.0 or ’CMPVAL’ of (0.0,0.0) returned.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.135 SCDONE

Use this API to restore the working case after 'SCINIT'.

Python syntax:

```
ierr = scdone()
```

IPLAN syntax:

```
CALL SCDONE(IERR)
```

Fortran syntax:

```
CALL SCDONE(IERR)
```

where:

Integer IERR	Is the error code (output):	
	IERR = 0	No error.
	IERR = 1	’SCINIT’ not successfully called.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.136 SCINIT

Use this API to initialize the short-circuit data retrieval APIs. It must be called when PSS®E is at the power flow activity level and must follow each execution of activity SCMU. Can be followed by SCBUS2, SCBRN2, SC3WND and SCMAC2 calls. SCDONE ***must*** be called after executing these and before any other calls or PSS®E activities can be executed.

Python syntax:

```
ierr = scinit()
```

IPLAN syntax:

```
CALL SCINIT(IERR)
```

Fortran syntax:

```
CALL SCINIT(IERR)
```

where:

Integer IERR	Is the error code (output):	
	IERR = 0	No error.
	IERR = 1	Sequence data not in case.
	IERR = 2	Unable to access SCMU results; one of the following occurred: • No SCMU results file is open. • SCMU results are not compatible with the working case. • Error reading the SCMU results file.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.137 SCMAC2

Use this API to return the machine short circuit currents following activity SCMU.

Python syntax:

```
ierr, cmpval = scmac2(ibus, ickt, string)
```

IPLAN syntax:

```
CALL SCMAC2(IBUS, ICKT, STRING, P, Q, IERR)
```

Fortran syntax:

```
CALL SCMAC2(IBUS, ICKT, STRING, CMPVAL, IERR)
```

where:

Integer IBUS	Bus number (input).	
Character ICKT	Machine identifier (input).	
Character STRING	String indicating the short circuit current desired (input):	
	'IPOS'	Positive sequence.
	'INEG'	Negative sequence.
	'IZERO'	Zero sequence.
	'IA'	A-phase.
	'IB'	B-phase.
	'IC'	C-phase.
Real P	Real component of the complex current (output).	
Real Q	Reactive component of the complex current (output).	
Complex CMPVAL	Complex current (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; 'P' and 'Q' or 'CMPVAL' returned.
	IERR = 1	'SCINIT' not successfully called; 'P' and 'Q' or 'CMPVAL' unchanged.
	IERR = 2	Bus not found; 'P' and 'Q' or 'CMPVAL' unchanged.
	IERR = 3	Machine not found; 'P' and 'Q' or 'CMPVAL' unchanged.
	IERR = 4	Bad value of 'STRING'; 'P' and 'Q' or 'CMPVAL' unchanged.
	IERR = 5	Machine out of service; 'P' and 'Q' of 0.0 or 'CMPVAL' of (0.0,0.0) returned.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.138 SELCTR

Use this API to return the activity selector indicator.

Python syntax:

```
ival = selctr()
```

IPLAN syntax:

```
CALL SELCTR(IVAL)
```

Fortran syntax:

```
CALL SELCTR(IVAL)
```

where:

Integer IVAL	Activity indicator (output):	
	IVAL = 0	Not at ’ACTIVITY’.
	IVAL = 1	At power flow activity selector.
	IVAL = 2	At dynamics activity selector.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.139 SFILES

Use this API to return the current Saved Case and Snapshot filenames.

Python syntax:

```
savfilnam, snpfilnam = sfiles()
```

IPLAN syntax:

```
CALL SFILES(SAVFILNAM, SNPFILNAM)
```

Fortran syntax:

```
CALL SFILES(SAVFILNAM, SNPFILNAM)
```

where:

Character SAVFILNAM*260	Current Saved Case File (output).
Character SNPFILNAM*260	Current Snapshot File (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.140 SLMODIND

Use this API to return the subsystem load-related model starting array indices and status.

Python syntax:

```
ierr, ival = slmodind(ctype, ibus, id, string1, string2)
```

IPLAN syntax:

```
CALL SLMODIND(CTYPE, ID, IBUS, STRING1, STRING2, IVAL, IERR)
```

Fortran syntax:

```
CALL SLMODIND(CTYPE, ID, IBUS, STRING1, STRING2, IVAL, IERR)
```

where:

Character CTYPE	Subsystem type (input):	
	’BUS’	Subsystem specified by buses.
	’AREA’	Subsystem specified by areas.
	’ZONE’	Subsystem specified by zones.
	’OWNER’	Subsystem specified by owners.
	’ALL’	Subsystem of all loads.
Integer IBUS	Number of bus, area, zone, or owner, depending on CTYPE. Not used if CTYPE=’ALL’ (input).	
Character ID	Load identifier (input).	
Character STRING1	String indicating the type of load model desired (input):	
	’CHARAC’	Load characteristic model.
	’RELAY’	Load relay model.
Character STRING2	String indicating the load model quantity desired (input):	
	’CON’	Starting CON index.
	’ICON’	Starting ICON index.
	’NCON’	Number of CON values used by the model.
	’NICON’	Number of ICON values used by the model.
	’STATUS’	Bypassed flag: 0 = bypassed; 1 = not bypassed. Does not reflect load status. For a model to be called during simulations, the load must be in-service, the bus type code must be 1 or 2, and the model's bypassed flag must be *not bypassed*.
Integer IVAL	Load model index or status (output).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’IVAL’ returned.
	IERR = 1	Invalid value of ’CTYPE’; ’IVAL’ unchanged.
	IERR = 2	Invalid value of ’STRING1’; ’IVAL’ unchanged.
	IERR = 3	No model of type ’STRING1’ for this load; ’IVAL’ unchanged.
	IERR = 4	Invalid value of ’STRING2’; ’IVAL’ unchanged.
	IERR = 5	Model does not use any of this data type; ’IVAL’ = 0 returned.
	IERR = 9	Dynamics data not present in working memory; ’IVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.141 SLMODNAM

Use this API to return the subsystem load-related model name.

Python syntax:

```
ierr, cval = slmodnam(ctype, ibus, id, string)
```

IPLAN syntax:

```
CALL SLMODNAM(CTYPE, IBUS, ID, STRING, CVAL, IERR)
```

Fortran syntax:

```
CALL SLMODNAM(CTYPE, IBUS, ID, STRING, CVAL, IERR)
```

where:

Character CTYPE	Subsystem type (input):	
	’BUS’	Subsystem specified by buses.
	’AREA’	Subsystem specified by areas.
	’ZONE’	Subsystem specified by zones.
	’OWNER’	Subsystem specified by owners.
	’ALL’	Subsystem of all loads.
Integer IBUS	Number of bus, area, zone, or owner, depending on CTYPE. Not used if CTYPE=’ALL’ (input).	
Character ID	Load identifier (input).	
Character STRING	String indicating the type of load model desired (input).	
	’CHARAC’	Load characteristic model.
	’RELAY’	Load relay model.
Character CVAL	Model name (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’CVAL’ returned.
	IERR = 1	Invalid value of ’CTYPE’; ’CVAL’ returned.
	IERR = 2	Invalid value of ’STRING’; ’CVAL’ unchanged
	IERR = 3	No model of type ’STRING’ for this load; ’CVAL’ unchanged
	IERR = 9	Dynamics data not present in working memory; ’CVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.142 SOLVED

Use this API to check whether the last solution attempt reached tolerance. This is not a test for whether the working case is a solved case.

Python syntax:

```
ival = solved()
```

IPLAN syntax:

```
CALL SOLVED(IVAL)
```

Fortran syntax:

```
CALL SOLVED(IVAL)
```

where:

Integer IVAL	Solution convergence indicator (output).	
	IVAL = 0	Met convergence tolerance.
	IVAL = 1	Iteration limit exceeded.
	IVAL = 2	Blown up (only when non-divergent option disabled).
	IVAL = 3	Terminated by non-divergent option.
	IVAL = 4	Terminated by console interrupt.
	IVAL = 5	Singular Jacobian matrix or voltage of 0.0 detected.
	IVAL = 6	Inertial power flow dispatch error (INLF).
	IVAL = 7	OPF solution met convergence tolerance (NOPF).
	IVAL = 9	Solution not attempted.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.143 SOLVED_CA

Use this API to check whether the last corrective action solution attempt reached tolerance.

Python syntax:

```
ival = solved_ca()
```

IPLAN syntax:

```
ival = solved_ca()
```

Fortran syntax:

```
CALL SOLVED_CA(IVAL)
```

where:

Integer IVAL	Corrective action solution convergence indicator (output).	
	IVAL=0	Removed all limit violations successfully
	IVAL=1	Exceeded iteration limit (Number of AC load flows)
	IVAL=2	Terminated with constraint violations.
	IVAL=3	Terminated because of insufficient controls
	IVAL=4	Terminated because linear programming model can not be setup
	IVAL=5	Terminated because Linear Programming solution diverged
	IVAL=6	Terminated because AC load flow exceeded iteration limit
	IVAL=7	Terminated because AC load flow blown up (only when non-divergent option disabled)
	IVAL=8	Terminated because AC load flow terminated by non-divergent option
	IVAL=9	Terminated because AC load flow terminated by console interrupt
	IVAL=10	Terminated because AC load flow terminated by singular jacobian matrix (FNSL) or voltage= 0.0 (FDNS)
	IVAL=13	Terminated because AC load flow converged with stalled and/or tripped induction machines
	IVAL=14	Solution not attempted

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.144 STAREA

Use this API to restrict 'NXTBUS' to a single area. It must be called after INIBUS / INIBUX / ININAM / ININAX.

Python syntax:

```
ierr = starea(iar)
```

IPLAN syntax:

```
CALL STAREA(IAR, IERR)
```

Fortran syntax:

```
CALL STAREA(IAR, IERR)
```

where:

Integer IAR	Area number (input).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error.
	IERR = 1	Invalid area number.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.145 STBSKV

Use this API to restrict 'NXTBUS' to a single base voltage level. It must be called after INIBUS / INIBUX / ININAM / ININAX.

Python syntax:

```
ierr = stbskv(bskv)
```

IPLAN syntax:

```
CALL STBSKV(BSKV, IERR)
```

Fortran syntax:

```
CALL STBSKV(BSKV, IERR)
```

where:

Real BSKV	Base kV level (input).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error.
	IERR = 1	Negative base voltage.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.146 STZONE

Use this API to restrict 'NXTBUS' to a single zone. It must be called after INIBUS / INIBUX / ININAM / ININAX.

Python syntax:

```
ierr = stzone(iar)
```

IPLAN syntax:

```
CALL STZONE(IAR, IERR)
```

Fortran syntax:

```
CALL STZONE(IAR, IERR)
```

where:

Integer IAR	Zone number (input).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error.
	IERR = 1	Invalid zone number.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.147 SWSBLK

Use this API to return the switched shunt number of steps and step size for block IBLK.

Python syntax:

```
ierr, ival, v = swsblk(ibus, iblk)
```

IPLAN syntax:

```
CALL SWSBLK(IBUS, IBLK, IVAL, V, IERR)
```

Fortran syntax:

```
CALL SWSBLK(IBUS, IBLK, IVAL, V, IERR)
```

where:

Integer IBUS	Bus number (input).
Integer IBLK	Switched shunt block number, 1 through 8 (input).
Integer IVAL	Number of steps in block ’IBLK’ (output).
Real V	Admittance in Mvar per step in block ’IBLK’ (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; ’IVAL’ and ’V’ returned.
	IERR = 1 — Bus not found; ’IVAL’ and ’V’ unchanged.
	IERR = 2 — No switched shunt at bus; ’IVAL’ and ’V’ unchanged.
	IERR = 3 — Bad value of ’IBLK’; ’IVAL’ and ’V’ unchanged.
	IERR = 4 — Bus type code greater than or equal to 4; ’IVAL’ and ’V’ returned.
	IERR = 5 — Status is 0; ’IVAL’ and ’V’ returned.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.148 SWSBLZ

Use this API to return the switched shunt number of steps and zero sequence step size in per unit for block IBLK.

Python syntax:

```
ierr, ival, v = swsblz(ibus, iblk)
```

IPLAN syntax:

```
CALL SWSBLZ(IBUS, IBLK, IVAL, V, IERR)
```

Fortran syntax:

```
CALL SWSBLZ(IBUS, IBLK, IVAL, V, IERR)
```

where:

Integer IBUS	Bus number (input).
Integer IBLK	Switched shunt block number (input).
Integer IVAL	Number of steps in block ’IBLK’ (output).
Real V	Admittance in pu per step in block ’IBLK’ (output).
Integer IERR	Is the error code (output):

IERR = 0	No error; ’IVAL’ and ’V’ returned.
IERR = 1	Bus not found; ’IVAL’ and ’V’ unchanged.
IERR = 2	No switched shunt at bus; ’IVAL’ and ’V’ unchanged.
IERR = 3	Bad value of ’IBLK’; ’IVAL’ and ’V’ unchanged.
IERR = 4	Bus type code greater than or equal to 4; ’IVAL’ and ’V’ returned.
IERR = 5	Status is 0; ’IVAL’ and ’V’ returned.
IERR = 6	Sequence data not in case; ’IVAL’ and ’V’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.149 SWSBLZ2

Use this API to return the switched shunt number of steps and zero sequence step size in Mvar for block IBLK.

Python syntax:

```
ierr, ival, v = swsblz2(ibus, iblk)
```

IPLAN syntax:

```
CALL SWSBLZ2(IBUS, IBLK, IVAL, V, IERR)
```

Fortran syntax:

```
CALL SWSBLZ2(IBUS, IBLK, IVAL, V, IERR)
```

where:

Integer IBUS	Bus number (input).
Integer IBLK	Switched shunt block number (input).
Integer IVAL	Number of steps in block ’IBLK’ (output).
Real V	Admittance in Mvar per step in block ’IBLK’ (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; ’IVAL’ and ’V’ returned.
	IERR = 1 — Bus not found; ’IVAL’ and ’V’ unchanged.
	IERR = 2 — No switched shunt at bus; ’IVAL’ and ’V’ unchanged.
	IERR = 3 — Bad value of ’IBLK’; ’IVAL’ and ’V’ unchanged.
	IERR = 4 — Bus type code greater than or equal to 4; ’IVAL’ and ’V’ returned.
	IERR = 5 — Status is 0; ’IVAL’ and ’V’ returned.
	IERR = 6 — Sequence data not in case; ’IVAL’ and ’V’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.150 SWSDAT

Use this API to return the switched shunt schedule voltage band.

Python syntax:

```
ierr, vhi, vlo = swsdat(ibus)
```

IPLAN syntax:

```
CALL SWSDAT(IBUS, VHI, VLO, IERR)
```

Fortran syntax:

```
CALL SWSDAT(IBUS, VHI, VLO, IERR)
```

where:

Integer IBUS	Bus number (input).	
Real VHI	Desired voltage upper limit (output).	
Real VLO	Desired voltage lower limit (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’VHI’ and ’VLO’ returned.
	IERR = 1	Bus not found; ’VHI’ and ’VLO’ unchanged.
	IERR = 2	No switched shunt at bus; ’VHI’ and ’VLO’ unchanged.
	IERR = 4	Bus type code greater than or equal to 4; ’VHI’ and ’VLO’ returned.
	IERR = 5	Status is 0; ’VHI’ and ’VLO’ returned.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.151 SWSDT1

Use this API to return switched shunt real data.

Python syntax:

```
ierr, rval = swsdt1(ibus, string)
```

IPLAN syntax:

```
CALL SWSDT1(IBUS, STRING, RVAL, IERR)
```

Fortran syntax:

```
CALL SWSDT1(IBUS, STRING, RVAL, IERR)
```

where:

Integer IBUS	Bus number (input).	
Character STRING	String indicating the switched shunt quantity desired (input):	
	'VSWHI'	Controlled quantity upper limit.
	'VSWLO'	Controlled quantity lower limit.
	'RMPCT'	Percentage of total Mvar required to regulate controlled bus voltage.
	'BINIT'	Present admittance switched on in Mvar at unity voltage.
	'O_BINIT'	Same as 'BINIT', units determined by power output option.
Real RVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; 'RVAL' returned.
	IERR = 1	Bus not found; 'RVAL' unchanged.
	IERR = 2	No switched shunt at bus; 'RVAL' unchanged.
	IERR = 3	Invalid 'STRING'; 'RVAL' unchanged.
	IERR = 4	Bus type code greater than or equal to 4; 'RVAL' returned.
	IERR = 5	Status is 0; 'RVAL' returned.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.152 SWSINT

Use this API to return switched shunt integer parameters.

Python syntax:

```
ierr, ival = swsint(ibus, string)
```

IPLAN syntax:

```
CALL SWSINT(IBUS, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL SWSINT(IBUS, STRING, IVAL, IERR)
```

where:

Integer IBUS	Bus number (input).	
Character STRING	String indicating the switched shunt quantity desired (input):	
	'MODE'	Control mode.
	'STATUS'	Status flag
	'ADJM'	Adjustment method.
	'REMOTE'	Bus number of regulated bus; zero for none.
	'BLOCKS'	Number of blocks.
Integer IVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; 'IVAL' returned.
	IERR = 1	Bus not found; 'IVAL' unchanged.
	IERR = 2	No switched shunt at bus; 'IVAL' unchanged.
	IERR = 3	Bad value of 'STRING'; 'IVAL' unchanged.
	IERR = 4	Bus type code greater than or equal to 4; 'IVAL' returned.
	IERR = 5	Status is 0; 'IVAL' returned.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.153 SWSMIND

Use this API to return switched shunt model starting array indices and status.

Python syntax:

```
ierr, ival = swsmind(ibus, string)
```

IPLAN syntax:

```
CALL swsmind(IBUS, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL SWSMIND(IBUS, STRING, IVAL, IERR)
```

where:

Integer IBUS	Bus number (input).	
Character STRING	String indicating the switched shunt model quantity desired (input):	
	’CON’	Starting CON index.
	’STATE’	Starting STATE index.
	’VAR’	Starting VAR index.
	’ICON’	Starting ICON index.
	’NCON’	Number of CON values used by the model.
	’NSTATE’	Number of STATE values used by the model.
	’NVAR’	Number of VAR values used by the model.
	’NICON’	Number of ICON values used by the model.
	’STATUS’	Bypassed flag: 0 = bypassed; 1 = not bypassed. Does not reflect switched shunt status. For a model to be called during simulations, the switched shunt must be in service, the model’s bypassed flag must be *not bypassed*.
Integer IVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’IVAL’ returned.
	IERR = 1	Switched shunt not in network but in model tables; ’IVAL’ returned.
	IERR = 2	Switched shunt not in network or in model tables; ’IVAL’ unchanged.
	IERR = 4	No model in model tables; ’IVAL’ unchanged
	IERR = 5	Invalid value of ’STRING’; ’IVAL’ unchanged.
	IERR = 6	Model does not use any of this data type; ’IVAL’ = 0 returned.
	IERR = 7	Bus not found; ’IVAL’ unchanged.
	IERR = 9	Dynamics data not present in working memory; ’IVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.154 SWSMNAM

Use this API to return switched shunt model name.

Python syntax:

```
ierr, cval = swsmnam(ibus)
```

IPLAN syntax:

```
CALL SWSMNAM(IBUS, CVAL, IERR)
```

Fortran syntax:

```
CALL SWSMNAM(IBUS, CVAL, IERR)
```

where:

Integer IBUS	Bus number (input).	
Character CVAL	Model name (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’CVAL’ returned.
	IERR = 1	Switched shunt not in network but in model tables; ’CVAL’ returned.
	IERR = 2	Switched shunt not in network or in model tables; ’CVAL’ unchanged.
	IERR = 4	No model in model tables; ’CVAL’ unchanged.
	IERR = 7	Bus not found; ’CVAL’ unchanged.
	IERR = 9	Dynamics data not present in working memory; ’CVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.155 SYSMSM

Use this API to return the total system MVA mismatch.

Python syntax:

```
mva = sysmsm()
```

IPLAN syntax:

```
CALL SYSMSM(MVA)
```

Fortran syntax:

```
CALL SYSMSM(MVA)
```

where:

Real MVA	Total system MVA mismatch (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.156 SYSMVA

Use this API to return the system base MVA.

Python syntax:

```
mvabas = sysmva()
```

IPLAN syntax:

```
CALL SYSMVA(MVABAS)
```

Fortran syntax:

```
CALL SYSMVA(MVABAS)
```

where:

Real MVABAS	System base MVA (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.157 SYSTOT

Use this API to return complex system-wide values.

Python syntax:

```
ierr, cmpval = systot(string)
```

IPLAN syntax:

```
CALL SYSTOT(STRING, P, Q, IERR)
```

Fortran syntax:

```
CALL SYSTOT(STRING, CMPVAL, IERR)
```

where:

Character STRING	String indicating the system total desired (input):	
	'LOAD'	Total system load.
	'GEN'	Total system generation.
	'LOSS'	Total system losses.
	'INDMAC'	Total system induction machine powers.
	'INDGEN'	Total system induction generator powers.
	'INDMOT'	Total system induction motor powers.
	'FACTSH'	Total of FACTS device shunt elements.
	'BSSHNT'	Total of bus fixed and switched shunt elements.
	'LNSHNT'	Total of line shunts and transformer magnetizing shunt elements.
	'ALSHNT'	Total of FACTS device, fixed, switched, line, and transformer magnetizing shunt elements..
Real P	Real component of the complex power (output).	
Real Q	Reactive component of the complex power (output).	
Complex CMPVAL	Complex power (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; 'P' and 'Q' or 'CMPVAL' returned.
	IERR = 1	Bad 'STRING' value; 'P' and 'Q' or 'CMPVAL' unchanged.
	IERR = 2	No corresponding in-service elements in the case; 'P' and 'Q' or 'CMPVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.158 TITLDT

Use this API to return the two line case title.

Python syntax:

```
line1, line2 = titldt()
```

IPLAN syntax:

```
CALL TITLDT(LINE1, LINE2)
```

Fortran syntax:

```
CALL TITLDT(LINE1, LINE2)
```

where:

Character LINE1	First 60-character line (output).
Character LINE2	Second 60-character line (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.159 TOTBUS

Use this API to return the total number of buses in the working case. Dummy buses of multisection lines and star point buses of three-winding transformers are included.

Python syntax:

```
n = totbus()
```

IPLAN syntax:

```
CALL TOTBUS(N)
```

Fortran syntax:

```
CALL TOTBUS(N)
```

where:

Integer N	Number of buses in the working case. If the working case is empty, then the number of buses returned is zero (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.160 TR3DAT

Use this API to return three-winding transformer real values.

Python syntax:

```
ierr, rval = tr3dat(ibus, jbus, kbus, ickt, string)
```

IPLAN syntax:

```
CALL TR3DAT(IBUS, JBUS, KBUS, ICKT, STRING, RVAL, IERR)
```

Fortran syntax:

```
CALL TR3DAT(IBUS, JBUS, KBUS, ICKT, STRING, RVAL, IERR)
```

where:

Integer IBUS	From bus number (input).	
Integer JBUS	To bus number (input).	
Integer KBUS	Third bus number (input).	
Character ICKT	Circuit identifier (input).	
Character STRING	String indicating the three-winding transformer quantity desired (input):	
	’FRACT1’	Fraction of total ownership assigned to the first owner.
	’FRACT2’	Fraction of total ownership assigned to the second owner.
	’FRACT3’	Fraction of total ownership assigned to the third owner.
	’FRACT4’	Fraction of total ownership assigned to the fourth owner.
	’VMSTAR’	Star bus voltage magnitude.
	’ANSTAR’	Star bus voltage angle in degrees.
Real RVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’RVAL’ returned.
	IERR = 1	Bus not found; ’RVAL’ unchanged.
	IERR = 2	Three-winding transformer not found; ’RVAL’ unchanged.
	IERR = 3	Invalid ’STRING’; ’RVAL’ unchanged.
	IERR = 4	Error fetching transformer data; ’RVAL’ unchanged.
	IERR = 6	With STRING = FRACTn, n > number of owners; ’RVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.161 TR3DT2

Use this API to return three-winding transformer complex values.

Python syntax:

```
ierr, cmpval = tr3dt2(ibus, jbus, kbus, ickt, string)
```

IPLAN syntax:

```
CALL TR3DT2(IBUS, JBUS, KBUS, ICKT, STRING, P, Q, IERR)
```

Fortran syntax:

```
CALL TR3DT2(IBUS, JBUS, KBUS, ICKT, STRING, CMPVAL, IERR)
```

where:

Integer IBUS	From bus number (input).	
Integer JBUS	To bus number (input).	
Integer KBUS	Third bus number (input).	
Character ICKT	Circuit identifier (input).	
Character STRING	String indicating the three-winding transformer quantity desired (input):	
	’RX1-2’	Measured impedance in per unit between the winding 1 and 2 buses.
	’RX2-3’	Measured impedance in per unit between the winding 2 and 3 buses.
	’RX3-1’	Measured impedance in per unit between the winding 3 and 1 buses.
	’YMAGNT’	Magnetizing admittance, in pu.
	’ZG1’, ’ZGRND’	Zero-sequence winding 1 grounding impedance.
	’Z01’	Zero-sequence winding 1, or winding 1 bus to winding 2 bus, leakage impedance.
	’ZG2’	Zero-sequence winding 2 grounding impedance.
	’Z02’	Zero-sequence winding 2, or winding 2 bus to winding 3 bus, leakage impedance.
	’ZG3’	Zero-sequence winding 3 grounding impedance.
	’Z03’	Zero-sequence winding 3, or winding 3 bus to winding 1 bus, leakage impedance.
	’ZNUTRL’	Zero-sequence common neutral grounding impedance.
Real P	Real component of complex value indicated by STRING (output).	
Real Q	Reactive component of complex value indicated by STRING (output).	
Complex CMPVAL	Complex value indicated by STRING (output).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’P’ and ’Q’ or ’CMPVAL’ returned.
	IERR = 1	Bus not found; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
	IERR = 2	Three-winding transformer not found; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
	IERR = 3	Invalid ’STRING’; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
	IERR = 4	Error fetching transformer data; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
	IERR = 5	Sequence data not in case for ’STRING’ = ’ZGRND’, ’ZG1’, ’Z01’, ’ZG2’, ,’Z02’, ’ZG3’, ’Z03’, or ’ZNUTRL’; ’P’ and ’Q’ or ’CMPVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.162 TR3INT

Use this API to return three-winding transformer integer values.

Python syntax:

```
ierr, ival = tr3int(ibus, jbus, kbus, ickt, string)
```

IPLAN syntax:

```
CALL TR3INT(IBUS, JBUS, KBUS, ICKT, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL TR3INT(IBUS, JBUS, KBUS, ICKT, STRING, IVAL, IERR)
```

where:

Integer IBUS	From bus number (input).
Integer JBUS	To bus number (input).
Integer KBUS	Third bus number (input).
Character ICKT	Circuit identifier (input).
Character STRING	String indicating the three-winding transformer quantity desired (input):
	'STATUS' Three-winding transformer status.
	'NMETER' Non-metered bus number.
	'OWNERS' Number of owners.
	'OWN1' Owner number of first owner.
	'OWN2' Owner number of second owner.
	'OWN3' Owner number of third owner.
	'OWN4' Owner number of fourth owner.
	'CW' Winding data I/O code.
	'CZ' Impedance data I/O code.
	'CM' Magnetizing admittance data I/O code.
	'CZ0' Leakage impedance data I/O code.
	'CZG' Grounding impedance data I/O code.
	'CNXCOD' Winding connection code.
Integer IVAL	Value indicated by STRING (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR — Is the error code (output):

IERR = 0	No error; ’IVAL’ returned.
IERR = 1	Bus not found; ’IVAL’ unchanged.
IERR = 2	Three-winding transformer not found; ’IVAL’ unchanged.
IERR = 3	Invalid ’STRING’; ’IVAL’ unchanged.
IERR = 4	Error fetching transformer data; ’IVAL’ unchanged.
IERR = 5	Sequence data not in case for ’STRING’ = ’CZ0’, ’CZG’ or ’CNXCOD’; ’IVAL’ unchanged.
IERR = 6	With ’STRING’ = OWNn, n > number of owners; ’IVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.163 TR3NAM

Use this API to return three-winding transformer name.

Python syntax:

```
ierr, cval = tr3nam(ibus, jbus, kbus, ickt)
```

IPLAN syntax:

```
CALL TR3NAM(IBUS, JBUS, KBUS, ICKT, CVAL, IERR)
```

Fortran syntax:

```
CALL TR3NAM(IBUS, JBUS, KBUS, ICKT, CVAL, IERR)
```

where:

Integer IBUS	From bus number (input).
Integer JBUS	To bus number (input).
Integer KBUS	Third bus number (input).
Character ICKT	Circuit identifier (input).
Character CVAL	Transformer name (output).
Integer IERR	Is the error code (output):

IERR = 0	No error; ’CVAL’ returned.
IERR = 1	Bus not found; ’CVAL’ unchanged.
IERR = 2	Three-winding transformer not found; ’CVAL’ unchanged.
IERR = 4	Error fetching transformer data; ’CVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.164 TRNDAT

This API is obsolete. It has been replaced by the API XFRDAT with STRING='RATIO' to get the RATIO value, and with STRING='ANGLE' to get the ANGLE value.

Python syntax:

```
ierr, ratio, angle = trndat(ibus, jbus, ickt)
```

IPLAN syntax:

```
CALL TRNDAT(IBUS, JBUS, ICKT, RATIO, ANGLE, IERR)
```

Fortran syntax:

```
CALL TRNDAT(IBUS, JBUS, ICKT, RATIO, ANGLE, IERR)
```

where:

Integer IBUS	From bus number (input).
Integer JBUS	To bus number (input).
Character ICKT	Circuit identifier (input).
Real RATIO	Tap position (output).
Real ANGLE	Phase shift angle, in degrees (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; 'RATIO' and 'ANGLE' returned.
	IERR = 1 — Bus not found; 'RATIO' and 'ANGLE' unchanged.
	IERR = 2 — Branch not found; 'RATIO' and 'ANGLE' unchanged.
	IERR = 3 — Branch not a transformer; 'RATIO' and 'ANGLE' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.165 TRXDAT

Use this API to return the inter-area transfer MW.

Python syntax:

```
ierr, rval = trxdat(iar, jar, id)
```

IPLAN syntax:

```
CALL TRXDAT(IAR, JAR, ID, RVAL, IERR)
```

Fortran syntax:

```
CALL TRXDAT(IAR, JAR, ID, RVAL, IERR)
```

where:

Integer IAR	From area number (input).
Integer JAR	To area number (input).
Character ID	Inter-area transfer identifier (input).
Real RVAL	Value (output).
Integer IERR	Is the error code (output):

IERR = 0	No error; 'RVAL' returned.
IERR = 1	Area number ≤ 0 or > largest allowable area number; 'RVAL' unchanged.
IERR = 2	Interarea transfer not found; 'RVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.166 VSCAUXMIND

Use this API to return the starting array indices and status of auxiliary signal model associated with VSC dc line for the specified signal index.

Python syntax:

```
ierr, ival = vscauxmind(name, isgndx, string)
```

IPLAN syntax:

```
CALL VSCAUXMIND(NAME, ISGNDX, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL VSCAUXMIND(NAME, ISGNDX, STRING, IVAL, IERR)
```

where:

Character NAME	VSC dc line name (input).	
Integer ISGNDX	Auxiliary signal index (input).	
Character STRING	String indicating the VSC dc line auxiliary signal model quantity desired (input):	
	’CON’	Starting CON index.
	’STATE’	Starting STATE index.
	’VAR’	Starting VAR index.
	’ICON’	Starting ICON index.
	’NCON’	Number of CON values used by the model.
	’NSTATE’	Number of STATE values used by the model.
	’NVAR’	Number of VAR values used by the model.
	’NICON’	Number of ICON values used by the model.
	’STATUS’	Bypassed flag: 0 = bypassed; 1 = not bypassed. For a model to be called during simulations, the model’s bypassed flag must be *not bypassed*.
Integer IVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’IVAL’ returned.
	IERR = 1	dc line not in network but auxiliary signal model in model tables; ’IVAL’ returned.
	IERR = 2	dc line not in network and auxiliary signal model not in model tables; ’IVAL’ unchanged.
	IERR = 4	No model in model tables for the given auxiliary signal index; ’IVAL’ unchanged
	IERR = 5	Invalid value of ’STRING’; ’IVAL’ unchanged.
	IERR = 6	Model does not use any of this data type; ’IVAL’ = 0 returned.
	IERR = 9	Dynamics data not present in working memory; ’IVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.167 VSCAUXMNAM

Use this API to return the auxiliary signal model name associated with a VSC dc line for the specified auxiliary signal index.

Python syntax:

```
ierr, cval = vscauxmnam(name, isgndx)
```

IPLAN syntax:

```
CALL VSCAUXMNAM(NAME, ISGNDX, CVAL, IERR)
```

Fortran syntax:

```
CALL VSCAUXMNAM(NAME, ISGNDX, CVAL, IERR)
```

where:

Character NAME	VSC dc line name (input).	
Integer ISGNDX	Auxiliary signal index (input).	
Character CVAL	Model name (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’CVAL’ returned.
	IERR = 1	dc line not in network but auxiliary signal model in model tables; ’CVAL’ returned.
	IERR = 2	dc line not in network and auxiliary signal model not in model tables; ’CVAL’ unchanged.
	IERR = 4	No model in model tables for the given auxiliary signal index; ’CVAL’ unchanged
	IERR = 9	Dynamics data not present in working memory; ’CVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.168 VSCCDT

Use this API to return VSC dc line real parameters.

Python syntax:

```
ierr, rval = vsccdt(id, ibus, string)
```

IPLAN syntax:

```
CALL VSCCDT(ID, IBUS, STRING, RVAL, IERR)
```

Fortran syntax:

```
CALL VSCCDT(ID, IBUS, STRING, RVAL, IERR)
```

where:

Character ID	VSC dc line name (input).	
Integer IBUS	Converter bus number (input).	
Character STRING	String indicating the VSC dc line quantity desired (input):	
	’PAC’	Real power into dc line at converter bus, in MW.
	’O_PAC’	Same as ’PAC’, in units determined by the power output option setting.
	’QAC’	Reactive power into dc line at converter bus, in MV.
	’O_QAC’	Same as ’QAC’, in units determined by the power output option setting.
	’MVA’	Apparent ac flow into the dc line, in MVA.
	’O_MVA’	Same as ’MVA’, in units determined by the power output option setting.
	’PCTMVA’	ac loading percent of limit.
	’ACAMPS’	ac current, in amps.
	’PUCUR’	ac current, in pu.
	’KVDC’	Voltage at controlling converter, in kV.
	’DCCUR’	dc current, in amps.
	’DCSET’	dc setpoint, in pu.
	’ACSET’	ac setpoint, in pu.
	’RDC’	dc line resistance, in ohms.
Real RVAL	Parameter value (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’RVAL’ returned.
	IERR = 1	VSC dc line ’ID’ not found; ’RVAL’ unchanged.
	IERR = 2	Invalid ’STRNG’; ’RVAL’ unchanged.
	IERR = 3	Bus ’IBUS’ not a converter of this dc line; ’RVAL’ unchanged.
	IERR = 4	dc line blocked; ’RVAL’ returned as 0.0.
	IERR = 5	No base kV at bus; ’RVAL’ returned as the per unit current.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.169 VSCCIN

Use this API to return VSC dc line converter integer parameters.

Python syntax:

```
ierr, ival = vsccin(id, ibus, string)
```

IPLAN syntax:

```
CALL VSCCIN(ID, IBUS, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL VSCCIN(ID, IBUS, STRING, IVAL, IERR)
```

where:

Character ID	VSC dc line name (input).
Integer IBUS	Converter bus number (input).
Character STRING	String indicating the VSC dc line quantity desired (input): 'DCTYPE' Converter dc control code. 'ACMODE' Converter ac control mode.
Integer IVAL	Parameter value (output).
Integer IERR	Is the error code (output): IERR = 0 No error; 'IVAL' returned. IERR = 1 VSC dc line 'ID' not found; 'IVAL' unchanged. IERR = 2 Invalid 'STRING'; 'IVAL' unchanged. IERR = 3 Bus 'IBUS' not a converter of this dc line; 'IVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.170 VSCINT

Use this API to return VSC dc line integer parameters.

Python syntax:

```
ierr, ival = vscint(id, string)
```

IPLAN syntax:

```
CALL VSCINT(ID, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL VSCINT(ID, STRING, IVAL, IERR)
```

where:

Character ID	VSC dc line name (input).
Character STRING	String indicating the VSC dc line quantity desired (input): ’CONV1’ Converter 1 bus number. ’CONV2’ Converter 2 bus number. ’MDC’ Control mode.
Integer IVAL	Parameter value (output).
Integer IERR	Is the error code (output): IERR = 0 No error; ’IVAL’ returned. IERR = 1 VSC dc line ’ID’ not found; ’IVAL’ unchanged. IERR = 2 Invalid ’STRING’; ’IVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.171 VSCMIND

Use this API to return VSC dc line model starting array indices and status.

Python syntax:

```
ierr, ival = vscmind(name, string)
```

IPLAN syntax:

```
CALL VSCMIND(NAME, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL VSCMIND(NAME, STRING, IVAL, IERR)
```

where:

Character NAME	VSC dc line name (input).	
Character STRING	String indicating the VSC dc line model quantity desired (input):	
	’CON’	Starting CON index.
	’STATE’	Starting STATE index.
	’VAR’	Starting VAR index.
	’ICON’	Starting ICON index.
	’NCON’	Number of CON values used by the model.
	’NSTATE’	Number of STATE values used by the model.
	’NVAR’	Number of VAR values used by the model.
	’NICON’	Number of ICON values used by the model.
	’STATUS’	Bypassed flag: 0 = bypassed; 1 = not bypassed. Does not reflect dc line status. For a model to be called during simulations, the dc line must be in service, the model’s bypassed flag must be *not bypassed*.
Integer IVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’IVAL’ returned.
	IERR = 1	dc line not in network but in model tables; ’IVAL’ returned.
	IERR = 2	dc line not in network or in model tables; ’IVAL’ unchanged.
	IERR = 4	No model in model tables; ’IVAL’ unchanged
	IERR = 5	Invalid value of ’STRING’; ’IVAL’ unchanged.
	IERR = 6	Model does not use any of this data type; ’IVAL’ = 0 returned.
	IERR = 9	Dynamics data not present in working memory; ’IVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.172 VSCMNAM

Use this API to return VSC dc line model name.

Python syntax:

```
ierr, cval = vscmnam(name)
```

IPLAN syntax:

```
CALL VSCMNAM(NAME, CVAL, IERR)
```

Fortran syntax:

```
CALL VSCMNAM(NAME, CVAL, IERR)
```

where:

Character NAME	VSC dc line name (input).	
Character CVAL	Model name (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’CVAL’ returned.
	IERR = 1	dc line not in network but in model tables; ’CVAL’ returned.
	IERR = 2	dc line not in network or in model tables; ’CVAL’ unchanged.
	IERR = 4	No model in model tables; ’CVAL’ unchanged.
	IERR = 9	Dynamics data not present in working memory; ’CVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.173 WINDMIND

Use this API to return wind model starting array indices and status.

Python syntax:

```
ierr, ival = windmind(ibus, id, string1, string2)
```

IPLAN syntax:

```
CALL WINDMIND(IBUS, ID, STRING1, STRING2, IVAL, IERR)
```

Fortran syntax:

```
CALL WINDMIND(IBUS, ID, STRING1, STRING2, IVAL, IERR)
```

where:

Integer IBUS	Bus number (input).	
Character ID	Machine identifier (input).	
Character STRING1	String indicating the type of wind model (input).	
	'WGEN'	Wind Generator model.
	'WELEC'	Wind Electrical model.
	'WMECH'	Wind Mechanical model.
	'WPICH'	Wind Pitch model.
	'WAERO'	Wind Aerodynamic model.
	'WGUST'	Wind Gust model
	'WAUX'	Wind Auxiliary Signal model.
Character STRING2	String indicating the plant-related model quantity desired (input):	
	'CON'	Starting CON index.
	'STATE'	Starting STATE index.
	'VAR'	Starting VAR index.
	'ICON'	Starting ICON index.
	'NCON'	Number of CON values used by the model.
	'NSTATE'	Number of STATE values used by the model.
	'NVAR'	Number of VAR values used by the model.
	'NICON'	Number of ICON values used by the model.
	'STATUS'	Bypassed flag: 0 = bypassed; 1 = not bypassed. Does not reflect machine status. For a model to be called during simulations, the machine must be in service, the type code of the bus must be 2, and the model's bypassed flag must be *not bypassed*.
Integer IVAL	Value indicated by STRING (output).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output):	
	IERR = 0	No error; 'IVAL' returned.
	IERR = 1	Machine not in network but in model tables; 'IVAL' returned.
	IERR = 2	Machine not in network or in model tables; 'IVAL' unchanged.
	IERR = 3	Invalid value of 'STRING1', 'IVAL' unchanged.
	IERR = 4	No model of type 'STRING1' for this machine; 'IVAL' unchanged.
	IERR = 5	Invalid value of 'STRING2'; 'IVAL' unchanged.
	IERR = 6	Model does not use any of this data type; 'IVAL' = 0 returned.
	IERR = 9	Dynamics data not present in working memory; 'IVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.174 WINDMNAM

Use this API to return wind model name.

Python syntax:

```
ierr, cval = windmnam(ibus, id, string)
```

IPLAN syntax:

```
CALL WINDMNAM(IBUS, ID, STRING, CVAL, IERR)
```

Fortran syntax:

```
CALL WINDMNAM(IBUS, ID, STRING, CVAL, IERR)
```

where:

Integer IBUS	Bus number (input).	
Character ID	Machine identifier (input).	
Character STRING	String indicating the type of plant-related model (input).	
	'WGEN'	Wind Generator model.
	'WELEC'	Wind Electrical model.
	'WMECH'	Wind Mechanical model.
	'WPICH'	Wind Pitch model.
	'WAERO'	Wind Aerodynamic model.
	'WGUST'	Wind Gust Model
	'WAUX'	Wind Auxiliary Signal model.
Character CVAL	Model name (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; 'CVAL' returned.
	IERR = 1	Machine not in network but in model tables; 'CVAL' returned.
	IERR = 2	Machine not in network or in model tables; 'CVAL' unchanged.
	IERR = 3	Invalid value of 'STRING', 'CVAL' unchanged.
	IERR = 4	No model of type 'STRING' for this machine; 'CVAL' unchanged.
	IERR = 9	Dynamics data not present in working memory; 'CVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.175 WNDDAT

Use this API to return three-winding transformer real values for winding connected to bus IBUS.

Python syntax:

```
ierr, rval = wnddat(ibus, jbus, kbus, ickt, string)
```

IPLAN syntax:

```
CALL WNDDAT(IBUS, JBUS, KBUS, ICKT, STRING, RVAL, IERR)
```

Fortran syntax:

```
CALL WNDDAT(IBUS, JBUS, KBUS, ICKT, STRING, RVAL, IERR)
```

where:

Integer IBUS	From bus number (input).
Integer JBUS	To bus number (input).
Integer KBUS	Third bus number (input).
Character ICKT	Circuit identifier (input).
Character STRING	String indicating the transformer winding quantity desired (input):

’RATEA’	Rating set A.
’RATEB’	Rating set B.
’RATEC’	Rating set C.
’RATE’	Default rating set.
’RATIO’	Winding tap ratio.
’ANGLE’	Winding phase angle shift, in degrees.
’SBASE’	Winding MVA base.
’NOMV’	Nominal winding voltage.
’RMAX’	Ratio or angle high limit.
’RMIN’	Ratio or angle low limit.
’VMAX’	Controlled quantity high limit.
’VMIN’	Controlled quantity low limit.
’STEP’	Derived tap step.
’CNXANG’	Winding connection angle, in degrees.
’MVA’	Winding MVA flow.
’O_MVA’	Same as ’MVA’, in units determined by the power output option setting.
’AMPS’	Winding current in amps.
’PUCUR’	Winding current in pu.
’PCTRTA’	Percent current loading of RATEA.
’PCTRTB’	Percent current loading of RATEB.
’PCTRTC’	Percent current loading of RATEC.
’PCTMVA’	Percent MVA loading of RATEA.
’PCTMVB’	Percent MVA loading of RATEB.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	’PCTMVC’	Percent MVA loading of RATEC.
	’PCTCPA’	Percent current or MVA loading (according to the transformer percent loading units program option setting) of RATEA.
	’PCTCPB’	Percent current or MVA loading (according to the transformer percent loading units program option setting) of RATEB.
	’PCTCPC’	Percent current or MVA loading (according to the transformer percent loading units program option setting) of RATEC.
Real RVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’RVAL’ returned.
	IERR = 1	Bus not found; ’RVAL’ unchanged.
	IERR = 2	Three-winding transformer not found; ’RVAL’ unchanged.
	IERR = 3	Invalid ’STRING’; ’RVAL’ unchanged.
	IERR = 4	Error fetching transformer data; ’RVAL’ unchanged.
	IERR = 7	Branch out of service; ’RVAL’ returned set to zero.
	IERR = 8	No base kV at IBUS; ’RVAL’ returned = per unit current.
	IERR = 9	Rating is zero; ’RVAL’ returned set to zero.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.176 WNDDT2

Use this API to return three-winding transformer complex values for winding connected to bus IBUS.

Python syntax:

```
ierr, cmpval = wnddt2(ibus, jbus, kbus, ickt, string)
```

IPLAN syntax:

```
CALL WNDDT2(IBUS, JBUS, KBUS, ICKT, STRING, P, Q, IERR)
```

Fortran syntax:

```
CALL WNDDT2(IBUS, JBUS, KBUS, ICKT, STRING, CMPVAL, IERR)
```

where:

Integer IBUS	From bus number (input).	
Integer JBUS	To bus number (input).	
Integer KBUS	Third bus number (input).	
Character ICKT	Circuit identifier (input).	
Character STRING	String indicating the transformer winding quantity desired (input):	
	’RX’	Winding impedance in per unit.
	’RXZ’	Zero-sequence winding impedance.
	’COMPRX’	Compensating impedance.
	’FLOW’	Complex flow, in MW/Mvar.
	’O_FLOW’	Same as ’FLOW’, units determined by power output option.
Real P	Real component of complex value indicated by STRING (output).	
Real Q	Reactive component of complex value indicated by STRING (output).	
Complex CMPVAL	Complex value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’P’ and ’Q’ or ’CMPVAL’ returned.
	IERR = 1	Bus not found; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
	IERR = 2	Three-winding transformer not found; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
	IERR = 3	Invalid ’STRING’; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
	IERR = 4	Error fetching transformer data; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
	IERR = 5	Sequence data not in case for ’STRING’ = ’RXZ’; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
	IERR = 7	Branch out of service; ’P’ and ’Q’ of 0.0 or ’CMPVAL’ of (0.0,0.0) returned.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.177 WNDINT

Use this API to return three-winding transformer integer values for winding connected to bus IBUS.

Python syntax:

```
ierr, ival = wndint(ibus, jbus, kbus, ickt, string)
```

IPLAN syntax:

```
CALL WNDINT(IBUS, JBUS, KBUS, ICKT, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL WNDINT(IBUS, JBUS, KBUS, ICKT, STRING, IVAL, IERR)
```

where:

Integer IBUS	From bus number (input).
Integer JBUS	To bus number (input).
Integer KBUS	Third bus number (input).
Character ICKT	Circuit identifier (input).
Character STRING	String indicating the transformer winding quantity desired (input):
	'STATUS' — Status.
	'CNXCOD' — Winding connection code.
	'WNDNUM' — Winding number.
	'ICONT — Controlled bus number.
	'TABLE' — Transformer impedance correction table number (0 if none).
	'CNTRL' — Adjustment flag.
	'CNTRL2' — Transformer adjustment code.
	'NTPOSN' — Number of tap positions.
	'TPSTT' — Transformer winding ratio limit code: (-1) low; (0) regulating; (1) high; (-2) locked.
	'ANSTT' — Transformer winding angle limit code: (-1) low; (0) regulating; (1) high; (-2) locked.
Integer IVAL	Value indicated by STRING (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; 'IVAL' returned.
	IERR = 1 — Bus not found; 'IVAL' unchanged.
	IERR = 2 — Three-winding transformer not found; 'IVAL' unchanged.
	IERR = 3 — Invalid 'STRING'; 'IVAL' unchanged.
	IERR = 4 — Error fetching transformer data; 'IVAL' unchanged.
	IERR = 5 — Sequence data not in case for 'STRING' = 'CNXCOD'; 'IVAL' unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.178 XFRDAT

Use this API to return two-winding transformer real parameters.

Python syntax:

```
ierr, rval = xfrdat(ibus, jbus, ickt, string)
```

IPLAN syntax:

```
CALL XFRDAT(IBUS, JBUS, ICKT, STRING, RVAL, IERR)
```

Fortran syntax:

```
CALL XFRDAT(IBUS, JBUS, ICKT, STRING, RVAL, IERR)
```

where:

Integer IBUS	From bus number (input).
Integer JBUS	To bus number (input).
Character ICKT	Circuit identifier (input).
Character STRING	String indicating the transformer quantity desired (input):

'RATIO'	Winding one off-nominal turns ratio.
'RATIO2'	Winding two off-nominal turns ratio.
'ANGLE'	Winding one-phase shift angle, in degrees.
'RMAX'	Winding one ratio or angle upper limit.
'RMIN'	Winding one ratio or angle lower limit.
'VMAX'	Controlled quantity upper limit.
'VMIN'	Controlled quantity lower limit.
'STEP'	Winding one ratio or angle step width.
'CR'	Real component of load drop compensation impedance.
'CX'	Reactive component of load drop compensation impedance.
'CNXANG'	Winding connection angle, in degrees.
'SBASE1'	Winding one base MVA.
'NOMV1'	Winding one nominal voltage, in kV.
'NOMV2'	Winding two nominal voltage, in kV.
'GMAGNT'	Conductance component of transformer magnetizing admittance, in pu.
'BMAGNT'	Susceptance component of transformer magnetizing admittance, in pu.
'RG1', 'RGRND'	Grounding resistance at the winding 1 bus.
'XG1', 'XGRND'	Grounding reactance at the winding 1 bus.
'R01'	Zero sequence leakage resistance connected to the winding 1 bus.
'X01'	Zero sequence leakage reactance connected to the winding 1 bus.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	’RG2’, ’RGRND2’	Grounding resistance at the winding 2 bus.
	’XG2’, ’XGRND2’	Grounding reactance at the winding 1 bus.
	’R02’	Zero sequence leakage resistance connected to the winding 2 bus.
	’X02’	Zero sequence leakage reactance connected to the winding 2 bus.
	’RNUTRL’	Common neutral grounding resistance.
	’XNUTRL’	Common neutral grounding reactance.
Real RVAL	Value indicated by STRING (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’RVAL’ returned.
	IERR = 1	Bus not found; ’RVAL’ unchanged.
	IERR = 2	Branch not found; ’RVAL’ unchanged.
	IERR = 3	Branch not a transformer; ’RVAL’ unchanged.
	IERR = 4	Invalid value of ’STRING’; ’RVAL’ unchanged.
	IERR = 5	Sequence data not in case for STRING = ’RGRND’, ’XGRND’, ’RGRND2’, ’XGRND2’, ’RG1’, ’XG1’, ’R01’, ’X01’, ’RG2’, ’XG2’, ’R02’, ’X02’, ’RNUTRL’ or ’XNUTRL’; ’RVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.179 XFRINT

Use this API to return two-winding transformer integer parameters.

Python syntax:

```
ierr, ival = xfrint(ibus, jbus, ickt, string)
```

IPLAN syntax:

```
CALL XFRINT(IBUS, JBUS, KBUS, ICKT, STRING, IVAL, IERR)
```

Fortran syntax:

```
CALL XFRINT(IBUS, JBUS, KBUS, ICKT, STRING, IVAL, IERR)
```

where:

Integer IBUS	From bus number (input).	
Integer JBUS	To bus number (input).	
Character ICKT	Circuit identifier (input).	
Character STRING	String indicating the transformer quantity desired (input):	
	’ICONT’	Controlled bus number.
	’TAPPED’	Tapped side bus number.
	’GCIBUS’	Bus IBUS old grounding code (derived from CNXCOD).
	’GCJBUS’	Bus JBUS old grounding code (derived from CNXCOD).
	’TABLE’	Transformer impedance correction table number (0 if none).
	’CNTRL’	Adjustment enable flag.
	’CNTRL2’	Transformer adjustment code.
	’NTPOSN’	Number of tap positions.
	’CNXCOD	Connection code.
	’CW’	Winding data I/O code.
	’CZ’	Impedance data I/O code.
	’CM’	Magnetizing admittance I/O code.
	’CZ0’	Leakage impedance data I/O code.
	’CZG’	Grounding impedance data I/O code.
	’TPSTT’	Transformer winding one ratio limit code: (-1) low; (0) regulating; (1) high; (-2) locked; (-3) controlling dc line.
	’ANSTT’	Transformer winding one angle limit code: (-1) low; (0) regulating; (1) high; (-2) locked.
Integer IVAL	Value indicated by STRING (output).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output):	
	IERR = 0	No error; 'IVAL' returned.
	IERR = 1	Bus not found; 'IVAL' unchanged.
	IERR = 2	Branch not found; 'IVAL' unchanged.
	IERR = 3	Branch not a transformer; 'IVAL' unchanged.
	IERR = 4	Invalid value of 'STRING'; 'IVAL' unchanged.
	IERR = 5	Sequence data not in case for STRING = 'CNXCOD', 'CZ0', 'CZG', 'GCIBUS' or 'GCJBUS'; IVAL unchanged.
	IERR = 6	Connection code is 6 or greater for STRING = 'GCIBUS' or 'GCJBUS'; IVAL unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.180 XFRNAM

Use this API to return two-winding transformer name.

Python syntax:

```
ierr, string = xfrnam(ibus, jbus, ickt)
```

IPLAN syntax:

```
CALL XFRNAM(IBUS, JBUS, ICKT, STRING, IERR)
```

Fortran syntax:

```
CALL XFRNAM(IBUS, JBUS, ICKT, STRING, IERR)
```

where:

Integer IBUS	From bus number (input).
Integer JBUS	To bus number (input).
Character ICKT	Circuit identifier (input).
Character STRING	Transformer name (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; ’STRING’ returned.
	IERR = 1 — Bus not found; ’STRING’ unchanged.
	IERR = 2 — Transformer not found; ’STRING’ unchanged.
	IERR = 3 — Branch not a transformer; ’STRING’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.181 ZNDAT

Use this API to return zone totals.

Python syntax:

```
ierr, cmpval = zndat(iar, string)
```

IPLAN syntax:

```
CALL ZNDAT(IAR, STRING, P, Q, IERR)
```

Fortran syntax:

```
CALL ZNDAT(IAR, STRING, CMPVAL, IERR)
```

where:

Integer IAR	Zone number (input).	
Character STRING	String indicating the zone total desired (input):	
	’LOAD’	Total zone load by bus zone assignment.
	’LOADLD’	Total zone load by load zone assignment.
	’GEN’	Total zone generation.
	’LOSS’	Total zone losses.
	’INT’	Net zone interchange.
	’INDMAC’	Total zone induction machine powers by bus zone assignment.
	’INDMACMC’	Total zone induction machine powers by machine zone assignment.
	’INDGEN’	Total zone induction generator powers by bus zone assignment.
	’INDGENMC’	Total zone induction generator powers by machine zone assignment.
	’INDMOT’	Total zone induction motor powers by bus zone assignment.
	’INDMOTMC’	Total zone induction motor powers by machine zone assignment.
Real P	Real component of the desired complex power (output).	
Real Q	Reactive component of the desired complex power (output).	
Complex CMPVAL	Desired complex power (output).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output):	
	IERR = 0	No error; ’P’ and ’Q’ or ’CMPVAL’ returned.
	IERR = 1	Zone number ≤ 0 or > largest allowable zone number; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
	IERR = 2	No in-service buses with in-service loads (for ’LOAD’), no in-service loads (for ’LOADLD’), no type 2 or type 3 buses (for ’GEN’), no in-service branches (for ’LOSS’), no in-service ties (for ’INT’), no in-service buses with in-service induction machines of the appropriate type (for ’INDMAC’, ’INDGEN’ and ’INDMOT’), or no in-service induction machines of the appropriate type (for ’INDMACMC’, ’INDGENMC’ and ’INDMOTMC’) in zone; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
	IERR = 3	Zone not found; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
	IERR = 4	Bad ’STRING’ value; ’P’ and ’Q’ or ’CMPVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.182 ZNITOJ

Use this API to return the interchange between two zones.

Python syntax:

```
ierr, cmpval = znitoj(iar, jar)
```

IPLAN syntax:

```
CALL ZNITOJ(IAR, JAR, P, Q, IERR)
```

Fortran syntax:

```
CALL ZNITOJ(IAR, JAR, CMPVAL, IERR)
```

where:

Integer IAR	From zone number (input).
Integer JAR	To zone number (input).
Real P	Real component of the complex interchange (output).
Real Q	Reactive component of the complex interchange (output).
Complex CMPVAL	Complex interchange (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error; ’P’ and ’Q’ or ’CMPVAL’ returned.
	IERR = 1 — Zone number ≤ 0 or > largest allowable zone number; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
	IERR = 2 — No in-service branches between zones; ’P’ and ’Q’ or ’CMPVAL’ unchanged.
	IERR = 3 — Zone not found; ’P’ and ’Q’ or ’CMPVAL’ unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.183 ZONNAM

Use this API to return the zone name.

Python syntax:

```
ierr, cval = zonnam(iar)
```

IPLAN syntax:

```
CALL ZONNAM(IAR, CVAL, IERR)
```

Fortran syntax:

```
CALL ZONNAM(IAR, CVAL, IERR)
```

where:

Integer IAR	Zone number (input).
Character CVAL	Zone name (output).
Integer IERR	Is the error code (output):
	IERR = 0 — No error, ’CVAL’ is returned.
	IERR = 1 — Zone number ≤ 0 or > largest allowable zone number; ’CVAL’ is unchanged.
	IERR = 2 — Zone not found; ’CVAL’ is unchanged.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.184 ZONNUM

Use this API to return the zone number.

Python syntax:

```
ierr, iar = zonnum(string)
```

IPLAN syntax:

```
CALL ZONNUM(IAR, STRING, IERR)
```

Fortran syntax:

```
CALL ZONNUM(IAR, STRING, IERR)
```

where:

Integer IAR	Zone number (integer output).
Character STRING	Zone name (string input).
Integer IERR	Is the error code (integer output): IERR = 0 — No error, 'IAR' is returned. IERR = 1 — Zone with name 'STRING' not found, 'IAR' is unchanged. IERR > 1 — Multiple zones with the name 'STRING' found, 'IAR' returned as lowest number of these.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

7.185 ZONUSE

Use this API to indicate whether a zone is in use.

Python syntax:

```
ival  = zonuse(iar)
```

IPLAN syntax:

```
CALL ZONUSE(IAR, IVAL)
```

Fortran syntax:

```
CALL ZONUSE(IAR, IVAL)
```

where:

Integer IAR	Zone number (input).	
Integer IVAL	Zone status (output):	
	IVAL = 1	If any equipment is assigned to zone IAR.
	IVAL = 0	If no equipment is assigned to zone IAR.
	IVAL = -1	Zone number ≤ 0 or > largest allowable zone number.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

SIEMENS

Chapter 8

Subsystem Data Retrieval

8.1 Introduction

The subsystem data retrieval APIs are used to retrieve data from the PSS®E engine for elements (e.g., buses, branches, etc.) in a subsystem. One or more input data items to each API allow the user to restrict the elements for which data is returned to a subset of subsystem elements (e.g., to omit out-of-service elements).

These APIs are grouped into families with each family containing up to six APIs:

- An API that returns the number of array entries required to hold one set of subsystem data items returned by the data retrieval APIs in the family (e.g., GET_BUS_COUNT returns the number of buses in subsystem SID that meet the editing criterion indicated by FLAG).
- A data retrieval API that returns an array of one or more sets of integer data values.
- A data retrieval API that returns an array of one or more sets of real data values.
- A data retrieval API that returns an array of one or more sets of complex data values.
- A data retrieval API that returns an array of one or more sets of character data values.
- An API that returns information on the data types corresponding to specified STRING values, each of which may be specified to one or more of the data retrieval APIs in the family. STRING values that are valid input values for both the real and complex data retrieval APIs of a family are indicated as being of type complex.

Each of the data retrieval APIs returns an array of one or more columns containing subsystem data items. Input arguments specify the name of the array to be used, the number data items to be retrieved (which is also the minimum number of columns required in the array), and the number of elements in each column (which must be at least as large as the number returned by the first API listed above). Array elements beyond those required to accommodate the requested data items are left unchanged by the data retrieval APIs.

These routines do not record themselves when called with recording enabled.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.2 Character Data

Prior to calling any of the subsystem data retrieval APIs, space for the array that is to receive the set(s) of data items that are to be returned must be allocated. For character data items, the number of characters required for each array entry for the data corresponding to any of the STRING arguments of the character data retrieval APIs (i.e., the _CHAR APIs) may be obtained from the GET_CHAR_LENGTH API.

8.2.1 GET_CHAR_LENGTH

Use this API to return the number of characters required to accommodate the data corresponding to the specified STRING entries.

Python syntax:

```
ierr, nchars = get_char_length(string)
```

Fortran syntax:

```
CALL GET_CHAR_LENGTH(NSTR, STRING, NCHARS, IERR)
```

where:

Integer NSTR	Is the number of elements in STRING and NCHARS (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR valid strings for the _CHAR API to be used (input; no default allowed).
Integer NCHARS	Is an array of NSTR entries containing the number of characters of each element of the array associated with the corresponding STRING value (output).
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 through 50 Position of an element in the array STRING that contains a value that is not recognized by any of the _CHAR APIs. IERR = 99 Invalid NSTR value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.3 Bus Data

The bus data family of subsystem data retrieval APIs returns arrays of data for buses in a specified subsystem. One of the input items to each API allows the user to elect either to retrieve data for all subsystem buses, or to restrict the array entries to data for in-service buses.

Data for subsystem buses is returned from all APIs in the bus data family in the same order: ascending bus number order or alphabetic extended bus name order, according to the bus output option setting in effect at the time the API is referenced.

8.3.1 aBusCount

Use this API to return the number of array entries required to accommodate the data to be returned by the remaining members of the bus data family.

Python syntax:

```
ierr, buses = abuscount(sid, flag)
```

Fortran syntax:

```
CALL GET_BUS_COUNT(SID, FLAG, BUSES, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem buses to include (input; 1 by default). FLAG = 1, for only in-service buses. FLAG = 2, for all buses.
Integer BUSES	Is the number of buses in the subsystem indicated by SID that meet the editing criteria indicated by FLAG (output).
Integer IERR	Is the error code (output): IERR = 0 No error, BUSES returned IERR = 1 Working case is empty; BUSES returned as 0. IERR = 2 Invalid SID value; BUSES returned as 0. IERR = 3 Invalid FLAG value; BUSES returned as 0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.3.2 aBusInt

Use this API to return an array of integer values for subsystem buses.

Python syntax:

```
ierr, iarray = abusint(sid, flag, string)
```

Fortran syntax:

```
CALL GET_BUS_INTEGER(SID,FLAG,NSTR,STRING,DIM,IARRAY,BUSES,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem buses to include (input; 1 by default). FLAG = 1, for only in-service buses. FLAG = 2, for all buses.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus quantities desired (input; no default allowed): 'NUMBER' Bus number. 'TYPE' Bus type code. 'AREA' Bus area number. 'ZONE' Bus zone number. 'OWNER' Bus owner number. 'DUMMY' Returns 1 if the bus is a dummy bus of a multisection line, or 0 if it is not.
Integer DIM	Is the number of elements in each column of the array IARRAY (input; no default allowed). DIM must be as least as large as the value of BUSES returned by the GET_BUS_COUNT API when it is called with the same values specified for SID and FLAG.
Integer IARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BUSES	Is the number of elements in each column of IARRAY that are set by this API (output). BUSES is identical to the value of BUSES returned by the GET_BUS_COUNT API when it is called with the same values specified for SID and FLAG.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output):	
	IERR = 0	No error.
	IERR = 1	Working case is empty.
	IERR = 2	Invalid SID value.
	IERR = 3	Invalid FLAG value.
	IERR = 4	Invalid NSTR value.
	IERR = 5	DIM, and hence the size of IARRAY, is not large enough.
	IERR = 6	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.3.3 aBusReal

Use this API to return an array of real values for subsystem buses.

Python syntax:

```
ierr, rarray = abusreal(sid, flag, string)
```

Fortran syntax:

```
CALL GET_BUS_REAL(SID,FLAG,NSTR,STRING,DIM,RARRAY,BUSES,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem buses to include (input; 1 by default). FLAG = 1, for only in-service buses. FLAG = 2, for all buses.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus quantities desired (input; no default allowed): ’BASE’ — Bus base voltage, in kV. ’PU’ — Actual bus voltage magnitude, in pu. ’KV’ — Actual bus voltage magnitude, in kV (in pu if base voltage is 0.0). ’ANGLE’ — Bus voltage phase angle, in radians. ’ANGLED’ — Bus voltage phase angle, in degrees. ’NVLMHI’ — Normal high voltage limit, in pu. ’NVLMLO’ — Normal low voltage limit, in pu. ’EVLMHI’ — Emergency high voltage limit, in pu. ’EVLMLO’ — Emergency low voltage limit, in pu. ’MISMATCH’ — Bus mismatch in MVA (0.0 if bus type code > 3). ’O_MISMATCH’ — Bus mismatch, in units determined by the power output option setting (0.0 if bus type code > 3).
Integer DIM	Is the number of elements in each column of the array RARRAY (input; no default allowed). DIM must be as least as large as the value of BUSES returned by the GET_BUS_COUNT API when it is called with the same values specified for SID and FLAG.
Real RARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer BUSES	Is the number of elements in each column of RARRAY that are set by this API (output). BUSES is identical to the value of BUSES returned by the GET_BUS_COUNT API when it is called with the same values specified for SID and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid FLAG value. IERR = 4 Invalid NSTR value. IERR = 5 DIM, and hence the size of RARRAY, is not large enough. IERR = 6 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.3.4 aBusCplx

Use this API to return an array of complex values for subsystem buses.

Python syntax:

```
ierr, xarray = abuscplx(sid, flag, string)
```

Fortran syntax:

```
CALL GET_BUS_COMPLEX(SID,FLAG,NSTR,STRING,DIM,XARRAY,BUSES,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem buses to include (input; 1 by default). FLAG = 1, for only in-service buses. FLAG = 2, for all buses.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus quantities desired (input; no default allowed): 'VOLTAGE' — Actual bus voltage, in pu, rectangular coordinates. 'SHUNTACT' — Actual total in-service fixed bus shunt, in MW and Mvar. 'O_SHUNTACT' — Actual total in-service fixed bus shunt, in units determined by the power output option setting. 'SHUNTNOM' — Nominal total in-service fixed bus shunt, in MW and Mvar at 1.0 pu voltage. 'O_SHUNTNOM' — Nominal total in-service fixed bus shunt, in units determined by the power output option setting at 1.0 pu voltage. 'SHUNTN' — Total in-service negative sequence load (pu, nominal). 'SHUNTZ' — Total in-service zero sequence load (pu, nominal). 'MISMATCH' — Bus mismatch, in MW and Mvar (0.0 if bus type code > 3). 'O_MISMATCH' — Bus mismatch, in units determined by the power output option setting (0.0 if bus type code > 3).
Integer DIM	Is the number of elements in each column of the array XARRAY (input; no default allowed). DIM must be as least as large as the value of BUSES returned by the GET_BUS_COUNT API when it is called with the same values specified for SID and FLAG.
Complex XARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer BUSES — Is the number of elements in each column of XARRAY that are set by this API (output).

BUSES is identical to the value of BUSES returned by the GET_BUS_COUNT API when it is called with the same values specified for SID and FLAG.

Integer IERR — Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid FLAG value.
IERR = 4	Invalid NSTR value.
IERR = 5	DIM, and hence the size of XARRAY, is not large enough.
IERR = 6	Invalid STRING value.
IERR = 7	Sequence data not in case (when STRING = ’SHUNTN’ or ’SHUNTZ’.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.3.5 aBusChar

Use this API to return an array of character values for subsystem buses.

Python syntax:

```
ierr, carray = abuschar(sid, flag, string)
```

Fortran syntax:

```
CALL GET_BUS_CHAR(SID,FLAG,NSTR,STRING,DIM,CARRAY,BUSES,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem buses to include (input; 1 by default). FLAG = 1, for only in-service buses. FLAG = 2, for all buses.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus quantities desired (input; no default allowed): 'NAME' Bus name (12 characters). 'EXNAME' Extended bus name (18 characters).
Integer DIM	Is the number of elements in each column of the array CARRAY (input; no default allowed). DIM must be as least as large as the value of BUSES returned by the GET_BUS_COUNT API when it is called with the same values specified for SID and FLAG.
Character CARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BUSES	Is the number of elements in each column of CARRAY that are set by this API (output). BUSES is identical to the value of BUSES returned by the GET_BUS_COUNT API when it is called with the same values specified for SID and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid FLAG value. IERR = 4 Invalid NSTR value. IERR = 5 DIM, and hence the size of CARRAY, is not large enough. IERR = 6 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.3.6 aBusTypes

Use this API to return an array of character values indicating the data types corresponding to a set of specified STRING values that are valid input values to any of the data retrieval routines of the bus data family (aBusInt, aBusReal, aBusCplx and aBusChar).

Python syntax:

```
ierr, types = abustypes(string)
```

Fortran syntax:

```
CALL GET_BUS_TYPES(NSTR,STRING,TYPES,IERR)
```

where:

Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR strings that are valid STRING values for any of the data retrieval routines of the bus data family (input; no default allowed):
Character TYPES*1	Is an array of at least NSTR elements, each of which contains one of the following characters indicating the data type of the data associated with the corresponding element of STRING (output): 'I' Integer. 'R' Real. 'X' Complex. 'C' Character. '?' Invalid STRING value.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = n Position of the first element in STRING that contains a value that is not a valid STRING value for any of the data retrieval routines in the bus data family (1 ≤ n ≤ NSTR). IERR = 99 Invalid NSTR value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.4 Plant Bus Data

The plant bus data family of subsystem data retrieval APIs returns arrays of data for buses in a specified subsystem. One of the input items to each API allows the user to elect either to retrieve data for all subsystem buses, or to restrict the array entries to data for a subset of buses according to their bus, plant, and/or machine configuration and statuses.

Data for subsystem buses is returned from all APIs in the plant bus data family in the same order: ascending bus number order or alphabetic extended bus name order, according to the bus output option setting in effect at the time the API is referenced.

8.4.1 aGenBusCount

Use this API to return the number of array entries required to accommodate the data to be returned by the remaining members of the plant bus data family.

Python syntax:

```
ierr, buses = agenbuscount(sid, flag)
```

Fortran syntax:

```
CALL GET_PLANT_BUS_COUNT(SID, FLAG, BUSES, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem buses to include (input; 1 by default). FLAG = 1 for only in-service plant buses (type code 2 or 3) with at least one in-service machine. FLAG = 2 for only in-service plant buses (type code 2 or 3), including those with no in-service machines. FLAG = 3 for all in-service buses, including those that are not plant buses. FLAG = 4 for all plant buses, including those with no in-service machines. FLAG = 5 for all buses, including those that are not plant buses.
Integer BUSES	Is the number of buses in the subsystem indicated by SID that meet the editing criteria indicated by FLAG (output).
Integer IERR	Is the error code (output): IERR = 0 No error, BUSES returned. IERR = 1 Working case is empty; BUSES returned as 0. IERR = 2 Invalid SID value; BUSES returned as 0. IERR = 3 Invalid FLAG value; BUSES returned as 0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.4.2 aGenBusInt

Use this API to return an array of integer values for subsystem buses.

Python syntax:

```
ierr, iarray = agenbusint(sid, flag, string)
```

Fortran syntax:

```
CALL
GET_PLANT_BUS_INTEGER(SID,FLAG,NSTR,STRING,DIM,IARRAY,BUSES,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem buses to include (input; 1 by default). FLAG = 1 — for only in-service plant buses (type code 2 or 3) with at least one in-service machine. FLAG = 2 — for only in-service plant buses (type code 2 or 3), including those with no in-service machines. FLAG = 3 — for all in-service buses, including those that are not plant buses. FLAG = 4 — for all plant buses, including those with no in-service machines. FLAG = 5 — for all buses, including those that are not plant buses.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or plant quantities desired (input; no default allowed): 'NUMBER' — Bus number. 'TYPE' — Bus type code. 'AREA' — Bus area number. 'ZONE' — Bus zone number. 'OWNER' — Bus owner number. 'DUMMY' — Returns 1 if the bus is a dummy bus of a multisection line, or 0 if it is not. 'STATUS' — Inferred plant status. Returns 1 if the bus type code is 2 or 3, and at least one in-service machine is present, or 0 otherwise. 'IREG' — Bus number of the regulated bus.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer DIM	Is the number of elements in each column of the array IARRAY (input; no default allowed). DIM must be as least as large as the value of BUSES returned by the GET_PLANT_BUS_COUNT API when it is called with the same values specified for SID and FLAG.
Integer IARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BUSES	Is the number of elements in each column of IARRAY that are set by this API (output). BUSES is identical to the value of BUSES returned by the GET_PLANT_BUS_COUNT API when it is called with the same values specified for SID and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid FLAG value. IERR = 4 Invalid NSTR value. IERR = 5 DIM, and hence the size of IARRAY, is not large enough. IERR = 6 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.4.3 aGenBusReal

Use this API to return an array of real values for subsystem buses.

Python syntax:

```
ierr, rarray = agenbusreal(sid, flag, string)
```

Fortran syntax:

```
CALL GET_PLANT_BUS_REAL(SID,FLAG,NSTR,STRING,DIM,RARRAY,BUSES,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem buses to include (input; 1 by default). FLAG = 1 for only in-service plant buses (type code 2 or 3) with at least one in-service machine. FLAG = 2 for only in-service plant buses (type code 2 or 3), including those with no in-service machines. FLAG = 3 for all in-service buses, including those that are not plant buses. FLAG = 4 for all plant buses, including those with no in-service machines. FLAG = 5 for all buses, including those that are not plant buses.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or plant quantities desired (input; no default allowed): 'BASE' Bus base voltage, in kV. 'PU' Actual bus voltage magnitude, in pu. 'KV' Actual bus voltage magnitude, in kV (in pu if base voltage is 0.0). 'ANGLE' Bus voltage phase angle, in radians. 'ANGLED' Bus voltage phase angle, in degrees. 'PERCENT' MVA loading as a percentage of total plant MVA base, MBASE. 'IREGBASE' Regulated bus base voltage, in kV. 'IREGPU' Actual regulated bus voltage magnitude, in pu. 'IREGKV' Actual regulated bus voltage magnitude, in kV (in pu if base voltage is 0.0). 'VSPU' Regulated bus scheduled voltage magnitude, in pu. 'VSKV' Regulated bus scheduled voltage magnitude, in kV (in pu if base voltage is 0.0). 'RMPCT' Percent of total Mvar to be contributed by this plant.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

For the following, values are returned in MW, Mvar, or MVA:

'PGEN'	Active power output of in-service machines at this plant.
'QGEN'	Reactive power output of in-service machines at this plant.
'MVA'	|PGEN + j QGEN| of in-service machines at this plant.
'PMAX'	Active power upper limit of in-service machines at this plant.
'PMIN'	Active power lower limit of in-service machines at this plant.
'QMAX'	Reactive power upper limit of in-service machines at this plant.
'QMIN'	Reactive power lower limit of in-service machines at this plant.
'MISMATCH'	Bus mismatch in MVA (0.0 if bus type code > 3).

For the following, values are returned in units determined by the power output option setting:

'O_PGEN'	Active power output of in-service machines at this plant.
'O_QGEN'	Reactive power output of in-service machines at this plant.
'O_MVA'	|PGEN + j QGEN| of in-service machines at this plant.
'O_PMAX'	Active power upper limit of in-service machines at this plant.
'O_PMIN'	Active power lower limit of in-service machines at this plant.
'O_QMAX'	Reactive power upper limit of in-service machines at this plant.
'O_QMIN'	Reactive power lower limit of in-service machines at this plant.
'O_MISMATCH'	Bus mismatch (0.0 if bus type code > 3).

Integer DIM

Is the number of elements in each column of the array RARRAY (input; no default allowed).

DIM must be as least as large as the value of BUSES returned by the GET_PLANT_BUS_COUNT API when it is called with the same values specified for SID and FLAG.

Real RARRAY

Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer BUSES

Is the number of elements in each column of RARRAY that are set by this API (output).

BUSES is identical to the value of BUSES returned by the GET_PLANT_BUS_COUNT API when it is called with the same values specified for SID and FLAG.

Integer IERR

Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid FLAG value.
IERR = 4	Invalid NSTR value.
IERR = 5	DIM, and hence the size of RARRAY, is not large enough.
IERR = 6	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.4.4 aGenBusCplx

Use this API to return an array of complex values for subsystem buses.

Python syntax:

```
ierr, xarray = agenbuscplx(sid, flag, string)
```

Fortran syntax:

```
CALL
GET_PLANT_BUS_COMPLEX(SID,FLAG,NSTR,STRING,DIM,XARRAY,BUSES,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem buses to include (input; 1 by default). FLAG = 1 — for only in-service plant buses (type code 2 or 3) with at least one in-service machine. FLAG = 2 — for only in-service plant buses (type code 2 or 3), including those with no in-service machines. FLAG = 3 — for all in-service buses, including those that are not plant buses. FLAG = 4 — for all plant buses, including those with no in-service machines. FLAG = 5 — for all buses, including those that are not plant buses.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or plant quantities desired (input; no default allowed): 'VOLTAGE' — Actual bus voltage, in pu, rectangular coordinates. For the following, values are returned in MW and Mvar: 'PQGEN' — Power output of in-service machines at this plant. 'MISMATCH' — Bus mismatch (0.0 if bus type code > 3). For the following, values are returned in units determined by the power output option setting: 'O_PQGEN' — Power output of in-service machines at this plant. 'O_MISMATCH' — Bus mismatch (0.0 if bus type code > 3).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer DIM	Is the number of elements in each column of the array XARRAY (input; no default allowed). DIM must be as least as large as the value of BUSES returned by the GET_PLANT_BUS_COUNT API when it is called with the same values specified for SID and FLAG.
Complex XARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BUSES	Is the number of elements in each column of XARRAY that are set by this API (output). BUSES is identical to the value of BUSES returned by the GET_PLANT_BUS_COUNT API when it is called with the same values specified for SID and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid FLAG value. IERR = 4 Invalid NSTR value. IERR = 5 DIM, and hence the size of XARRAY, is not large enough. IERR = 6 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.4.5 aGenBusChar

Use this API to return an array of character values for subsystem buses.

Python syntax:

```
ierr, carray = agenbuschar(sid, flag, string)
```

Fortran syntax:

```
CALL GET_PLANT_BUS_CHAR(SID,FLAG,NSTR,STRING,DIM,CARRAY,BUSES,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default).	
	SID =	a negative value, to instruct the API to assume a subsystem containing all buses in the working case.
	SID =	a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem buses to include (input; 1 by default).	
	FLAG = 1	for only in-service plant buses (type code 2 or 3) with at least one in-service machine.
	FLAG = 2	for only in-service plant buses (type code 2 or 3), including those with no in-service machines.
	FLAG = 3	for all in-service buses, including those that are not plant buses.
	FLAG = 4	for all plant buses, including those with no in-service machines.
	FLAG = 5	for all buses, including those that are not plant buses.
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).	
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or plant quantities desired (input; no default allowed):	
	'NAME'	Bus name (12 characters).
	'EXNAME'	Extended bus name (18 characters).
	'IREGNAME'	Bus name of regulated bus (12 characters).
	'IREGEXNAME'	Extended bus name of regulated bus (18 characters).
Integer DIM	Is the number of elements in each column of the array CARRAY (input; no default allowed). DIM must be as least as large as the value of BUSES returned by the GET_PLANT_BUS_COUNT API when it is called with the same values specified for SID and FLAG.	
Character CARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer BUSES	Is the number of elements in each column of CARRAY that are set by this API (output). BUSES is identical to the value of BUSES returned by the GET_PLANT_BUS_COUNT API when it is called with the same values specified for SID and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid FLAG value. IERR = 4 Invalid NSTR value. IERR = 5 DIM, and hence the size of CARRAY, is not large enough. IERR = 6 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.4.6 aGenBusTypes

Use this API to return an array of character values indicating the data types corresponding to a set of specified STRING values that are valid input values to any of the data retrieval routines of the plant bus data family (aGenBusInt, aGenBusReal, aGenBusCplx and aGenBusChar).

Python syntax:

```
ierr, types = agenbustypes(string)
```

Fortran syntax:

```
CALL GET_PLANT_BUS_TYPES(NSTR,STRING,TYPES,IERR)
```

where:

Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR strings that are valid STRING values for any of the data retrieval routines of the plant bus data family (input; no default allowed):
Character TYPES*1	Is an array of at least NSTR elements, each of which contains one of the following characters indicating the data type of the data associated with the corresponding element of STRING (output): 'I' Integer. 'R' Real. 'X' Complex. 'C' Character. '?' Invalid STRING value.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = n Position of the first element in STRING that contains a value that is not a valid STRING value for any of the data retrieval routines in the plant bus data family (1 ≤ n ≤ NSTR). IERR = 99 Invalid NSTR value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.5 Machine Data

The machine data family of subsystem data retrieval APIs returns arrays of data for machines in a specified subsystem. One of the input items to each API allows the user to elect either to retrieve data for all subsystem machines, or to restrict the array entries to data for a subset of machines according to their bus, plant, and/or machine configuration and statuses.

Data for subsystem machines is returned from all APIs in the machine data family in the same order: ascending bus number order or alphabetic extended bus name order, according to the bus output option setting in effect at the time the API is referenced; multiple machines at the same bus are in ascending machine identifier order.

8.5.1 aMachCount

Use this API to return the number of array entries required to accommodate the data to be returned by the remaining members of the machine data family.

Python syntax:

```
ierr, machs = amachcount(sid, flag)
```

Fortran syntax:

```
CALL GET_MACHINE_COUNT(SID, FLAG, MACHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default).	
	SID =	a negative value, to instruct the API to assume a subsystem containing all buses in the working case.
	SID =	a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem machines to include (input; 1 by default).	
	FLAG = 1	for only in-service machines at in-service plants (type code 2 or 3).
	FLAG = 2	for all machines at in-service plants (type code 2 or 3).
	FLAG = 3	for only in-service machines, including those at type 1 and 4 buses.
	FLAG = 4	for all machines.
Integer MACHS	Is the number of machines in the subsystem indicated by SID that meet the editing criteria indicated by FLAG (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error, MACHS returned.
	IERR = 1	Working case is empty; MACHS returned as 0.
	IERR = 2	Invalid SID value; MACHS returned as 0.
	IERR = 3	Invalid FLAG value; MACHS returned as 0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.5.2 aMachInt

Use this API to return an array of integer values for subsystem machines.

Python syntax:

```
ierr, iarray = amachint(sid, flag, string)
```

Fortran syntax:

```
CALL GET_MACHINE_INTEGER(SID,FLAG,NSTR,STRING,DIM,IARRAY,MACHS,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem machines to include (input; 1 by default). FLAG = 1 — for only in-service machines at in-service plants (type code 2 or 3). FLAG = 2 — for all machines at in-service plants (type code 2 or 3). FLAG = 3 — for only in-service machines, including those at Type 1 and 4 buses. FLAG = 4 — for all machines.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or machine quantities desired (input; no default allowed): 'NUMBER' — Bus number. 'STATUS' — Machine status. 'WMOD' — Wind machine reactive power limits mode; 0 if the machine is not a wind machine. 'OWNERS' — Number of owners. 'OWN1' — First owner. 'OWN2' — Second owner. 'OWN3' — Third owner. 'OWN4' — Fourth owner. 'CZG' — Grounding impedance data input/outout (I/O) code (1 for per unit, 2 for ohms).
Integer DIM	Is the number of elements in each column of the array IARRAY (input; no default allowed). DIM must be as least as large as the value of MACHS returned by the GET_MACHINE_COUNT API when it is called with the same values specified for SID and FLAG.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer MACHS	Is the number of elements in each column of IARRAY that are set by this API (output). MACHS is identical to the value of MACHS returned by the GET_MACHINE_COUNT API when it is called with the same values specified for SID and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid FLAG value. IERR = 4 Invalid NSTR value. IERR = 5 DIM, and hence the size of IARRAY, is not large enough. IERR = 6 Invalid STRING value. IERR = 7 Sequence data not in case (when STRING = ’CZG’).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.5.3 aMachReal

Use this API to return an array of real values for subsystem machines.

Python syntax:

```
ierr, rarray = amachreal(sid, flag, string)
```

Fortran syntax:

```
CALL GET_MACHINE_REAL(SID,FLAG,NSTR,STRING,DIM,RARRAY,MACHS,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem machines to include (input; 1 by default). FLAG = 1 for only in-service machines at in-service plants (type code 2 or 3). FLAG = 2 for all machines at in-service plants (type code 2 or 3). FLAG = 3 for only in-service machines, including those at Type 1 and 4 buses. FLAG = 4 for all machines.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)

Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or machine quantities desired (input; no default allowed):

'FRACT1'	First owner fraction.
'FRACT2'	Second owner fraction.
'FRACT3'	Third owner fraction.
'FRACT4'	Fourth owner fraction.
'PERCENT'	MVA loading as a percentage of machine MVA base, MBASE.
'MBASE'	Machine MVA base.
'GENTAP'	Step-up transformer tap ratio.
'WPF'	Power factor used in setting reactive power limits for the wind machine when its WMOD is 2 or 3; 1.0 if the machine is not a wind machine.
'RPOS'	Positive sequence fault analysis machine resistance.
'XSUBTR'	Positive sequence fault analysis machine subtransient reactance.
'XTRANS'	Positive sequence fault analysis machine transient reactance.
'XSYNCH'	Positive sequence fault analysis machine synchronous reactance.

For the following, values are returned in MW, Mvar or MVA:

'PGEN'	Active power output.
'QGEN'	Reactive power output.
'MVA'	|PGEN + j QGEN|.
'PMAX'	Active power upper limit.
'PMIN'	Active power lower limit.
'QMAX'	Reactive power upper limit.
'QMIN'	Reactive power lower limit.

For the following, values are returned in units determined by the power output option setting:

'O_PGEN'	Active power output.
'O_QGEN'	Reactive power output.
'O_MVA'	|PGEN + j QGEN|.
'O_PMAX'	Active power upper limit.
'O_PMIN'	Active power lower limit.
'O_QMAX'	Reactive power upper limit.
'O_QMIN'	Reactive power lower limit.

Integer DIM

Is the number of elements in each column of the array RARRAY (input; no default allowed).

DIM must be as least as large as the value of MACHS returned by the GET_MACHINE_COUNT API when it is called with the same values specified for SID and FLAG.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real RARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer MACHS	Is the number of elements in each column of RARRAY that are set by this API (output). MACHS is identical to the value of MACHS returned by the GET_MACHINE_COUNT API when it is called with the same values specified for SID and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid FLAG value. IERR = 4 Invalid NSTR value. IERR = 5 DIM, and hence the size of RARRAY, is not large enough. IERR = 6 Invalid STRING value. IERR = 7 Sequence data not in case (when STRING = ’RPOS’, ’XSUBTR’, ’XTRANS’, or ’XSYNCH’).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.5.4 aMachCplx

Use this API to return an array of complex values for subsystem machines.

Python syntax:

```
ierr, xarray = amachcplx(sid, flag, string)
```

Fortran syntax:

```
CALL GET_MACHINE_COMPLEX(SID,FLAG,NSTR,STRING,DIM,XARRAY,MACHS,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem machines to include (input; 1 by default). FLAG = 1 for only in-service machines at in-service plants (type code 2 or 3). FLAG = 2 for all machines at in-service plants (type code 2 or 3). FLAG = 3 for only in-service machines, including those at Type 1 and 4 buses. FLAG = 4 for all machines.
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or machine quantities desired (input; no default allowed): ’ZSORCE’ — Machine impedance (ZR + j ZX). ’XTRAN’ — Step-up transformer impedance (RT + j XT). ’ZPOS’ — Positive sequence fault analysis machine impedance (Rpos + j X"). ’ZNEG’ — Negative sequence machine impedance (ZRNEG + j ZXNEG). ’ZZERO’ — Zero sequence machine impedance (RZERO + j XZERO). ’ZGRND’ — Grounding impedance. If the grounding impedance data I/O code is 2 and the bus base voltage is positive, the impedance is returned in ohms; otherwise, it is return in per unit.. ’ZGRNDPU’ — Grounding impedance in per unit. For the following, values are returned in MW and Mvar: ’PQGEN’ — Power output. For the following, values are returned in units determined by the power output option setting: ’O_PQGEN’ — Power output.
Integer DIM	Is the number of elements in each column of the array XARRAY (input; no default allowed). DIM must be as least as large as the value of MACHS returned by the GET_MACHINE_COUNT API when it is called with the same values specified for SID and FLAG.
Complex XARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer MACHS	Is the number of elements in each column of XARRAY that are set by this API (output). MACHS is identical to the value of MACHS returned by the GET_MACHINE_COUNT API when it is called with the same values specified for SID and FLAG.
Integer IERR	Is the error code (output): IERR = 0 — No error. IERR = 1 — Working case is empty. IERR = 2 — Invalid SID value. IERR = 3 — Invalid FLAG value. IERR = 4 — Invalid NSTR value. IERR = 5 — DIM, and hence the size of XARRAY, is not large enough. IERR = 6 — Invalid STRING value. IERR = 7 — Sequence data not in case (when STRING = ’ZPOS’, ’ZNEG’, ’ZZERO’, ’ZGRND’ or ’ZGRNDPU’).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.5.5 aMachChar

Use this API to return an array of character values for subsystem machines.

Python syntax:

```
ierr, carray = amachchar(sid, flag, string)
```

Fortran syntax:

```
CALL GET_MACHINE_CHAR(SID,FLAG,NSTR,STRING,DIM,CARRAY,MACHS,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem machines to include (input; 1 by default). FLAG = 1 for only in-service machines at in-service plants (type code 2 or 3). FLAG = 2 for all machines at in-service plants (type code 2 or 3). FLAG = 3 for only in-service machines, including those at Type 1 and 4 buses. FLAG = 4 for all machines.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or machine quantities desired (input; no default allowed): 'ID' Machine identifier (2 characters). 'NAME' Bus name (12 characters). 'EXNAME' Extended bus name (18 characters).
Integer DIM	Is the number of elements in each column of the array CARRAY (input; no default allowed). DIM must be as least as large as the value of MACHS returned by the GET_MACHINE_COUNT API when it is called with the same values specified for SID and FLAG.
Character CARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer MACHS	Is the number of elements in each column of CARRAY that are set by this API (output). MACHS is identical to the value of MACHS returned by the GET_MACHINE_COUNT API when it is called with the same values specified for SID and FLAG.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output):	
	IERR = 0	No error.
	IERR = 1	Working case is empty.
	IERR = 2	Invalid SID value.
	IERR = 3	Invalid FLAG value.
	IERR = 4	Invalid NSTR value.
	IERR = 5	DIM, and hence the size of CARRAY, is not large enough.
	IERR = 6	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.5.6 aMachTypes

Use this API to return an array of character values indicating the data types corresponding to a set of specified STRING values that are valid input values to any of the data retrieval routines of the machine data family (aMachInt, aMachReal, aMachCplx and aMachChar).

Python syntax:

```
ierr, types = amachtypes(string)
```

Fortran syntax:

```
CALL GET_MACHINE_TYPES(NSTR,STRING,TYPES,IERR)
```

where:

Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR strings that are valid STRING values for any of the data retrieval routines of the machine data family (input; no default allowed):
Character TYPES*1	Is an array of at least NSTR elements, each of which contains one of the following characters indicating the data type of the data associated with the corresponding element of STRING (output): 'I' Integer. 'R' Real. 'X' Complex. 'C' Character. '?' Invalid STRING value.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = n Position of the first element in STRING that contains a value that is not a valid STRING value for any of the data retrieval routines in the machine data family (1 ≤ n ≤ NSTR). IERR = 99 Invalid NSTR value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.6 Load Bus Data

The load bus data family of subsystem data retrieval APIs returns arrays of data for buses in a specified subsystem. One of the input items to each API allows the user to elect either to retrieve data for all subsystem buses, or to restrict the array entries to data for a subset of buses according to their bus and/or load configuration and statuses.

Data for subsystem buses is returned from all APIs in the load bus data family in the same order: ascending bus number order or alphabetic extended bus name order, according to the bus output option setting in effect at the time the API is referenced.

8.6.1 aLodBusCount

Use this API to return the number of array entries required to accommodate the data to be returned by the remaining members of the load bus data family.

Python syntax:

```
ierr, buses = alodbuscount(sid, flag)
```

Fortran syntax:

```
CALL GET_LOAD_BUS_COUNT(SID, FLAG, BUSES, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem buses to include (input; 1 by default).

FLAG = 1	for only in-service load buses (type code 1, 2 or 3) with at least one in-service load.
FLAG = 2	for only in-service load buses (type code 1, 2 or 3), including those with only out-of-service loads.
FLAG = 3	for all in-service buses, including those that are not load buses.
FLAG = 4	for all load buses, including those with only out-of-service loads.
FLAG = 5	for all buses, including those that are not load buses.
FLAG = 10	for only in-service load buses (type code 1, 2 or 3) with at least one in-service scalable load.
FLAG = 11	for only in-service load buses (type code 1, 2 or 3) with at least one scalable load, including those with all such loads out-of-service.
FLAG = 12	for all load buses with at least one scalable load, including those with all such loads out-of-service.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	FLAG = 20	for only in-service load buses (type code 1, 2 or 3) with at least one in-service fixed load.
	FLAG = 21	for only in-service load buses (type code 1, 2 or 3) with at least one fixed load, including those with all such loads out-of-service.
	FLAG = 22	for all load buses with at least one fixed load, including those with all such loads out-of-service.
Integer BUSES	Is the number of buses in the subsystem indicated by SID that meet the editing criteria indicated by FLAG (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error, BUSES returned.
	IERR = 1	Working case is empty; BUSES returned as 0.
	IERR = 2	Invalid SID value; BUSES returned as 0.
	IERR = 3	Invalid FLAG value; BUSES returned as 0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.6.2 aLodBusInt

Use this API to return an array of integer values for subsystem buses.

Python syntax:

```
ierr, iarray = alodbusint(sid, flag, string)
```

Fortran syntax:

```
CALL GET_LOAD_BUS_INTEGER(SID,FLAG,NSTR,STRING,DIM,IARRAY,BUSES,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.	
Integer FLAG	Is a flag indicating which subsystem buses to include (input; 1 by default).	
	FLAG = 1	for only in-service load buses (type code 1, 2 or 3) with at least one in-service load.
	FLAG = 2	for only in-service load buses (type code 1, 2 or 3), including those with only out-of-service loads.
	FLAG = 3	for all in-service buses, including those that are not load buses.
	FLAG = 4	for all load buses, including those with only out-of-service loads.
	FLAG = 5	for all buses, including those that are not load buses.
	FLAG = 10	for only in-service load buses (type code 1, 2 or 3) with at least one in-service scalable load.
	FLAG = 11	for only in-service load buses (type code 1, 2 or 3) with at least one scalable load, including those with all such loads out-of-service.
	FLAG = 12	for all load buses with at least one scalable load, including those with all such loads out-of-service.
	FLAG = 20	for only in-service load buses (type code 1, 2 or 3) with at least one in-service fixed load.
	FLAG = 21	for only in-service load buses (type code 1, 2 or 3) with at least one fixed load, including those with all such loads out-of-service.
	FLAG = 22	for all load buses with at least one fixed load, including those with all such loads out-of-service.
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or load quantities desired (input; no default allowed): 'NUMBER' Bus number. 'TYPE' Bus type code. 'AREA' Bus area number. 'ZONE' Bus zone number. 'OWNER' Bus owner number. 'DUMMY' Returns 1 if the bus is a dummy bus of a multisection line, or 0 if it is not. 'STATUS' Inferred load status. Returns 1 if the bus type code is 1, 2 or 3, and at least one in-service load is present, or 0 otherwise.
Integer DIM	Is the number of elements in each column of the array IARRAY (input; no default allowed). DIM must be as least as large as the value of BUSES returned by the GET_LOAD_BUS_COUNT API when it is called with the same values specified for SID and FLAG.
Integer IARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BUSES	Is the number of elements in each column of IARRAY that are set by this API (output). BUSES is identical to the value of BUSES returned by the GET_LOAD_BUS_COUNT API when it is called with the same values specified for SID and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid FLAG value. IERR = 4 Invalid NSTR value. IERR = 5 DIM, and hence the size of IARRAY, is not large enough. IERR = 6 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.6.3 aLodBusReal

Use this API to return an array of real values for subsystem buses.

Python syntax:

```
ierr, rarray = alodbusreal(sid, flag, string)
```

Fortran syntax:

```
CALL GET_LOAD_BUS_REAL(SID,FLAG,NSTR,STRING,DIM,RARRAY,BUSES,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem buses to include (input; 1 by default).
	FLAG = 1 — for only in-service load buses (type code 1, 2 or 3) with at least one in-service load.
	FLAG = 2 — for only in-service load buses (type code 1, 2 or 3), including those with only out-of-service loads.
	FLAG = 3 — for all in-service buses, including those that are not load buses.
	FLAG = 4 — for all load buses, including those with only out-of-service loads.
	FLAG = 5 — for all buses, including those that are not load buses.
	FLAG = 10 — for only in-service load buses (type code 1, 2 or 3) with at least one in-service scalable load.
	FLAG = 11 — for only in-service load buses (type code 1, 2 or 3) with at least one scalable load, including those with all such loads out-of-service.
	FLAG = 12 — for all load buses with at least one scalable load, including those with all such loads out-of-service.
	FLAG = 20 — for only in-service load buses (type code 1, 2 or 3) with at least one in-service fixed load.
	FLAG = 21 — for only in-service load buses (type code 1, 2 or 3) with at least one fixed load, including those with all such loads out-of-service.
	FLAG = 22 — for all load buses with at least one fixed load, including those with all such loads out-of-service.
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or load quantities desired (input; no default allowed): 'BASE' Bus base voltage, in kV. 'PU' Actual bus voltage magnitude, in pu. 'KV' Actual bus voltage magnitude, in kV (in pu if base voltage is 0.0). 'ANGLE' Bus voltage phase angle, in radians. 'ANGLED' Bus voltage phase angle, in degrees. For the following, values are returned in MVA: 'MVAACT' Actual in-service constant MVA load. 'MVANOM' Nominal in-service constant MVA load at 1.0 pu voltage. 'ILACT' Actual in-service constant current load. 'ILNOM' Nominal in-service constant current load at 1.0 pu voltage. 'YLACT' Actual total in-service constant admittance load. 'YLNOM' Nominal total in-service constant admittance load at 1.0 pu voltage. 'TOTALACT' Actual in-service load. 'TOTALNOM' Nominal in-service load at 1.0 pu voltage. 'MISMATCH' Bus mismatch (0.0 if bus type code > 3). For the following, values are returned in units determined by the power output option setting: 'O_MVAACT' Actual in-service constant MVA load. 'O_MVANOM' Nominal in-service constant MVA load at 1.0 pu voltage. 'O_ILACT' Actual in-service constant current load. 'O_ILNOM' Nominal in-service constant current load at 1.0 pu voltage. 'O_YLACT' Actual total in-service constant admittance load. 'O_YLNOM' Nominal total in-service constant admittance load at 1.0 pu voltage. 'O_TOTALACT' Actual in-service load. 'O_TOTALNOM' Nominal in-service load at 1.0 pu voltage. 'O_MISMATCH' Bus mismatch (0.0 if bus type code > 3).
Integer DIM	Is the number of elements in each column of the array RARRAY (input; no default allowed). DIM must be as least as large as the value of BUSES returned by the GET_LOAD_BUS_COUNT API when it is called with the same values specified for SID and FLAG.
Real RARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer BUSES	Is the number of elements in each column of RARRAY that are set by this API (output). BUSES is identical to the value of BUSES returned by the GET_LOAD_BUS_COUNT API when it is called with the same values specified for SID and FLAG.
Integer IERR	Is the error code (output): IERR = 0 — No error. IERR = 1 — Working case is empty. IERR = 2 — Invalid SID value. IERR = 3 — Invalid FLAG value. IERR = 4 — Invalid NSTR value. IERR = 5 — DIM, and hence the size of RARRAY, is not large enough. IERR = 6 — Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.6.4 aLodBusCplx

Use this API to return an array of complex values for subsystem buses.

Python syntax:

```
ierr, xarray = alodbuscplx(sid, flag, string)
```

Fortran syntax:

```
CALL GET_LOAD_BUS_COMPLEX(SID,FLAG,NSTR,STRING,DIM,XARRAY,BUSES,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem buses to include (input; 1 by default). FLAG = 1 for only in-service load buses (type code 1, 2 or 3) with at least one in-service load. FLAG = 2 for only in-service load buses (type code 1, 2 or 3), including those with only out-of-service loads. FLAG = 3 for all in-service buses, including those that are not load buses. FLAG = 4 for all load buses, including those with only out-of-service loads. FLAG = 5 for all buses, including those that are not load buses. FLAG = 10 for only in-service load buses (type code 1, 2 or 3) with at least one in-service scalable load. FLAG = 11 for only in-service load buses (type code 1, 2 or 3) with at least one scalable load, including those with all such loads out-of-service. FLAG = 12 for all load buses with at least one scalable load, including those with all such loads out-of-service. FLAG = 20 for only in-service load buses (type code 1, 2 or 3) with at least one in-service fixed load. FLAG = 21 for only in-service load buses (type code 1, 2 or 3) with at least one fixed load, including those with all such loads out-of-service. FLAG = 22 for all load buses with at least one fixed load, including those with all such loads out-of-service.
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)

Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or load quantities desired (input; no default allowed):

'VOLTAGE'	Actual bus voltage, in pu, rectangular coordinates.

For the following, values are returned in MW and Mvar:

'MVAACT'	Actual in-service constant MVA load.
'MVANOM'	Nominal in-service constant MVA load at 1.0 pu voltage.
'ILACT'	Actual in-service constant current load.
'ILNOM'	Nominal in-service constant current load at 1.0 pu voltage.
'YLACT'	Actual total in-service constant admittance load.
'YLNOM'	Nominal total in-service constant admittance load at 1.0 pu voltage.
'TOTALACT'	Actual in-service load.
'TOTALNOM'	Nominal in-service load at 1.0 pu voltage.
'MISMATCH'	Bus mismatch (0.0 if bus type code > 3).

For the following, values are returned in units determined by the power output option setting:

'O_MVAACT'	Actual in-service constant MVA load.
'O_MVANOM'	Nominal in-service constant MVA load at 1.0 pu voltage.
'O_ILACT'	Actual in-service constant current load.
'O_ILNOM'	Nominal in-service constant current load at 1.0 pu voltage.
'O_YLACT'	Actual total in-service constant admittance load.
'O_YLNOM'	Nominal total in-service constant admittance load at 1.0 pu voltage.
'O_TOTALACT'	Actual in-service load.
'O_TOTALNOM'	Nominal in-service load at 1.0 pu voltage.
'O_MISMATCH'	Bus mismatch (0.0 if bus type code > 3).

Integer DIM

Is the number of elements in each column of the array XARRAY (input; no default allowed).

DIM must be as least as large as the value of BUSES returned by the GET_LOAD_BUS_COUNT API when it is called with the same values specified for SID and FLAG.

Complex XARRAY

Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer BUSES

Is the number of elements in each column of XARRAY that are set by this API (output).

BUSES is identical to the value of BUSES returned by the GET_LOAD_BUS_COUNT API when it is called with the same values specified for SID and FLAG.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output):	
	IERR = 0	No error.
	IERR = 1	Working case is empty.
	IERR = 2	Invalid SID value.
	IERR = 3	Invalid FLAG value.
	IERR = 4	Invalid NSTR value.
	IERR = 5	DIM, and hence the size of XARRAY, is not large enough.
	IERR = 6	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.6.5 aLodBusChar

Use this API to return an array of character values for subsystem buses.

Python syntax:

```
ierr, carray = alodbuschar(sid, flag, string)
```

Fortran syntax:

```
CALL GET_LOAD_BUS_CHAR(SID,FLAG,NSTR,STRING,DIM,CARRAY,BUSES,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem buses to include (input; 1 by default).
	FLAG = 1 — for only in-service load buses (type code 1, 2 or 3) with at least one in-service load.
	FLAG = 2 — for only in-service load buses (type code 1, 2 or 3), including those with only out-of-service loads.
	FLAG = 3 — for all in-service buses, including those that are not load buses.
	FLAG = 4 — for all load buses, including those with only out-of-service loads.
	FLAG = 5 — for all buses, including those that are not load buses.
	FLAG = 10 — for only in-service load buses (type code 1, 2 or 3) with at least one in-service scalable load.
	FLAG = 11 — for only in-service load buses (type code 1, 2 or 3) with at least one scalable load, including those with all such loads out-of-service.
	FLAG = 12 — for all load buses with at least one scalable load, including those with all such loads out-of-service.
	FLAG = 20 — for only in-service load buses (type code 1, 2 or 3) with at least one in-service fixed load.
	FLAG = 21 — for only in-service load buses (type code 1, 2 or 3) with at least one fixed load, including those with all such loads out-of-service.
	FLAG = 22 — for all load buses with at least one fixed load, including those with all such loads out-of-service.
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or load quantities desired (input; no default allowed): 'NAME' Bus name (12 characters). 'EXNAME' Extended bus name (18 characters).
Integer DIM	Is the number of elements in each column of the array CARRAY (input; no default allowed). DIM must be as least as large as the value of BUSES returned by the GET_LOAD_BUS_COUNT API when it is called with the same values specified for SID and FLAG.
Character CARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BUSES	Is the number of elements in each column of CARRAY that are set by this API (output). BUSES is identical to the value of BUSES returned by the GET_LOAD_BUS_COUNT API when it is called with the same values specified for SID and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid FLAG value. IERR = 4 Invalid NSTR value. IERR = 5 DIM, and hence the size of CARRAY, is not large enough. IERR = 6 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.6.6 aLodBusTypes

Use this API to return an array of character values indicating the data types corresponding to a set of specified STRING values that are valid input values to any of the data retrieval routines of the load bus data family (aLodBusInt, aLodBusReal, aLodBusCplx and aLodBusChar).

Python syntax:

```
ierr, types = alodbustypes(string)
```

Fortran syntax:

```
CALL GET_LOAD_BUS_TYPES(NSTR,STRING,TYPES,IERR)
```

where:

Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR strings that are valid STRING values for any of the data retrieval routines of the load bus data family (input; no default allowed):
Character TYPES*1	Is an array of at least NSTR elements, each of which contains one of the following characters indicating the data type of the data associated with the corresponding element of STRING (output): 'I' Integer. 'R' Real. 'X' Complex. 'C' Character. '?' Invalid STRING value.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = n Position of the first element in STRING that contains a value that is not a valid STRING value for any of the data retrieval routines in the load bus data family (1 ≤ n ≤ NSTR). IERR = 99 Invalid NSTR value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.7 Load Data

The load data family of subsystem data retrieval APIs returns arrays of data for loads in a specified subsystem. One of the input items to each API allows the user to elect either to retrieve data for all subsystem loads, or to restrict the array entries to data for a subset of loads according to their bus and/or load configuration and statuses.

Data for subsystem loads is returned from all APIs in the load data family in the same order: ascending bus number order or alphabetic extended bus name order, according to the bus output option setting in effect at the time the API is referenced; multiple loads at the same bus are in ascending load identifier order.

8.7.1 aLoadCount

Use this API to return the number of array entries required to accommodate the data to be returned by the remaining members of the load data family.

Python syntax:

```
ierr, loads = aloadcount(sid, flag)
```

Fortran syntax:

```
CALL GET_LOAD_COUNT(SID, FLAG, LOADS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem loads to include (input; 1 by default). FLAG = 1 for only in-service loads at in-service buses (type code < 4). FLAG = 2 for all loads at in-service buses (type code < 4). FLAG = 3 for only in-service loads, including those at type 4 buses. FLAG = 4 for all loads.
Integer LOADS	Is the number of loads in the subsystem indicated by SID that meet the editing criteria indicated by FLAG (output).
Integer IERR	Is the error code (output): IERR = 0 No error, LOADS returned. IERR = 1 Working case is empty; LOADS returned as 0. IERR = 2 Invalid SID value; LOADS returned as 0. IERR = 3 Invalid FLAG value; LOADS returned as 0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.7.2 aLoadInt

Use this API to return an array of integer values for subsystem loads.

Python syntax:

```
ierr, iarray = aloadint(sid, flag, string)
```

Fortran syntax:

```
CALL GET_LOAD_INTEGER(SID,FLAG,NSTR,STRING,DIM,IARRAY,LOADS,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem loads to include (input; 1 by default). FLAG = 1 for only in-service loads at in-service buses (type code < 4). FLAG = 2 for all loads at in-service buses (type code < 4). FLAG = 3 for only in-service loads, including those at type 4 buses. FLAG = 4 for all loads.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or load quantities desired (input; no default allowed): 'NUMBER' Bus number. 'AREA' Bus area number. 'ZONE' Bus zone number. 'OWNER' Bus owner number. 'STATUS' Load status. 'SCALE' Scalable load flag. 'CGR' Grounding code (0 for an ungrounded load, 1 for a grounded load).
Integer DIM	Is the number of elements in each column of the array IARRAY (input; no default allowed). DIM must be as least as large as the value of LOADS returned by the GET_LOAD_COUNT API when it is called with the same values specified for SID and FLAG.
Integer IARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer LOADS	Is the number of elements in each column of IARRAY that are set by this API (output). LOADS is identical to the value of LOADS returned by the GET_LOAD_COUNT API when it is called with the same values specified for SID and FLAG.
Integer IERR	Is the error code (output):
	IERR = 0 — No error.
	IERR = 1 — Working case is empty.
	IERR = 2 — Invalid SID value.
	IERR = 3 — Invalid FLAG value.
	IERR = 4 — Invalid NSTR value.
	IERR = 5 — DIM, and hence the size of IARRAY, is not large enough.
	IERR = 6 — Invalid STRING value.
	IERR = 7 — Sequence data not in case (when STRING = ’CGR’).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.7.3 aLoadReal

Use this API to return an array of real values for subsystem loads.

Python syntax:

```
ierr, rarray = aloadreal(sid, flag, string)
```

Fortran syntax:

```
CALL GET_LOAD_REAL(SID,FLAG,NSTR,STRING,DIM,RARRAY,LOADS,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem loads to include (input; 1 by default). FLAG = 1 for only in-service loads at in-service buses (type code < 4). FLAG = 2 for all loads at in-service buses (type code < 4). FLAG = 3 for only in-service loads, including those at type 4 buses. FLAG = 4 for all loads.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or load quantities desired (input; no default allowed): For the following, values are returned in MVA: 'MVAACT' Actual in-service constant MVA load. 'MVANOM' Nominal in-service constant MVA load at 1.0 pu voltage. 'ILACT' Actual in-service constant current load. 'ILNOM' Nominal in-service constant current load at 1.0 pu voltage. 'YLACT' Actual total in-service constant admittance load. 'YLNOM' Nominal total in-service constant admittance load at 1.0 pu voltage. 'TOTALACT' Actual in-service load. 'TOTALNOM' Nominal in-service load at 1.0 pu voltage. 'YNEG' Nominal negative sequence total load at 1.0 pu voltage. 'YZERO' Nominal zero sequence total load at 1.0 pu voltage.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

For the following, values are returned in units determined by the power output option setting:

'O_MVAACT'	Actual in-service constant MVA load.
'O_MVANOM'	Nominal in-service constant MVA load at 1.0 pu voltage.
'O_ILACT'	Actual in-service constant current load.
'O_ILNOM'	Nominal in-service constant current load at 1.0 pu voltage.
'O_YLACT'	Actual total in-service constant admittance load.
'O_YLNOM'	Nominal total in-service constant admittance load at 1.0 pu voltage.
'O_TOTALACT'	Actual in-service load.
'O_TOTALNOM'	Nominal in-service load at 1.0 pu voltage.
'O_YNEG'	Nominal negative sequence total load at 1.0 pu voltage.
'O_YZERO'	Nominal zero sequence total load at 1.0 pu voltage.

Integer DIM

Is the number of elements in each column of the array RARRAY (input; no default allowed).

DIM must be as least as large as the value of LOADS returned by the GET_LOAD_COUNT API when it is called with the same values specified for SID and FLAG.

Real RARRAY

Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer LOADS

Is the number of elements in each column of RARRAY that are set by this API (output).

LOADS is identical to the value of LOADS returned by the GET_LOAD_COUNT API when it is called with the same values specified for SID and FLAG.

Integer IERR

Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid FLAG value.
IERR = 4	Invalid NSTR value.
IERR = 5	DIM, and hence the size of RARRAY, is not large enough.
IERR = 6	Invalid STRING value.
IERR = 7	Sequence data not in case (when STRING = 'YNEG', 'O_YNEG', 'YZERO' or 'O_YZERO').

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.7.4 aLoadCplx

Use this API to return an array of complex values for subsystem loads.

Python syntax:

```
ierr, xarray = aloadcplx(sid, flag, string)
```

Fortran syntax:

```
CALL GET_LOAD_COMPLEX(SID,FLAG,NSTR,STRING,DIM,XARRAY,LOADS,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem loads to include (input; 1 by default). FLAG = 1 for only in-service loads at in-service buses (type code < 4). FLAG = 2 for all loads at in-service buses (type code < 4). FLAG = 3 for only in-service loads, including those at type 4 buses. FLAG = 4 for all loads.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or load quantities desired (input; no default allowed): For the following, values are returned in MW and Mvar: 'MVAACT' Actual in-service constant MVA load. 'MVANOM' Nominal in-service constant MVA load at 1.0 pu voltage. 'ILACT' Actual in-service constant current load. 'ILNOM' Nominal in-service constant current load at 1.0 pu voltage. 'YLACT' Actual total in-service constant admittance load. 'YLNOM' Nominal total in-service constant admittance load at 1.0 pu voltage. 'TOTALACT' Actual in-service load. 'TOTALNOM' Nominal in-service load at 1.0 pu voltage.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

For the following, values are returned in units determined by the power output option setting:

'O_MVAACT'	Actual in-service constant MVA load.
'O_MVANOM'	Nominal in-service constant MVA load at 1.0 pu voltage.
'O_ILACT'	Actual in-service constant current load.
'O_ILNOM'	Nominal in-service constant current load at 1.0 pu voltage.
'O_YLACT'	Actual total in-service constant admittance load.
'O_YLNOM'	Nominal total in-service constant admittance load at 1.0 pu voltage.
'O_TOTALACT'	Actual in-service load.
'O_TOTALNOM'	Nominal in-service load at 1.0 pu voltage.

Integer DIM

Is the number of elements in each column of the array XARRAY (input; no default allowed).

DIM must be as least as large as the value of LOADS returned by the GET_LOAD_COUNT API when it is called with the same values specified for SID and FLAG.

Complex XARRAY

Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer LOADS

Is the number of elements in each column of XARRAY that are set by this API (output).

LOADS is identical to the value of LOADS returned by the GET_LOAD_COUNT API when it is called with the same values specified for SID and FLAG.

Integer IERR

Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid FLAG value.
IERR = 4	Invalid NSTR value.
IERR = 5	DIM, and hence the size of XARRAY, is not large enough.
IERR = 6	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.7.5 aLoadChar

Use this API to return an array of character values for subsystem loads.

Python syntax:

```
ierr, carray = aloadchar(sid, flag, string)
```

Fortran syntax:

```
CALL GET_LOAD_CHAR(SID,FLAG,NSTR,STRING,DIM,CARRAY,LOADS,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem loads to include (input; 1 by default). FLAG = 1 for only in-service loads at in-service buses (type code < 4). FLAG = 2 for all loads at in-service buses (type code < 4). FLAG = 3 for only in-service loads, including those at type 4 buses. FLAG = 4 for all loads.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or load quantities desired (input; no default allowed): 'ID' Load identifier (2 characters). 'NAME' Bus name (12 characters). 'EXNAME' Extended bus name (18 characters).
Integer DIM	Is the number of elements in each column of the array CARRAY (input; no default allowed). DIM must be as least as large as the value of LOADS returned by the GET_LOAD_COUNT API when it is called with the same values specified for SID and FLAG.
Character CARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer LOADS	Is the number of elements in each column of CARRAY that are set by this API (output). LOADS is identical to the value of LOADS returned by the GET_LOAD_COUNT API when it is called with the same values specified for SID and FLAG.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output):	
	IERR = 0	No error.
	IERR = 1	Working case is empty.
	IERR = 2	Invalid SID value.
	IERR = 3	Invalid FLAG value.
	IERR = 4	Invalid NSTR value.
	IERR = 5	DIM, and hence the size of CARRAY, is not large enough.
	IERR = 6	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.7.6 aLoadTypes

Use this API to return an array of character values indicating the data types corresponding to a set of specified STRING values that are valid input values to any of the data retrieval routines of the load data family (aLoadInt, aLoadReal, aLoadCplx and aLoadChar).

Python syntax:

```
ierr, types = aloadtypes(string)
```

Fortran syntax:

```
CALL GET_LOAD_TYPES(NSTR,STRING,TYPES,IERR)
```

where:

Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR strings that are valid STRING values for any of the data retrieval routines of the load data family (input; no default allowed):
Character TYPES*1	Is an array of at least NSTR elements, each of which contains one of the following characters indicating the data type of the data associated with the corresponding element of STRING (output): 'I' Integer. 'R' Real. 'X' Complex. 'C' Character. '?' Invalid STRING value.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = n Position of the first element in STRING that contains a value that is not a valid STRING value for any of the data retrieval routines in the load data family (1 ≤ n ≤ NSTR). IERR = 99 Invalid NSTR value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.8 Fixed Shunt Bus Data

The fixed shunt bus data family of subsystem data retrieval APIs returns arrays of data for buses in a specified subsystem. One of the input items to each API allows the user to elect either to retrieve data for all subsystem buses, or to restrict the array entries to data for a subset of buses according to their bus and/or fixed shunt configuration and statuses.

Data for subsystem buses is returned from all APIs in the fixed shunt bus data family in the same order: ascending bus number order or alphabetic extended bus name order, according to the bus output option setting in effect at the time the API is referenced.

8.8.1 aFxShntBusCount

Use this API to return the number of array entries required to accommodate the data to be returned by the remaining members of the fixed shunt bus data family.

Python syntax:

```
ierr, buses = afxshntbuscount(sid, flag)
```

Fortran syntax:

```
CALL GET_SHUNT_BUS_COUNT(SID, FLAG, BUSES, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem buses to include (input; 1 by default). FLAG = 1 for only in-service fixed shunt buses (type code 1, 2 or 3) with at least one in-service fixed shunt. FLAG = 2 for only in-service fixed shunt buses (type code 1, 2 or 3), including those with only out-of-service fixed shunts. FLAG = 3 for all in-service buses, including those that are not fixed shunt buses. FLAG = 4 for all fixed shunt buses, including those with only out-of-service fixed shunts. FLAG = 5 for all buses, including those that are not fixed shunt buses.
Integer BUSES	Is the number of buses in the subsystem indicated by SID that meet the editing criteria indicated by FLAG (output).
Integer IERR	Is the error code (output): IERR = 0 No error, BUSES returned. IERR = 1 Working case is empty; BUSES returned as 0. IERR = 2 Invalid SID value; BUSES returned as 0. IERR = 3 Invalid FLAG value; BUSES returned as 0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.8.2 aFxShntBusInt

Use this API to return an array of integer values for subsystem buses.

Python syntax:

```
ierr, iarray = afxshntbusint(sid, flag, string)
```

Fortran syntax:

```
CALL
GET_SHUNT_BUS_INTEGER(SID,FLAG,NSTR,STRING,DIM,IARRAY,BUSES,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem buses to include (input; 1 by default). FLAG = 1 — for only in-service fixed shunt buses (type code 1, 2 or 3) with at least one in-service fixed shunt. FLAG = 2 — for only in-service fixed shunt buses (type code 1, 2 or 3), including those with only out-of-service fixed shunts. FLAG = 3 — for all in-service buses, including those that are not fixed shunt buses. FLAG = 4 — for all fixed shunt buses, including those with only out-of-service fixed shunts. FLAG = 5 — for all buses, including those that are not fixed shunt buses.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or fixed shunt quantities desired (input; no default allowed): 'NUMBER' — Bus number. 'TYPE' — Bus type code. 'AREA' — Bus area number. 'ZONE' — Bus zone number. 'OWNER' — Bus owner number. 'DUMMY' — Returns 1 if the bus is a dummy bus of a multisection line, or 0 if it is not. 'STATUS' — Inferred shunt status. Returns 1 if the bus type code is 1, 2 or 3, and at least one in-service fixed shunt is present, or 0 otherwise.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer DIM	Is the number of elements in each column of the array IARRAY (input; no default allowed). DIM must be as least as large as the value of BUSES returned by the GET_SHUNT_BUS_COUNT API when it is called with the same values specified for SID and FLAG.
Integer IARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BUSES	Is the number of elements in each column of IARRAY that are set by this API (output). BUSES is identical to the value of BUSES returned by the GET_SHUNT_BUS_COUNT API when it is called with the same values specified for SID and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid FLAG value. IERR = 4 Invalid NSTR value. IERR = 5 DIM, and hence the size of IARRAY, is not large enough. IERR = 6 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.8.3 aFxShntBusReal

Use this API to return an array of real values for subsystem buses.

Python syntax:

```
ierr, rarray = afxshntbusreal(sid, flag, string)
```

Fortran syntax:

```
CALL GET_SHUNT_BUS_REAL(SID,FLAG,NSTR,STRING,DIM,RARRAY,BUSES,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem buses to include (input; 1 by default). FLAG = 1 — for only in-service fixed shunt buses (type code 1, 2 or 3) with at least one in-service fixed shunt. FLAG = 2 — for only in-service fixed shunt buses (type code 1, 2 or 3), including those with only out-of-service fixed shunts. FLAG = 3 — for all in-service buses, including those that are not fixed shunt buses. FLAG = 4 — for all fixed shunt buses, including those with only out-of-service fixed shunts. FLAG = 5 — for all buses, including those that are not fixed shunt buses.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or fixed shunt quantities desired (input; no default allowed): 'BASE' — Bus base voltage, in kV. 'PU' — Actual bus voltage magnitude, in pu. 'KV' — Actual bus voltage magnitude, in kV (in pu if base voltage is 0.0). 'ANGLE' — Bus voltage phase angle, in radians. 'ANGLED' — Bus voltage phase angle, in degrees. 'PU_GBZERO' — Zero sequence total nominal in-service fixed bus shunt, in per unit.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

For the following, values are returned in MVA:

'SHUNTACT'	Total actual in-service fixed bus shunt.
'SHUNTNOM'	Total nominal in-service fixed bus shunt at 1.0 pu voltage.
'GBZERO'	Zero sequence total nominal in-service fixed bus shunt.
'MISMATCH'	Bus mismatch (0.0 if bus type code > 3).

For the following, values are returned in units determined by the power output option setting:

'O_SHUNTACT'	Total actual in-service fixed bus shunt.
'O_SHUNTNOM'	Total nominal in-service fixed bus shunt at 1.0 pu voltage.
'O_GBZERO'	Zero sequence total nominal in-service fixed bus shunt.
'O_MISMATCH'	Bus mismatch (0.0 if bus type code > 3).

Integer DIM

Is the number of elements in each column of the array RARRAY (input; no default allowed).

DIM must be as least as large as the value of BUSES returned by the GET_SHUNT_BUS_COUNT API when it is called with the same values specified for SID and FLAG.

Real RARRAY

Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer BUSES

Is the number of elements in each column of RARRAY that are set by this API (output).

BUSES is identical to the value of BUSES returned by the GET_SHUNT_BUS_COUNT API when it is called with the same values specified for SID and FLAG.

Integer IERR

Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid FLAG value.
IERR = 4	Invalid NSTR value.
IERR = 5	DIM, and hence the size of RARRAY, is not large enough.
IERR = 6	Invalid STRING value.
IERR = 7	Sequence data not in case; when STRING = 'PU_GBZERO', 'GBZERO' and 'O_GBZERO'.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.8.4 aFxShntBusCplx

Use this API to return an array of complex values for subsystem buses.

Python syntax:

```
ierr, xarray = afxshntbuscplx(sid, flag, string)
```

Fortran syntax:

```
CALL
GET_SHUNT_BUS_COMPLEX(SID,FLAG,NSTR,STRING,DIM,XARRAY,BUSES,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem buses to include (input; 1 by default). FLAG = 1 — for only in-service fixed shunt buses (type code 1, 2 or 3) with at least one in-service fixed shunt. FLAG = 2 — for only in-service fixed shunt buses (type code 1, 2 or 3), including those with only out-of-service fixed shunts. FLAG = 3 — for all in-service buses, including those that are not fixed shunt buses. FLAG = 4 — for all fixed shunt buses, including those with only out-of-service fixed shunts. FLAG = 5 — for all buses, including those that are not fixed shunt buses.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or fixed shunt quantities desired (input; no default allowed): 'VOLTAGE' — Actual bus voltage, in pu, rectangular coordinates. 'PU_GBZERO' — Zero sequence total nominal in-service fixed bus shunt, in per unit. For the following, values are returned in MW and Mvar: 'SHUNTACT' — Total actual in-service fixed bus shunt. 'SHUNTNOM' — Total nominal in-service fixed bus shunt at 1.0 pu voltage. 'GBZERO' — Zero sequence total nominal in-service fixed bus shunt. 'MISMATCH' — Bus mismatch (0.0 if bus type code > 3).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

For the following, values are returned in units determined by the power output option setting:

'O_SHUNTACT'	Total actual in-service fixed bus shunt.
'O_SHUNTNOM'	Total nominal in-service fixed bus shunt at 1.0 pu voltage.
'O_GBZERO'	Zero sequence total nominal in-service fixed bus shunt.
'O_MISMATCH'	Bus mismatch (0.0 if bus type code > 3).

Integer DIM

Is the number of elements in each column of the array XARRAY (input; no default allowed).

DIM must be as least as large as the value of BUSES returned by the GET_SHUNT_BUS_COUNT API when it is called with the same values specified for SID and FLAG.

Complex XARRAY

Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer BUSES

Is the number of elements in each column of XARRAY that are set by this API (output).

BUSES is identical to the value of BUSES returned by the GET_SHUNT_BUS_COUNT API when it is called with the same values specified for SID and FLAG.

Integer IERR

Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid FLAG value.
IERR = 4	Invalid NSTR value.
IERR = 5	DIM, and hence the size of XARRAY, is not large enough.
IERR = 6	Invalid STRING value.
IERR = 7	Sequence data not in case; when STRING = 'PU_GBZERO', 'GBZERO' and 'O_GBZERO'.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.8.5 aFxShntBusChar

Use this API to return an array of character values for subsystem buses.

Python syntax:

```
ierr, carray = afxshntbuschar(sid, flag, string)
```

Fortran syntax:

```
CALL GET_SHUNT_BUS_CHAR(SID,FLAG,NSTR,STRING,DIM,CARRAY,BUSES,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem buses to include (input; 1 by default). FLAG = 1 — for only in-service fixed shunt buses (type code 1, 2 or 3) with at least one in-service fixed shunt. FLAG = 2 — for only in-service fixed shunt buses (type code 1, 2 or 3), including those with only out-of-service fixed shunts. FLAG = 3 — for all in-service buses, including those that are not fixed shunt buses. FLAG = 4 — for all fixed shunt buses, including those with only out-of-service fixed shunts. FLAG = 5 — for all buses, including those that are not fixed shunt buses.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or fixed shunt quantities desired (input; no default allowed): 'NAME' — Bus name (12 characters). 'EXNAME' — Extended bus name (18 characters).
Integer DIM	Is the number of elements in each column of the array CARRAY (input; no default allowed). DIM must be as least as large as the value of BUSES returned by the GET_SHUNT_BUS_COUNT API when it is called with the same values specified for SID and FLAG.
Character CARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer BUSES	Is the number of elements in each column of CARRAY that are set by this API (output). BUSES is identical to the value of BUSES returned by the GET_SHUNT_BUS_COUNT API when it is called with the same values specified for SID and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid FLAG value. IERR = 4 Invalid NSTR value. IERR = 5 DIM, and hence the size of CARRAY, is not large enough. IERR = 6 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.8.6 aFxShntBusTypes

Use this API to return an array of character values indicating the data types corresponding to a set of specified STRING values that are valid input values to any of the data retrieval routines of the fixed shunt bus data family (aFxShntBusInt, aFxShntBusReal, aFxShntBusCplx and aFxShntBusChar).

Python syntax:

```
ierr, types = afxshntbustypes(string)
```

Fortran syntax:

```
CALL GET_SHUNT_BUS_TYPES(NSTR,STRING,TYPES,IERR)
```

where:

Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR strings that are valid STRING values for any of the data retrieval routines of the fixed shunt bus data family (input; no default allowed):
Character TYPES*1	Is an array of at least NSTR elements, each of which contains one of the following characters indicating the data type of the data associated with the corresponding element of STRING (output): 'I' Integer. 'R' Real. 'X' Complex. 'C' Character. '?' Invalid STRING value.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = n Position of the first element in STRING that contains a value that is not a valid STRING value for any of the data retrieval routines in the fixed shunt bus data family (1 ≤ n ≤ NSTR). IERR = 99 Invalid NSTR value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.9 Fixed Shunt Data

The fixed shunt data family of subsystem data retrieval APIs returns arrays of data for fixed bus shunts in a specified subsystem. One of the input items to each API allows the user to elect either to retrieve data for all subsystem fixed bus shunts, or to restrict the array entries to data for a subset of fixed bus shunts according to their bus and/or fixed shunt configuration and statuses.

Data for subsystem fixed bus shunts is returned from all APIs in the fixed shunt data family in the same order: ascending bus number order or alphabetic extended bus name order, according to the bus output option setting in effect at the time the API is referenced; multiple fixed shunts at the same bus are in ascending shunt identifier order.

8.9.1 aFxShuntCount

Use this API to return the number of array entries required to accommodate the data to be returned by the remaining members of the fixed shunt data family.

Python syntax:

```
ierr, shunts = afxshuntcount(sid, flag)
```

Fortran syntax:

```
CALL GET_SHUNT_COUNT(SID, FLAG, SHUNTS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem fixed bus shunts to include (input; 1 by default). FLAG = 1 for only in-service fixed shunts at in-service buses (type code < 4). FLAG = 2 for all fixed shunts at in-service buses (type code < 4). FLAG = 3 for only in-service fixed shunts, including those at type 4 buses. FLAG = 4 for all fixed bus shunts.
Integer SHUNTS	Is the number of fixed bus shunts in the subsystem indicated by SID that meet the editing criteria indicated by FLAG (output).
Integer IERR	Is the error code (output): IERR = 0 No error, SHUNTS returned. IERR = 1 Working case is empty; SHUNTS returned as 0. IERR = 2 Invalid SID value; SHUNTS returned as 0. IERR = 3 Invalid FLAG value; SHUNTS returned as 0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.9.2 aFxShuntInt

Use this API to return an array of integer values for subsystem fixed shunts.

Python syntax:

```
ierr, iarray = afxshuntint(sid, flag, string)
```

Fortran syntax:

```
CALL GET_SHUNT_INTEGER(SID,FLAG,NSTR,STRING,DIM,IARRAY,SHUNTS,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem fixed bus shunts to include (input; 1 by default). FLAG = 1 for only in-service fixed shunts at in-service buses (type code < 4). FLAG = 2 for all fixed shunts at in-service buses (type code < 4). FLAG = 3 for only in-service fixed shunts, including those at type 4 buses. FLAG = 4 for all fixed bus shunts.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or fixed shunt quantities desired (input; no default allowed): 'NUMBER' Bus number. 'STATUS' Shunt status.
Integer DIM	Is the number of elements in each column of the array IARRAY (input; no default allowed). DIM must be as least as large as the value of SHUNTS returned by the GET_SHUNT_COUNT API when it is called with the same values specified for SID and FLAG.
Integer IARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer SHUNTS	Is the number of elements in each column of IARRAY that are set by this API (output). SHUNTS is identical to the value of SHUNTS returned by the GET_SHUNT_COUNT API when it is called with the same values specified for SID and FLAG.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output):	
	IERR = 0	No error.
	IERR = 1	Working case is empty.
	IERR = 2	Invalid SID value.
	IERR = 3	Invalid FLAG value.
	IERR = 4	Invalid NSTR value.
	IERR = 5	DIM, and hence the size of IARRAY, is not large enough.
	IERR = 6	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.9.3 aFxShuntReal

Use this API to return an array of real values for subsystem fixed shunts.

Python syntax:

```
ierr, rarray = afxshuntreal(sid, flag, string)
```

Fortran syntax:

```
CALL GET_SHUNT_REAL(SID,FLAG,NSTR,STRING,DIM,RARRAY,SHUNTS,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem fixed bus shunts to include (input; 1 by default). FLAG = 1 for only in-service fixed shunts at in-service buses (type code < 4). FLAG = 2 for all fixed shunts at in-service buses (type code < 4). FLAG = 3 for only in-service fixed shunts, including those at type 4 buses. FLAG = 4 for all fixed bus shunts.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or fixed shunt quantities desired (input; no default allowed): 'PU_GBZERO' Zero sequence nominal fixed bus shunt, in per unit. For the following, values are returned in MVA: 'SHUNTACT' Actual in-service fixed bus shunt. 'SHUNTNOM' Nominal fixed bus shunt at 1.0 pu voltage. 'GBZERO' Zero sequence nominal fixed bus shunt. For the following, values are returned in units determined by the power output option setting: 'O_SHUNTACT' Actual in-service fixed bus shunt. 'O_SHUNTNOM' Nominal fixed bus shunt at 1.0 pu voltage. 'O_GBZERO' Zero sequence nominal fixed bus shunt.
Integer DIM	Is the number of elements in each column of the array RARRAY (input; no default allowed). DIM must be as least as large as the value of SHUNTS returned by the GET_SHUNT_COUNT API when it is called with the same values specified for SID and FLAG.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real RARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer SHUNTS	Is the number of elements in each column of RARRAY that are set by this API (output). SHUNTS is identical to the value of SHUNTS returned by the GET_SHUNT_COUNT API when it is called with the same values specified for SID and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid FLAG value. IERR = 4 Invalid NSTR value. IERR = 5 DIM, and hence the size of RARRAY, is not large enough. IERR = 6 Invalid STRING value. IERR = 7 Sequence data not in case; when STRING = ’PU_GBZERO’, ’GBZERO’ and ’O_GBZERO’.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.9.4 aFxShuntCplx

Use this API to return an array of complex values for subsystem fixed shunts.

Python syntax:

```
ierr, xarray = afxshuntcplx(sid, flag, string)
```

Fortran syntax:

```
CALL GET_SHUNT_COMPLEX(SID,FLAG,NSTR,STRING,DIM,XARRAY,SHUNTS,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem fixed bus shunts to include (input; 1 by default). FLAG = 1 for only in-service fixed shunts at in-service buses (type code < 4). FLAG = 2 for all fixed shunts at in-service buses (type code < 4). FLAG = 3 for only in-service fixed shunts, including those at type 4 buses. FLAG = 4 for all fixed bus shunts.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or fixed shunt quantities desired (input; no default allowed): 'PU_GBZERO' Zero sequence nominal fixed bus shunt, in per unit. For the following, values are returned in MW and Mvar: 'SHUNTACT' Actual in-service fixed bus shunt. 'SHUNTNOM' Nominal fixed bus shunt at 1.0 pu voltage. 'GBZERO' Zero sequence nominal fixed bus shunt. For the following, values are returned in units determined by the power output option setting: 'O_SHUNTACT' Actual in-service fixed bus shunt. 'O_SHUNTNOM' Nominal fixed bus shunt at 1.0 pu voltage. 'O_GBZERO' Zero sequence nominal fixed bus shunt.
Integer DIM	Is the number of elements in each column of the array XARRAY (input; no default allowed). DIM must be as least as large as the value of SHUNTS returned by the GET_SHUNT_COUNT API when it is called with the same values specified for SID and FLAG.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Complex XARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer SHUNTS	Is the number of elements in each column of XARRAY that are set by this API (output). SHUNTS is identical to the value of SHUNTS returned by the GET_SHUNT_COUNT API when it is called with the same values specified for SID and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid FLAG value. IERR = 4 Invalid NSTR value. IERR = 5 DIM, and hence the size of XARRAY, is not large enough. IERR = 6 Invalid STRING value. IERR = 7 Sequence data not in case; when STRING = ’PU_GBZERO’, ’GBZERO’ and ’O_GBZERO’.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.9.5 aFxShuntChar

Use this API to return an array of character values for subsystem fixed shunts.

Python syntax:

```
ierr, carray = afxshuntchar(sid, flag, string)
```

Fortran syntax:

```
CALL GET_SHUNT_CHAR(SID,FLAG,NSTR,STRING,DIM,CARRAY,SHUNTS,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem fixed bus shunts to include (input; 1 by default). FLAG = 1 for only in-service fixed shunts at in-service buses (type code < 4). FLAG = 2 for all fixed shunts at in-service buses (type code < 4). FLAG = 3 for only in-service fixed shunts, including those at type 4 buses. FLAG = 4 for all fixed bus shunts.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or fixed shunt quantities desired (input; no default allowed): 'ID' Shunt identifier (2 characters). 'NAME' Bus name (12 characters). 'EXNAME' Extended bus name (18 characters).
Integer DIM	Is the number of elements in each column of the array CARRAY (input; no default allowed). DIM must be as least as large as the value of SHUNTS returned by the GET_SHUNT_COUNT API when it is called with the same values specified for SID and FLAG.
Character CARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer SHUNTS	Is the number of elements in each column of CARRAY that are set by this API (output). SHUNTS is identical to the value of SHUNTS returned by the GET_SHUNT_COUNT API when it is called with the same values specified for SID and FLAG.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output):	
	IERR = 0	No error.
	IERR = 1	Working case is empty.
	IERR = 2	Invalid SID value.
	IERR = 3	Invalid FLAG value.
	IERR = 4	Invalid NSTR value.
	IERR = 5	DIM, and hence the size of CARRAY, is not large enough.
	IERR = 6	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.9.6 aFxShuntTypes

Use this API to return an array of character values indicating the data types corresponding to a set of specified STRING values that are valid input values to any of the data retrieval routines of the fixed shunt data family (aFxShuntInt, aFxShuntReal, aFxShuntCplx and aFxShuntChar).

Python syntax:

```
ierr, types = afxshunttypes(string)
```

Fortran syntax:

```
CALL GET_SHUNT_TYPES(NSTR,STRING,TYPES,IERR)
```

where:

Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR strings that are valid STRING values for any of the data retrieval routines of the fixed shunt data family (input; no default allowed):
Character TYPES*1	Is an array of at least NSTR elements, each of which contains one of the following characters indicating the data type of the data associated with the corresponding element of STRING (output): 'I' Integer. 'R' Real. 'X' Complex. 'C' Character. '?' Invalid STRING value.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = n Position of the first element in STRING that contains a value that is not a valid STRING value for any of the data retrieval routines in the fixed shunt data family (1 ≤ n ≤ NSTR). IERR = 99 Invalid NSTR value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.10 Switched Shunt Data

The switched shunt data family of subsystem data retrieval APIs returns arrays of data for switched shunts in a specified subsystem. One of the input items to each API allows the user to elect either to retrieve data for all subsystem switched shunts, or to restrict the array entries to data for switched shunts at in-service buses.

Data for subsystem switched shunts is returned from all APIs in the switched shunt data family in the same order: ascending bus number order or alphabetic extended bus name order, according to the bus output option setting in effect at the time the API is referenced.

8.10.1 aSwshCount

Use this API to return the number of array entries required to accommodate the data to be returned by the remaining members of the switched shunt data family.

Python syntax:

```
ierr, shnts = aswshcount(sid, flag)
```

Fortran syntax:

```
CALL GET_SWITCHED_SHUNT_COUNT(SID, FLAG, SHNTS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem switched shunts to include (input; 1 by default). FLAG = 1 for only in-service switched shunts at in-service buses (type code < 4). FLAG = 2 for all switched shunts at in-service buses (type code < 4). FLAG = 3 for only in-service switched shunts, including those at type 4 buses. FLAG = 4 for all switched shunts.
Integer SHNTS	Is the number of switched shunts in the subsystem indicated by SID that meet the editing criteria indicated by FLAG (output).
Integer IERR	Is the error code (output): IERR = 0 No error, SHNTS returned. IERR = 1 Working case is empty; SHNTS returned as 0. IERR = 2 Invalid SID value; SHNTS returned as 0. IERR = 3 Invalid FLAG value; SHNTS returned as 0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.10.2 aSwshInt

Use this API to return an array of integer values for subsystem switched shunts.

Python syntax:

```
ierr, iarray = aswshint(sid, flag, string)
```

Fortran syntax:

```
CALL GET_SWITCHED_SHUNT_INTEGER(SID, FLAG, NSTR, STRING, DIM, IARRAY,
SHNTS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem switched shunts to include (input; 1 by default). FLAG = 1 for only in-service switched shunts at in-service buses (type code < 4). FLAG = 2 for all switched shunts at in-service buses (type code < 4). FLAG = 3 for only in-service switched shunts, including those at type 4 buses. FLAG = 4 for all switched shunts.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)

Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or switched shunt quantities desired (input; no default allowed):

'NUMBER'	Bus number.
'TYPE'	Bus type code.
'AREA'	Bus area number.
'ZONE'	Bus zone number.
'OWNER'	Bus owner number.
'DUMMY'	Returns 1 if the bus is a dummy bus of a multisection line, or 0 if it is not.
'MODE'	Switched shunt control mode.
'STATUS'	Switched shunt status flag.
'ADJMETHOD'	Switched shunt adjustment method.
'IREG'	Controlled bus.
'BLOCKS'	Number of blocks.
'STEPSBLOCK1'	Number of steps in block 1.
'STEPSBLOCK2'	Number of steps in block 2.
'STEPSBLOCK3'	Number of steps in block 3.
'STEPSBLOCK4'	Number of steps in block 4.
'STEPSBLOCK5'	Number of steps in block 5.
'STEPSBLOCK6'	Number of steps in block 6.
'STEPSBLOCK7'	Number of steps in block 7.
'STEPSBLOCK8'	Number of steps in block 8.

Integer DIM

Is the number of elements in each column of the array IARRAY (input; no default allowed).

DIM must be as least as large as the value of SHNTS returned by the GET_SWITCHED_SHUNT_COUNT API when it is called with the same values specified for SID and FLAG.

Integer IARRAY

Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer SHNTS

Is the number of elements in each column of IARRAY that are set by this API (output).

SHNTS is identical to the value of SHNTS returned by the GET_SWITCHED_SHUNT_COUNT API when it is called with the same values specified for SID and FLAG.

Integer IERR

Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid FLAG value.
IERR = 4	Invalid NSTR value.
IERR = 5	DIM, and hence the size of IARRAY, is not large enough.
IERR = 6	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.10.3 aSwshReal

Use this API to return an array of real values for subsystem switched shunts.

Python syntax:

```
ierr, rarray = aswshreal(sid, flag, string)
```

Fortran syntax:

```
CALL
GET_SWITCHED_SHUNT_REAL(SID,FLAG,NSTR,STRING,DIM,RARRAY,SHNTS,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem switched shunts to include (input; 1 by default). FLAG = 1 for only in-service switched shunts at in-service buses (type code < 4). FLAG = 2 for all switched shunts at in-service buses (type code < 4). FLAG = 3 for only in-service switched shunts, including those at type 4 buses. FLAG = 4 for all switched shunts.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR) Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or switched shunt quantities desired (input; no default allowed):

'BASE'	Bus base voltage, in kV.
'PU'	Actual bus voltage magnitude, in pu.
'KV'	Actual bus voltage magnitude, in kV (in pu if base voltage is 0.0).
'ANGLE'	Bus voltage phase angle, in radians.
'ANGLED'	Bus voltage phase angle, in degrees.
'VSWHI'	Controlled quantity upper limit.
'VSWLO'	Controlled quantity lower limit.
'RMPCT'	Percent of total Mvar to be contributed by this switched shunt.
'PU_BSWZERO'	Present zero sequence susceptance switched on in pu.
'PUBZERBLOCK1'	Zero sequence pu step size in block 1.
'PUBZERBLOCK2'	Zero sequence pu step size in block 2.
'PUBZERBLOCK3'	Zero sequence pu step size in block 3.
'PUBZERBLOCK4'	Zero sequence pu step size in block 4.
'PUBZERBLOCK5'	Zero sequence pu step size in block 5.
'PUBZERBLOCK6'	Zero sequence pu step size in block 6.
'PUBZERBLOCK7'	Zero sequence pu step size in block 7.
'PUBZERBLOCK8'	Zero sequence pu step size in block 8.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

For the following, values are returned in MVA, Mvar or susceptance:

'BSWNOM'	Present susceptance switched on (susceptance).
'BSWMAX'	Admittance with all reactors off, all capacitors on.
'BSWMIN'	Admittance with all reactors on, all capacitors off.
'BSWACT'	Present susceptance switched on (Mvar).
'BSWZERO'	Present zero sequence susceptance switched on in pu.
'BSTPBLOCK1'	Step size in block 1.
'BSTPBLOCK2'	Step size in block 2.
'BSTPBLOCK3'	Step size in block 3.
'BSTPBLOCK4'	Step size in block 4.
'BSTPBLOCK5'	Step size in block 5.
'BSTPBLOCK6'	Step size in block 6.
'BSTPBLOCK7'	Step size in block 7.
'BSTPBLOCK8'	Step size in block 8.
'BZERBLOCK1'	Zero sequence step size in block 1.
'BZERBLOCK2'	Zero sequence step size in block 2.
'BZERBLOCK3'	Zero sequence step size in block 3.
'BZERBLOCK4'	Zero sequence step size in block 4.
'BZERBLOCK5'	Zero sequence step size in block 5.
'BZERBLOCK6'	Zero sequence step size in block 6.
'BZERBLOCK7'	Zero sequence step size in block 7.
'BZERBLOCK8'	Zero sequence step size in block 8.
'MISMATCH'	Bus mismatch (0.0 if bus type code > 3).

For the following, values are returned in units determined by the power output option setting:

'O_BSWNOM'	Present susceptance switched on (susceptance).
'O_BSWMAX'	Admittance with all reactors off, all capacitors on.
'O_BSWMIN'	Admittance with all reactors on, all capacitors off.
'O_BSWACT'	Present susceptance switched on (Mvar).
'O_BSWZERO'	Present zero sequence susceptance switched on in pu.
'O_BSTPBLOCK1'	Step size in block 1.
'O_BSTPBLOCK2'	Step size in block 2.
'O_BSTPBLOCK3'	Step size in block 3.
'O_BSTPBLOCK4'	Step size in block 4.
'O_BSTPBLOCK5'	Step size in block 5.
'O_BSTPBLOCK6'	Step size in block 6.
'O_BSTPBLOCK7'	Step size in block 7.
'O_BSTPBLOCK8'	Step size in block 8.
'O_BZERBLOCK1'	Zero sequence step size in block 1.
'O_BZERBLOCK2'	Zero sequence step size in block 2.
'O_BZERBLOCK3'	Zero sequence step size in block 3.
'O_BZERBLOCK4'	Zero sequence step size in block 4.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

'O_BZERBLOCK5'Zero sequence step size in block 5.

'O_BZERBLOCK6'Zero sequence step size in block 6.

'O_BZERBLOCK7'Zero sequence step size in block 7.

'O_BZERBLOCK8'Zero sequence step size in block 8.

'O_MISMATCH' Bus mismatch (0.0 if bus type code > 3).

Integer DIM — Is the number of elements in each column of the array RARRAY (input; no default allowed).

DIM must be as least as large as the value of SHNTS returned by the GET_SWITCHED_SHUNT_COUNT API when it is called with the same values specified for SID and FLAG.

Real RARRAY — Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer SHNTS — Is the number of elements in each column of RARRAY that are set by this API (output).

SHNTS is identical to the value of SHNTS returned by the GET_SWITCHED_SHUNT_COUNT API when it is called with the same values specified for SID and FLAG.

Integer IERR — Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid FLAG value.
IERR = 4	Invalid NSTR value.
IERR = 5	DIM, and hence the size of RARRAY, is not large enough.
IERR = 6	Invalid STRING value.
IERR = 7	Sequence data not in case (when STRING = 'BSWZERO', 'PU_BSWZERO', 'O_BSWZERO', 'BZERBLOCKn', 'PUBZERBLOCKn', 'O_BZERBLOCKn').

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.10.4 aSwshCplx

Use this API to return an array of complex values for subsystem switched shunts.

Python syntax:

```
ierr, carray = aswshcplx(sid, flag, string)
```

Fortran syntax:

```
CALL GET_SWITCHED_SHUNT_COMPLEX(SID, FLAG, NSTR, STRING, DIM, CARRAY,
SHNTS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem switched shunts to include (input; 1 by default). FLAG = 1 — for only in-service switched shunts at in-service buses (type code < 4). FLAG = 2 — for all switched shunts at in-service buses (type code < 4). FLAG = 3 — for only in-service switched shunts, including those at type 4 buses. FLAG = 4 — for all switched shunts.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or switched shunt quantities desired (input; no default allowed): 'VOLTAGE' — Actual bus voltage, in pu, rectangular coordinates. For the following, values are returned in MW and Mvar: 'YSWACT' — Present shunt switched on. 'MISMATCH' — Bus mismatch (0.0 if bus type code > 3). For the following, values are returned in units determined by the power output option setting: 'O_YSWACT' — Present shunt switched on. 'O_MISMATCH' — Bus mismatch (0.0 if bus type code > 3).
Integer DIM	Is the number of elements in each column of the array XARRAY (input; no default allowed). DIM must be as least as large as the value of SHNTS returned by the GET_SWITCHED_SHUNT_COUNT API when it is called with the same values specified for SID and FLAG.
Complex XARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer SHNTS	Is the number of elements in each column of XARRAY that are set by this API (output). SHNTS is identical to the value of SHNTS returned by the GET_SWITCHED_SHUNT_COUNT API when it is called with the same values specified for SID and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid FLAG value. IERR = 4 Invalid NSTR value. IERR = 5 DIM, and hence the size of XARRAY, is not large enough. IERR = 6 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.10.5 aSwshChar

Use this API to return an array of character values for subsystem switched shunts.

Python syntax:

```
ierr, carray = aswshchar(sid, flag, string)
```

Fortran syntax:

```
CALL
GET_SWITCHED_SHUNT_CHAR(SID,FLAG,NSTR,STRING,DIM,CARRAY,SHNTS,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem switched shunts to include (input; 1 by default). FLAG = 1 — for only in-service switched shunts at in-service buses (type code < 4). FLAG = 2 — for all switched shunts at in-service buses (type code < 4). FLAG = 3 — for only in-service switched shunts, including those at type 4 buses. FLAG = 4 — for all switched shunts.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or switched shunt quantities desired (input; no default allowed): 'VSCNAME' — VSC dc line name; non-blank name is returned only for those switched shunts for which the control mode, MODSW, is 4 (12 characters). 'FACTSNAME' — FACTS device name; non-blank name is returned only for those switched shunts for which the control mode, MODSW, is 6 (12 characters). 'DEVICENAME' — VSC dc line name or FACTS device name; non-blank name is returned only for those switched shunts for which the control mode, MODSW, is 4 or 6 (12 characters). 'NAME' — Bus name (12 characters). 'EXNAME' — Extended bus name (18 characters). 'IREGNAME' — Bus name of regulated bus (12 characters). 'IREGEXNAME' — Extended bus name of regulated bus (18 characters).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer DIM	Is the number of elements in each column of the array CARRAY (input; no default allowed). DIM must be as least as large as the value of SHNTS returned by the GET_SWITCHED_SHUNT_COUNT API when it is called with the same values specified for SID and FLAG.
Character CARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer SHNTS	Is the number of elements in each column of CARRAY that are set by this API (output). SHNTS is identical to the value of SHNTS returned by the GET_SWITCHED_SHUNT_COUNT API when it is called with the same values specified for SID and FLAG.
Integer IERR	Is the error code (output): IERR = 0 — No error. IERR = 1 — Working case is empty. IERR = 2 — Invalid SID value. IERR = 3 — Invalid FLAG value. IERR = 4 — Invalid NSTR value. IERR = 5 — DIM, and hence the size of CARRAY, is not large enough. IERR = 6 — Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.10.6 aSwshTypes

Use this API to return an array of character values indicating the data types corresponding to a set of specified STRING values that are valid input values to any of the data retrieval routines of the switched shunt data family (aSwshInt, aSwshReal, aSwshCplx and aSwshChar).

Python syntax:

```
ierr, types = aswshtypes(string)
```

Fortran syntax:

```
CALL GET_SWITCHED_SHUNT_TYPES(NSTR,STRING,TYPES,IERR)
```

where:

Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR strings that are valid STRING values for any of the data retrieval routines of the switched shunt data family (input; no default allowed):
Character TYPES*1	Is an array of at least NSTR elements, each of which contains one of the following characters indicating the data type of the data associated with the corresponding element of STRING (output): 'I' Integer. 'R' Real. 'X' Complex. 'C' Character. '?' Invalid STRING value.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = n Position of the first element in STRING that contains a value that is not a valid STRING value for any of the data retrieval routines in the switched shunt data family (1 ≤ n ≤ NSTR). IERR = 99 Invalid NSTR value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.11 Branch Flow Data

The branch flow data family of subsystem data retrieval APIs returns arrays of loading data for ac branches in a specified subsystem. These APIs return loading data for non-transformer branches, two-winding transformers, and three-winding transformers.

One of the input items to each API allows the user to elect either to retrieve data for all subsystem branches, for subsystem tie branches, or for both subsystem and tie branches. Another input item allows the user to elect either to retrieve data for all such branches, or to restrict the array entries to data for in-service branches.

Data for subsystem branches is returned from all APIs in the branch flow data family in the same order. Unless otherwise restricted for tie branches, each branch appears in both directions (in all three directions for three-winding transformers) in the arrays returned by the APIs of the branch flow data family. Branches are ordered in ascending numerical or alphabetical order by from bus, according to the bus output option setting in effect at the time the API is referenced. For each from bus, branches are ordered in ascending order by to bus. Parallel branches are ordered in ascending circuit identifier order.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.11.1 aFlowCount

Use this API to return the number of array entries required to accommodate the data to be returned by the remaining members of the branch flow data family.

Python syntax:

```
ierr, brnchs = aflowcount(sid, owner, ties, flag)
```

Fortran syntax:

```
CALL GET_BRANCH_FLOW_COUNT(SID, OWNER, TIES, FLAG, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer OWNER	Is a flag indicating owner usage if ownership is a subsystem selection criterion (ignored if SID is negative) (input; 1 by default). OWNER = 1 to use bus ownership. OWNER = 2 to use branch ownership.
Integer TIES	Is a flag indicating which subsystem branches to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for each end of interior subsystem branches only. TIES = 2 for the subsystem bus end of tie branches only. TIES = 3 for the non-subsystem bus end of tie branches only. TIES = 4 for each end of tie branches only. TIES = 5 for each end of interior subsystem branches and the subsystem bus end of tie branches. TIES = 6 for each end of interior subsystem branches and tie branches.
Integer FLAG	Is a flag indicating which subsystem branches to include (input; 1 by default). FLAG = 1 for only in-service branches. FLAG = 2 for all branches.
Integer BRNCHS	Is the number of array entries required for the subsystem indicated by SID and OWNER that meet the editing criteria indicated by TIES and FLAG (output).
Integer IERR	Is the error code (output): IERR = 0 No error, BRNCHS returned. IERR = 1 Working case is empty; BRNCHS returned as 0. IERR = 2 Invalid SID value; BRNCHS returned as 0. IERR = 3 Invalid OWNER value; BRNCHS returned as 0. IERR = 4 Invalid TIES value; BRNCHS returned as 0. IERR = 5 Invalid FLAG value; BRNCHS returned as 0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.11.2 aFlowInt

Use this API to return an array of integer values for subsystem branches.

Python syntax:

```
ierr, iarray = aflowint(sid, owner, ties, flag, string)
```

Fortran syntax:

```
CALL GET_BRANCH_FLOW_INTEGER(SID, OWNER, TIES, FLAG, NSTR, STRING,
DIM, IARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer OWNER	Is a flag indicating owner usage if ownership is a subsystem selection criterion (ignored if SID is negative) (input; 1 by default). OWNER = 1 to use bus ownership. OWNER = 2 to use branch ownership.
Integer TIES	Is a flag indicating which subsystem branches to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for each end of interior subsystem branches only. TIES = 2 for the subsystem bus end of tie branches only. TIES = 3 for the non-subsystem bus end of tie branches only. TIES = 4 for each end of tie branches only. TIES = 5 for each end of interior subsystem branches and the subsystem bus end of tie branches. TIES = 6 for each end of interior subsystem branches and tie branches.
Integer FLAG	Is a flag indicating which subsystem branches to include (input; 1 by default). FLAG = 1 for only in-service branches. FLAG = 2 for all branches.
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the branch quantities desired (input; no default allowed): 'FROMNUMBER' From bus number. 'TONUMBER' To bus number (>10000000 for a three-winding transformer winding). 'STATUS' Branch status. 'NMETERNUMBER' Non-metered end bus number. 'OWNERS' Number of owners. 'OWN1' First owner. 'OWN2' Second owner. 'OWN3' Third owner. 'OWN4' Fourth owner.
Integer DIM	Is the number of elements in each column of the array IARRAY (input; no default allowed). DIM must be as least as large as the value of BRNCHS returned by the GET_BRANCH_FLOW_COUNT API when it is called with the same values specified for SID, OWNER, TIES and FLAG.
Integer IARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BRNCHS	Is the number of elements in each column of IARRAY that are set by this API (output). BRNCHS is identical to the value of BRNCHS returned by the GET_BRANCH_FLOW_COUNT API when it is called with the same values specified for SID, OWNER, TIES and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid OWNER value. IERR = 4 Invalid TIES value. IERR = 5 Invalid FLAG value. IERR = 6 Invalid NSTR value. IERR = 7 DIM, and hence the size of IARRAY, is not large enough. IERR = 8 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.11.3 aFlowReal

Use this API to return an array of real values for subsystem branches.

Python syntax:

```
ierr, rarray = aflowreal(sid, owner, ties, flag, string)
```

Fortran syntax:

```
CALL GET_BRANCH_FLOW_REAL(SID, OWNER, TIES, FLAG, NSTR, STRING, DIM,
RARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer OWNER	Is a flag indicating owner usage if ownership is a subsystem selection criterion (ignored if SID is negative) (input; 1 by default). OWNER = 1 to use bus ownership. OWNER = 2 to use branch ownership.
Integer TIES	Is a flag indicating which subsystem branches to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for each end of interior subsystem branches only. TIES = 2 for the subsystem bus end of tie branches only. TIES = 3 for the non-subsystem bus end of tie branches only. TIES = 4 for each end of tie branches only. TIES = 5 for each end of interior subsystem branches and the subsystem bus end of tie branches. TIES = 6 for each end of interior subsystem branches and tie branches.
Integer FLAG	Is a flag indicating which subsystem branches to include (input; 1 by default). FLAG = 1 for only in-service branches. FLAG = 2 for all branches.
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)

Is an array of NSTR elements specifying NSTR of the following strings indicating the branch quantities desired (input; no default allowed):

'AMPS'	Branch current in amps (0.0 if bus base voltage is 0.0).
'PUCUR'	Branch current in pu.
'PCTRATE'	Percent from bus current of default rating set.
'PCTRATEA'	Percent from bus current of rating set A.
'PCTRATEB'	Percent from bus current of rating set B.
'PCTRATEC'	Percent from bus current of rating set C.
'PCTMVARATE'	Percent from bus MVA of default rating set.
'PCTMVARATEA'	Percent from bus MVA of rating set A.
'PCTMVARATEB'	Percent from bus MVA of rating set B.
'PCTMVARATEC'	Percent from bus MVA of rating set C.
'PCTCORPRATE'	Percent from bus current or MVA loading (according to the appropriate percent loading units program option setting) of default rating set.
'PCTCORPRATEA'	Percent from bus current or MVA loading (according to the appropriate percent loading units program option setting) of rating set A.
'PCTCORPRATEB'	Percent from bus current or MVA loading (according to the appropriate percent loading units program option setting) of rating set B.
'PCTCORPRATEC'	Percent from bus current or MVA loading (according to the appropriate percent loading units program option setting) of rating set C.
'FRACT1'	First owner fraction.
'FRACT2'	Second owner fraction.
'FRACT3'	Third owner fraction.
'FRACT4'	Fourth owner fraction.
'RATE'	Rating from default rating set.
'RATEA'	Rating from rating set A.
'RATEB'	Rating from rating set B.
'RATEC'	Rating from rating set C.

For the following, values are returned in MW, Mvar, or MVA:

'P'	Active power flow.
'Q'	Reactive power flow.
'MVA'	|P + j Q|.
'PLOSS'	Active power losses.
'QLOSS'	Reactive power losses.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

For the following, values are returned in units determined by the power output option setting:

'O_P'	Active power flow.
'O_Q'	Reactive power flow.
'O_MVA'	|P + j Q|.
'O_PLOSS'	Active power losses.
'O_QLOSS'	Reactive power losses.

Integer DIM — Is the number of elements in each column of the array RARRAY (input; no default allowed).

DIM must be as least as large as the value of BRNCHS returned by the GET_BRANCH_FLOW_COUNT API when it is called with the same values specified for SID, OWNER, TIES and FLAG.

Real RARRAY — Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer BRNCHS — Is the number of elements in each column of RARRAY that are set by this API (output).

BRNCHS is identical to the value of BRNCHS returned by the GET_BRANCH_FLOW_COUNT API when it is called with the same values specified for SID, OWNER, TIES and FLAG.

Integer IERR — Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid OWNER value.
IERR = 4	Invalid TIES value.
IERR = 5	Invalid FLAG value.
IERR = 6	Invalid NSTR value.
IERR = 7	DIM, and hence the size of RARRAY, is not large enough.
IERR = 8	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.11.4 aFlowCplx

Use this API to return an array of complex values for subsystem branches.

Python syntax:

```
ierr, xarray = aflowcplx(sid, owner, ties, flag, string)
```

Fortran syntax:

```
CALL GET_BRANCH_FLOW_COMPLEX(SID, OWNER, TIES, FLAG, NSTR, STRING,
DIM, XARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer OWNER	Is a flag indicating owner usage if ownership is a subsystem selection criterion (ignored if SID is negative) (input; 1 by default). OWNER = 1 to use bus ownership. OWNER = 2 to use branch ownership.
Integer TIES	Is a flag indicating which subsystem branches to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for each end of interior subsystem branches only. TIES = 2 for the subsystem bus end of tie branches only. TIES = 3 for the non-subsystem bus end of tie branches only. TIES = 4 for each end of tie branches only. TIES = 5 for each end of interior subsystem branches and the subsystem bus end of tie branches. TIES = 6 for each end of interior subsystem branches and tie branches.
Integer FLAG	Is a flag indicating which subsystem branches to include (input; 1 by default). FLAG = 1 for only in-service branches. FLAG = 2 for all branches.
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the branch quantities desired (input; no default allowed): For the following, values are returned in MW and Mvar: 'PQ' P + j Q flow at from bus end. 'PQLOSS' Losses. For the following, values are returned in units determined by the power output option setting: 'O_PQ' P + j Q flow at from bus end. 'O_PQLOSS' Losses.
Integer DIM	Is the number of elements in each column of the array XARRAY (input; no default allowed). DIM must be as least as large as the value of BRNCHS returned by the GET_BRANCH_FLOW_COUNT API when it is called with the same values specified for SID, OWNER, TIES and FLAG.
Complex XARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BRNCHS	Is the number of elements in each column of XARRAY that are set by this API (output). BRNCHS is identical to the value of BRNCHS returned by the GET_BRANCH_FLOW_COUNT API when it is called with the same values specified for SID, OWNER, TIES and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid OWNER value. IERR = 4 Invalid TIES value. IERR = 5 Invalid FLAG value. IERR = 6 Invalid NSTR value. IERR = 7 DIM, and hence the size of XARRAY, is not large enough. IERR = 8 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.11.5 aFlowChar

Use this API to return an array of character values for subsystem branches.

Python syntax:

```
ierr, carray = aflowchar(sid, owner, ties, flag, string)
```

Fortran syntax:

```
CALL GET_BRANCH_FLOW_CHAR(SID, OWNER, TIES, FLAG, NSTR, STRING, DIM,
CARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer OWNER	Is a flag indicating owner usage if ownership is a subsystem selection criterion (ignored if SID is negative) (input; 1 by default). OWNER = 1 to use bus ownership. OWNER = 2 to use branch ownership.
Integer TIES	Is a flag indicating which subsystem branches to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for each end of interior subsystem branches only. TIES = 2 for the subsystem bus end of tie branches only. TIES = 3 for the non-subsystem bus end of tie branches only. TIES = 4 for each end of tie branches only. TIES = 5 for each end of interior subsystem branches and the subsystem bus end of tie branches. TIES = 6 for each end of interior subsystem branches and tie branches.
Integer FLAG	Is a flag indicating which subsystem branches to include (input; 1 by default). FLAG = 1 for only in-service branches. FLAG = 2 for all branches.
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the branch quantities desired (input; no default allowed): 'ID' Circuit identifier (2 characters). 'FROMNAME' From bus name (12 characters). 'FROMEXNAME' From bus extended bus name (18 characters). 'TONAME' To bus name (three-winding transformer name for a three-winding transformer winding) (12 characters). 'TOEXNAME' To bus extended bus name (three-winding transformer name and winding number for a three-winding transformer winding) (18 characters). 'NMETERNAME' Non-metered bus name (12 characters). 'NMETEREXNAME' Non-metered bus extended bus name (18 characters).
Integer DIM	Is the number of elements in each column of the array CARRAY (input; no default allowed). DIM must be as least as large as the value of BRNCHS returned by the GET_BRANCH_FLOW_COUNT API when it is called with the same values specified for SID, OWNER, TIES and FLAG.
Character CARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BRNCHS	Is the number of elements in each column of CARRAY that are set by this API (output). BRNCHS is identical to the value of BRNCHS returned by the GET_BRANCH_FLOW_COUNT API when it is called with the same values specified for SID, OWNER, TIES and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid OWNER value. IERR = 4 Invalid TIES value. IERR = 5 Invalid FLAG value. IERR = 6 Invalid NSTR value. IERR = 7 DIM, and hence the size of CARRAY, is not large enough. IERR = 8 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.11.6 aFlowTypes

Use this API to return an array of character values indicating the data types corresponding to a set of specified STRING values that are valid input values to any of the data retrieval routines of the branch flow data family (aFlowInt, aFlowReal, aFlowCplx and aFlowChar).

Python syntax:

```
ierr, types = aflowtypes(string)
```

Fortran syntax:

```
CALL GET_BRANCH_FLOW_TYPES(NSTR,STRING,TYPES,IERR)
```

where:

Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR strings that are valid STRING values for any of the data retrieval routines of the branch flow data family (input; no default allowed):
Character TYPES*1	Is an array of at least NSTR elements, each of which contains one of the following characters indicating the data type of the data associated with the corresponding element of STRING (output): 'I' Integer. 'R' Real. 'X' Complex. 'C' Character. '?' Invalid STRING value.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = n Position of the first element in STRING that contains a value that is not a valid STRING value for any of the data retrieval routines in the branch flow data family (1 ≤ n ≤ NSTR). IERR = 99 Invalid NSTR value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.12 Branch Data

The branch data family of subsystem data retrieval APIs returns arrays of data for two-terminal ac branches in a specified subsystem. These APIs return data for non-transformer branches and/or two-winding transformers.

One of the input items to each API allows the user to elect either to retrieve data for all subsystem branches, for subsystem tie branches, or for both subsystem and tie branches. Another input item allows the user to elect to retrieve data for only non-transformer branches, only two-winding transformers, or both non-transformer branches and two-winding transformers, and either to retrieve data for all such branches, or to restrict the array entries to data for in-service branches.

Data for subsystem branches is returned from all APIs in the branch data family in the same order. One of the input items to each API allows the user to elect that either single entry arrays (each branch appears once) or double entry arrays (each branch appears in both directions) are to be returned. Branches are ordered in ascending numerical or alphabetical order by from bus, according to the bus output option setting in effect at the time the API is referenced. For each from bus, branches are ordered in ascending order by to bus. Parallel branches are ordered in ascending circuit identifier order.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.12.1 aBrnCount

Use this API to return the number of array entries required to accommodate the data to be returned by the remaining members of the branch data family.

Python syntax:

```
ierr, brnchs = abrncount(sid, owner, ties, flag, entry)
```

Fortran syntax:

```
CALL GET_BRANCH_COUNT(SID, OWNER, TIES, FLAG, ENTRY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer OWNER	Is a flag indicating owner usage if ownership is a subsystem selection criterion (ignored if SID is negative) (input; 1 by default). OWNER = 1 to use bus ownership. OWNER = 2 to use branch ownership.
Integer TIES	Is a flag indicating which subsystem branches to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem branches only. TIES = 2 for subsystem tie branches only. TIES = 3 for both interior subsystem branches and tie branches.
Integer FLAG	Is a flag indicating which subsystem branches to include (input; 1 by default). FLAG = 1 for only in-service non-transformer branches. FLAG = 2 for all non-transformer branches. FLAG = 3 for only in-service non-transformer branches and two-winding transformers. FLAG = 4 for all non-transformer branches and two-winding transformers. FLAG = 5 for only in-service two-winding transformers. FLAG = 6 for all two-winding transformers.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for single entry (each branch once). ENTRY = 2 for double entry (each branch in both directions).
Integer BRNCHS	Is the number of array entries required for the subsystem indicated by SID and OWNER that meet the editing criteria indicated by TIES, FLAG and ENTRY (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output):	
	IERR = 0	No error, BRNCHS returned.
	IERR = 1	Working case is empty; BRNCHS returned as 0.
	IERR = 2	Invalid SID value; BRNCHS returned as 0.
	IERR = 3	Invalid OWNER value; BRNCHS returned as 0.
	IERR = 4	Invalid TIES value; BRNCHS returned as 0.
	IERR = 5	Invalid FLAG value; BRNCHS returned as 0.
	IERR = 6	Invalid ENTRY value; BRNCHS returned as 0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.12.2 aBrnInt

Use this API to return an array of integer values for subsystem branches.

Python syntax:

```
ierr, iarray = abrnint(sid, owner, ties, flag, entry, string)
```

Fortran syntax:

```
CALL GET_BRANCH_INTEGER(SID, OWNER, TIES, FLAG, ENTRY, NSTR, STRING,
DIM, IARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer OWNER	Is a flag indicating owner usage if ownership is a subsystem selection criterion (ignored if SID is negative) (input; 1 by default). OWNER = 1 to use bus ownership. OWNER = 2 to use branch ownership.
Integer TIES	Is a flag indicating which subsystem branches to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem branches only. TIES = 2 for subsystem tie branches only. TIES = 3 for both interior subsystem branches and tie branches.
Integer FLAG	Is a flag indicating which subsystem branches to include (input; 1 by default). FLAG = 1 for only in-service non-transformer branches. FLAG = 2 for all non-transformer branches. FLAG = 3 for only in-service non-transformer branches and two-winding transformers. FLAG = 4 for all non-transformer branches and two-winding transformers. FLAG = 5 for only in-service two-winding transformers. FLAG = 6 for all two-winding transformers.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for single entry (each branch once). ENTRY = 2 for double entry (each branch in both directions).
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the branch quantities desired (input; no default allowed): 'FROMNUMBER' From bus number. 'TONUMBER' To bus number. 'STATUS' Branch status. 'METERNUMBER' Metered end bus number. 'NMETERNUMBER' Non-metered end bus number. 'OWNERS' Number of owners. 'OWN1' First owner. 'OWN2' Second owner. 'OWN3' Third owner. 'OWN4' Fourth owner.
Integer DIM	Is the number of elements in each column of the array IARRAY (input; no default allowed). DIM must be as least as large as the value of BRNCHS returned by the GET_BRANCH_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.
Integer IARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BRNCHS	Is the number of elements in each column of IARRAY that are set by this API (output). BRNCHS is identical to the value of BRNCHS returned by the GET_BRANCH_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid OWNER value. IERR = 4 Invalid TIES value. IERR = 5 Invalid FLAG value. IERR = 6 Invalid ENTRY value. IERR = 7 Invalid NSTR value. IERR = 8 DIM, and hence the size of IARRAY, is not large enough. IERR = 9 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.12.3 aBrnReal

Use this API to return an array of real values for subsystem branches.

Python syntax:

```
ierr, rarray = abrnreal(sid, owner, ties, flag, entry, string)
```

Fortran syntax:

```
CALL GET_BRANCH_REAL(SID, OWNER, TIES, FLAG, ENTRY, NSTR, STRING,
DIM, RARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer OWNER	Is a flag indicating owner usage if ownership is a subsystem selection criterion (ignored if SID is negative) (input; 1 by default). OWNER = 1 to use bus ownership. OWNER = 2 to use branch ownership.
Integer TIES	Is a flag indicating which subsystem branches to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem branches only. TIES = 2 for subsystem tie branches only. TIES = 3 for both interior subsystem branches and tie branches.
Integer FLAG	Is a flag indicating which subsystem branches to include (input; 1 by default). FLAG = 1 for only in-service non-transformer branches. FLAG = 2 for all non-transformer branches. FLAG = 3 for only in-service non-transformer branches and two-winding transformers. FLAG = 4 for all non-transformer branches and two-winding transformers. FLAG = 5 for only in-service two-winding transformers. FLAG = 6 for all two-winding transformers.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for single entry (each branch once). ENTRY = 2 for double entry (each branch in both directions).
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)

Is an array of NSTR elements specifying NSTR of the following strings indicating the branch quantities desired (input; no default allowed):

'AMPS'	Branch current in amps (0.0 if bus base voltage is 0.0).
'PUCUR'	Branch current in pu.
'PCTRATE'	Percent from bus current of default rating set.
'PCTRATEA'	Percent from bus current of rating set A.
'PCTRATEB'	Percent from bus current of rating set B.
'PCTRATEC'	Percent from bus current of rating set C.
'PCTMVARATE'	Percent from bus MVA of default rating set.
'PCTMVARATEA'	Percent from bus MVA of rating set A.
'PCTMVARATEB'	Percent from bus MVA of rating set B.
'PCTMVARATEC'	Percent from bus MVA of rating set C.
'PCTCORPRATE'	Percent from bus current or MVA loading (according to the appropriate percent loading units program option setting) of default rating set.
'PCTCORPRATEA'	Percent from bus current or MVA loading (according to the appropriate percent loading units program option setting) of rating set A.
'PCTCORPRATEB'	Percent from bus current or MVA loading (according to the appropriate percent loading units program option setting) of rating set B.
'PCTCORPRATEC'	Percent from bus current or MVA loading (according to the appropriate percent loading units program option setting) of rating set C.
'MAXPCTRATE'	Larger of percent from/to bus current of default rating set.
'MAXPCTRATEA'	Larger of percent from/to bus current of rating set A.
'MAXPCTRATEB'	Larger of percent from/to bus current of rating set B.
'MAXPCTRATEC'	Larger of percent from/to bus current of rating set C.
'MXPCTMVARAT'	Larger of percent from/to bus MVA of default rating set.
'MXPCTMVARATA'	Larger of percent from/to bus MVA of rating set A.
'MXPCTMVARATB'	Larger of percent from/to bus MVA of rating set B.
'MXPCTMVARATC'	Larger of percent from/to bus MVA of rating set C.
'MXPCTCRPRAT'	Larger of percent from/to bus current or MVA loading (according to the appropriate percent loading units program option setting) of default rating set.
'MXPCTCRPRATA'	Larger of percent from/to bus current or MVA loading (according to the appropriate percent loading units program option setting) of rating set A.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

'MXPCTCRPRATB'
Larger of percent from/to bus current or MVA loading (according to the appropriate percent loading units program option setting) of rating set B.

'MXPCTCRPRATC'
Larger of percent from/to bus current or MVA loading (according to the appropriate percent loading units program option setting) of rating set C.

'FRACT1'	First owner fraction.
'FRACT2'	Second owner fraction.
'FRACT3'	Third owner fraction.
'FRACT4'	Fourth owner fraction.
'RATE'	Rating from default rating set.
'RATEA'	Rating from rating set A.
'RATEB'	Rating from rating set B.
'RATEC'	Rating from rating set C.
'LENGTH'	Line length.
'CHARGING'	Total charging capacitance in pu.
'CHARGINGZER0'	Zero sequence total charging capacitance in pu.

For the following, values are returned in MW, Mvar, or MVA:

'P'	Active power flow at from bus end.
'Q'	Reactive power flow at from bus end.
'MVA'	|P + j Q| at from bus end.
'MAXMVA'	|P + j Q| at from bus or to bus end, whichever is larger.
'PLOSS'	Active power losses.
'QLOSS'	Reactive power losses.

For the following, values are returned in units determined by the power output option setting:

'O_P'	Active power flow at from bus end.
'O_Q'	Reactive power flow at from bus end.
'O_MVA'	|P + j Q| at from bus end.
'O_MAXMVA'	||P + j Q| at from bus or to bus end, whichever is larger.
'O_PLOSS'	Active power losses.
'O_QLOSS'	Reactive power losses.

Integer DIM — Is the number of elements in each column of the array RARRAY (input; no default allowed).

DIM must be as least as large as the value of BRNCHS returned by the GET_BRANCH_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.

Real RARRAY — Is an array of at least NSTR columns, each of which contains DIM elements (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer BRNCHS	Is the number of elements in each column of RARRAY that are set by this API (output). BRNCHS is identical to the value of BRNCHS returned by the GET_BRANCH_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.
Integer IERR	Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid OWNER value.
IERR = 4	Invalid TIES value.
IERR = 5	Invalid FLAG value.
IERR = 6	Invalid ENTRY value.
IERR = 7	Invalid NSTR value.
IERR = 8	DIM, and hence the size of RARRAY, is not large enough.
IERR = 9	Invalid STRING value.
IERR =10	Sequence data not in case (when STRING = ’CHARGINGZERO’).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.12.4 aBrnCplx

Use this API to return an array of complex values for subsystem branches.

Python syntax:

```
ierr, xarray = abrncplx(sid, owner, ties, flag, entry, string)
```

Fortran syntax:

```
CALL GET_BRANCH_COMPLEX(SID, OWNER, TIES, FLAG, ENTRY, NSTR, STRING,
DIM, XARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer OWNER	Is a flag indicating owner usage if ownership is a subsystem selection criterion (ignored if SID is negative) (input; 1 by default). OWNER = 1 to use bus ownership. OWNER = 2 to use branch ownership.
Integer TIES	Is a flag indicating which subsystem branches to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem branches only. TIES = 2 for subsystem tie branches only. TIES = 3 for both interior subsystem branches and tie branches.
Integer FLAG	Is a flag indicating which subsystem branches to include (input; 1 by default). FLAG = 1 for only in-service non-transformer branches. FLAG = 2 for all non-transformer branches. FLAG = 3 for only in-service non-transformer branches and two-winding transformers. FLAG = 4 for all non-transformer branches and two-winding transformers. FLAG = 5 for only in-service two-winding transformers. FLAG = 6 for all two-winding transformers.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for single entry (each branch once). ENTRY = 2 for double entry (each branch in both directions).
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the branch quantities desired (input; no default allowed): 'RX' Branch impedance in pu. 'FROMSHNT' Line shunt at from bus end in pu. 'TOSHNT' Line shunt at to bus end in pu. 'RXZERO' Zero sequence branch impedance in pu. 'FROMSHNTZERO' Zero sequence line shunt at from bus end in pu. 'TOSHNTZERO' Zero sequence line shunt at to bus end in pu. For the following values are returned in MW and Mvar: 'PQ' P + j Q flow at from bus end. 'PQLOSS' Losses. For the following, values are returned in units determined by the power output option setting: 'O_PQ' P + j Q flow at from bus end. 'O_PQLOSS' Losses.
Integer DIM	Is the number of elements in each column of the array XARRAY (input; no default allowed). DIM must be as least as large as the value of BRNCHS returned by the GET_BRANCH_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.
Complex XARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BRNCHS	Is the number of elements in each column of XARRAY that are set by this API (output). BRNCHS is identical to the value of BRNCHS returned by the GET_BRANCH_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid OWNER value. IERR = 4 Invalid TIES value. IERR = 5 Invalid FLAG value. IERR = 6 Invalid ENTRY value. IERR = 7 Invalid NSTR value. IERR = 8 DIM, and hence the size of XARRAY, is not large enough. IERR = 9 Invalid STRING value. IERR = 10 Sequence data not in case (when STRING = 'RXZERO', 'FROMSHNTZERO' or 'TOSHNTZERO').

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.12.5 aBrnChar

Use this API to return an array of character values for subsystem branches.

Python syntax:

```
ierr, carray = abrnchar(sid, owner, ties, flag, entry, string)
```

Fortran syntax:

```
CALL GET_BRANCH_CHAR(SID, OWNER, TIES, FLAG, ENTRY, NSTR, STRING,
DIM, CARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer OWNER	Is a flag indicating owner usage if ownership is a subsystem selection criterion (ignored if SID is negative) (input; 1 by default). OWNER = 1 to use bus ownership. OWNER = 2 to use branch ownership.
Integer TIES	Is a flag indicating which subsystem branches to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem branches only. TIES = 2 for subsystem tie branches only. TIES = 3 for both interior subsystem branches and tie branches.
Integer FLAG	Is a flag indicating which subsystem branches to include (input; 1 by default). FLAG = 1 for only in-service non-transformer branches. FLAG = 2 for all non-transformer branches. FLAG = 3 for only in-service non-transformer branches and two-winding transformers. FLAG = 4 for all non-transformer branches and two-winding transformers. FLAG = 5 for only in-service two-winding transformers. FLAG = 6 for all two-winding transformers.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for single entry (each branch once). ENTRY = 2 for double entry (each branch in both directions).
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the branch quantities desired (input; no default allowed): 'ID' Circuit identifier (2 characters). 'FROMNAME' From bus name (12 characters). 'FROMEXNAME' From bus extended bus name (18 characters). 'TONAME' To bus name (three-winding transformer name for a three-winding transformer winding) (12 characters). 'TOEXNAME' To bus extended bus name (three-winding transformer name and winding number for a three-winding transformer winding) (18 characters). 'METERNAME' Metered bus name (12 characters). 'METEREXNAME' Metered bus extended bus name (18 characters). 'NMETERNAME' Non-metered bus name (12 characters). 'NMETEREXNAME' Non-metered bus extended bus name (18 characters).
Integer DIM	Is the number of elements in each column of the array CARRAY (input; no default allowed). DIM must be as least as large as the value of BRNCHS returned by the GET_BRANCH_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.
Character CARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BRNCHS	Is the number of elements in each column of CARRAY that are set by this API (output). BRNCHS is identical to the value of BRNCHS returned by the GET_BRANCH_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid OWNER value. IERR = 4 Invalid TIES value. IERR = 5 Invalid FLAG value. IERR = 6 Invalid ENTRY value. IERR = 7 Invalid NSTR value. IERR = 8 DIM, and hence the size of CARRAY, is not large enough. IERR = 9 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.12.6 aBrnTypes

Use this API to return an array of character values indicating the data types corresponding to a set of specified STRING values that are valid input values to any of the data retrieval routines of the branch data family (aBrnInt, aBrnReal, aBrnCplx and aBrnChar).

Python syntax:

```
ierr, types = abrntypes(string)
```

Fortran syntax:

```
CALL GET_BRANCH_TYPES(NSTR,STRING,TYPES,IERR)
```

where:

Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR strings that are valid STRING values for any of the data retrieval routines of the branch data family (input; no default allowed):
Character TYPES*1	Is an array of at least NSTR elements, each of which contains one of the following characters indicating the data type of the data associated with the corresponding element of STRING (output): 'I' Integer. 'R' Real. 'X' Complex. 'C' Character. '?' Invalid STRING value.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = n Position of the first element in STRING that contains a value that is not a valid STRING value for any of the data retrieval routines in the branch data family (1 ≤ n ≤ NSTR). IERR = 99 Invalid NSTR value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.13 Two-Winding Transformer Data

The two-winding transformer data family of subsystem data retrieval APIs returns arrays of data for two-winding transformers in a specified subsystem.

One of the input items to each API allows the user to elect either to retrieve data for all subsystem two-winding transformers, for subsystem tie two-winding transformers, or for both subsystem and tie two-winding transformers. Another input item allows the user to elect either to retrieve data for all such transformers, or to restrict the array entries to data for in-service transformers.

Data for subsystem two-winding transformers is returned from all APIs in the two-winding transformer data family in the same order. One of the input items to each API allows the user to elect that either single entry arrays (each transformer appears once) or double entry arrays (each transformer appears in both directions) are to be returned. Transformers are ordered in ascending numerical or alphabetical order by from bus, according to the bus output option setting in effect at the time the API is referenced. For each from bus, transformers are ordered in ascending order by to bus. Parallel transformers are ordered in ascending circuit identifier order.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.13.1 aTrnCount

Use this API to return the number of array entries required to accommodate the data to be returned by the remaining members of the two-winding transformer data family.

Python syntax:

```
ierr, brnchs = atrncount(sid, owner, ties, flag, entry)
```

Fortran syntax:

```
CALL GET_2WINDING_COUNT(SID,OWNER,TIES,FLAG,ENTRY,BRNCHS,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer OWNER	Is a flag indicating owner usage if ownership is a subsystem selection criterion (ignored if SID is negative) (input; 1 by default). OWNER = 1 to use bus ownership. OWNER = 2 to use transformer ownership.
Integer TIES	Is a flag indicating which subsystem two-winding transformers to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem two-winding transformers only. TIES = 2 for subsystem tie two-winding transformers only. TIES = 3 for both interior subsystem and tie two-winding transformers.
Integer FLAG	Is a flag indicating which subsystem two-winding transformers to include (input; 1 by default). FLAG = 1 for only in-service two-winding transformers. FLAG = 2 for all two-winding transformers.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for single entry (each branch once). ENTRY = 2 for double entry (each branch in both directions).
Integer BRNCHS	Is the number of array entries required for the subsystem indicated by SID and OWNER that meet the editing criteria indicated by TIES, FLAG and ENTRY (output).
Integer IERR	Is the error code (output): IERR = 0 No error, BRNCHS returned. IERR = 1 Working case is empty; BRNCHS returned as 0. IERR = 2 Invalid SID value; BRNCHS returned as 0. IERR = 3 Invalid OWNER value; BRNCHS returned as 0. IERR = 4 Invalid TIES value; BRNCHS returned as 0. IERR = 5 Invalid FLAG value; BRNCHS returned as 0. IERR = 6 Invalid ENTRY value; BRNCHS returned as 0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.13.2 aTrnInt

Use this API to return an array of integer values for subsystem two-winding transformers.

Python syntax:

```
ierr, iarray = atrnint(sid, owner, ties, flag, entry, string)
```

Fortran syntax:

```
CALL GET_2WINDING_INTEGER(SID, OWNER, TIES, FLAG, ENTRY, NSTR,
STRING, DIM, IARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer OWNER	Is a flag indicating owner usage if ownership is a subsystem selection criterion (ignored if SID is negative) (input; 1 by default). OWNER = 1 to use bus ownership. OWNER = 2 to use transformer ownership.
Integer TIES	Is a flag indicating which subsystem two-winding transformers to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem two-winding transformers only. TIES = 2 for subsystem tie two-winding transformers only. TIES = 3 for both interior subsystem and tie two-winding transformers.
Integer FLAG	Is a flag indicating which subsystem two-winding transformers to include (input; 1 by default). FLAG = 1 for only in-service two-winding transformers. FLAG = 2 for all two-winding transformers.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for single entry (each branch once). ENTRY = 2 for double entry (each branch in both directions).
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the transformer quantities desired (input; no default allowed): 'FROMNUMBER' From bus number. 'TONUMBER' To bus number. 'STATUS' Transformer status. 'METERNUMBER' Metered end bus number. 'NMETERNUMBER' Non-metered end bus number. 'OWNERS' Number of owners.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

'OWN1'	First owner.
'OWN2'	Second owner.
'OWN3'	Third owner.
'OWN4'	Fourth owner.
'ICONTNUMBER'	Controlled bus number.
'WIND1NUMBER'	Winding 1 bus number.
'WIND2NUMBER'	Winding 2 bus number.
'TABLE'	Transformer impedance adjustment table number.
'CODE'	Transformer control mode for automatic adjustments.
'NTPOSN'	Number of tap positions.
'CW'	Winding data I/O code.
'CZ'	Impedance data I/O code.
'CM'	Magnetizing admittance I/O code.
'CZ0'	Leakage impedance data I/O code.
'CZG'	Grounding impedance data I/O code.
'CNXCOD'	Connection code.
'TPSTT'	Winding 1 ratio limit code: -1 for low, 0 for regulating, 1 for high, -2 for locked, -3 for controlling a dc line.
'ANSTT'	Winding 1 angle limit code: -1 for low, 0 for regulating, 1 for high, -2 for locked.

Integer DIM — Is the number of elements in each column of the array IARRAY (input; no default allowed).

DIM must be as least as large as the value of BRNCHS returned by the GET_2WINDING_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.

Integer IARRAY — Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer BRNCHS — Is the number of elements in each column of IARRAY that are set by this API (output).

BRNCHS is identical to the value of BRNCHS returned by the GET_2WINDING_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output):	
	IERR = 0	No error.
	IERR = 1	Working case is empty.
	IERR = 2	Invalid SID value.
	IERR = 3	Invalid OWNER value.
	IERR = 4	Invalid TIES value.
	IERR = 5	Invalid FLAG value.
	IERR = 6	Invalid ENTRY value.
	IERR = 7	Invalid NSTR value.
	IERR = 8	DIM, and hence the size of IARRAY, is not large enough.
	IERR = 9	Invalid STRING value.
	IERR = 10	Sequence data not in case (when STRING = ’CZ0’, ’CZG’ or ’CNXCOD’).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.13.3 aTrnReal

Use this API to return an array of real values for subsystem two-winding transformers.

Python syntax:

```
ierr, rarray = atrnreal(sid, owner, ties, flag, entry, string)
```

Fortran syntax:

```
CALL GET_2WINDING_REAL(SID, OWNER, TIES, FLAG, ENTRY, NSTR, STRING,
DIM, RARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer OWNER	Is a flag indicating owner usage if ownership is a subsystem selection criterion (ignored if SID is negative) (input; 1 by default). OWNER = 1 to use bus ownership. OWNER = 2 to use transformer ownership.
Integer TIES	Is a flag indicating which subsystem two-winding transformers to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem two-winding transformers only. TIES = 2 for subsystem tie two-winding transformers only. TIES = 3 for both interior subsystem and tie two-winding transformers.
Integer FLAG	Is a flag indicating which subsystem two-winding transformers to include (input; 1 by default). FLAG = 1 for only in-service two-winding transformers. FLAG = 2 for all two-winding transformers.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for single entry (each branch once). ENTRY = 2 for double entry (each branch in both directions).
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the transformer quantities desired (input; no default allowed): 'AMPS' Branch current in amps (0.0 if bus base voltage is 0.0). 'PUCUR' Branch current in pu. 'PCTRATE' Percent from bus current of default rating set. 'PCTRATEA' Percent from bus current of rating set A. 'PCTRATEB' Percent from bus current of rating set B. 'PCTRATEC' Percent from bus current of rating set C.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

’PCTMVARATE’ Percent from bus MVA of default rating set.

’PCTMVARATEA’ Percent from bus MVA of rating set A.

’PCTMVARATEB’ Percent from bus MVA of rating set B.

’PCTMVARATEC’ Percent from bus MVA of rating set C.

’PCTCORPRATE’ Percent from bus current or MVA loading (according to the transformer percent loading units program option setting) of default rating set.

’PCTCORPRATEA’Percent from bus current or MVA loading (according to the transformer percent loading units program option setting) of rating set A.

’PCTCORPRATEB’Percent from bus current or MVA loading (according to the transformer percent loading units program option setting) of rating set B.

’PCTCORPRATEC’Percent from bus current or MVA loading (according to the transformer percent loading units program option setting) of rating set C.

’MAXPCTRATE’ Larger of percent from/to bus current of default rating set.

’MAXPCTRATEA’ Larger of percent from/to bus current of rating set A.

’MAXPCTRATEB’ Larger of percent from/to bus current of rating set B.

’MAXPCTRATEC’ Larger of percent from/to bus current of rating set C.

’MXPCTMVARAT’ Larger of percent from/to bus MVA of default rating set.

’MXPCTMVARATA’
Larger of percent from/to bus MVA of rating set A.

’MXPCTMVARATB’
Larger of percent from/to bus MVA of rating set B.

’MXPCTMVARATC’
Larger of percent from/to bus MVA of rating set C.

’MXPCTCRPRAT’
Larger of percent from/to bus current or MVA loading (according to the transformer percent loading units program option setting) of default rating set.

’MXPCTCRPRATA’Larger of percent from/to bus current or MVA loading (according to the transformer percent loading units program option setting) of rating set A.

’MXPCTCRPRATB’
Larger of percent from/to bus current or MVA loading (according to the transformer percent loading units program option setting) of rating set B.

’MXPCTCRPRATC’
Larger of percent from/to bus current or MVA loading (according to the transformer percent loading units program option setting) of rating set C.

’FRACT1’ First owner fraction.

’FRACT2’ Second owner fraction.

’FRACT3’ Third owner fraction.

’FRACT4’ Fourth owner fraction.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

’RATE’	Rating from default rating set.
’RATEA’	Rating from rating set A.
’RATEB’	Rating from rating set B.
’RATEC’	Rating from rating set C.
’RATIO’	Winding 1 tap ratio in pu.
’RATIOCW’	Winding 1 tap ratio in units governed by CW.
’RATIO2’	Winding 2 tap ratio in pu.
’RATIO2CW’	Winding 2 tap ratio in units governed by CW.
’ANGLE’	Winding 1 phase shift angle in degrees.
’RMAX’	Winding 1 ratio or angle upper limit (in pu if ratio).
’RMAXCW’	Winding 1 ratio or angle upper limit (in units governed by CW if ratio).
’RMIN’	Winding 1 ratio or angle lower limit (in pu if ratio).
’RMINCW’	Winding 1 ratio or angle lower limit (in units governed by CW if ratio).
’VMAX’	Controlled quantity upper limit (in pu if voltage).
’VMAXKV’	Controlled quantity upper limit (in units governed by CW if voltage).
’VMIN’	Controlled quantity lower limit (in pu if voltage).
’VMINKV’	Controlled quantity lower limit (in units governed by CW if voltage).
’STEP’	Winding 1 ratio or angle step width (in pu if ratio).
’STEPCW’	Winding 1 ratio or angle step width (in units governed by CW if ratio).
'CNXANG'	Winding 1 connection angle in degrees.
’NOMV1’	Winding 1 nominal voltage in kv (0.0 for bus base voltage).
’NOMV2’	Winding 2 nominal voltage in kv (0.0 for bus base voltage).
’SBASE1’	Winding 1 base MVA.

For the following, values are returned in MW, Mvar, or MVA:

’P’	Active power flow at from bus end.
’Q’	Reactive power flow at from bus end.
’MVA’	|P + j Q| at from bus end.
’MAXMVA’	|P + j Q| at from bus or to bus end, whichever is larger.
’PLOSS’	Active power losses.
’QLOSS’	Reactive power losses.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

For the following, values are returned in units determined by the power output option setting:

'O_P'	Active power flow at from bus end.
'O_Q'	Reactive power flow at from bus end.
'O_MVA'	|P + j Q| at from bus end.
'O_MAXMVA'	|P + j Q| at from bus or to bus end, whichever is larger.
'O_PLOSS'	Active power losses.
'O_QLOSS'	Reactive power losses.

Integer DIM — Is the number of elements in each column of the array RARRAY (input; no default allowed).

DIM must be as least as large as the value of BRNCHS returned by the GET_2WINDING_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.

Real RARRAY — Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer BRNCHS — Is the number of elements in each column of RARRAY that are set by this API (output).

BRNCHS is identical to the value of BRNCHS returned by the GET_2WINDING_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.

Integer IERR — Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid OWNER value.
IERR = 4	Invalid TIES value.
IERR = 5	Invalid FLAG value.
IERR = 6	Invalid ENTRY value.
IERR = 7	Invalid NSTR value.
IERR = 8	DIM, and hence the size of RARRAY, is not large enough.
IERR = 9	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.13.4 aTrnCplx

Use this API to return an array of complex values for subsystem two-winding transformers.

Python syntax:

```
ierr, xarray = atrncplx(sid, owner, ties, flag, entry, string)
```

Fortran syntax:

```
CALL GET_2WINDING_COMPLEX(SID, OWNER, TIES, FLAG, ENTRY, NSTR,
STRING, DIM, XARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer OWNER	Is a flag indicating owner usage if ownership is a subsystem selection criterion (ignored if SID is negative) (input; 1 by default). OWNER = 1 to use bus ownership. OWNER = 2 to use transformer ownership.
Integer TIES	Is a flag indicating which subsystem two-winding transformers to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem two-winding transformers only. TIES = 2 for subsystem tie two-winding transformers only. TIES = 3 for both interior subsystem and tie two-winding transformers.
Integer FLAG	Is a flag indicating which subsystem two-winding transformers to include (input; 1 by default). FLAG = 1 for only in-service two-winding transformers. FLAG = 2 for all two-winding transformers.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for single entry (each branch once). ENTRY = 2 for double entry (each branch in both directions).
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the transformer quantities desired (input; no default allowed): 'RXACT' Actual transformer impedance in pu. 'RXACTCZ' Actual transformer impedance in units governed by CZ. 'RXNOM' Nominal transformer impedance in pu. 'RXNOMCZ' Nominal transformer impedance in units governed by CZ.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

'YMAG'	Magnetizing admittance in pu.
'YMAGCM'	Magnetizing admittance in units governed by CM.
'COMPRX'	Load drop compensation impedance in pu.
'RXZERO'	Zero sequence branch impedance in pu.
'ZG1, 'ZGRND'	Zero sequence winding 1 grounding impedance in pu.
'ZG1CZG'	Zero sequence winding 1 grounding impedance in units governed by CZG.
'ZG2', 'ZGRND2'	Zero sequence winding 2 grounding impedance in pu.
'ZG2CZG'	Zero sequence winding 2 grounding impedance in units governed by CZG.
'Z01'	Zero sequence leakage impedance connected to the winding 1 bus in pu.
'Z01CZ0'	Zero sequence leakage impedance connected to the winding 1 bus in units governed by CZ0.
'Z02'	Zero sequence leakage impedance connected to the winding 2 bus in pu.
'Z02CZ0'	Zero sequence leakage impedance connected to the winding 2 bus in units governed by CZ0.
'ZNUTRL'	Zero sequence common neutral grounding impedance in pu.
'ZNUTRLCZG'	Zero sequence common neutral grounding impedance in units governed by CZG.

For the following, values are returned in MW and Mvar:

'PQ'	P + j Q flow at from bus end.
'PQLOSS'	Losses.

For the following, values are returned in units determined by the power output option setting:

'O_PQ'	P + j Q flow at from bus end.
'O_PQLOSS'	Losses.

Integer DIM Is the number of elements in each column of the array XARRAY (input; no default allowed).

DIM must be as least as large as the value of BRNCHS returned by the GET_2WINDING_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.

Complex XARRAY Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer BRNCHS Is the number of elements in each column of XARRAY that are set by this API (output).

BRNCHS is identical to the value of BRNCHS returned by the GET_2WINDING_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output):	
	IERR = 0	No error.
	IERR = 1	Working case is empty.
	IERR = 2	Invalid SID value.
	IERR = 3	Invalid OWNER value.
	IERR = 4	Invalid TIES value.
	IERR = 5	Invalid FLAG value.
	IERR = 6	Invalid ENTRY value.
	IERR = 7	Invalid NSTR value.
	IERR = 8	DIM, and hence the size of XARRAY, is not large enough.
	IERR = 9	Invalid STRING value.
	IERR = 10	Sequence data not in case (when STRING = ’RXZERO’, ’ZGRND’, ’ZGRND2’, ’ZG1’, ’ZG1CZG’, ’ZG2’, ’ZG2CZG’, ’Z01’, ’Z01CZ0’, ’Z02’, ’Z02CZ0’, ’ZNUTRL’, or ’ZNUTRLCZG’).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.13.5 aTrnChar

Use this API to return an array of character values for subsystem two-winding transformers.

Python syntax:

```
ierr, carray = atrnchar(sid, owner, ties, flag, entry, string)
```

Fortran syntax:

```
CALL GET_2WINDING_CHAR(SID, OWNER, TIES, FLAG, ENTRY, NSTR, STRING,
DIM, CARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer OWNER	Is a flag indicating owner usage if ownership is a subsystem selection criterion (ignored if SID is negative) (input; 1 by default). OWNER = 1 to use bus ownership. OWNER = 2 to use transformer ownership.
Integer TIES	Is a flag indicating which subsystem two-winding transformers to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem two-winding transformers only. TIES = 2 for subsystem tie two-winding transformers only. TIES = 3 for both interior subsystem and tie two-winding transformers.
Integer FLAG	Is a flag indicating which subsystem two-winding transformers to include (input; 1 by default). FLAG = 1 for only in-service two-winding transformers. FLAG = 2 for all two-winding transformers.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for single entry (each branch once). ENTRY = 2 for double entry (each branch in both directions).
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the transformer quantities desired (input; no default allowed): 'ID' Circuit identifier (2 characters). 'FROMNAME' From bus name (12 characters). 'FROMEXNAME' From bus extended bus name (18 characters). 'TONAME' To bus name (three-winding transformer name for a three-winding transformer winding) (12 characters).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

'TOEXNAME'	To bus extended bus name (three-winding transformer name and winding number for a three-winding transformer winding) (18 characters).
'METERNAME'	Metered bus name (12 characters).
'METEREXNAME'	Metered bus extended bus name (18 characters).
'NMETERNAME'	Non-metered bus name (12 characters).
'NMETEREXNAME'	Non-metered bus extended bus name (18 characters).
'ICONTNAME'	Controlled bus name (12 characters).
'ICONTEXNAME'	Controlled bus extended bus name (18 characters).
'WIND1NAME'	Winding 1 bus name (12 characters).
'WIND1EXNAME'	Winding 1 bus extended bus name (18 characters).
'WIND2NAME'	Winding 2 bus name (12 characters).
'WIND2EXNAME'	Winding 2 bus extended bus name (18 characters).
'XFRNAME'	Transformer name (12 characters).
'VECTORGROUP'	Vector group name (12 characters).

Integer DIM — Is the number of elements in each column of the array CARRAY (input; no default allowed).

DIM must be as least as large as the value of BRNCHS returned by the GET_2WINDING_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.

Character CARRAY — Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer BRNCHS — Is the number of elements in each column of CARRAY that are set by this API (output).

BRNCHS is identical to the value of BRNCHS returned by the GET_2WINDING_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.

Integer IERR — Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid OWNER value.
IERR = 4	Invalid TIES value.
IERR = 5	Invalid FLAG value.
IERR = 6	Invalid ENTRY value.
IERR = 7	Invalid NSTR value.
IERR = 8	DIM, and hence the size of CARRAY, is not large enough.
IERR = 9	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.13.6 aTrnTypes

Use this API to return an array of character values indicating the data types corresponding to a set of specified STRING values that are valid input values to any of the data retrieval routines of the two-winding transformer data family (aTrnInt, aTrnReal, aTrnCplx and aTrnChar).

Python syntax:

```
ierr, types = atrntypes(string)
```

Fortran syntax:

```
CALL GET_2WINDING_TYPES(NSTR,STRING,TYPES,IERR)
```

where:

Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR strings that are valid STRING values for any of the data retrieval routines of the two-winding transformer data family (input; no default allowed):
Character TYPES*1	Is an array of at least NSTR elements, each of which contains one of the following characters indicating the data type of the data associated with the corresponding element of STRING (output): 'I' Integer. 'R' Real. 'X' Complex. 'C' Character. '?' Invalid STRING value.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = n Position of the first element in STRING that contains a value that is not a valid STRING value for any of the data retrieval routines in the two-winding transformer data family (1 ≤ n ≤ NSTR). IERR = 99 Invalid NSTR value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.14 Three-Winding Transformer Data

The three-winding transformer data family of subsystem data retrieval APIs returns arrays of data for three-winding transformers in a specified subsystem.

One of the input items to each API allows the user to elect either to retrieve data for all subsystem three-winding transformers, for subsystem tie three-winding transformers, or for both subsystem and tie three-winding transformers. Another input item allows the user to elect either to retrieve data for all such transformers, or to restrict the array entries to data for in-service transformers.

Data for subsystem three-winding transformers is returned from all APIs in the three-winding transformer data family in the same order. One of the input items to each API allows the user to elect that transformers be ordered either by transformer name, or in ascending numerical or alphabetical order by winding 1 bus, according to the bus output option setting in effect at the time the API is referenced. For each winding 1 bus, transformers are ordered in ascending order by winding 2 bus, then winding 3 bus, then in ascending circuit identifier order.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.14.1 aTr3Count

Use this API to return the number of array entries required to accommodate the data to be returned by the remaining members of the three-winding transformer data family.

Python syntax:

```
ierr, brnchs = atr3count(sid, owner, ties, flag, entry)
```

Fortran syntax:

```
CALL GET_3WINDING_COUNT(SID,OWNER,TIES,FLAG,ENTRY,BRNCHS,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer OWNER	Is a flag indicating owner usage if ownership is a subsystem selection criterion (ignored if SID is negative) (input; 1 by default). OWNER = 1 to use bus ownership. OWNER = 2 to use transformer ownership.
Integer TIES	Is a flag indicating which subsystem three-winding transformers to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem three-winding transformers only. TIES = 2 for subsystem tie three-winding transformers only. TIES = 3 for both interior subsystem and tie three-winding transformers.
Integer FLAG	Is a flag indicating which subsystem three-winding transformers to include (input; 1 by default). FLAG = 1 for only in-service three-winding transformers. FLAG = 2 for all three-winding transformers.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for winding 1 bus order. ENTRY = 2 for transformer name order.
Integer BRNCHS	Is the number of array entries required for the subsystem indicated by SID and OWNER that meet the editing criteria indicated by TIES, FLAG and ENTRY (output).
Integer IERR	Is the error code (output): IERR = 0 No error, BRNCHS returned. IERR = 1 Working case is empty; BRNCHS returned as 0. IERR = 2 Invalid SID value; BRNCHS returned as 0. IERR = 3 Invalid OWNER value; BRNCHS returned as 0. IERR = 4 Invalid TIES value; BRNCHS returned as 0. IERR = 5 Invalid FLAG value; BRNCHS returned as 0. IERR = 6 Invalid ENTRY value; BRNCHS returned as 0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.14.2 aTr3Int

Use this API to return an array of integer values for subsystem three-winding transformers.

Python syntax:

```
ierr, iarray = atr3int(sid, owner, ties, flag, entry, string)
```

Fortran syntax:

```
CALL GET_3WINDING_INTEGER(SID, OWNER, TIES, FLAG, ENTRY, NSTR,
STRING, DIM, IARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer OWNER	Is a flag indicating owner usage if ownership is a subsystem selection criterion (ignored if SID is negative) (input; 1 by default). OWNER = 1 to use bus ownership. OWNER = 2 to use transformer ownership.
Integer TIES	Is a flag indicating which subsystem three-winding transformers to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem three-winding transformers only. TIES = 2 for subsystem tie three-winding transformers only. TIES = 3 for both interior subsystem and tie three-winding transformers.
Integer FLAG	Is a flag indicating which subsystem three-winding transformers to include (input; 1 by default). FLAG = 1 for only in-service three-winding transformers. FLAG = 2 for all three-winding transformers.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for winding 1 bus order. ENTRY = 2 for transformer name order.
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the transformer quantities desired (input; no default allowed): 'WIND1NUMBER' Winding 1 bus number. 'WIND2NUMBER' Winding 2 bus number. 'WIND3NUMBER' Winding 3 bus number. 'STATUS' Transformer status. 'NMETERNUMBER' Non-metered end bus number.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

'OWNERS'	Number of owners.
'OWN1'	First owner.
'OWN2'	Second owner.
'OWN3'	Third owner.
'OWN4'	Fourth owner.
'CW'	Winding data I/O code.
'CZ'	Impedance data I/O code.
'CM'	Magnetizing admittance I/O code.
'CZ0'	Leakage impedance data I/O code.
'CZG'	Grounding impedance data I/O code.
'CNXCOD'	Connection code.

Integer DIM

Is the number of elements in each column of the array IARRAY (input; no default allowed).

DIM must be as least as large as the value of BRNCHS returned by the GET_3WINDING_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.

Integer IARRAY

Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer BRNCHS

Is the number of elements in each column of IARRAY that are set by this API (output).

BRNCHS is identical to the value of BRNCHS returned by the GET_3WINDING_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.

Integer IERR

Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid OWNER value.
IERR = 4	Invalid TIES value.
IERR = 5	Invalid FLAG value.
IERR = 6	Invalid ENTRY value.
IERR = 7	Invalid NSTR value.
IERR = 8	DIM, and hence the size of IARRAY, is not large enough.
IERR = 9	Invalid STRING value.
IERR = 10	Sequence data not in case (when STRING = 'CZ0', 'CZG' or 'CNXCOD').

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.14.3 aTr3Real

Use this API to return an array of real values for subsystem three-winding transformers.

Python syntax:

```
ierr, rarray = atr3real(sid, owner, ties, flag, entry, string)
```

Fortran syntax:

```
CALL GET_3WINDING_REAL(SID, OWNER, TIES, FLAG, ENTRY, NSTR, STRING,
DIM, RARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer OWNER	Is a flag indicating owner usage if ownership is a subsystem selection criterion (ignored if SID is negative) (input; 1 by default). OWNER = 1 to use bus ownership. OWNER = 2 to use transformer ownership.
Integer TIES	Is a flag indicating which subsystem three-winding transformers to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem three-winding transformers only. TIES = 2 for subsystem tie three-winding transformers only. TIES = 3 for both interior subsystem and tie three-winding transformers.
Integer FLAG	Is a flag indicating which subsystem three-winding transformers to include (input; 1 by default). FLAG = 1 for only in-service three-winding transformers. FLAG = 2 for all three-winding transformers.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for winding 1 bus order. ENTRY = 2 for transformer name order.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the transformer quantities desired (input; no default allowed): ’FRACT1’ First owner fraction. ’FRACT2’ Second owner fraction. ’FRACT3’ Third owner fraction. ’FRACT4’ Fourth owner fraction. ’VMSTAR’ Actual bus voltage magnitude at the star point bus, in pu. ’ANSTAR’ Bus voltage phase angle at the star point bus, in radians.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

For the following, values are returned in MW or Mvar:

'PLOSS'	Active power losses.
'QLOSS'	Reactive power losses.

For the following, values are returned in units determined by the power output option setting:

'O_PLOSS'	Active power losses.
'O_QLOSS'	Reactive power losses.

Integer DIM

Is the number of elements in each column of the array RARRAY (input; no default allowed).

DIM must be as least as large as the value of BRNCHS returned by the GET_3WINDING_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.

Real RARRAY

Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer BRNCHS

Is the number of elements in each column of RARRAY that are set by this API (output).

BRNCHS is identical to the value of BRNCHS returned by the GET_3WINDING_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.

Integer IERR

Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid OWNER value.
IERR = 4	Invalid TIES value.
IERR = 5	Invalid FLAG value.
IERR = 6	Invalid ENTRY value.
IERR = 7	Invalid NSTR value.
IERR = 8	DIM, and hence the size of RARRAY, is not large enough.
IERR = 9	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.14.4 aTr3Cplx

Use this API to return an array of complex values for subsystem three-winding transformers.

Python syntax:

```
ierr, xarray = atr3cplx(sid, owner, ties, flag, entry, string)
```

Fortran syntax:

```
CALL GET_3WINDING_COMPLEX(SID, OWNER, TIES, FLAG, ENTRY, NSTR,
STRING, DIM, XARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer OWNER	Is a flag indicating owner usage if ownership is a subsystem selection criterion (ignored if SID is negative) (input; 1 by default). OWNER = 1 to use bus ownership. OWNER = 2 to use transformer ownership.
Integer TIES	Is a flag indicating which subsystem three-winding transformers to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem three-winding transformers only. TIES = 2 for subsystem tie three-winding transformers only. TIES = 3 for both interior subsystem and tie three-winding transformers.
Integer FLAG	Is a flag indicating which subsystem three-winding transformers to include (input; 1 by default). FLAG = 1 for only in-service three-winding transformers. FLAG = 2 for all three-winding transformers.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for winding 1 bus order. ENTRY = 2 for transformer name order.
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the transformer quantities desired (input; no default allowed): 'RX1-2ACT' Actual measured impedance between the winding 1 and winding 2 buses in pu. 'RX1-2ACTCZ' Actual measured impedance between the winding 1 and winding 2 buses in units governed by CZ. 'RX1-2NOM' Nominal measured impedance between the winding 1 and winding 2 buses in pu.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

’RX1-2NOMCZ’	Nominal measured impedance between the winding 1 and winding 2 buses in units governed by CZ.
’RX2-3ACT’	Actual measured impedance between the winding 2 and winding 3 buses in pu.
’RX2-3ACTCZ’	Actual measured impedance between the winding 2 and winding 3 buses in units governed by CZ.
’RX2-3NOM’	Nominal measured impedance between the winding 2 and winding 3 buses in pu.
’RX2-3NOMCZ’	Nominal measured impedance between the winding 2 and winding 3 buses in units governed by CZ.
’RX3-1ACT’	Actual measured impedance between the winding 3 and winding 1 buses in pu.
’RX3-1ACTCZ’	Actual measured impedance between the winding 3 and winding 1 buses in units governed by CZ.
’RX3-1NOM’	Nominal measured impedance between the winding 3 and winding 1 buses in pu.
’RX3-1NOMCZ’	Nominal measured impedance between the winding 3 and winding 1 buses in units governed by CZ.
’YMAG’	Magnetizing admittance in pu.
’YMAGCM’	Magnetizing admittance in units governed by CM.
’ZG1, ’ZGRND’	Zero sequence winding 1 grounding impedance in pu.
’ZG1CZG’	Zero sequence winding 1 grounding impedance in units governed by CZG.
’ZG2’, ’ZGRND2’	Zero sequence winding 2 grounding impedance in pu.
’ZG2CZG’	Zero sequence winding 2 grounding impedance in units governed by CZG.
’ZG3’, ’ZGRND3’	Zero sequence winding 3 grounding impedance in pu.
’ZG3CZG’	Zero sequence winding 3 grounding impedance in units governed by CZG.
’Z01’	Zero sequence winding 1, or bus 1 to bus 2, leakage impedance in pu.
’Z01CZ0’	Zero sequence winding 1, or bus 1 to bus 2, leakage impedance in units governed by CZ0.
’Z02’	Zero sequence winding 2, or bus 2 to bus 3, leakage impedance in pu.
’Z02CZ0’	Zero sequence winding 2, or bus 2 to bus 3, leakage impedance in units governed by CZ0.
’Z03’	Zero sequence winding 3, or bus 3 to bus 1, leakage impedance in pu.
’Z03CZ0’	Zero sequence winding 3, or bus 3 to bus 1, leakage impedance in units governed by CZ0.
’ZNUTRL’	Zero sequence common neutral grounding impedance in pu.
’ZNUTRLCZG’	Zero sequence common neutral grounding impedance in units governed by CZG.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

For the following, values are returned in MW and Mvar:

'PQLOSS'	Losses.

For the following, values are returned in units determined by the power output option setting:

'O_PQLOSS'	Losses.

Integer DIM

Is the number of elements in each column of the array XARRAY (input; no default allowed).

DIM must be as least as large as the value of BRNCHS returned by the GET_3WINDING_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.

Complex XARRAY

Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer BRNCHS

Is the number of elements in each column of XARRAY that are set by this API (output).

BRNCHS is identical to the value of BRNCHS returned by the GET_3WINDING_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.

Integer IERR

Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid OWNER value.
IERR = 4	Invalid TIES value.
IERR = 5	Invalid FLAG value.
IERR = 6	Invalid ENTRY value.
IERR = 7	Invalid NSTR value.
IERR = 8	DIM, and hence the size of XARRAY, is not large enough.
IERR = 9	Invalid STRING value.
IERR = 10	Sequence data not in case (when STRING = 'ZGRND', 'ZG1', 'ZG1CZG', 'ZG2', 'ZG2CZG', 'ZG3', 'ZG3CZG', 'Z01', 'Z01CZ0', 'Z02', 'Z02CZ0', 'Z03', 'Z03CZ0', 'ZNUTRL', or 'ZNUTRLCZG').

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.14.5 aTr3Char

Use this API to return an array of character values for subsystem three-winding transformers.

Python syntax:

```
ierr, carray = atr3char(sid, owner, ties, flag, entry, string)
```

Fortran syntax:

```
CALL GET_3WINDING_CHAR(SID, OWNER, TIES, FLAG, ENTRY, NSTR, STRING,
DIM, CARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer OWNER	Is a flag indicating owner usage if ownership is a subsystem selection criterion (ignored if SID is negative) (input; 1 by default). OWNER = 1 to use bus ownership. OWNER = 2 to use transformer ownership.
Integer TIES	Is a flag indicating which subsystem three-winding transformers to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem three-winding transformers only. TIES = 2 for subsystem tie three-winding transformers only. TIES = 3 for both interior subsystem and tie three-winding transformers.
Integer FLAG	Is a flag indicating which subsystem three-winding transformers to include (input; 1 by default). FLAG = 1 for only in-service three-winding transformers. FLAG = 2 for all three-winding transformers.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for winding 1 bus order. ENTRY = 2 for transformer name order.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the transformer quantities desired (input; no default allowed): 'ID' Circuit identifier (2 characters). 'WIND1NAME' Winding 1 bus name (12 characters). 'WIND1EXNAME' Winding 1 bus extended bus name (18 characters). 'WIND2NAME' Winding 2 bus name (12 characters).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	'WIND2EXNAME' Winding 2 bus extended bus name (18 characters). 'WIND3NAME' Winding 3 bus name (12 characters). 'WIND3EXNAME' Winding 3 bus extended bus name (18 characters). 'NMETERNAME' Non-metered bus name (12 characters). 'NMETEREXNAME' Non-metered bus extended bus name (18 characters). 'XFRNAME' Transformer name (12 characters). 'VECTORGROUP' Vector group name (12 characters).
Integer DIM	Is the number of elements in each column of the array CARRAY (input; no default allowed). DIM must be as least as large as the value of BRNCHS returned by the GET_3WINDING_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.
Character CARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BRNCHS	Is the number of elements in each column of CARRAY that are set by this API (output). BRNCHS is identical to the value of BRNCHS returned by the GET_3WINDING_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid OWNER value. IERR = 4 Invalid TIES value. IERR = 5 Invalid FLAG value. IERR = 6 Invalid ENTRY value. IERR = 7 Invalid NSTR value. IERR = 8 DIM, and hence the size of CARRAY, is not large enough. IERR = 9 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.14.6 aTr3Types

Use this API to return an array of character values indicating the data types corresponding to a set of specified STRING values that are valid input values to any of the data retrieval routines of the three-winding transformer data family (aTr3Int, aTr3Real, aTr3Cplx and aTr3Char).

Python syntax:

```
ierr, types = atr3types(string)
```

Fortran syntax:

```
CALL GET_3WINDING_TYPES(NSTR,STRING,TYPES,IERR)
```

where:

Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR strings that are valid STRING values for any of the data retrieval routines of the three-winding transformer data family (input; no default allowed):
Character TYPES*1	Is an array of at least NSTR elements, each of which contains one of the following characters indicating the data type of the data associated with the corresponding element of STRING (output): 'I' Integer. 'R' Real. 'X' Complex. 'C' Character. '?' Invalid STRING value.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = n Position of the first element in STRING that contains a value that is not a valid STRING value for any of the data retrieval routines in the three-winding transformer data family (1 ≤ n ≤ NSTR). IERR = 99 Invalid NSTR value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.15 Three-Winding Transformer Winding Data

The three-winding transformer winding data family of subsystem data retrieval APIs returns arrays of winding data for three-winding transformers in a specified subsystem.

One of the input items to each API allows the user to elect either to retrieve data for all subsystem three-winding transformers, for subsystem tie three-winding transformers, or for both subsystem and tie three-winding transformers. Another input item allows the user to elect either to retrieve data for all such transformers, to restrict the array entries to data for all windings of in-service transformers, or to restrict the array entries to data for in-service windings of in-service transformers.

Data for subsystem three-winding transformers is returned from all APIs in the three-winding transformer winding data family in the same order. One of the input items to each API allows the user to elect that transformers be ordered either by transformer name with all windings of each transformer grouped together, or in ascending numerical or alphabetical order by winding bus, according to the bus output option setting in effect at the time the API is referenced.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.15.1 aWndCount

Use this API to return the number of array entries required to accommodate the data to be returned by the remaining members of the three-winding transformer winding data family.

Python syntax:

```
ierr, brnchs = awndcount(sid, owner, ties, flag, entry)
```

Fortran syntax:

```
CALL
GET_3WINDING_WINDINGS_COUNT(SID,OWNER,TIES,FLAG,ENTRY,BRNCHS,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer OWNER	Is a flag indicating owner usage if ownership is a subsystem selection criterion (ignored if SID is negative) (input; 1 by default). OWNER = 1 to use bus ownership. OWNER = 2 to use transformer ownership.
Integer TIES	Is a flag indicating which subsystem three-winding transformers to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem three-winding transformers only. TIES = 2 for subsystem tie three-winding transformers only. TIES = 3 for both interior subsystem and tie three-winding transformers.
Integer FLAG	Is a flag indicating which subsystem three-winding transformers to include (input; 1 by default). FLAG = 1 for only in-service windings of in-service three-winding transformers. FLAG = 2 for all windings of in-service three-winding transformers. FLAG = 3 for all three-winding transformer windings.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for winding 1 bus order. ENTRY = 2 for transformer name order.
Integer BRNCHS	Is the number of array entries required for the subsystem indicated by SID and OWNER that meet the editing criteria indicated by TIES, FLAG and ENTRY (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output):	
	IERR = 0	No error, BRNCHS returned.
	IERR = 1	Working case is empty; BRNCHS returned as 0.
	IERR = 2	Invalid SID value; BRNCHS returned as 0.
	IERR = 3	Invalid OWNER value; BRNCHS returned as 0.
	IERR = 4	Invalid TIES value; BRNCHS returned as 0.
	IERR = 5	Invalid FLAG value; BRNCHS returned as 0.
	IERR = 6	Invalid ENTRY value; BRNCHS returned as 0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.15.2 aWndInt

Use this API to return an array of integer values for subsystem three-winding transformer windings.

Python syntax:

```
ierr, iarray = awndint(sid, owner, ties, flag, entry, string)
```

Fortran syntax:

```
CALL GET_3WINDING_WINDINGS_INTEGER(SID, OWNER, TIES, FLAG, ENTRY,
NSTR, STRING, DIM, IARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer OWNER	Is a flag indicating owner usage if ownership is a subsystem selection criterion (ignored if SID is negative) (input; 1 by default). OWNER = 1 to use bus ownership. OWNER = 2 to use transformer ownership.
Integer TIES	Is a flag indicating which subsystem three-winding transformers to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem three-winding transformers only. TIES = 2 for subsystem tie three-winding transformers only. TIES = 3 for both interior subsystem and tie three-winding transformers.
Integer FLAG	Is a flag indicating which subsystem three-winding transformers to include (input; 1 by default). FLAG = 1 for only in-service windings of in-service three-winding transformers. FLAG = 2 for all windings of in-service three-winding transformers. FLAG = 3 for all three-winding transformer windings.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for winding 1 bus order. ENTRY = 2 for transformer name order.
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)

Is an array of NSTR elements specifying NSTR of the following strings indicating the transformer quantities desired (input; no default allowed):

'WNDBUSNUMBER'
Number of the bus connected to this winding.

'OTHER1NUMBER'
Number of another bus connected to this transformer.

'OTHER2NUMBER'
Number of another bus connected to this transformer.

'WNDNUM' Winding number.

'WIND1NUMBER' Winding 1 bus number.

'WIND2NUMBER' Winding 2 bus number.

'WIND3NUMBER' Winding 3 bus number.

'NMETERNUMBER'
Non-metered end bus number.

'STATUS' Winding status.

'OWNERS' Number of owners.

'OWN1' First owner.

'OWN2' Second owner.

'OWN3' Third owner.

'OWN4' Fourth owner.

'ICONTNUMBER' Controlled bus number.

'TABLE' Transformer impedance adjustment table number.

'CODE' Transformer control mode for automatic adjustments.

'NTPOSN' Number of tap positions.

'CW' Winding data I/O code.

'CZ' Impedance data I/O code.

'CM' Magnetizing admittance I/O code.

'CZ0' Leakage impedance data I/O code.

'CZG' Grounding impedance data I/O code.

'CNXCOD' Connection code.

'TPSTT' Winding 1 ratio limit code: -1 for low, 0 for regulating, 1 for high, -2 for locked.

'ANSTT' Winding 1 angle limit code: -1 for low, 0 for regulating, 1 for high, -2 for locked.

Integer DIM

Is the number of elements in each column of the array IARRAY (input; no default allowed).

DIM must be as least as large as the value of BRNCHS returned by the GET_3WINDING_WINDINGS_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.

Integer IARRAY

Is an array of at least NSTR columns, each of which contains DIM elements (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer BRNCHS	Is the number of elements in each column of IARRAY that are set by this API (output). BRNCHS is identical to the value of BRNCHS returned by the GET_3WINDING_WINDINGS_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.
Integer IERR	Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid OWNER value.
IERR = 4	Invalid TIES value.
IERR = 5	Invalid FLAG value.
IERR = 6	Invalid ENTRY value.
IERR = 7	Invalid NSTR value.
IERR = 8	DIM, and hence the size of IARRAY, is not large enough.
IERR = 9	Invalid STRING value.
IERR = 10	Sequence data not in case (when STRING = ’CZ0’, ’CZG’ or ’CNXCOD’).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.15.3 aWndReal

Use this API to return an array of real values for subsystem three-winding transformer windings.

Python syntax:

```
ierr, rarray = awndreal(sid, owner, ties, flag, entry, string)
```

Fortran syntax:

```
CALL GET_3WINDING_WINDINGS_REAL(SID, OWNER, TIES, FLAG, ENTRY, NSTR,
STRING, DIM, RARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer OWNER	Is a flag indicating owner usage if ownership is a subsystem selection criterion (ignored if SID is negative) (input; 1 by default). OWNER = 1 to use bus ownership. OWNER = 2 to use transformer ownership.
Integer TIES	Is a flag indicating which subsystem three-winding transformers to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem three-winding transformers only. TIES = 2 for subsystem tie three-winding transformers only. TIES = 3 for both interior subsystem and tie three-winding transformers.
Integer FLAG	Is a flag indicating which subsystem three-winding transformers to include (input; 1 by default). FLAG = 1 for only in-service windings of in-service three-winding transformers. FLAG = 2 for all windings of in-service three-winding transformers. FLAG = 3 for all three-winding transformer windings.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for winding 1 bus order. ENTRY = 2 for transformer name order.
Integer NSTR	Is the number of elements in STRING ($1 \leq$ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the transformer quantities desired (input; no default allowed): 'AMPS' Branch current in amps (0.0 if bus base voltage is 0.0). 'PUCUR' Branch current in pu. 'PCTRATE' Percent winding bus current of default rating set.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

'PCTRATEA' Percent winding bus current of rating set A.

'PCTRATEB' Percent winding bus current of rating set B.

'PCTRATEC' Percent winding bus current of rating set C.

'PCTMVARATE' Percent winding bus MVA of default rating set.

'PCTMVARATEA' Percent winding bus MVA of rating set A.

'PCTMVARATEB' Percent winding bus MVA of rating set B.

'PCTMVARATEC' Percent winding bus MVA of rating set C.

'PCTCORPRATE'
Percent winding bus current or MVA loading (according to the transformer percent loading units program option setting) of default rating set.

'PCTCORPRATEA'
Percent winding bus current or MVA loading (according to the transformer percent loading units program option setting) of rating set A.

'PCTCORPRATEB'
Percent winding bus current or MVA loading (according to the transformer percent loading units program option setting) of rating set B.

'PCTCORPRATEC'
Percent winding bus current or MVA loading (according to the transformer percent loading units program option setting) of rating set C.

'MAXPCTRATE' Larger of percent winding/star point bus current of default rating set.

'MAXPCTRATEA' Larger of percent winding/star point bus current of rating set A.

'MAXPCTRATEB' Larger of percent winding/star point bus current of rating set B.

'MAXPCTRATEC' Larger of percent winding/star point bus current of rating set C.

'MXPCTMVARAT' Larger of percent winding/star point bus MVA of default rating set.

'MXPCTMVARATA'
Larger of percent winding/star point bus MVA of rating set A.

'MXPCTMVARATB'
Larger of percent winding/star point bus MVA of rating set B.

'MXPCTMVARATC'
Larger of percent winding/star point bus MVA of rating set C.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

'MXPCTCRPRAT'	Larger of percent winding/star point bus current or MVA loading (according to the transformer percent loading units program option setting) of default rating set.
'MXPCTCRPRATA'	Larger of percent winding/star point bus current or MVA loading (according to the transformer percent loading units program option setting) of rating set A.
'MXPCTCRPRATB'	Larger of percent winding/star point bus current or MVA loading (according to the transformer percent loading units program option setting) of rating set B.
'MXPCTCRPRATC'	Larger of percent winding/star point bus current or MVA loading (according to the transformer percent loading units program option setting) of rating set C.
'FRACT1'	First owner fraction.
'FRACT2'	Second owner fraction.
'FRACT3'	Third owner fraction.
'FRACT4'	Fourth owner fraction.
'RATE'	Rating from default rating set.
'RATEA'	Rating from rating set A.
'RATEB'	Rating from rating set B.
'RATEC'	Rating from rating set C.
'RATIO'	Winding tap ratio in pu.
'RATIOCW'	Winding tap ratio in units governed by CW.
'ANGLE'	Winding phase shift angle in degrees.
'RMAX'	Winding ratio or angle upper limit (in pu if ratio).
'RMAXCW'	Winding ratio or angle upper limit (in units governed by CW if ratio).
'RMIN'	Winding ratio or angle lower limit (in pu if ratio).
'RMINCW'	Winding ratio or angle lower limit (in units governed by CW if ratio).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

'VMAX'	Controlled quantity upper limit (in pu if voltage).
'VMAXKV'	Controlled quantity upper limit (in units governed by CW if voltage).
'VMIN'	Controlled quantity lower limit (in pu if voltage).
'VMINKV'	Controlled quantity lower limit (in units governed by CW if voltage).
'STEP'	Winding ratio or angle step width (in pu if ratio).
'STEPCW'	Winding ratio or angle step width (in units governed by CW if ratio).
'CNXANG'	Winding connection angle in degrees.
'NOMV'	Winding nominal voltage in kv (0.0 for bus base voltage).
'SBASE'	Winding base MVA.

For the following, values are returned in MW, Mvar, or MVA:

'P'	Active power flow at winding bus end.
'Q'	Reactive power flow at winding bus end.
'MVA'	|P + j Q| at winding bus end.
'MAXMVA'	|P + j Q| at winding bus or star point bus end, whichever is larger.
'PLOSS'	Active power losses.
'QLOSS'	Reactive power losses.

For the following, values are returned in units determined by the power output option setting:

'O_P'	Active power flow at winding bus end.
'O_Q'	Reactive power flow at winding bus end.
'O_MVA'	|P + j Q| at winding bus end.
'O_MAXMVA'	|P + j Q| at winding bus or star point bus end, whichever is larger.
'O_PLOSS'	Active power losses.
'O_QLOSS'	Reactive power losses.

Integer DIM — Is the number of elements in each column of the array RARRAY (input; no default allowed).

DIM must be as least as large as the value of BRNCHS returned by the GET_3WINDING_WINDINGS_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.

Real RARRAY — Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer BRNCHS — Is the number of elements in each column of RARRAY that are set by this API (output).

BRNCHS is identical to the value of BRNCHS returned by the GET_3WINDING_WINDINGS_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output):	
	IERR = 0	No error.
	IERR = 1	Working case is empty.
	IERR = 2	Invalid SID value.
	IERR = 3	Invalid OWNER value.
	IERR = 4	Invalid TIES value.
	IERR = 5	Invalid FLAG value.
	IERR = 6	Invalid ENTRY value.
	IERR = 7	Invalid NSTR value.
	IERR = 8	DIM, and hence the size of RARRAY, is not large enough.
	IERR = 9	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.15.4 aWndCplx

Use this API to return an array of complex values for subsystem three-winding transformer windings.

Python syntax:

```
ierr, xarray = awndcplx(sid, owner, ties, flag, entry, string)
```

Fortran syntax:

```
CALL GET_3WINDING_WINDINGS_COMPLEX(SID, OWNER, TIES, FLAG, ENTRY,
NSTR, STRING, DIM, XARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer OWNER	Is a flag indicating owner usage if ownership is a subsystem selection criterion (ignored if SID is negative) (input; 1 by default). OWNER = 1 to use bus ownership. OWNER = 2 to use transformer ownership.
Integer TIES	Is a flag indicating which subsystem three-winding transformers to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem three-winding transformers only. TIES = 2 for subsystem tie three-winding transformers only. TIES = 3 for both interior subsystem and tie three-winding transformers.
Integer FLAG	Is a flag indicating which subsystem three-winding transformers to include (input; 1 by default). FLAG = 1 for only in-service windings of in-service three-winding transformers. FLAG = 2 for all windings of in-service three-winding transformers. FLAG = 3 for all three-winding transformer windings.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for winding bus order. ENTRY = 2 for transformer name order.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the transformer quantities desired (input; no default allowed): 'RXACT' Actual transformer impedance in pu. 'RXNOM' Nominal transformer impedance in pu. 'COMPRX' Load drop compensation impedance in pu. 'RXZERO' Zero sequence winding leakage impedance in pu. 'ZGRND' Zero sequence winding grounding impedance in pu.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

For the following, values are returned in MW and Mvar:

'PQ'	P + j Q flow at winding bus end.
'PQLOSS'	Losses.

For the following, values are returned in units determined by the power output option setting:

'O_PQ'	P + j Q flow at winding bus end.
'O_PQLOSS'	Losses.

Integer DIM — Is the number of elements in each column of the array XARRAY (input; no default allowed).

DIM must be as least as large as the value of BRNCHS returned by the GET_3WINDING_WINDINGS_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.

Complex XARRAY — Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer BRNCHS — Is the number of elements in each column of XARRAY that are set by this API (output).

BRNCHS is identical to the value of BRNCHS returned by the GET_3WINDING_WINDINGS_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.

Integer IERR — Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid OWNER value.
IERR = 4	Invalid TIES value.
IERR = 5	Invalid FLAG value.
IERR = 6	Invalid ENTRY value.
IERR = 7	Invalid NSTR value.
IERR = 8	DIM, and hence the size of XARRAY, is not large enough.
IERR = 9	Invalid STRING value.
IERR = 10	Sequence data not in case (when STRING = 'RXZERO' or 'ZGRND').

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.15.5 aWndChar

Use this API to return an array of character values for subsystem three-winding transformer windings.

Python syntax:

```
ierr, carray = awndchar(sid, owner, ties, flag, entry, string)
```

Fortran syntax:

```
CALL GET_3WINDING_WINDINGS_CHAR(SID, OWNER, TIES, FLAG, ENTRY, NSTR,
STRING, DIM, CARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer OWNER	Is a flag indicating owner usage if ownership is a subsystem selection criterion (ignored if SID is negative) (input; 1 by default). OWNER = 1 to use bus ownership. OWNER = 2 to use transformer ownership.
Integer TIES	Is a flag indicating which subsystem three-winding transformers to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem three-winding transformers only. TIES = 2 for subsystem tie three-winding transformers only. TIES = 3 for both interior subsystem and tie three-winding transformers.
Integer FLAG	Is a flag indicating which subsystem three-winding transformers to include (input; 1 by default). FLAG = 1 for only in-service windings of in-service three-winding transformers. FLAG = 2 for all windings of in-service three-winding transformers. FLAG = 3 for all three-winding transformer windings.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for winding bus order. ENTRY = 2 for transformer name order.
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the transformer quantities desired (input; no default allowed): 'ID' Circuit identifier (2 characters). 'WNDBUSNAME' Name of the bus connected to this winding (12 characters). 'WNDBUSEXNAME' Extended bus name of the bus connected to this winding (18 characters). 'OTHER1NAME' Name of another bus connected to this winding (12 characters). 'OTHER1EXNAME' Extended bus name of another bus connected to this winding (18 characters). 'OTHER2NAME' Name of another bus connected to this winding (12 characters). 'OTHER2EXNAME' Extended bus name of another bus connected to this winding (18 characters). 'WIND1NAME' Winding 1 bus name (12 characters). 'WIND1EXNAME' Winding 1 bus extended bus name (18 characters). 'WIND2NAME' Winding 2 bus name (12 characters). 'WIND2EXNAME' Winding 2 bus extended bus name (18 characters). 'WIND3NAME' Winding 3 bus name (12 characters). 'WIND3EXNAME' Winding 3 bus extended bus name (18 characters). 'NMETERNAME' Non-metered bus name (12 characters). 'NMETEREXNAME' Non-metered bus extended bus name (18 characters). 'XFRNAME' Transformer name (12 characters).
Integer DIM	Is the number of elements in each column of the array CARRAY (input; no default allowed). DIM must be as least as large as the value of BRNCHS returned by the GET_3WINDING_WINDINGS_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.
Character CARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BRNCHS	Is the number of elements in each column of CARRAY that are set by this API (output). BRNCHS is identical to the value of BRNCHS returned by the GET_3WINDING_WINDINGS_COUNT API when it is called with the same values specified for SID, OWNER, TIES, FLAG and ENTRY.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output):	
	IERR = 0	No error.
	IERR = 1	Working case is empty.
	IERR = 2	Invalid SID value.
	IERR = 3	Invalid OWNER value.
	IERR = 4	Invalid TIES value.
	IERR = 5	Invalid FLAG value.
	IERR = 6	Invalid ENTRY value.
	IERR = 7	Invalid NSTR value.
	IERR = 8	DIM, and hence the size of CARRAY, is not large enough.
	IERR = 9	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.15.6 aWndTypes

Use this API to return an array of character values indicating the data types corresponding to a set of specified STRING values that are valid input values to any of the data retrieval routines of the three-winding transformer winding data family (aWndInt, aWndReal, aWndCplx and aWndChar).

Python syntax:

```
ierr, types = awndtypes(string)
```

Fortran syntax:

```
CALL GET_3WINDING_WINDINGS_TYPES(NSTR,STRING,TYPES,IERR)
```

where:

Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR strings that are valid STRING values for any of the data retrieval routines of the three-winding transformer winding data family (input; no default allowed):
Character TYPES*1	Is an array of at least NSTR elements, each of which contains one of the following characters indicating the data type of the data associated with the corresponding element of STRING (output): 'I' Integer. 'R' Real. 'X' Complex. 'C' Character. '?' Invalid STRING value.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = n Position of the first element in STRING that contains a value that is not a valid STRING value for any of the data retrieval routines in the three-winding transformer winding data family (1 ≤ n ≤ NSTR). IERR = 99 Invalid NSTR value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.16 Two-Terminal dc Line Data

The two-terminal dc line data family of subsystem data retrieval APIs returns arrays of data for two-terminal dc lines in a specified subsystem.

One of the input items to each API allows the user to elect to retrieve data either for all subsystem two-terminal dc lines, for subsystem two-terminal dc lines that are tie branches, or for both subsystem and tie two-terminal dc lines. Another input item allows the user to elect to retrieve data for all such two-terminal dc lines, or to restrict the array entries to data for in-service two-terminal dc lines.

Data for subsystem two-terminal dc lines is returned from all APIs in the two-terminal dc line data family in dc line name alphabetical order.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.16.1 a2TrmDcCount

Use this API to return the number of array entries required to accommodate the data to be returned by the remaining members of the two-terminal dc line data family.

Python syntax:

```
ierr, brnchs = a2trmdccount(sid, ties, flag)
```

Fortran syntax:

```
CALL GET_TWO_TERM_DC_LINE_COUNT(SID, TIES, FLAG, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem two-terminal dc lines to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem two-terminal dc lines only. TIES = 2 for subsystem tie two-terminal dc lines only. TIES = 3 for both interior subsystem two-terminal dc lines and tie two-terminal dc lines.
Integer FLAG	Is a flag indicating which subsystem two-terminal dc lines to include (input; 1 by default). FLAG = 1 for only in-service two-terminal dc lines. FLAG = 2 for all two-terminal dc lines.
Integer BRNCHS	Is the number of array entries required for the subsystem indicated by SID that meet the editing criteria indicated by TIES and FLAG (output).
Integer IERR	Is the error code (output): IERR = 0 No error, BRNCHS returned. IERR = 1 Working case is empty; BRNCHS returned as 0. IERR = 2 Invalid SID value; BRNCHS returned as 0. IERR = 3 Invalid TIES value; BRNCHS returned as 0. IERR = 4 Invalid FLAG value; BRNCHS returned as 0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.16.2 a2TrmDcInt

Use this API to return an array of integer values for subsystem two-terminal dc lines.

Python syntax:

```
ierr, iarray = a2trmdcint(sid, ties, flag, string)
```

Fortran syntax:

```
CALL GET_TWO_TERM_DC_LINE_INTEGER(SID, TIES, FLAG, NSTR, STRING, DIM,
IARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem two-terminal dc lines to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem two-terminal dc lines only. TIES = 2 for subsystem tie two-terminal dc lines only. TIES = 3 for both interior subsystem two-terminal dc lines and tie two-terminal dc lines.
Integer FLAG	Is a flag indicating which subsystem two-terminal dc lines to include (input; 1 by default). FLAG = 1 for only in-service two-terminal dc lines. FLAG = 2 for all two-terminal dc lines.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the two-terminal dc line quantities desired (input; no default allowed): 'FROMNUMBER' Rectifier bus number. 'TONUMBER' Inverter bus number. 'METERNUMBER' Metered end bus number. 'NMETERNUMBER' Non-metered end bus number. 'IFRNUMBER' Rectifier firing angle measuring bus number. 'IFINUMBER' Inverter extinction angle measuring bus number. 'MDC' Control mode. 'NBR' Rectifier number of bridges in series. 'NBI' Inverter number of bridges in series. 'ANSTT_R' Rectifier angle limit code: -1 (at or below low limit), 0 (in range), or 1 (at or above high limit).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

'ANSTT_I'	Inverter angle limit code: -1 (at or below low limit), 0 (in range), or 1 (at or above high limit).
'TPSTT_R'	Rectifier tap ratio limit code: -1 (at or below low limit), 0 (in range), 1 (at or above high limit), or -2 (locked).
'TPSTT_I'	Inverter tap ratio limit code: -1 (at or below low limit), 0 (in range), 1 (at or above high limit), or -2 (locked).
'CCCITMX'	Iteration limit for capacitor commutated converter solution.

Integer DIM

Is the number of elements in each column of the array IARRAY (input; no default allowed).

DIM must be as least as large as the value of BRNCHS returned by the GET_TWO_TERM_DC_LINE_COUNT API when it is called with the same values specified for SID, TIES and FLAG.

Integer IARRAY

Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer BRNCHS

Is the number of elements in each column of IARRAY that are set by this API (output).

BRNCHS is identical to the value of BRNCHS returned by the GET_TWO_TERM_DC_LINE_COUNT API when it is called with the same values specified for SID, TIES and FLAG.

Integer IERR

Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid TIES value.
IERR = 4	Invalid FLAG value.
IERR = 5	Invalid NSTR value.
IERR = 6	DIM, and hence the size of IARRAY, is not large enough.
IERR = 7	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.16.3 a2TrmDcReal

Use this API to return an array of real values for subsystem two-terminal dc lines.

Python syntax:

```
ierr, rarray = a2trmdcreal(sid, ties, flag, string)
```

Fortran syntax:

```
CALL GET_TWO_TERM_DC_LINE_REAL(SID, TIES, FLAG, NSTR, STRING, DIM,
RARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem two-terminal dc lines to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem two-terminal dc lines only. TIES = 2 for subsystem tie two-terminal dc lines only. TIES = 3 for both interior subsystem two-terminal dc lines and tie two-terminal dc lines.
Integer FLAG	Is a flag indicating which subsystem two-terminal dc lines to include (input; 1 by default). FLAG = 1 for only in-service two-terminal dc lines. FLAG = 2 for all two-terminal dc lines.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the two-terminal dc line quantities desired (input; no default allowed): 'DCCUR' dc line current in amps. 'RDC' dc line resistance in ohms. 'SETVAL' Desired current (in amps) or power (in MW) demand. 'VSCHD' Scheduled compounded dc voltage in kV. 'VCMOD' Mode switch dc voltage in kV. 'RCOMP' Compounding resistance in ohms. 'DELTI' Margin in pu of desired dc power or current. 'DCVMIN' Minimum compounded dc voltage in kV. 'ACAMPS_R' ac current in amps at the rectifier. 'ACAMPS_I' ac current in amps at the inverter.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

’PUCUR_R’	ac current in pu at the rectifier.
’PUCUR_I’	ac current in pu at the inverter.
’KVDC_R’	Rectifier dc voltage in kV.
’KVDC_I’	Inverter dc voltage in kV.
’ANGLE_R’	Rectifier converter angle in degrees.
’ANGLE_I’	Inverter converter angle in degrees.
’ANGMIN_R’	Rectifier minimum steady-state converter angle in degrees.
’ANGMIN_I’	Inverter minimum steady-state converter angle in degrees.
’ANGMAX_R’	Rectifier nominal maximum converter angle in degrees.
’ANGMAX_I’	Inverter nominal maximum converter angle in degrees.
’RC_R’	Rectifier commutating transformer resistance per bridge in ohms.
’RC_I’	Inverter commutating transformer resistance per bridge in ohms.
’XC_R’	Rectifier commutating transformer reactance per bridge in ohms.
’XC_I’	Inverter commutating transformer reactance per bridge in ohms.
’EBASE_R’	Rectifier primary base ac voltage in kV.
’EBASE_I’	Inverter primary base ac voltage in kV.
’TR_R’	Rectifier transformer ratio in pu.
’TR_I’	Inverter transformer ratio in pu.
’TAP_R’	Rectifier tap setting in pu.
’TAP_I’	Inverter tap setting in pu.
’TPMX_R’	Rectifier maximum tap setting in pu.
’TPMX_I’	Inverter maximum tap setting in pu.
’TPMN_R’	Rectifier minimum tap setting in pu.
’TPMN_I’	Inverter minimum tap setting in pu.
’STP_R’	Rectifier tap step in pu.
’STP_I’	Inverter tap step in pu.
’XCAP_R’	Rectifier commutating capacitor reactance magnitude per bridge in pu.
’XCAP_I’	Inverter commutating capacitor reactance magnitude per bridge in pu.
’CCCACC’	Acceleration factor for capacitor commutated converter solution.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

For the following, values are returned in MW, Mvar, or MVA:

'PAC_R'	Apparent active power flow into the dc line at the rectifier.
'PAC_I'	Apparent active power flow into the dc line at the inverter.
'QAC_R'	Apparent reactive power flow into the dc line at the rectifier.
'QAC_I'	Apparent reactive power flow into the dc line at the inverter.
'MVA_R'	Apparent ac power flow into the dc line at the rectifier.
'MVA_I'	Apparent ac power flow into the dc line at the inverter.
'PLOSS'	Active power losses.
'QLOSS'	Reactive power losses.

For the following, values are returned in units determined by the power output option setting:

'O_PAC_R'	Apparent active power flow into the dc line at the rectifier.
'O_PAC_I'	Apparent active power flow into the dc line at the inverter.
'O_QAC_R'	Apparent reactive power flow into the dc line at the rectifier.
'O_QAC_I'	Apparent reactive power flow into the dc line at the inverter.
'O_MVA_R'	Apparent ac power flow into the dc line at the rectifier.
'O_MVA_I'	Apparent ac power flow into the dc line at the inverter.
'O_PLOSS'	Active power losses.
'O_QLOSS'	Reactive power losses.

Integer DIM

Is the number of elements in each column of the array RARRAY (input; no default allowed).

DIM must be as least as large as the value of BRNCHS returned by the GET_TWO_TERM_DC_LINE_COUNT API when it is called with the same values specified for SID, TIES and FLAG.

Real RARRAY

Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer BRNCHS

Is the number of elements in each column of RARRAY that are set by this API (output).

BRNCHS is identical to the value of BRNCHS returned by the GET_TWO_TERM_DC_LINE_COUNT API when it is called with the same values specified for SID, TIES and FLAG.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output):	
	IERR = 0	No error.
	IERR = 1	Working case is empty.
	IERR = 2	Invalid SID value.
	IERR = 3	Invalid TIES value.
	IERR = 4	Invalid FLAG value.
	IERR = 5	Invalid NSTR value.
	IERR = 6	DIM, and hence the size of RARRAY, is not large enough.
	IERR = 7	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.16.4 a2TrmDcCplx

Use this API to return an array of complex values for subsystem two-terminal dc lines.

Python syntax:

```
ierr, xarray = a2trmdccplx(sid, ties, flag, string)
```

Fortran syntax:

```
CALL GET_TWO_TERM_DC_LINE_COMPLEX(SID, TIES, FLAG, NSTR, STRING, DIM,
XARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem two-terminal dc lines to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem two-terminal dc lines only. TIES = 2 for subsystem tie two-terminal dc lines only. TIES = 3 for both interior subsystem two-terminal dc lines and tie two-terminal dc lines.
Integer FLAG	Is a flag indicating which subsystem two-terminal dc lines to include (input; 1 by default). FLAG = 1 for only in-service two-terminal dc lines. FLAG = 2 for all two-terminal dc lines.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the two-terminal dc line quantities desired (input; no default allowed): For the following, values are returned in MW and Mvar: 'PQAC_R' Apparent ac power flow into the dc line at the rectifier. 'PQAC_I' Apparent ac power flow into the dc line at the inverter. 'PQLOSS' ac power losses. For the following, values are returned in units determined by the power output option setting: 'O_PQAC_R' Apparent ac power flow into the dc line at the rectifier. 'O_PQAC_I' Apparent ac power flow into the dc line at the inverter. 'O_PQLOSS' ac power losses.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer DIM	Is the number of elements in each column of the array XARRAY (input; no default allowed). DIM must be as least as large as the value of BRNCHS returned by the GET_TWO_TERM_DC_LINE_COUNT API when it is called with the same values specified for SID, TIES and FLAG.
Complex XARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BRNCHS	Is the number of elements in each column of XARRAY that are set by this API (output). BRNCHS is identical to the value of BRNCHS returned by the GET_TWO_TERM_DC_LINE_COUNT API when it is called with the same values specified for SID, TIES and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid TIES value. IERR = 4 Invalid FLAG value. IERR = 5 Invalid NSTR value. IERR = 6 DIM, and hence the size of XARRAY, is not large enough. IERR = 7 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.16.5 a2TrmDcChar

Use this API to return an array of character values for subsystem two-terminal dc lines.

Python syntax:

```
ierr, carray = a2trmdcchar(sid, ties, flag, string)
```

Fortran syntax:

```
CALL GET_TWO_TERM_DC_LINE_CHAR(SID, TIES, FLAG, NSTR, STRING, DIM,
CARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem two-terminal dc lines to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem two-terminal dc lines only. TIES = 2 for subsystem tie two-terminal dc lines only. TIES = 3 for both interior subsystem two-terminal dc lines and tie two-terminal dc lines.
Integer FLAG	Is a flag indicating which subsystem two-terminal dc lines to include (input; 1 by default). FLAG = 1 for only in-service two-terminal dc lines. FLAG = 2 for all two-terminal dc lines.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the two-terminal dc line quantities desired (input; no default allowed): 'DCNAME' dc line name. 'FROMNAME' Rectifier bus name (12 characters). 'FROMEXNAME' Rectifier bus extended bus name (18 characters). 'TONAME' Inverter bus name (12 characters). 'TOEXNAME' Inverter bus extended bus name (18 characters). 'METERNAME' Metered bus name (12 characters). 'METEREXNAME' Metered bus extended bus name (18 characters). 'NMETERNAME' Non-metered bus name (12 characters). 'NMETEREXNAME' Non-metered bus extended bus name (18 characters).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	'IFRNAME'	Rectifier firing angle measuring bus name (12 characters).
	'IFREXNAME'	Rectifier firing angle measuring bus extended bus name (18 characters).
	'IFINAME'	Inverter firing angle measuring bus name (12 characters).
	'IFIEXNAME'	Inverter firing angle measuring bus extended bus name (18 characters).

Integer DIM

Is the number of elements in each column of the array CARRAY (input; no default allowed).

DIM must be as least as large as the value of BRNCHS returned by the GET_TWO_TERM_DC_LINE_COUNT API when it is called with the same values specified for SID, TIES and FLAG.

Character CARRAY

Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer BRNCHS

Is the number of elements in each column of CARRAY that are set by this API (output).

BRNCHS is identical to the value of BRNCHS returned by the GET_TWO_TERM_DC_LINE_COUNT API when it is called with the same values specified for SID, TIES and FLAG.

Integer IERR

Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid TIES value.
IERR = 4	Invalid FLAG value.
IERR = 5	Invalid NSTR value.
IERR = 6	DIM, and hence the size of CARRAY, is not large enough.
IERR = 7	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.16.6 a2TrmDcTypes

Use this API to return an array of character values indicating the data types corresponding to a set of specified STRING values that are valid input values to any of the data retrieval routines of the two-terminal dc line data family (a2TrmDcInt, a2TrmDcReal, a2TrmDcCplx and a2TrmDcChar).

Python syntax:

```
ierr, types = a2trmdctypes(string)
```

Fortran syntax:

```
CALL GET_TWO_TERM_DC_LINE_TYPES(NSTR,STRING,TYPES,IERR)
```

where:

Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR strings that are valid STRING values for any of the data retrieval routines of the two-terminal dc line data family (input; no default allowed):
Character TYPES*1	Is an array of at least NSTR elements, each of which contains one of the following characters indicating the data type of the data associated with the corresponding element of STRING (output): 'I' Integer. 'R' Real. 'X' Complex. 'C' Character. '?' Invalid STRING value.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = n Position of the first element in STRING that contains a value that is not a valid STRING value for any of the data retrieval routines in the two-terminal dc line data family (1 ≤ n ≤ NSTR). IERR = 99 Invalid NSTR value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.17 Two-Terminal dc Line Converter Data

The two-terminal dc line converter data family of subsystem data retrieval APIs returns arrays of data for two-terminal dc line converters in a specified subsystem.

One of the input items to each API allows the user to elect to retrieve data either for all subsystem two-terminal dc line converters, for subsystem two-terminal dc line converters of two-terminal dc lines that are tie branches, or for both subsystem and tie two-terminal dc line converters. Another input item allows the user to elect to retrieve data for all such two-terminal dc line converters, or to restrict the array entries to data for in-service two-terminal dc lines.

Data for subsystem converter buses is returned from all APIs in the two-terminal dc line converter data family in the same order. One of the input items to each API allows the user to elect that converter buses be ordered either in dc line name order with all converter buses of each two-terminal dc line grouped together, or in numerical or alphabetical order by converter bus, according to the bus output option setting in effect at the time the API is referenced.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.17.1 a2TrmDcConvCount

Use this API to return the number of array entries required to accommodate the data to be returned by the remaining members of the two-terminal dc line converter data family.

Python syntax:

```
ierr, buses = a2trmdcconvcount(sid, ties, flag)
```

Fortran syntax:

```
CALL GET_TWO_TERM_DC_CONV_COUNT(SID, TIES, FLAG, BUSES, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem converter buses of two-terminal dc lines to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for each end of interior subsystem two-terminal dc lines only. TIES = 2 for the subsystem bus end of tie two-terminal dc lines only. TIES = 3 for the non-subsystem bus end of tie two-terminal dc lines only. TIES = 4 for each end of tie two-terminal dc lines only. TIES = 5 for each end of interior subsystem two-terminal dc lines and the subsystem bus end of tie two-terminal dc lines. TIES = 6 for each end of interior subsystem two-terminal dc lines and tie two-terminal dc lines.
Integer FLAG	Is a flag indicating which subsystem two-terminal dc line converters to include (input; 1 by default). FLAG = 1 for only in-service two-terminal dc lines. FLAG = 2 for all two-terminal dc lines.
Integer BUSES	Is the number of array entries required for the subsystem indicated by SID that meet the editing criteria indicated by TIES and FLAG (output).
Integer IERR	Is the error code (output): IERR = 0 No error, BUSES returned. IERR = 1 Working case is empty; BUSES returned as 0. IERR = 2 Invalid SID value; BUSES returned as 0. IERR = 3 Invalid TIES value; BUSES returned as 0. IERR = 4 Invalid FLAG value; BUSES returned as 0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.17.2 a2TrmDcConvInt

Use this API to return an array of integer values for subsystem two-terminal dc line converters.

Python syntax:

```
ierr, iarray = a2trmdcconvint(sid, ties, flag, entry, string)
```

Fortran syntax:

```
CALL GET_TWO_TERM_DC_CONV_INTEGER(SID, TIES, FLAG, ENTRY, NSTR,
STRING, DIM, IARRAY, BUSES, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem converter buses of two-terminal dc lines to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for each end of interior subsystem two-terminal dc lines only. TIES = 2 for the subsystem bus end of tie two-terminal dc lines only. TIES = 3 for the non-subsystem bus end of tie two-terminal dc lines only. TIES = 4 for each end of tie two-terminal dc lines only. TIES = 5 for each end of interior subsystem two-terminal dc lines and the subsystem bus end of tie two-terminal dc lines. TIES = 6 for each end of interior subsystem two-terminal dc lines and tie two-terminal dc lines.
Integer FLAG	Is a flag indicating which subsystem two-terminal dc line converters to include (input; 1 by default). FLAG = 1 for only in-service two-terminal dc lines. FLAG = 2 for all two-terminal dc lines.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for converter bus order. ENTRY = 2 for dc line name order.
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the two-terminal dc line converter quantities desired (input; no default allowed): 'FROMNUMBER' From bus number. 'TONUMBER' To bus number. 'METERNUMBER' Metered end bus number. 'NMETERNUMBER' Non-metered end bus number. 'CONVERTERTYP' 1 if from bus is the rectifier, 2 if the inverter. 'MDC' Control mode. 'ANSTT' From bus converter angle limit code: -1 (at or below low limit), 0 (in range), or 1 (at or above high limit). 'TPSTT' From bus converter tap ratio limit code: -1 (at or below low limit), 0 (in range), 1 (at or above high limit) or -2 (locked).
Integer DIM	Is the number of elements in each column of the array IARRAY (input; no default allowed). DIM must be as least as large as the value of BUSES returned by the GET_TWO_TERM_DC_CONV_COUNT API when it is called with the same values specified for SID, TIES and FLAG.
Integer IARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BUSES	Is the number of elements in each column of IARRAY that are set by this API (output). BUSES is identical to the value of BUSES returned by the GET_TWO_TERM_DC_CONV_COUNT API when it is called with the same values specified for SID, TIES and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid TIES value. IERR = 4 Invalid FLAG value. IERR = 5 Invalid ENTRY value. IERR = 6 Invalid NSTR value. IERR = 7 DIM, and hence the size of IARRAY, is not large enough. IERR = 8 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.17.3 a2TrmDcConvReal

Use this API to return an array of real values for subsystem two-terminal dc line converters.

Python syntax:

```
ierr, rarray = a2trmdcconvreal(sid, ties, flag, entry, string)
```

Fortran syntax:

```
CALL GET_TWO_TERM_DC_CONV_REAL(SID, TIES, FLAG, ENTRY, NSTR, STRING,
DIM, RARRAY, BUSES, IERR)
```

where:

Integer SID — Defines the bus subsystem to be used (input; -1 by default).

SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case.

SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.

Integer TIES — Is a flag indicating which subsystem converter buses of two-terminal dc lines to include (ignored if SID is negative) (input; 1 by default).

- TIES = 1 — for each end of interior subsystem two-terminal dc lines only.
- TIES = 2 — for the subsystem bus end of tie two-terminal dc lines only.
- TIES = 3 — for the non-subsystem bus end of tie two-terminal dc lines only.
- TIES = 4 — for each end of tie two-terminal dc lines only.
- TIES = 5 — for each end of interior subsystem two-terminal dc lines and the subsystem bus end of tie two-terminal dc lines.
- TIES = 6 — for each end of interior subsystem two-terminal dc lines and tie two-terminal dc lines.

Integer FLAG — Is a flag indicating which subsystem two-terminal dc line converters to include (input; 1 by default).

- FLAG = 1 — for only in-service two-terminal dc lines.
- FLAG = 2 — for all two-terminal dc lines.

Integer ENTRY — Is a flag indicating output organization (input; 1 by default).

- ENTRY = 1 — for converter bus order.
- ENTRY = 2 — for dc line name order.

Integer NSTR — Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the two-terminal dc line converter quantities desired (input; no default allowed): 'DCCUR' dc line current in amps. 'ACAMPS' ac current in amps at the from bus. 'PUCUR' ac current in pu at the from bus. 'KVDC' From bus dc voltage in kV. 'ANGLE' From bus converter angle in degrees. 'TAP' From bus tap setting in pu. For the following, values are returned in MW, Mvar, or MVA: 'PAC' Apparent active power flow into the dc line at the from bus. 'QAC' Apparent reactive power flow into the dc line at the from bus. 'MVA' Apparent ac power flow into the dc line at the from bus. 'PLOSS' Active power losses. 'QLOSS' Reactive power losses. For the following, values are returned in units determined by the power output option setting: 'O_PAC' Apparent active power flow into the dc line at the from bus. 'O_QAC' Apparent reactive power flow into the dc line at the from bus. 'O_MVA' Apparent ac power flow into the dc line at the from bus. 'O_PLOSS' Active power losses. 'O_QLOSS' Reactive power losses.
Integer DIM	Is the number of elements in each column of the array RARRAY (input; no default allowed). DIM must be as least as large as the value of BUSES returned by the GET_TWO_TERM_DC_CONV_COUNT API when it is called with the same values specified for SID, TIES and FLAG.
Real RARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BUSES	Is the number of elements in each column of RARRAY that are set by this API (output). BUSES is identical to the value of BUSES returned by the GET_TWO_TERM_DC_CONV_COUNT API when it is called with the same values specified for SID, TIES and FLAG.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid TIES value.
IERR = 4	Invalid FLAG value.
IERR = 5	Invalid ENTRY value.
IERR = 6	Invalid NSTR value.
IERR = 7	DIM, and hence the size of RARRAY, is not large enough.
IERR = 8	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.17.4 a2TrmDcConvCplx

Use this API to return an array of complex values for subsystem two-terminal dc line converters.

Python syntax:

```
ierr, xarray = a2trmdcconvcplx(sid, ties, flag, entry, string)
```

Fortran syntax:

```
CALL GET_TWO_TERM_DC_CONV_COMPLEX(SID, TIES, FLAG, ENTRY, NSTR,
STRING, DIM, XARRAY, BUSES, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem converter buses of two-terminal dc lines to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for each end of interior subsystem two-terminal dc lines only. TIES = 2 for the subsystem bus end of tie two-terminal dc lines only. TIES = 3 for the non-subsystem bus end of tie two-terminal dc lines only. TIES = 4 for each end of tie two-terminal dc lines only. TIES = 5 for each end of interior subsystem two-terminal dc lines and the subsystem bus end of tie two-terminal dc lines. TIES = 6 for each end of interior subsystem two-terminal dc lines and tie two-terminal dc lines.
Integer FLAG	Is a flag indicating which subsystem two-terminal dc line converters to include (input; 1 by default). FLAG = 1 for only in-service two-terminal dc lines. FLAG = 2 for all two-terminal dc lines.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for converter bus order. ENTRY = 2 for dc line name order.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the two-terminal dc line converter quantities desired (input; no default allowed): For the following, values are returned in MW and Mvar: 'PQAC' Apparent ac power flow into the dc line at the from bus. 'PQLOSS' ac power losses.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

For the following, values are returned in units determined by the power output option setting:

'O_PQAC'	Apparent ac power flow into the dc line at the from bus.
'O_PQLOSS'	ac power losses.

Integer DIM

Is the number of elements in each column of the array XARRAY (input; no default allowed).

DIM must be as least as large as the value of BUSES returned by the GET_TWO_TERM_DC_CONV_COUNT API when it is called with the same values specified for SID, TIES and FLAG.

Complex XARRAY

Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer BUSES

Is the number of elements in each column of XARRAY that are set by this API (output).

BUSES is identical to the value of BUSES returned by the GET_TWO_TERM_DC_CONV_COUNT API when it is called with the same values specified for SID, TIES and FLAG.

Integer IERR

Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid TIES value.
IERR = 4	Invalid FLAG value.
IERR = 5	Invalid ENTRY value.
IERR = 6	Invalid NSTR value.
IERR = 7	DIM, and hence the size of XARRAY, is not large enough.
IERR = 8	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.17.5 a2TrmDcConvChar

Use this API to return an array of character values for subsystem two-terminal dc line converters.

Python syntax:

```
ierr, carray = a2trmdcconvchar(sid, ties, flag, entry, string)
```

Fortran syntax:

```
CALL GET_TWO_TERM_DC_CONV_CHAR(SID, TIES, FLAG, ENTRY, NSTR, STRING,
DIM, CARRAY, BUSES, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem converter buses of two-terminal dc lines to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for each end of interior subsystem two-terminal dc lines only. TIES = 2 for the subsystem bus end of tie two-terminal dc lines only. TIES = 3 for the non-subsystem bus end of tie two-terminal dc lines only. TIES = 4 for each end of tie two-terminal dc lines only. TIES = 5 for each end of interior subsystem two-terminal dc lines and the subsystem bus end of tie two-terminal dc lines. TIES = 6 for each end of interior subsystem two-terminal dc lines and tie two-terminal dc lines.
Integer FLAG	Is a flag indicating which subsystem two-terminal dc line converters to include (input; 1 by default). FLAG = 1 for only in-service two-terminal dc lines. FLAG = 2 for all two-terminal dc lines.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for converter bus order. ENTRY = 2 for dc line name order.
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the two-terminal dc line converter quantities desired (input; no default allowed): 'DCNAME' dc line name. 'FROMNAME' From bus name (12 characters). 'FROMEXNAME' From bus extended bus name (18 characters). 'TONAME' To bus name (12 characters). 'TOEXNAME' To bus extended bus name (18 characters). 'METERNAME' Metered bus name (12 characters). 'METEREXNAME' Metered bus extended bus name (18 characters). 'NMETERNAME' Non-metered bus name (12 characters). 'NMETEREXNAME' Non-metered bus extended bus name (18 characters).
Integer DIM	Is the number of elements in each column of the array CARRAY (input; no default allowed). DIM must be as least as large as the value of BUSES returned by the GET_TWO_TERM_DC_CONV_COUNT API when it is called with the same values specified for SID, TIES and FLAG.
Character CARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BUSES	Is the number of elements in each column of CARRAY that are set by this API (output). BUSES is identical to the value of BUSES returned by the GET_TWO_TERM_DC_CONV_COUNT API when it is called with the same values specified for SID, TIES and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid TIES value. IERR = 4 Invalid FLAG value. IERR = 5 Invalid ENTRY value. IERR = 6 Invalid NSTR value. IERR = 7 DIM, and hence the size of CARRAY, is not large enough. IERR = 8 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.17.6 a2TrmDcConvTypes

Use this API to return an array of character values indicating the data types corresponding to a set of specified STRING values that are valid input values to any of the data retrieval routines of the two-terminal dc line converter data family (a2TrmDcConvInt, a2TrmDcConvReal, a2TrmDcConvCplx and a2TrmDcConvChar).

Python syntax:

```
ierr, types = a2trmdcconvtypes(string)
```

Fortran syntax:

```
CALL GET_TWO_TERM_DC_CONV_TYPES(NSTR,STRING,TYPES,IERR)
```

where:

Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR strings that are valid STRING values for any of the data retrieval routines of the two-terminal dc line converter data family (input; no default allowed):
Character TYPES*1	Is an array of at least NSTR elements, each of which contains one of the following characters indicating the data type of the data associated with the corresponding element of STRING (output):
	'I' Integer.
	'R' Real.
	'X' Complex.
	'C' Character.
	'?' Invalid STRING value.
Integer IERR	Is the error code (output):
	IERR = 0 No error.
	IERR = n Position of the first element in STRING that contains a value that is not a valid STRING value for any of the data retrieval routines in the two-terminal dc line converter data family (1 ≤ n ≤ NSTR).
	IERR = 99 Invalid NSTR value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.18 Multi-Terminal dc Line Data

The multi-terminal dc line data family of subsystem data retrieval APIs returns arrays of data for multi-terminal dc lines in a specified subsystem.

One of the input items to each API allows the user to elect to retrieve data for either all subsystem multi-terminal dc lines, for subsystem multi-terminal dc lines that are tie branches, or for both subsystem and tie multi-terminal dc lines. Another input item allows the user to elect to retrieve data for all such multi-terminal dc lines, or to restrict the array entries to data for in-service multi-terminal dc lines.

Data for subsystem multi-terminal dc lines is returned from all APIs in the multi-terminal dc line data family in dc line name alphabetical order.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.18.1 aMultiTrmDcCount

Use this API to return the number of array entries required to accommodate the data to be returned by the remaining members of the multi-terminal dc line data family.

Python syntax:

```
ierr, brnchs = amultitrmdccount(sid, ties, flag)
```

Fortran syntax:

```
CALL GET_MULTI_TERM_DC_LINE_COUNT(SID, TIES, FLAG, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem multi-terminal dc lines to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem multi-terminal dc lines only. TIES = 2 for subsystem tie multi-terminal dc lines only. TIES = 3 for both interior subsystem multi-terminal dc lines and tie multi-terminal dc lines.
Integer FLAG	Is a flag indicating which subsystem multi-terminal dc lines to include (input; 1 by default). FLAG = 1 for only in-service multi-terminal dc lines. FLAG = 2 for all multi-terminal dc lines.
Integer BRNCHS	Is the number of array entries required for the subsystem indicated by SID that meet the editing criteria indicated by TIES and FLAG (output).
Integer IERR	Is the error code (output): IERR = 0 No error, BRNCHS returned. IERR = 1 Working case is empty; BRNCHS returned as 0. IERR = 2 Invalid SID value; BRNCHS returned as 0. IERR = 3 Invalid TIES value; BRNCHS returned as 0. IERR = 4 Invalid FLAG value; BRNCHS returned as 0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.18.2 aMultiTrmDcInt

Use this API to return an array of integer values for subsystem multi-terminal dc lines.

Python syntax:

```
ierr, iarray = amultitrmdcint(sid, ties, flag, string)
```

Fortran syntax:

```
CALL GET_MULTI_TERM_DC_LINE_INTEGER(SID, TIES, FLAG, NSTR, STRING,
DIM, IARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem multi-terminal dc lines to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem multi-terminal dc lines only. TIES = 2 for subsystem tie multi-terminal dc lines only. TIES = 3 for both interior subsystem multi-terminal dc lines and tie multi-terminal dc lines.
Integer FLAG	Is a flag indicating which subsystem multi-terminal dc lines to include (input; 1 by default). FLAG = 1 for only in-service multi-terminal dc lines. FLAG = 2 for all multi-terminal dc lines.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the multi-terminal dc line quantities desired (input; no default allowed): 'CONVERTERS' Number of converters. 'DCBUSES' Number of dc buses. 'DCLINKS' Number of dc links. 'VCNPOSNUMBER' Positive pole voltage controlling converter bus number. 'VCNNEGNUMBER' Negative pole voltage controlling converter bus number. 'MDC' Control mode.

Integer DIM	Is the number of elements in each column of the array IARRAY (input; no default allowed). DIM must be as least as large as the value of BRNCHS returned by the GET_MULTI_TERM_DC_LINE_COUNT API when it is called with the same values specified for SID, TIES and FLAG.
Integer IARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BRNCHS	Is the number of elements in each column of IARRAY that are set by this API (output). BRNCHS is identical to the value of BRNCHS returned by the GET_MULTI_TERM_DC_LINE_COUNT API when it is called with the same values specified for SID, TIES and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid TIES value. IERR = 4 Invalid FLAG value. IERR = 5 Invalid NSTR value. IERR = 6 DIM, and hence the size of IARRAY, is not large enough. IERR = 7 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.18.3 aMultiTrmDcReal

Use this API to return an array of real values for subsystem multi-terminal dc lines.

Python syntax:

```
ierr, rarray = amultitrmdcreal(sid, ties, flag, string)
```

Fortran syntax:

```
CALL GET_MULTI_TERM_DC_LINE_REAL(SID, TIES, FLAG, NSTR, STRING, DIM,
RARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem multi-terminal dc lines to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem multi-terminal dc lines only. TIES = 2 for subsystem tie multi-terminal dc lines only. TIES = 3 for both interior subsystem multi-terminal dc lines and tie multi-terminal dc lines.
Integer FLAG	Is a flag indicating which subsystem multi-terminal dc lines to include (input; 1 by default). FLAG = 1 for only in-service multi-terminal dc lines. FLAG = 2 for all multi-terminal dc lines.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the multi-terminal dc line quantities desired (input; no default allowed): 'VCMOD' Mode switch dc voltage in kV. For the following, values are returned in MW or Mvar: 'PLOSS' Active power losses. 'QLOSS' Reactive power losses. For the following, values are returned in units determined by the power output option setting: 'O_PLOSS' Active power losses. 'O_QLOSS' Reactive power losses.
Integer DIM	Is the number of elements in each column of the array RARRAY (input; no default allowed). DIM must be as least as large as the value of BRNCHS returned by the GET_MULTI_TERM_DC_LINE_COUNT API when it is called with the same values specified for SID, TIES and FLAG.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real RARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BRNCHS	Is the number of elements in each column of RARRAY that are set by this API (output). BRNCHS is identical to the value of BRNCHS returned by the GET_MULTI_TERM_DC_LINE_COUNT API when it is called with the same values specified for SID, TIES and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid TIES value. IERR = 4 Invalid FLAG value. IERR = 5 Invalid NSTR value. IERR = 6 DIM, and hence the size of RARRAY, is not large enough. IERR = 7 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.18.4 aMultiTrmDcCplx

Use this API to return an array of complex values for subsystem multi-terminal dc lines.

Python syntax:

```
ierr, xarray = amultitrmdccplx(sid, ties, flag, string)
```

Fortran syntax:

```
CALL GET_MULTI_TERM_DC_LINE_COMPLEX(SID, TIES, FLAG, NSTR, STRING,
DIM, XARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem multi-terminal dc lines to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem multi-terminal dc lines only. TIES = 2 for subsystem tie multi-terminal dc lines only. TIES = 3 for both interior subsystem multi-terminal dc lines and tie multi-terminal dc lines.
Integer FLAG	Is a flag indicating which subsystem multi-terminal dc lines to include (input; 1 by default). FLAG = 1 for only in-service multi-terminal dc lines. FLAG = 2 for all multi-terminal dc lines.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the multi-terminal dc line quantities desired (input; no default allowed): For the following, values are returned in MW and Mvar: 'PQLOSS' ac power losses. For the following, values are returned in units determined by the power output option setting: 'O_PQLOSS' ac power losses.
Integer DIM	Is the number of elements in each column of the array XARRAY (input; no default allowed). DIM must be as least as large as the value of BRNCHS returned by the GET_MULTI_TERM_DC_LINE_COUNT API when it is called with the same values specified for SID, TIES and FLAG.
Complex XARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer BRNCHS	Is the number of elements in each column of XARRAY that are set by this API (output). BRNCHS is identical to the value of BRNCHS returned by the GET_MULTI_TERM_DC_LINE_COUNT API when it is called with the same values specified for SID, TIES and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid TIES value. IERR = 4 Invalid FLAG value. IERR = 5 Invalid NSTR value. IERR = 6 DIM, and hence the size of XARRAY, is not large enough. IERR = 7 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.18.5 aMultiTrmDcChar

Use this API to return an array of character values for subsystem multi-terminal dc lines.

Python syntax:

```
ierr, carray = amultitrmdcchar(sid, ties, flag, string)
```

Fortran syntax:

```
CALL GET_MULTI_TERM_DC_LINE_CHAR(SID, TIES, FLAG, NSTR, STRING, DIM,
CARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem multi-terminal dc lines to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem multi-terminal dc lines only. TIES = 2 for subsystem tie multi-terminal dc lines only. TIES = 3 for both interior subsystem multi-terminal dc lines and tie multi-terminal dc lines.
Integer FLAG	Is a flag indicating which subsystem multi-terminal dc lines to include (input; 1 by default). FLAG = 1 for only in-service multi-terminal dc lines. FLAG = 2 for all multi-terminal dc lines.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the multi-terminal dc line quantities desired (input; no default allowed): 'DCNAME' dc line name. 'VCNPOSNAME' Positive pole voltage controlling converter bus name (12 characters). 'VCNPOSEXNAME' Positive pole voltage controlling converter bus extended bus name (18 characters). 'VCNNEGNAME' Negative pole voltage controlling converter bus name (12 characters). 'VCNNEGEXNAME' Negative pole voltage controlling converter bus extended bus name (18 characters).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer DIM	Is the number of elements in each column of the array CARRAY (input; no default allowed). DIM must be as least as large as the value of BRNCHS returned by the GET_MULTI_TERM_DC_LINE_COUNT API when it is called with the same values specified for SID, TIES and FLAG.
Character CARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BRNCHS	Is the number of elements in each column of CARRAY that are set by this API (output). BRNCHS is identical to the value of BRNCHS returned by the GET_MULTI_TERM_DC_LINE_COUNT API when it is called with the same values specified for SID, TIES and FLAG.
Integer IERR	Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid TIES value.
IERR = 4	Invalid FLAG value.
IERR = 5	Invalid NSTR value.
IERR = 6	DIM, and hence the size of CARRAY, is not large enough.
IERR = 7	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.18.6 aMultiTrmDcTypes

Use this API to return an array of character values indicating the data types corresponding to a set of specified STRING values that are valid input values to any of the data retrieval routines of the multi-terminal dc line data family (aMultiTrmDcInt, aMultiTrmDcReal, aMultiTrmDcCplx and aMultiTrmDcChar).

Python syntax:

```
ierr, types = amultitrmdctypes(string)
```

Fortran syntax:

```
CALL GET_MULTI_TERM_DC_LINE_TYPES(NSTR,STRING,TYPES,IERR)
```

where:

Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR strings that are valid STRING values for any of the data retrieval routines of the multi-terminal dc line data family (input; no default allowed):
Character TYPES*1	Is an array of at least NSTR elements, each of which contains one of the following characters indicating the data type of the data associated with the corresponding element of STRING (output):
	'I' Integer.
	'R' Real.
	'X' Complex.
	'C' Character.
	'?' Invalid STRING value.
Integer IERR	Is the error code (output):
	IERR = 0 No error.
	IERR = n Position of the first element in STRING that contains a value that is not a valid STRING value for any of the data retrieval routines in the multi-terminal dc line data family (1 ≤ n ≤ NSTR).
	IERR = 99 Invalid NSTR value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.19 Multi-Terminal dc Line Converter Data

The multi-terminal dc line converter data family of subsystem data retrieval APIs returns arrays of data for multi-terminal dc line converters in a specified subsystem.

One of the input items to each API allows the user to elect to retrieve data either for all subsystem multi-terminal dc line converters, for subsystem multi-terminal dc line converters of multi-terminal dc lines that are tie branches, or for both subsystem and tie multi-terminal dc line converters. Another input item allows the user to elect to retrieve data for all such multi-terminal dc line converters, or to restrict the array entries to data for in-service multi-terminal dc lines.

Data for subsystem converter buses is returned from all APIs in the multi-terminal dc line converter data family in the same order. One of the input items to each API allows the user to elect that converter buses be ordered either in dc line name order with all converter buses of each multi-terminal dc line grouped together, or in numerical or alphabetical order by converter bus, according to the bus output option setting in effect at the time the API is referenced.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.19.1 aMultiTrmDcConvCount

Use this API to return the number of array entries required to accommodate the data to be returned by the remaining members of the multi-terminal dc line converter data family.

Python syntax:

```
ierr, buses = amultitrmdcconvcount(sid, ties, flag)
```

Fortran syntax:

```
CALL GET_MULTI_TERM_DC_CONV_COUNT(SID, TIES, FLAG, BUSES, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem converter buses of multi-terminal dc lines to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for each end of interior subsystem multi-terminal dc lines only. TIES = 2 for the subsystem bus ends of tie multi-terminal dc lines only. TIES = 3 for the non-subsystem bus ends of tie multi-terminal dc lines only. TIES = 4 for each end of tie multi-terminal dc lines only. TIES = 5 for each end of interior subsystem multi-terminal dc lines and the subsystem bus ends of tie multi-terminal dc lines. TIES = 6 for each end of interior subsystem multi-terminal dc lines and tie multi-terminal dc lines.
Integer FLAG	Is a flag indicating which subsystem multi-terminal dc line converters to include (input; 1 by default). FLAG = 1 for only in-service multi-terminal dc lines. FLAG = 2 for all multi-terminal dc lines.
Integer BUSES	Is the number of array entries required for the subsystem indicated by SID that meet the editing criteria indicated by TIES and FLAG (output).
Integer IERR	Is the error code (output): IERR = 0 No error, BUSES returned. IERR = 1 Working case is empty; BUSES returned as 0. IERR = 2 Invalid SID value; BUSES returned as 0. IERR = 3 Invalid TIES value; BUSES returned as 0. IERR = 4 Invalid FLAG value; BUSES returned as 0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.19.2 aMultiTrmDcConvInt

Use this API to return an array of integer values for subsystem multi-terminal dc line converters.

Python syntax:

```
ierr, iarray = amultitrmdcconvint(sid, ties, flag, entry, string)
```

Fortran syntax:

```
CALL GET_MULTI_TERM_DC_CONV_INTEGER(SID, TIES, FLAG, ENTRY, NSTR,
STRING, DIM, IARRAY, BUSES, IERR)
```

where:

Integer SID — Defines the bus subsystem to be used (input; -1 by default).

SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case.

SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.

Integer TIES — Is a flag indicating which subsystem converter buses of multi-terminal dc lines to include (ignored if SID is negative) (input; 1 by default).

- TIES = 1 — for each end of interior subsystem multi-terminal dc lines only.
- TIES = 2 — for the subsystem bus ends of tie multi-terminal dc lines only.
- TIES = 3 — for the non-subsystem bus ends of tie multi-terminal dc lines only.
- TIES = 4 — for each end of tie multi-terminal dc lines only.
- TIES = 5 — for each end of interior subsystem multi-terminal dc lines and the subsystem bus ends of tie multi-terminal dc lines.
- TIES = 6 — for each end of interior subsystem multi-terminal dc lines and tie multi-terminal dc lines.

Integer FLAG — Is a flag indicating which subsystem multi-terminal dc line converters to include (input; 1 by default).

- FLAG = 1 — for only in-service multi-terminal dc lines.
- FLAG = 2 — for all multi-terminal dc lines.

Integer ENTRY — Is a flag indicating output organization (input; 1 by default).

- ENTRY = 1 — for converter bus order.
- ENTRY = 2 — for dc line name order.

Integer NSTR — Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)

Is an array of NSTR elements specifying NSTR of the following strings indicating the multi-terminal dc line converter quantities desired (input; no default allowed):

'NUMBER'	Converter bus number.
'CONVERTERTYP'	1 for a voltage controlling converter, 3 for a rectifier, 4 for an inverter.
'POLE'	1 for a positive pole converter, -1 for a negative pole converter.
'DCBUSNUMBER'	Number of the dc bus connected to this converter bus.
'DCBUS2NUMBER'	Number of the optional second dc bus connected to this converter bus.
'NBRIDGES'	Number of bridges in series.
'ANSTT'	Converter angle limit code: -1 (at or below low limit), 0 (in range), or 1 (at or above high limit).
'TPSTT'	Converter tap ratio limit code: -1 (at or below low limit), 0 (in range), 1 (at or above high limit), or -2 (locked).

Integer DIM

Is the number of elements in each column of the array IARRAY (input; no default allowed).

DIM must be as least as large as the value of BUSES returned by the GET_MULTI_TERM_DC_CONV_COUNT API when it is called with the same values specified for SID, TIES and FLAG.

Integer IARRAY

Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer BUSES

Is the number of elements in each column of IARRAY that are set by this API (output).

BUSES is identical to the value of BUSES returned by the GET_MULTI_TERM_DC_CONV_COUNT API when it is called with the same values specified for SID, TIES and FLAG.

Integer IERR

Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid TIES value.
IERR = 4	Invalid FLAG value.
IERR = 5	Invalid ENTRY value.
IERR = 6	Invalid NSTR value.
IERR = 7	DIM, and hence the size of IARRAY, is not large enough.
IERR = 8	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.19.3 aMultiTrmDcConvReal

Use this API to return an array of real values for subsystem multi-terminal dc line converters.

Python syntax:

```
ierr, rarray = amultitrmdcconvreal(sid, ties, flag, entry, string)
```

Fortran syntax:

```
CALL GET_MULTI_TERM_DC_CONV_REAL(SID, TIES, FLAG, ENTRY, NSTR,
STRING, DIM, RARRAY, BUSES, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem converter buses of multi-terminal dc lines to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for each end of interior subsystem multi-terminal dc lines only. TIES = 2 for the subsystem bus ends of tie multi-terminal dc lines only. TIES = 3 for the non-subsystem bus ends of tie multi-terminal dc lines only. TIES = 4 for each end of tie multi-terminal dc lines only. TIES = 5 for each end of interior subsystem multi-terminal dc lines and the subsystem bus ends of tie multi-terminal dc lines. TIES = 6 for each end of interior subsystem multi-terminal dc lines and tie multi-terminal dc lines.
Integer FLAG	Is a flag indicating which subsystem multi-terminal dc line converters to include (input; 1 by default). FLAG = 1 for only in-service multi-terminal dc lines. FLAG = 2 for all multi-terminal dc lines.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for converter bus order. ENTRY = 2 for dc line name order.
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)

Is an array of NSTR elements specifying NSTR of the following strings indicating the multi-terminal dc line converter quantities desired (input; no default allowed):

'DCCUR'	Converter dc current in amps.
'SETVAL'	Voltage (kV), current (amps) or power (MW) setpoint.
'ACAMPS'	ac current in amps at the converter bus.
'PUCUR'	ac current in pu at the converter bus.
'KVDC'	Converter bus dc voltage in kV.
'ANGLE'	Converter angle in degrees.
'ANGMIN'	Minimum steady-state converter angle, in degrees.
'ANGMAX'	Nominal maximum converter angle, in degrees.
'TAP'	Converter bus tap setting in pu.
'TPMX'	Maximum tap setting in pu.
'TPMN'	Minimum tap setting in pu.
'STP'	Tap step in pu.
'TR'	Transformer ratio in pu.
'RC'	Commutating transformer resistance per bridge in ohms.
'XC'	Commutating transformer reactance per bridge in ohms.
'EBASE'	Primary base ac voltage in kV.
'DELTI'	Rectifier margin in pu of desired dc power or current.
'DCPF'	Converter participation factor.

For the following, values are returned in MW, Mvar, or MVA:

'PAC'	Apparent active power flow into the dc line at the converter bus.
'QAC'	Apparent reactive power flow into the dc line at the converter bus.
'MVA'	Apparent ac power flow into the dc line at the converter bus.
'PLOSS'	Total dc line active power losses.
'QLOSS'	Total dc line reactive power losses.

For the following, values are returned in units determined by the power output option setting:

'O_PAC'	Apparent active power flow into the dc line at the converter bus.
'O_QAC'	Apparent reactive power flow into the dc line at the converter bus.
'O_MVA'	Apparent ac power flow into the dc line at the converter bus.
'O_PLOSS'	Total dc line active power losses.
'O_QLOSS'	Total dc line reactive power losses.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer DIM	Is the number of elements in each column of the array RARRAY (input; no default allowed). DIM must be as least as large as the value of BUSES returned by the GET_MULTI_TERM_DC_CONV_COUNT API when it is called with the same values specified for SID, TIES and FLAG.
Real RARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BUSES	Is the number of elements in each column of RARRAY that are set by this API (output). BUSES is identical to the value of BUSES returned by the GET_MULTI_TERM_DC_CONV_COUNT API when it is called with the same values specified for SID, TIES and FLAG.
Integer IERR	Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid TIES value.
IERR = 4	Invalid FLAG value.
IERR = 5	Invalid ENTRY value.
IERR = 6	Invalid NSTR value.
IERR = 7	DIM, and hence the size of RARRAY, is not large enough.
IERR = 8	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.19.4 aMultiTrmDcConvCplx

Use this API to return an array of complex values for subsystem multi-terminal dc line converters.

Python syntax:

```
ierr, xarray = amultitrmdcconvcplx(sid, ties, flag, entry, string)
```

Fortran syntax:

```
CALL GET_MULTI_TERM_DC_CONV_COMPLEX(SID, TIES, FLAG, ENTRY, NSTR,
STRING, DIM, XARRAY, BUSES, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem converter buses of multi-terminal dc lines to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for each end of interior subsystem multi-terminal dc lines only. TIES = 2 for the subsystem bus ends of tie multi-terminal dc lines only. TIES = 3 for the non-subsystem bus ends of tie multi-terminal dc lines only. TIES = 4 for each end of tie multi-terminal dc lines only. TIES = 5 for each end of interior subsystem multi-terminal dc lines and the subsystem bus ends of tie multi-terminal dc lines. TIES = 6 for each end of interior subsystem multi-terminal dc lines and tie multi-terminal dc lines.
Integer FLAG	Is a flag indicating which subsystem multi-terminal dc line converters to include (input; 1 by default). FLAG = 1 for only in-service multi-terminal dc lines. FLAG = 2 for all multi-terminal dc lines.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for converter bus order. ENTRY = 2 for dc line name order.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the multi-terminal dc line converter quantities desired (input; no default allowed). For the following, values are returned in MW and Mvar: 'PQAC' Apparent ac power flow into the dc line at the from bus. 'PQLOSS' Total dc line ac power losses.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

For the following, values are returned in units determined by the power output option setting:

'O_PQAC'	Apparent ac power flow into the dc line at the from bus.
'O_PQLOSS'	Total dc line ac power losses.

Integer DIM

Is the number of elements in each column of the array XARRAY (input; no default allowed).

DIM must be as least as large as the value of BUSES returned by the GET_MULTI_TERM_DC_CONV_COUNT API when it is called with the same values specified for SID, TIES and FLAG.

Complex XARRAY

Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer BUSES

Is the number of elements in each column of XARRAY that are set by this API (output).

BUSES is identical to the value of BUSES returned by the GET_MULTI_TERM_DC_CONV_COUNT API when it is called with the same values specified for SID, TIES and FLAG.

Integer IERR

Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid TIES value.
IERR = 4	Invalid FLAG value.
IERR = 5	Invalid ENTRY value.
IERR = 6	Invalid NSTR value.
IERR = 7	DIM, and hence the size of XARRAY, is not large enough.
IERR = 8	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.19.5 aMultiTrmDcConvChar

Use this API to return an array of character values for subsystem multi-terminal dc line converters.

Python syntax:

```
ierr, carray = amultitrmdcconvchar(sid, ties, flag, entry, string)
```

Fortran syntax:

```
CALL GET_MULTI_TERM_DC_CONV_CHAR(SID, TIES, FLAG, ENTRY, NSTR,
STRING, DIM, CARRAY, BUSES, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem converter buses of multi-terminal dc lines to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for each end of interior subsystem multi-terminal dc lines only. TIES = 2 for the subsystem bus ends of tie multi-terminal dc lines only. TIES = 3 for the non-subsystem bus ends of tie multi-terminal dc lines only. TIES = 4 for each end of tie multi-terminal dc lines only. TIES = 5 for each end of interior subsystem multi-terminal dc lines and the subsystem bus ends of tie multi-terminal dc lines. TIES = 6 for each end of interior subsystem multi-terminal dc lines and tie multi-terminal dc lines.
Integer FLAG	Is a flag indicating which subsystem multi-terminal dc line converters to include (input; 1 by default). FLAG = 1 for only in-service multi-terminal dc lines. FLAG = 2 for all multi-terminal dc lines.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for converter bus order. ENTRY = 2 for dc line name order.
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the multi-terminal dc line converter quantities desired (input; no default allowed): 'DCNAME' dc line name. 'NAME' Converter bus name (12 characters). 'EXNAME' Converter bus extended bus name (18 characters). 'DCBUSNAME' Name of the dc bus connected to this converter bus (12 characters). 'DCBUS2NAME' Name of the optional second dc bus connected to this converter bus (12 characters).
Integer DIM	Is the number of elements in each column of the array CARRAY (input; no default allowed). DIM must be as least as large as the value of BUSES returned by the GET_MULTI_TERM_DC_CONV_COUNT API when it is called with the same values specified for SID, TIES and FLAG.
Character CARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BUSES	Is the number of elements in each column of CARRAY that are set by this API (output). BUSES is identical to the value of BUSES returned by the GET_MULTI_TERM_DC_CONV_COUNT API when it is called with the same values specified for SID, TIES and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid TIES value. IERR = 4 Invalid FLAG value. IERR = 5 Invalid ENTRY value. IERR = 6 Invalid NSTR value. IERR = 7 DIM, and hence the size of CARRAY, is not large enough. IERR = 8 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.19.6 aMultiTrmDcConvTypes

Use this API to return an array of character values indicating the data types corresponding to a set of specified STRING values that are valid input values to any of the data retrieval routines of the two-terminal dc line converter data family (aMultiTrmDcConvInt, aMultiTrmDcConvReal, aMultiTrmDcConvCplx and aMultiTrmDcConvChar).

Python syntax:

```
ierr, types = amultitrmdcconvtypes(string)
```

Fortran syntax:

```
CALL GET_MULTI_TERM_DC_CONV_TYPES(NSTR,STRING,TYPES,IERR)
```

where:

Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR strings that are valid STRING values for any of the data retrieval routines of the two-terminal dc line converter data family (input; no default allowed):
Character TYPES*1	Is an array of at least NSTR elements, each of which contains one of the following characters indicating the data type of the data associated with the corresponding element of STRING (output): 'I' Integer. 'R' Real. 'X' Complex. 'C' Character. '?' Invalid STRING value.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = n Position of the first element in STRING that contains a value that is not a valid STRING value for any of the data retrieval routines in the two-terminal dc line converter data family (1 ≤ n ≤ NSTR). IERR = 99 Invalid NSTR value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.20 VSC dc Line Data

The VSC dc line data family of subsystem data retrieval APIs returns arrays of data for VSC dc lines in a specified subsystem.

One of the input items to each API allows the user to elect to retrieve data either for all subsystem VSC dc lines, for subsystem VSC dc lines that are tie branches, or for both subsystem and tie VSC dc lines. Another input item allows the user to elect to retrieve data for all such VSC dc lines, to restrict the array entries to data for in-service VSC dc lines with one or both converters in-service, or to restrict the array entries to data for in-service VSC dc lines with both converters in-service.

Data for subsystem VSC dc lines is returned from all APIs in the VSC dc line data family in dc line name alphabetical order.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.20.1 aVscDcCount

Use this API to return the number of array entries required to accommodate the data to be returned by the remaining members of the VSC dc line data family.

Python syntax:

```
ierr, brnchs = avscdccount(sid, ties, flag)
```

Fortran syntax:

```
CALL GET_VSC_DC_LINE_COUNT(SID, TIES, FLAG, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem VSC dc lines to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem VSC dc lines only. TIES = 2 for subsystem tie VSC dc lines only. TIES = 3 for both interior subsystem VSC dc lines and tie VSC dc lines.
Integer FLAG	Is a flag indicating which subsystem VSC dc lines to include (input; 1 by default). FLAG = 1 for only in-service VSC dc lines with one or both converters in-service. FLAG = 2 for only in-service VSC dc lines with both converters in-service. FLAG = 3 for all VSC dc lines.
Integer BRNCHS	Is the number of array entries required for the subsystem indicated by SID that meet the editing criteria indicated by TIES and FLAG (output).
Integer IERR	Is the error code (output): IERR = 0 No error, BRNCHS returned. IERR = 1 Working case is empty; BRNCHS returned as 0. IERR = 2 Invalid SID value; BRNCHS returned as 0. IERR = 3 Invalid TIES value; BRNCHS returned as 0. IERR = 4 Invalid FLAG value; BRNCHS returned as 0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.20.2 aVscDcInt

Use this API to return an array of integer values for subsystem VSC dc lines.

Python syntax:

```
ierr, iarray = avscdcint(sid, ties, flag, string)
```

Fortran syntax:

```
CALL GET_VSC_DC_LINE_INTEGER(SID, TIES, FLAG, NSTR, STRING, DIM,
IARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem VSC dc lines to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem VSC dc lines only. TIES = 2 for subsystem tie VSC dc lines only. TIES = 3 for both interior subsystem VSC dc lines and tie VSC dc lines.
Integer FLAG	Is a flag indicating which subsystem VSC dc lines to include (input; 1 by default). FLAG = 1 for only in-service VSC dc lines with one or both converters in-service. FLAG = 2 for only in-service VSC dc lines with both converters in-service. FLAG = 3 for all VSC dc lines.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the VSC dc line quantities desired (input; no default allowed): 'FROMNUMBER' Converter 1 bus number. 'TONUMBER' Converter 2 bus number. 'MDC' Control mode. 'OWNERS' Number of owners. 'OWN1' First owner. 'OWN2' Second owner. 'OWN3' Third owner. 'OWN4' Fourth owner.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer DIM	Is the number of elements in each column of the array IARRAY (input; no default allowed). DIM must be as least as large as the value of BRNCHS returned by the GET_VSC_DC_LINE_COUNT API when it is called with the same values specified for SID, TIES and FLAG.
Integer IARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BRNCHS	Is the number of elements in each column of IARRAY that are set by this API (output). BRNCHS is identical to the value of BRNCHS returned by the GET_VSC_DC_LINE_COUNT API when it is called with the same values specified for SID, TIES and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid TIES value. IERR = 4 Invalid FLAG value. IERR = 5 Invalid NSTR value. IERR = 6 DIM, and hence the size of IARRAY, is not large enough. IERR = 7 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.20.3 aVscDcReal

Use this API to return an array of real values for subsystem VSC dc lines.

Python syntax:

```
ierr, rarray = avscdcreal(sid, ties, flag, string)
```

Fortran syntax:

```
CALL GET_VSC_DC_LINE_REAL(SID, TIES, FLAG, NSTR, STRING, DIM, RARRAY,
BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem VSC dc lines to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem VSC dc lines only. TIES = 2 for subsystem tie VSC dc lines only. TIES = 3 for both interior subsystem VSC dc lines and tie VSC dc lines.
Integer FLAG	Is a flag indicating which subsystem VSC dc lines to include (input; 1 by default). FLAG = 1 for only in-service VSC dc lines with one or both converters in-service. FLAG = 2 for only in-service VSC dc lines with both converters in-service. FLAG = 3 for all VSC dc lines.
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the VSC dc line quantities desired (input; no default allowed): 'DCCUR' dc line current in amps. 'RDC' dc line resistance in ohms. 'FRACT1' First owner fraction. 'FRACT2' Second owner fraction. 'FRACT3' Third owner fraction. 'FRACT4' Fourth owner fraction. For the following, values are returned in MW or Mvar: 'PLOSS' Active power losses. 'QLOSS' Reactive power losses.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

For the following, values are returned in units determined by the power output option setting:

'O_PLOSS'	Active power losses.
'O_QLOSS'	Reactive power losses.

Integer DIM

Is the number of elements in each column of the array RARRAY (input; no default allowed).

DIM must be as least as large as the value of BRNCHS returned by the GET_VSC_DC_LINE_COUNT API when it is called with the same values specified for SID, TIES and FLAG.

Real RARRAY

Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer BRNCHS

Is the number of elements in each column of RARRAY that are set by this API (output).

BRNCHS is identical to the value of BRNCHS returned by the GET_VSC_DC_LINE_COUNT API when it is called with the same values specified for SID, TIES and FLAG.

Integer IERR

Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid TIES value.
IERR = 4	Invalid FLAG value.
IERR = 5	Invalid NSTR value.
IERR = 6	DIM, and hence the size of RARRAY, is not large enough.
IERR = 7	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.20.4 aVscDcCplx

Use this API to return an array of complex values for subsystem VSC dc lines.

Python syntax:

```
ierr, xarray = avscdccplx(sid, ties, flag, string)
```

Fortran syntax:

```
CALL GET_VSC_DC_LINE_COMPLEX(SID, TIES, FLAG, NSTR, STRING, DIM,
XARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem VSC dc lines to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem VSC dc lines only. TIES = 2 for subsystem tie VSC dc lines only. TIES = 3 for both interior subsystem VSC dc lines and tie VSC dc lines.
Integer FLAG	Is a flag indicating which subsystem VSC dc lines to include (input; 1 by default). FLAG = 1 for only in-service VSC dc lines with one or both converters in-service. FLAG = 2 for only in-service VSC dc lines with both converters in-service. FLAG = 3 for all VSC dc lines.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the VSC dc line quantities desired (input; no default allowed): For the following, values are returned in MW and Mvar: 'PQLOSS' ac power losses. For the following, values are returned in units determined by the power output option setting: 'O_PQLOSS' ac power losses.
Integer DIM	Is the number of elements in each column of the array XARRAY (input; no default allowed). DIM must be as least as large as the value of BRNCHS returned by the GET_VSC_DC_LINE_COUNT API when it is called with the same values specified for SID, TIES and FLAG.
Complex XARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer BRNCHS	Is the number of elements in each column of XARRAY that are set by this API (output). BRNCHS is identical to the value of BRNCHS returned by the GET_VSC_DC_LINE_COUNT API when it is called with the same values specified for SID, TIES and FLAG.
Integer IERR	Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid TIES value.
IERR = 4	Invalid FLAG value.
IERR = 5	Invalid NSTR value.
IERR = 6	DIM, and hence the size of XARRAY, is not large enough.
IERR = 7	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.20.5 aVscDcChar

Use this API to return an array of character values for subsystem VSC dc lines.

Python syntax:

```
ierr, carray = avscdcchar(sid, ties, flag, string)
```

Fortran syntax:

```
CALL GET_VSC_DC_LINE_CHAR(SID, TIES, FLAG, NSTR, STRING, DIM, CARRAY,
BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem VSC dc lines to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem VSC dc lines only. TIES = 2 for subsystem tie VSC dc lines only. TIES = 3 for both interior subsystem VSC dc lines and tie VSC dc lines.
Integer FLAG	Is a flag indicating which subsystem VSC dc lines to include (input; 1 by default). FLAG = 1 for only in-service VSC dc lines with one or both converters in-service. FLAG = 2 for only in-service VSC dc lines with both converters in-service. FLAG = 3 for all VSC dc lines.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the VSC dc line quantities desired (input; no default allowed): 'VSCNAME' dc line name (12 characters). 'FROMNAME' Converter 1 bus name (12 characters). 'FROMEXNAME' Converter 1 bus extended bus name (18 characters). 'TONAME' Converter 2 bus name (12 characters). 'TOEXNAME' Converter 2 bus extended bus name (18 characters).
Integer DIM	Is the number of elements in each column of the array CARRAY (input; no default allowed). DIM must be as least as large as the value of BRNCHS returned by the GET_VSC_DC_LINE_COUNT API when it is called with the same values specified for SID, TIES and FLAG.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character CARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BRNCHS	Is the number of elements in each column of CARRAY that are set by this API (output). BRNCHS is identical to the value of BRNCHS returned by the GET_VSC_DC_LINE_COUNT API when it is called with the same values specified for SID, TIES and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid TIES value. IERR = 4 Invalid FLAG value. IERR = 5 Invalid NSTR value. IERR = 6 DIM, and hence the size of CARRAY, is not large enough. IERR = 7 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.20.6 aVscDcTypes

```
Use this API to return an array of character values indicating the
data types corresponding to a set of specified STRING values that are
valid input values to any of the data retrieval routines of the VSC
dc line data family (aVscDcInt, aVscDcReal, aVscDcCplx and
aVscDcChar).
```

Python syntax:

```
ierr, types = avscdctypes(string)
```

Fortran syntax:

```
CALL GET_VSC_DC_LINE_TYPES(NSTR,STRING,TYPES,IERR)
```

where:

Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR strings that are valid STRING values for any of the data retrieval routines of the VSC dc line data family (input; no default allowed):
Character TYPES*1	Is an array of at least NSTR elements, each of which contains one of the following characters indicating the data type of the data associated with the corresponding element of STRING (output):
	'I' Integer.
	'R' Real.
	'X' Complex.
	'C' Character.
	'?' Invalid STRING value.
Integer IERR	Is the error code (output):
	IERR = 0 No error.
	IERR = n Position of the first element in STRING that contains a value that is not a valid STRING value for any of the data retrieval routines in the VSC dc line data family (1 ≤ n ≤ NSTR).
	IERR = 99 Invalid NSTR value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.21 VSC dc Line Converter Data

The VSC dc line converter data family of subsystem data retrieval APIs returns arrays of data for VSC dc line converters in a specified subsystem.

One of the input items to each API allows the user to elect to retrieve data either for all subsystem VSC dc line converters, for subsystem VSC dc line converters of VSC dc lines that are tie branches, or for both subsystem and tie VSC dc line converters. Another input item allows the user to elect to retrieve data for all such VSC dc line converters, to restrict the array entries to data for all converters of in-service VSC dc lines, or to restrict the array entries to data for in-service converters of in-service VSC dc lines.

Data for subsystem converter buses is returned from all APIs in the VSC dc line converter data family in the same order. One of the input items to each API allows the user to elect that converter buses be ordered either in dc line name order with all converter buses of each VSC dc line grouped together, or in numerical or alphabetical order by converter bus, according to the bus output option setting in effect at the time the API is referenced.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.21.1 aVscDcConvCount

Use this API to return the number of array entries required to accommodate the data to be returned by the remaining members of the VSC dc line converter data family.

Python syntax:

```
ierr, buses = avscdcconvcount(sid, ties, flag)
```

Fortran syntax:

```
CALL GET_VSC_DC_CONV_COUNT(SID, TIES, FLAG, BUSES, IERR)
```

where:

Integer SID — Defines the bus subsystem to be used (input; -1 by default).

SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case.

SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.

Integer TIES — Is a flag indicating which subsystem converter buses of VSC dc lines to include (ignored if SID is negative) (input; 1 by default).

- TIES = 1 — for each end of interior subsystem VSC dc lines only.
- TIES = 2 — for the subsystem bus end of tie VSC dc lines only.
- TIES = 3 — for the non-subsystem bus end of tie VSC dc lines only.
- TIES = 4 — for each end of tie VSC dc lines only.
- TIES = 5 — for each end of interior subsystem VSC dc lines and the subsystem bus end of tie VSC dc lines.
- TIES = 6 — for each end of interior subsystem VSC dc lines and tie VSC dc lines.

Integer FLAG — Is a flag indicating which subsystem VSC dc line converters to include (input; 1 by default).

- FLAG = 1 — for only in-service converters of in-service VSC dc lines.
- FLAG = 2 — for all converters of in-service VSC dc lines.
- FLAG = 3 — for all converters of all VSC dc lines.

Integer BUSES — Is the number of array entries required for the subsystem indicated by SID that meet the editing criteria indicated by TIES and FLAG (output).

Integer IERR — Is the error code (output):

- IERR = 0 — No error, BUSES returned.
- IERR = 1 — Working case is empty; BUSES returned as 0.
- IERR = 2 — Invalid SID value; BUSES returned as 0.
- IERR = 3 — Invalid TIES value; BUSES returned as 0.
- IERR = 4 — Invalid FLAG value; BUSES returned as 0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.21.2 aVscDcConvInt

Use this API to return an array of integer values for subsystem VSC dc line converters.

Python syntax:

```
ierr, iarray = avscdcconvint(sid, ties, flag, entry, string)
```

Fortran syntax:

```
CALL GET_VSC_DC_CONV_INTEGER(SID, TIES, FLAG, ENTRY, NSTR, STRING,
DIM, IARRAY, BUSES, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem converter buses of VSC dc lines to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for each end of interior subsystem VSC dc lines only. TIES = 2 for the subsystem bus end of tie VSC dc lines only. TIES = 3 for the non-subsystem bus end of tie VSC dc lines only. TIES = 4 for each end of tie VSC dc lines only. TIES = 5 for each end of interior subsystem VSC dc lines and the subsystem bus end of tie VSC dc lines. TIES = 6 for each end of interior subsystem VSC dc lines and tie VSC dc lines.
Integer FLAG	Is a flag indicating which subsystem VSC dc line converters to include (input; 1 by default). FLAG = 1 for only in-service converters of in-service VSC dc lines. FLAG = 2 for all converters of in-service VSC dc lines. FLAG = 3 for all converters of all VSC dc lines.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for converter bus order. ENTRY = 2 for dc line name order.
Integer NSTR	Is the number of elements in STRING ($1 \leq$ NSTR ≤ 50) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the VSC dc line converter quantities desired (input; no default allowed): 'FROMNUMBER' From bus number. 'TONUMBER' To bus number. 'IREG' Bus number of the remote voltage controlled bus. 'CONVERTER' 1 if the from bus is converter 1, 2 if converter 2. 'DCTYPE' Converter dc control code. 'ACMODE' Converter ac control mode.
Integer DIM	Is the number of elements in each column of the array IARRAY (input; no default allowed). DIM must be as least as large as the value of BUSES returned by the GET_VSC_DC_CONV_COUNT API when it is called with the same values specified for SID, TIES and FLAG.
Integer IARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BUSES	Is the number of elements in each column of IARRAY that are set by this API (output). BUSES is identical to the value of BUSES returned by the GET_VSC_DC_CONV_COUNT API when it is called with the same values specified for SID, TIES and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid TIES value. IERR = 4 Invalid FLAG value. IERR = 5 Invalid ENTRY value. IERR = 6 Invalid NSTR value. IERR = 7 DIM, and hence the size of IARRAY, is not large enough. IERR = 8 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.21.3 aVscDcConvReal

Use this API to return an array of real values for subsystem VSC dc line converters.

Python syntax:

```
ierr, rarray = avscdcconvreal(sid, ties, flag, entry, string)
```

Fortran syntax:

```
CALL GET_VSC_DC_CONV_REAL(SID, TIES, FLAG, ENTRY, NSTR, STRING, DIM,
RARRAY, BUSES, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem converter buses of VSC dc lines to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for each end of interior subsystem VSC dc lines only. TIES = 2 for the subsystem bus end of tie VSC dc lines only. TIES = 3 for the non-subsystem bus end of tie VSC dc lines only. TIES = 4 for each end of tie VSC dc lines only. TIES = 5 for each end of interior subsystem VSC dc lines and the subsystem bus end of tie VSC dc lines. TIES = 6 for each end of interior subsystem VSC dc lines and tie VSC dc lines.
Integer FLAG	Is a flag indicating which subsystem VSC dc line converters to include (input; 1 by default). FLAG = 1 for only in-service converters of in-service VSC dc lines. FLAG = 2 for all converters of in-service VSC dc lines. FLAG = 3 for all converters of all VSC dc lines.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for converter bus order. ENTRY = 2 for dc line name order.
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)

Is an array of NSTR elements specifying NSTR of the following strings indicating the VSC dc line converter quantities desired (input; no default allowed):

'ACAMPS'	ac current in amps at the from bus.
'PUCUR'	ac current in pu at the from bus..
'PCTMVA'	MVA as a percent of MVA rating (more restrictive of SMAX and IMAX) at the from bus.
'KVDC'	From bus dc voltage in kV.
'DCSET'	dc setpoint (kV or MW) at the from bus.
'ACSET'	ac setpoint (power factor or pu voltage) at the from bus.
'ALOSS'	From bus A loss coefficient in kW.
'BLOSS'	From bus B loss coefficient in kW/amp.
'MINLOSS'	From bus minimum converter losses in kW.
'SMAX'	From bus converter MVA rating.
'IMAX'	From bus converter current rating in amps.
'PWF'	From bus power weighting factor fraction.
'RMPCT'	Percent of total Mvar to be contributed by this converter.

For the following, values are returned in MW, Mvar, or MVA:

'PAC'	Apparent active power flow into the dc line at the from bus.
'QAC'	Apparent reactive power flow into the dc line at the from bus.
'MVA'	Apparent ac power flow into the dc line at the from bus.
'PLOSS'	Total active power losses.
'QLOSS'	Total reactive power losses.
'MAXQ'	Reactive power upper limit.
'MINQ'	Reactive power lower limit.

For the following, values are returned in units determined by the power output option setting:

'O_PAC'	Apparent active power flow into the dc line at the from bus.
'O_QAC'	Apparent reactive power flow into the dc line at the from bus.
'O_MVA'	Apparent ac power flow into the dc line at the from bus.
'O_PLOSS'	Total active power losses.
'O_QLOSS'	Total reactive power losses.
'O_MAXQ'	Reactive power upper limit.
'O_MINQ'	Reactive power lower limit.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer DIM	Is the number of elements in each column of the array RARRAY (input; no default allowed). DIM must be as least as large as the value of BUSES returned by the GET_VSC_DC_CONV_COUNT API when it is called with the same values specified for SID, TIES and FLAG.
Real RARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BUSES	Is the number of elements in each column of RARRAY that are set by this API (output). DIM must be as least as large as the value of BUSES returned by the GET_VSC_DC_CONV_COUNT API when it is called with the same values specified for SID, TIES and FLAG.
Integer IERR	Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid TIES value.
IERR = 4	Invalid FLAG value.
IERR = 5	Invalid ENTRY value.
IERR = 6	Invalid NSTR value.
IERR = 7	DIM, and hence the size of RARRAY, is not large enough.
IERR = 8	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.21.4 aVscDcConvCplx

Use this API to return an array of complex values for subsystem VSC dc line converters.

Python syntax:

```
ierr, xarray = avscdcconvcplx(sid, ties, flag, entry, string)
```

Fortran syntax:

```
CALL GET_VSC_DC_CONV_COMPLEX(SID, TIES, FLAG, ENTRY, NSTR, STRING,
DIM, XARRAY, BUSES, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem converter buses of VSC dc lines to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for each end of interior subsystem VSC dc lines only. TIES = 2 for the subsystem bus end of tie VSC dc lines only. TIES = 3 for the non-subsystem bus end of tie VSC dc lines only. TIES = 4 for each end of tie VSC dc lines only. TIES = 5 for each end of interior subsystem VSC dc lines and the subsystem bus end of tie VSC dc lines. TIES = 6 for each end of interior subsystem VSC dc lines and tie VSC dc lines.
Integer FLAG	Is a flag indicating which subsystem VSC dc line converters to include (input; 1 by default). FLAG = 1 for only in-service converters of in-service VSC dc lines. FLAG = 2 for all converters of in-service VSC dc lines. FLAG = 3 for all converters of all VSC dc lines.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for converter bus order. ENTRY = 2 for dc line name order.
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the VSC dc line converter quantities desired (input; no default allowed): For the following, values are returned in MW and Mvar: 'PQAC' Apparent ac power flow into the dc line at the from bus. 'PQLOSS' ac power losses. For the following, values are returned in units determined by the power output option setting: 'O_PQAC' Apparent ac power flow into the dc line at the from bus. 'O_PQLOSS' ac power losses.
Integer DIM	Is the number of elements in each column of the array XARRAY (input; no default allowed). DIM must be as least as large as the value of BUSES returned by the GET_VSC_DC_CONV_COUNT API when it is called with the same values specified for SID, TIES and FLAG.
Complex XARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BUSES	Is the number of elements in each column of XARRAY that are set by this API (output). DIM must be as least as large as the value of BUSES returned by the GET_VSC_DC_CONV_COUNT API when it is called with the same values specified for SID, TIES and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid TIES value. IERR = 4 Invalid FLAG value. IERR = 5 Invalid ENTRY value. IERR = 6 Invalid NSTR value. IERR = 7 DIM, and hence the size of XARRAY, is not large enough. IERR = 8 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.21.5 aVscDcConvChar

Use this API to return an array of character values for subsystem VSC dc line converters.

Python syntax:

```
ierr, carray = avscdcconvchar(sid, ties, flag, entry, string)
```

Fortran syntax:

```
CALL GET_VSC_DC_CONV_CHAR(SID, TIES, FLAG, ENTRY, NSTR, STRING, DIM,
CARRAY, BUSES, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem converter buses of VSC dc lines to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for each end of interior subsystem VSC dc lines only. TIES = 2 for the subsystem bus end of tie VSC dc lines only. TIES = 3 for the non-subsystem bus end of tie VSC dc lines only. TIES = 4 for each end of tie VSC dc lines only. TIES = 5 for each end of interior subsystem VSC dc lines and the subsystem bus end of tie VSC dc lines. TIES = 6 for each end of interior subsystem VSC dc lines and tie VSC dc lines.
Integer FLAG	Is a flag indicating which subsystem VSC dc line converters to include (input; 1 by default). FLAG = 1 for only in-service converters of in-service VSC dc lines. FLAG = 2 for all converters of in-service VSC dc lines. FLAG = 3 for all converters of all VSC dc lines.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for converter bus order. ENTRY = 2 for dc line name order.
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the VSC dc line converter quantities desired (input; no default allowed): 'VSCNAME' dc line name (12 characters). 'FROMNAME' From bus name (12 characters). 'FROMEXNAME' From bus extended bus name (18 characters). 'TONAME' To bus name (12 characters). 'TOEXNAME' To bus extended bus name (18 characters). 'IREGNAME' Bus name of the remote voltage controlled bus (12 characters). 'IREGEXNAME' Extended bus name of the remote voltage controlled bus (18 characters).
Integer DIM	Is the number of elements in each column of the array CARRAY (input; no default allowed). DIM must be as least as large as the value of BUSES returned by the GET_VSC_DC_CONV_COUNT API when it is called with the same values specified for SID, TIES and FLAG.
Character CARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BUSES	Is the number of elements in each column of CARRAY that are set by this API (output). DIM must be as least as large as the value of BUSES returned by the GET_VSC_DC_CONV_COUNT API when it is called with the same values specified for SID, TIES and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid TIES value. IERR = 4 Invalid FLAG value. IERR = 5 Invalid ENTRY value. IERR = 6 Invalid NSTR value. IERR = 7 DIM, and hence the size of CARRAY, is not large enough. IERR = 8 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.21.6 aVscDcConvTypes

Use this API to return an array of character values indicating the data types corresponding to a set of specified STRING values that are valid input values to any of the data retrieval routines of the VSC dc line converter data family (aVscDcConvInt, aVscDcConvReal, aVscDcConvCplx and aVscDcConvChar).

Python syntax:

```
ierr, types = avscdcconvtypes(string)
```

Fortran syntax:

```
CALL GET_VSC_DC_CONV_TYPES(NSTR,STRING,TYPES,IERR)
```

where:

Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR strings that are valid STRING values for any of the data retrieval routines of the VSC dc line converter data family (input; no default allowed):
Character TYPES*1	Is an array of at least NSTR elements, each of which contains one of the following characters indicating the data type of the data associated with the corresponding element of STRING (output): 'I' Integer. 'R' Real. 'X' Complex. 'C' Character. '?' Invalid STRING value.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = n Position of the first element in STRING that contains a value that is not a valid STRING value for any of the data retrieval routines in the VSC dc line converter data family (1 ≤ n ≤ NSTR). IERR = 99 Invalid NSTR value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.22 FACTS Device Data

The FACTS device data family of subsystem data retrieval APIs returns arrays of data for FACTS devices in a specified subsystem.

FACTS devices with a series element may be subsystem tie branches. One of the input items to each API allows the user to elect to retrieve data either for all subsystem FACTS devices, for subsystem FACTS devices that are tie branches, or for both subsystem and tie FACTS devices. Another input item allows the user to elect to retrieve data only for FACTS devices without a series element, only for FACTS devices with a series element, or all types of FACTS devices.

Another input item allows the user to elect to retrieve data for all such FACTS devices, or to restrict the array entries to data for in-service FACTS devices.

Data for subsystem FACTS devices is returned from all APIs in the FACTS device data family in ascending FACTS device name alphabetical order.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.22.1 aFactsCount

Use this API to return the number of array entries required to accommodate the data to be returned by the remaining members of the FACTS device data family.

Python syntax:

```
ierr, brnchs = afactscount(sid, ties, flag, fcttyp)
```

Fortran syntax:

```
CALL GET_FACTS_COUNT(SID, TIES, FLAG, FCTTYP, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem FACTS devices with a series element to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem FACTS devices only. TIES = 2 for subsystem tie FACTS devices only. TIES = 3 for both interior subsystem FACTS devices and tie FACTS devices.
Integer FLAG	Is a flag indicating which subsystem FACTS devices to include (input; 1 by default). FLAG = 1 for only in-service FACTS devices. FLAG = 2 for both in-service and out-of-service FACTS devices.
Integer FCTTYP	Is a flag indicating which types of FACTS devices to include (input; 1 by default). FCTTYP = 1 for only FACTS devices without a series element. FCTTYP = 2 for only FACTS devices with a series element. FCTTYP = 3 for both types of FACTS devices.
Integer BRNCHS	Is the number of array entries required for the subsystem indicated by SID that meet the editing criteria indicated by TIES, FLAG and FCTTYP (output).
Integer IERR	Is the error code (output): IERR = 0 No error, BRNCHS returned. IERR = 1 Working case is empty; BRNCHS returned as 0. IERR = 2 Invalid SID value; BRNCHS returned as 0. IERR = 3 Invalid TIES value; BRNCHS returned as 0. IERR = 4 Invalid FLAG value; BRNCHS returned as 0. IERR = 5 Invalid FCTTYP value; BRNCHS returned as 0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.22.2 aFactsInt

Use this API to return an array of integer values for subsystem FACTS devices.

Python syntax:

```
ierr, iarray = afactsint(sid, ties, flag, fcttyp, string)
```

Fortran syntax:

```
CALL GET_FACTS_INTEGER(SID, TIES, FLAG, FCTTYP, NSTR, STRING, DIM,
IARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem FACTS devices with a series element to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem FACTS devices only. TIES = 2 for subsystem tie FACTS devices only. TIES = 3 for both interior subsystem FACTS devices and tie FACTS devices.
Integer FLAG	Is a flag indicating which subsystem FACTS devices to include (input; 1 by default). FLAG = 1 for only in-service FACTS devices. FLAG = 2 for both in-service and out-of-service FACTS devices.
Integer FCTTYP	Is a flag indicating which types of FACTS devices to include (input; 1 by default). FCTTYP = 1 for only FACTS devices without a series element. FCTTYP = 2 for only FACTS devices with a series element. FCTTYP = 3 for both types of FACTS devices.
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the FACTS device quantities desired (input; no default allowed): 'SENDNUMBER' Sending end bus number. 'TERMNUMBER' Terminal end bus number. 'MODE' Control mode (negative if blocked during solution). 'OWNER' Owner number. 'VSREF' Series voltage reference code. 'STATVT' Terminal end voltage condition code: -1 (holding low limit), 0 (in range), or 1 (holding high limit). 'STATVS' Series voltage limit code: 0 (in range), or 1 (holding high limit). 'STATBR' Bridge active power transfer limit code: 0 (in range), or 1 (holding high limit).
Integer DIM	Is the number of elements in each column of the array IARRAY (input; no default allowed). DIM must be as least as large as the value of BRNCHS returned by the GET_FACTS_COUNT API when it is called with the same values specified for SID, TIES, FLAG and FCTTYP.
Integer IARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BRNCHS	Is the number of elements in each column of IARRAY that are set by this API (output). BRNCHS is identical to the value of BRNCHS returned by the GET_FACTS_COUNT API when it is called with the same values specified for SID, TIES, FLAG and FCTTYP.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid TIES value. IERR = 4 Invalid FLAG value. IERR = 5 Invalid FCTTYP value. IERR = 6 Invalid NSTR value. IERR = 7 DIM, and hence the size of IARRAY, is not large enough. IERR = 8 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.22.3 aFactsReal

Use this API to return an array of real values for subsystem FACTS devices.

Python syntax:

```
ierr, rarray = afactsreal(sid, ties, flag, fcttyp, string)
```

Fortran syntax:

```
CALL GET_FACTS_REAL(SID, TIES, FLAG, FCTTYP, NSTR, STRING, DIM,
RARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem FACTS devices with a series element to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem FACTS devices only. TIES = 2 for subsystem tie FACTS devices only. TIES = 3 for both interior subsystem FACTS devices and tie FACTS devices.
Integer FLAG	Is a flag indicating which subsystem FACTS devices to include (input; 1 by default). FLAG = 1 for only in-service FACTS devices. FLAG = 2 for both in-service and out-of-service FACTS devices.
Integer FCTTYP	Is a flag indicating which types of FACTS devices to include (input; 1 by default). FCTTYP = 1 for only FACTS devices without a series element. FCTTYP = 2 for only FACTS devices with a series element. FCTTYP = 3 for both types of FACTS devices.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the FACTS device quantities desired (input; no default allowed): 'VSET' Voltage setpoint at the sending end bus in pu. 'VTMN' Minimum voltage at the terminal end bus in pu. 'VTMX' Maximum voltage at the terminal end bus in pu. 'VSMX' Maximum series voltage magnitude in pu. 'LINX' Reactance of the dummy series element. 'RMPCT' Percentage of the total MVAR required to regulate the controlled bus voltage that are contributed by this device. 'VSERES' Series voltage magnitude in pu. 'IPUSND' Current flow into series element at sending end bus in pu.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

’IPUTRM’	Current flow into series element at terminal end bus in pu.
’IPUSHN’	Current flow into shunt element in pu.
’AMPSND’	Current flow into series element at sending end bus in amps.
’AMPTRM’	Current flow into series element at terminal end bus in amps.
’AMPSHN’	Current flow into shunt element in amps.
’PCTSND’	Percent series current loading at sending bus end.
’PCTTRM’	Percent series current loading at terminal bus end.
’PCTSHN’	Percent shunt current loading.
’PCTPBR’	Percent bridge active power loading.

For the following, values are returned in MW, Mvar, or MVA:

’PDES’	Desired active power flow at the terminal bus end.
’QDES’	Desired reactive power flow at the terminal bus end.
’SHMX’	Maximum shunt current at the sending end bus at 1.0 pu voltage.
’TRMX’	Maximum bridge active power transfer.
’IMX’	Maximum series current at 1.0 pu voltage.
’PSEND’	Active power flow into the series element at the sending end bus.
’QSEND’	Reactive power flow into the series element at the sending end bus.
’PTERM’	Active power flow into the series element at the terminal end bus.
’QTERM’	Reactive power flow into the series element at the terminal end bus.
’PBRDG’	Bridge active power exchange.
’QSHNT’	Reactive power flow into the shunt element.
’MVASND’	Power flow into the series element at the sending end bus.
’MVATRM’	Power flow into the series element at the terminal end bus.
’MVASHN’	Power flow into the shunt element.
’QLOSS’	Reactive power losses.

For the following, values are returned in units determined by the power output option setting:

’O_PDES’	Desired active power flow at the terminal bus end.
’O_QDES’	Desired reactive power flow at the terminal bus end.
’O_SHMX’	Maximum shunt current at the sending end bus at 1.0 pu voltage.
’O_TRMX’	Maximum bridge active power transfer.
’O_IMX’	Maximum series current at 1.0 pu voltage.
’O_PSEND’	Active power flow into the series element at the sending end bus.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

'O_QSEND'	Reactive power flow into the series element at the sending end bus.
'O_PTERM'	Active power flow into the series element at the terminal end bus.
'O_QTERM'	Reactive power flow into the series element at the terminal end bus.
'O_PBRDG'	Bridge active power exchange.
'O_QSHNT'	Reactive power flow into the shunt element.
'O_MVASND'	Power flow into the series element at the sending end bus.
'O_MVATRM'	Power flow into the series element at the terminal end bus.
'O_MVASHN'	Power flow into the shunt element.
'O_QLOSS'	Reactive power losses.

Integer DIM — Is the number of elements in each column of the array RARRAY (input; no default allowed).

DIM must be as least as large as the value of BRNCHS returned by the GET_FACTS_COUNT API when it is called with the same values specified for SID, TIES, FLAG and FCTTYP.

Real RARRAY — Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer BRNCHS — Is the number of elements in each column of RARRAY that are set by this API (output).

BRNCHS is identical to the value of BRNCHS returned by the GET_FACTS_COUNT API when it is called with the same values specified for SID, TIES, FLAG and FCTTYP.

Integer IERR — Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid TIES value.
IERR = 4	Invalid FLAG value.
IERR = 5	Invalid FCTTYP value.
IERR = 6	Invalid NSTR value.
IERR = 7	DIM, and hence the size of RARRAY, is not large enough.
IERR = 8	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.22.4 aFactsCplx

Use this API to return an array of complex values for subsystem FACTS devices.

Python syntax:

```
ierr, xarray = afactscplx(sid, ties, flag, fcttyp, string)
```

Fortran syntax:

```
CALL GET_FACTS_COMPLEX(SID, TIES, FLAG, FCTTYP, NSTR, STRING, DIM,
XARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem FACTS devices with a series element to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem FACTS devices only. TIES = 2 for subsystem tie FACTS devices only. TIES = 3 for both interior subsystem FACTS devices and tie FACTS devices.
Integer FLAG	Is a flag indicating which subsystem FACTS devices to include (input; 1 by default). FLAG = 1 for only in-service FACTS devices. FLAG = 2 for both in-service and out-of-service FACTS devices.
Integer FCTTYP	Is a flag indicating which types of FACTS devices to include (input; 1 by default). FCTTYP = 1 for only FACTS devices without a series element. FCTTYP = 2 for only FACTS devices with a series element. FCTTYP = 3 for both types of FACTS devices.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the FACTS device quantities desired (input; no default allowed): For the following, values are returned in MW and Mvar: 'PQDES' Desired complex power flow at the terminal bus end. 'PQSEND' Complex power flow into the series element at the sending end bus. 'PQTERM' Complex power flow into the series element at the terminal end bus. 'PQSHNT' Complex bridge (active) and shunt (reactive) power flow into the shunt element. 'PQLOSS' Losses.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

For the following, values are returned in units determined by the power output option setting:

'O_PQDES'	Desired complex power flow at the terminal bus end.
'O_PQSEND'	Complex power flow into the series element at the sending end bus.
'O_PQTERM'	Complex power flow into the series element at the terminal end bus.
'O_PQSHNT'	Complex bridge (active) and shunt (reactive) power flow into the shunt element.
'O_PQLOSS'	Losses.

Integer DIM — Is the number of elements in each column of the array XARRAY (input; no default allowed).

DIM must be as least as large as the value of BRNCHS returned by the GET_FACTS_COUNT API when it is called with the same values specified for SID, TIES, FLAG and FCTTYP.

Complex XARRAY — Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer BRNCHS — Is the number of elements in each column of XARRAY that are set by this API (output).

BRNCHS is identical to the value of BRNCHS returned by the GET_FACTS_COUNT API when it is called with the same values specified for SID, TIES, FLAG and FCTTYP.

Integer IERR — Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid TIES value.
IERR = 4	Invalid FLAG value.
IERR = 5	Invalid FCTTYP value.
IERR = 6	Invalid NSTR value.
IERR = 7	DIM, and hence the size of RARRAY, is not large enough.
IERR = 8	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.22.5 aFactsChar

Use this API to return an array of character values for subsystem FACTS devices.

Python syntax:

```
ierr, carray = afactschar(sid, ties, flag, fcttyp, string)
```

Fortran syntax:

```
CALL GET_FACTS_CHAR(SID, TIES, FLAG, FCTTYP, STRING, DIM, CARRAY,
BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem FACTS devices with a series element to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for interior subsystem FACTS devices only. TIES = 2 for subsystem tie FACTS devices only. TIES = 3 for both interior subsystem FACTS devices and tie FACTS devices.
Integer FLAG	Is a flag indicating which subsystem FACTS devices to include (input; 1 by default). FLAG = 1 for only in-service FACTS devices. FLAG = 2 for both in-service and out-of-service FACTS devices.
Integer FCTTYP	Is a flag indicating which types of FACTS devices to include (input; 1 by default). FCTTYP = 1 for only FACTS devices without a series element. FCTTYP = 2 for only FACTS devices with a series element. FCTTYP = 3 for both types of FACTS devices.
Integer NSTR	Is the number of elements in STRING ($1 \leq$ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the FACTS device quantities desired (input; no default allowed): 'FACTSNAME' FACTS device name. 'FROMNAME' Sending end bus name (12 characters). 'FROMEXNAME' Sending end bus extended bus name (18 characters). 'TONAME' Terminal end bus name (12 characters). 'TOEXNAME' Terminal end bus extended bus name (18 characters).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer DIM	Is the number of elements in each column of the array CARRAY (input; no default allowed). DIM must be as least as large as the value of BRNCHS returned by the GET_FACTS_COUNT API when it is called with the same values specified for SID, TIES, FLAG and FCTTYP.
Character CARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BRNCHS	Is the number of elements in each column of CARRAY that are set by this API (output). BRNCHS is identical to the value of BRNCHS returned by the GET_FACTS_COUNT API when it is called with the same values specified for SID, TIES, FLAG and FCTTYP.
Integer IERR	Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid TIES value.
IERR = 4	Invalid FLAG value.
IERR = 5	Invalid FCTTYP value.
IERR = 6	Invalid NSTR value.
IERR = 7	DIM, and hence the size of CARRAY, is not large enough.
IERR = 8	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.22.6 aFactsTypes

Use this API to return an array of character values indicating the data types corresponding to a set of specified STRING values that are valid input values to any of the data retrieval routines of the FACTS device data family (aFactsInt, aFactsReal, aFactsCplx and aFactsChar).

Python syntax:

```
ierr, types = afactstypes(string)
```

Fortran syntax:

```
CALL GET_FACTS_TYPES(NSTR,STRING,TYPES,IERR)
```

where:

Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR strings that are valid STRING values for any of the data retrieval routines of the FACTS device data family (input; no default allowed):
Character TYPES*1	Is an array of at least NSTR elements, each of which contains one of the following characters indicating the data type of the data associated with the corresponding element of STRING (output): 'I' Integer. 'R' Real. 'X' Complex. 'C' Character. '?' Invalid STRING value.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = n Position of the first element in STRING that contains a value that is not a valid STRING value for any of the data retrieval routines in the FACTS device data family (1 ≤ n ≤ NSTR). IERR = 99 Invalid NSTR value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.23 FACTS Device Bus Data

The FACTS device bus data family of subsystem data retrieval APIs returns arrays of data for buses that are in a specified subsystem and to which FACTS devices are connected.

One of the input items to each API allows the user to elect to retrieve data either for all subsystem FACTS device buses, for subsystem buses connected to FACTS devices with a series element that is a tie branch, or for both subsystem and tie FACTS device buses. Another input item allows the user to elect to retrieve data either for FACTS devices without a series element, for FACTS devices with a series element, or for both types of FACTS devices.

Another input item allows the user to elect to retrieve data for all such FACTS device buses, or to restrict the array entries to data for buses connected to in-service FACTS devices.

Data for subsystem buses is returned from all APIs in the FACTS device bus data family in the same order. One of the input items to each API allows the user to elect that FACTS device buses be ordered either in FACTS device name order with all buses of each FACTS device grouped together, or in numerical or alphabetical order by FACTS device bus, according to the bus output option setting in effect at the time the API is referenced.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.23.1 aFactsBusCount

Use this API to return the number of array entries required to accommodate the data to be returned by the remaining members of the FACTS device bus data family.

Python syntax:

```
ierr, buses = afactsbuscount(sid, ties, flag, fcttyp)
```

Fortran syntax:

```
CALL GET_FACTS_BUS_COUNT(SID, TIES, FLAG, FCTTYP, BUSES, IERR)
```

where:

Integer SID — Defines the bus subsystem to be used (input; -1 by default).

SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case.

SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.

Integer TIES — Is a flag indicating which subsystem buses connected to FACTS devices with a series element to include (ignored if SID is negative) (input; 1 by default).

TIES = 1	for each end of interior subsystem FACTS devices only.
TIES = 2	for the subsystem bus end of tie FACTS devices only.
TIES = 3	for the non-subsystem bus end of tie FACTS devices only.
TIES = 4	for each end of tie FACTS devices only.
TIES = 5	for each end of interior subsystem FACTS devices and the subsystem bus end of tie FACTS devices.
TIES = 6	for each end of interior subsystem FACTS devices and tie FACTS devices.

Integer FLAG — Is a flag indicating which subsystem FACTS devices to include (input; 1 by default).

FLAG = 1	for only in-service FACTS devices.
FLAG = 2	for both in-service and out-of-service FACTS devices.

Integer FCTTYP — Is a flag indicating which types of FACTS devices to include (input; 1 by default).

FCTTYP = 1	for only FACTS devices without a series element.
FCTTYP = 2	for only FACTS devices with a series element.
FCTTYP = 3	for both types of FACTS devices.

Integer BUSES — Is the number of array entries required for the subsystem indicated by SID that meet the editing criteria indicated by TIES, FLAG and FCTTYP (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output):	
	IERR = 0	No error, BUSES returned.
	IERR = 1	Working case is empty; BUSES returned as 0.
	IERR = 2	Invalid SID value; BUSES returned as 0.
	IERR = 3	Invalid TIES value; BUSES returned as 0.
	IERR = 4	Invalid FLAG value; BUSES returned as 0.
	IERR = 5	Invalid FCTTYP value; BUSES returned as 0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.23.2 aFactsBusInt

Use this API to return an array of integer values for subsystem FACTS device buses.

Python syntax:

```
ierr, iarray = afactsbusint(sid, ties, flag, fcttyp, entry, string)
```

Fortran syntax:

```
CALL GET_FACTS_BUS_INTEGER(SID, TIES, FLAG, FCTTYP, ENTRY, NSTR,
STRING, DIM, IARRAY, BUSES, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem buses connected to FACTS devices with a series element to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for each end of interior subsystem FACTS devices only. TIES = 2 for the subsystem bus end of tie FACTS devices only. TIES = 3 for the non-subsystem bus end of tie FACTS devices only. TIES = 4 for each end of tie FACTS devices only. TIES = 5 for each end of interior subsystem FACTS devices and the subsystem bus end of tie FACTS devices. TIES = 6 for each end of interior subsystem FACTS devices and tie FACTS devices.
Integer FLAG	Is a flag indicating which subsystem FACTS devices to include (input; 1 by default). FLAG = 1 for only in-service FACTS devices. FLAG = 2 for both in-service and out-of-service FACTS devices.
Integer FCTTYP	Is a flag indicating which types of FACTS devices to include (input; 1 by default). FCTTYP = 1 for only FACTS devices without a series element. FCTTYP = 2 for only FACTS devices with a series element. FCTTYP = 3 for both types of FACTS devices.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for bus order. ENTRY = 2 for FACTS device name order.
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)

Is an array of NSTR elements specifying NSTR of the following strings indicating the FACTS device quantities desired (input; no default allowed):

'FROMNUMBER'	Bus number of this FACTS bus.
'TONUMBER'	Bus number of the other end FACTS bus.
'FACTSEND'	FACTS device end code of this FACTS bus: 1 (sending end bus) or 2 (terminal end bus).
'MODE'	Control mode (negative if blocked during solution).
'OWNER'	Owner number.
'VSREF'	Series voltage reference code.
'STATVT'	Terminal end voltage condition code: -1 (holding low limit), 0 (in range), or 1 (holding high limit).
'STATVS'	Series voltage limit code: 0 (in range), or 1 (holding high limit).
'STATBR'	Bridge active power transfer limit code: 0 (in range), or 1 (holding high limit).

Integer DIM

Is the number of elements in each column of the array IARRAY (input; no default allowed).

DIM must be as least as large as the value of BUSES returned by the GET_FACTS_BUS_COUNT API when it is called with the same values specified for SID, TIES, FLAG and FCTTYP.

Integer IARRAY

Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer BUSES

Is the number of elements in each column of IARRAY that are set by this API (output).

BUSES is identical to the value of BUSES returned by the GET_FACTS_BUS_COUNT API when it is called with the same values specified for SID, TIES, FLAG and FCTTYP.

Integer IERR

Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid TIES value.
IERR = 4	Invalid FLAG value.
IERR = 5	Invalid FCTTYP value.
IERR = 6	Invalid ENTRY value.
IERR = 7	Invalid NSTR value.
IERR = 8	DIM, and hence the size of IARRAY, is not large enough.
IERR = 9	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.23.3 aFactsBusReal

Use this API to return an array of real values for subsystem FACTS device buses.

Python syntax:

```
ierr, rarray = afactsbusreal(sid, ties, flag, fcttyp, entry, string)
```

Fortran syntax:

```
CALL GET_FACTS_BUS_REAL(SID, TIES, FLAG, FCTTYP, ENTRY, NSTR, STRING,
DIM, RARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem buses connected to FACTS devices with a series element to include (ignored if SID is negative) (input; 1 by default). TIES = 1 for each end of interior subsystem FACTS devices only. TIES = 2 for the subsystem bus end of tie FACTS devices only. TIES = 3 for the non-subsystem bus end of tie FACTS devices only. TIES = 4 for each end of tie FACTS devices only. TIES = 5 for each end of interior subsystem FACTS devices and the subsystem bus end of tie FACTS devices. TIES = 6 for each end of interior subsystem FACTS devices and tie FACTS devices.
Integer FLAG	Is a flag indicating which subsystem FACTS devices to include (input; 1 by default). FLAG = 1 for only in-service FACTS devices. FLAG = 2 for both in-service and out-of-service FACTS devices.
Integer FCTTYP	Is a flag indicating which types of FACTS devices to include (input; 1 by default). FCTTYP = 1 for only FACTS devices without a series element. FCTTYP = 2 for only FACTS devices with a series element. FCTTYP = 3 for both types of FACTS devices.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 for bus order. ENTRY = 2 for FACTS device name order.
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)

Is an array of NSTR elements specifying NSTR of the following strings indicating the FACTS device quantities desired (input; no default allowed):

'VSERES'	Series voltage magnitude in pu.
'IPUFROM'	Current flow into series element at the from bus end in pu.
'IPUSHN'	Current flow into shunt element in pu (0.0 if the from bus is the terminal end bus).
'AMPSFROM'	Current flow into series element at the from bus end in amps.
'AMPSHN'	Current flow into shunt element in amps (0.0 if the from bus is the terminal end bus).
'PCTFROM'	Percent series current loading at the from bus end.
'PCTSHN'	Percent shunt current loading (0.0 if the from bus is the terminal end bus).
'PCTPBR'	Percent bridge active power loading (0.0 if the from bus is the terminal end bus).

For the following, values are returned in MW, Mvar, or MVA:

'PFROM'	Active power flow into the series element at the from bus end.
'QFROM'	Reactive power flow into the series element at the from bus end.
'PBRDG'	Bridge active power exchange (0.0 if the from bus is the terminal end bus).
'QSHNT'	Reactive power flow into the shunt element (0.0 if the from bus is the terminal end bus).
'MVAFROM'	Power flow into the series element at the from bus end.
'MVASHN'	Power flow into the shunt element (0.0 if the from bus is the terminal end bus).
'QLOSS'	Reactive power losses.

For the following, values are returned in units determined by the power output option setting:

'O_PFROM'	Active power flow into the series element at the from bus end.
'O_QFROM'	Reactive power flow into the series element at the from bus end.
'O_PBRDG'	Bridge active power exchange (0.0 if the from bus is the terminal end bus).
'O_QSHNT'	Reactive power flow into the shunt element (0.0 if the from bus is the terminal end bus).
'O_MVAFROM'	Power flow into the series element at the from bus end.
'O_MVASHN'	Power flow into the shunt element (0.0 if the from bus is the terminal end bus).
'O_QLOSS'	Reactive power losses.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer DIM	Is the number of elements in each column of the array RARRAY (input; no default allowed). DIM must be as least as large as the value of BUSES returned by the GET_FACTS_BUS_COUNT API when it is called with the same values specified for SID, TIES, FLAG and FCTTYP.
Real RARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BUSES	Is the number of elements in each column of RARRAY that are set by this API (output). BUSES is identical to the value of BUSES returned by the GET_FACTS_BUS_COUNT API when it is called with the same values specified for SID, TIES, FLAG and FCTTYP.
Integer IERR	Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid TIES value.
IERR = 4	Invalid FLAG value.
IERR = 5	Invalid FCTTYP value.
IERR = 6	Invalid ENTRY value.
IERR = 7	Invalid NSTR value.
IERR = 8	DIM, and hence the size of RARRAY, is not large enough.
IERR = 9	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.23.4 aFactsBusCplx

Use this API to return an array of complex values for subsystem FACTS device buses.

Python syntax:

```
ierr, xarray = afactsbuscplx(sid, ties, flag, fcttyp, entry, string)
```

Fortran syntax:

```
CALL GET_FACTS_BUS_COMPLEX(SID, TIES, FLAG, FCTTYP, ENTRY, NSTR,
STRING, DIM, XARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem buses connected to FACTS devices with a series element to include (ignored if SID is negative) (input; 1 by default). TIES = 1 — for each end of interior subsystem FACTS devices only. TIES = 2 — for the subsystem bus end of tie FACTS devices only. TIES = 3 — for the non-subsystem bus end of tie FACTS devices only. TIES = 4 — for each end of tie FACTS devices only. TIES = 5 — for each end of interior subsystem FACTS devices and the subsystem bus end of tie FACTS devices. TIES = 6 — for each end of interior subsystem FACTS devices and tie FACTS devices.
Integer FLAG	Is a flag indicating which subsystem FACTS devices to include (input; 1 by default). FLAG = 1 — for only in-service FACTS devices. FLAG = 2 — for both in-service and out-of-service FACTS devices.
Integer FCTTYP	Is a flag indicating which types of FACTS devices to include (input; 1 by default). FCTTYP = 1 — for only FACTS devices without a series element. FCTTYP = 2 — for only FACTS devices with a series element. FCTTYP = 3 — for both types of FACTS devices.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 — for bus order. ENTRY = 2 — for FACTS device name order.
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the FACTS device quantities desired (input; no default allowed): For the following, values are returned in MW and Mvar: 'PQFROM' Complex power flow into the series element at the from bus end. 'PQSHNT' Complex bridge (active) and shunt (reactive) power flow into the shunt element (0.0 if the from bus is the terminal end bus). 'PQLOSS' Losses. For the following, values are returned in units determined by the power output option setting: 'O_PQFROM' Complex power flow into the series element at the from bus end. 'O_PQSHNT' Complex bridge (active) and shunt (reactive) power flow into the shunt element (0.0 if the from bus is the terminal end bus). 'O_PQLOSS' Losses.
Integer DIM	Is the number of elements in each column of the array XARRAY (input; no default allowed). DIM must be as least as large as the value of BUSES returned by the GET_FACTS_BUS_COUNT API when it is called with the same values specified for SID, TIES, FLAG and FCTTYP.
Complex XARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BUSES	Is the number of elements in each column of XARRAY that are set by this API (output). BUSES is identical to the value of BUSES returned by the GET_FACTS_BUS_COUNT API when it is called with the same values specified for SID, TIES, FLAG and FCTTYP.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid TIES value. IERR = 4 Invalid FLAG value. IERR = 5 Invalid FCTTYP value. IERR = 6 Invalid ENTRY value. IERR = 7 Invalid NSTR value. IERR = 8 DIM, and hence the size of RARRAY, is not large enough. IERR = 9 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.23.5 aFactsBusChar

Use this API to return an array of character values for subsystem FACTS device buses.

Python syntax:

```
ierr, carray = afactsbuschar(sid, ties, flag, fcttyp, entry, string)
```

Fortran syntax:

```
CALL GET_FACTS_BUS_CHAR(SID, TIES, FLAG, FCTTYP, ENTRY, NSTR, STRING,
DIM, CARRAY, BRNCHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer TIES	Is a flag indicating which subsystem buses connected to FACTS devices with a series element to include (ignored if SID is negative) (input; 1 by default). TIES = 1 — for each end of interior subsystem FACTS devices only. TIES = 2 — for the subsystem bus end of tie FACTS devices only. TIES = 3 — for the non-subsystem bus end of tie FACTS devices only. TIES = 4 — for each end of tie FACTS devices only. TIES = 5 — for each end of interior subsystem FACTS devices and the subsystem bus end of tie FACTS devices. TIES = 6 — for each end of interior subsystem FACTS devices and tie FACTS devices.
Integer FLAG	Is a flag indicating which subsystem FACTS devices to include (input; 1 by default). FLAG = 1 — for only in-service FACTS devices. FLAG = 2 — for both in-service and out-of-service FACTS devices.
Integer FCTTYP	Is a flag indicating which types of FACTS devices to include (input; 1 by default). FCTTYP = 1 — for only FACTS devices without a series element. FCTTYP = 2 — for only FACTS devices with a series element. FCTTYP = 3 — for both types of FACTS devices.
Integer ENTRY	Is a flag indicating output organization (input; 1 by default). ENTRY = 1 — for bus order. ENTRY = 2 — for FACTS device name order.
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the FACTS device quantities desired (input; no default allowed): 'FACTSNAME' FACTS device name. 'FROMNAME' Bus name of this FACTS bus (12 characters). 'FROMEXNAME' Extended bus name of this FACTS bus (18 characters). 'TONAME' Bus name of the other end FACTS bus (12 characters). 'TOEXNAME' Extended bus name of the other end FACTS bus (18 characters).
Integer DIM	Is the number of elements in each column of the array CARRAY (input; no default allowed). DIM must be as least as large as the value of BUSES returned by the GET_FACTS_BUS_COUNT API when it is called with the same values specified for SID, TIES, FLAG and FCTTYP.
Character CARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BUSES	Is the number of elements in each column of CARRAY that are set by this API (output). BUSES is identical to the value of BUSES returned by the GET_FACTS_BUS_COUNT API when it is called with the same values specified for SID, TIES, FLAG and FCTTYP.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid TIES value. IERR = 4 Invalid FLAG value. IERR = 5 Invalid FCTTYP value. IERR = 6 Invalid ENTRY value. IERR = 7 Invalid NSTR value. IERR = 8 DIM, and hence the size of CARRAY, is not large enough. IERR = 9 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.23.6 aFactsBusTypes

Use this API to return an array of character values indicating the data types corresponding to a set of specified STRING values that are valid input values to any of the data retrieval routines of the FACTS device bus data family (aFactsBusInt, aFactsBusReal, aFactsBusCplx and aFactsBusChar).

Python syntax:

```
ierr, types = afactsbustypes(string)
```

Fortran syntax:

```
CALL GET_FACTS_BUS_TYPES(NSTR,STRING,TYPES,IERR)
```

where:

Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR strings that are valid STRING values for any of the data retrieval routines of the FACTS device bus data family (input; no default allowed):
Character TYPES*1	Is an array of at least NSTR elements, each of which contains one of the following characters indicating the data type of the data associated with the corresponding element of STRING (output):
	'I' Integer.
	'R' Real.
	'X' Complex.
	'C' Character.
	'?' Invalid STRING value.
Integer IERR	Is the error code (output):
	IERR = 0 No error.
	IERR = n Position of the first element in STRING that contains a value that is not a valid STRING value for any of the data retrieval routines in the FACTS device bus data family (1 ≤ n ≤ NSTR).
	IERR = 99 Invalid NSTR value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.24 Area Data

The area data family of subsystem data retrieval APIs returns arrays of data for areas in a specified area subsystem. One of the input items to each API allows the user to elect either to retrieve data for areas with any ac buses, loads, induction machines, or dc buses assigned to them, or to restrict the array entries to data for areas with at least one ac bus assigned to them.

Data for subsystem areas is returned from all APIs in the area data family in the same order: ascending area number order.

8.24.1 aAreaCount

Use this API to return the number of array entries required to accommodate the data to be returned by the remaining members of the area data family.

Python syntax:

```
ierr, areas = aareacount(sid, flag)
```

Fortran syntax:

```
CALL GET_AREA_COUNT(SID, FLAG, AREAS, IERR)
```

where:

Integer SID	Defines the area subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all areas in the working case. SID = a valid area subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem areas to include (input; 1 by default). FLAG = 1 for only subsystem areas with at least one ac bus assigned to them. FLAG = 2 for subsystem areas with any equipment (ac buses, loads, induction machines, and/or dc buses) assigned to them.
Integer AREAS	Is the number of areas in the subsystem indicated by SID that meet the editing criteria indicated by FLAG (output).
Integer IERR	Is the error code (output): IERR = 0 No error, AREAS returned. IERR = 1 Working case is empty; AREAS returned as 0. IERR = 2 Invalid SID value; AREAS returned as 0. IERR = 3 Invalid FLAG value; AREAS returned as 0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.24.2 aAreaInt

Use this API to return an array of integer values for subsystem areas.

Python syntax:

```
ierr, iarray = aareaint(sid, flag, string)
```

Fortran syntax:

```
CALL GET_AREA_INTEGER(SID,FLAG,NSTR,STRING,DIM,IARRAY,AREAS,IERR)
```

where:

Integer SID	Defines the area subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all areas in the working case. SID = a valid area subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem areas to include (input; 1 by default). FLAG = 1 — for only subsystem areas with at least one ac bus assigned to them. FLAG = 2 — for subsystem areas with any equipment (ac buses, loads, induction machines, and/or dc buses) assigned to them.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the area quantities desired (input; no default allowed): 'NUMBER' — Area number. 'SWING' — Number of the area swing bus used for area interchange control purposes. 'BUSES' — Number of ac buses assigned to the area. 'LOADS' — Number of loads assigned to the area. 'INDMACS' — Number of induction machines assigned to the area. 'DCBUSES' — Number of dc buses assigned to the area.
Integer DIM	Is the number of elements in each column of the array IARRAY (input; no default allowed). DIM must be as least as large as the value of AREAS returned by the GET_AREA_COUNT API when it is called with the same values specified for SID and FLAG.
Integer IARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer AREAS	Is the number of elements in each column of IARRAY that are set by this API (output). AREAS is identical to the value of AREAS returned by the GET_AREA_COUNT API when it is called with the same values specified for SID and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid FLAG value. IERR = 4 Invalid NSTR value. IERR = 5 DIM, and hence the size of IARRAY, is not large enough. IERR = 6 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.24.3 aAreaReal

Use this API to return an array of real values for subsystem areas.

Python syntax:

```
ierr, rarray = aareareal(sid, flag, string)
```

Fortran syntax:

```
CALL GET_AREA_REAL(SID,FLAG,NSTR,STRING,DIM,RARRAY,AREAS,IERR)
```

where:

Integer SID	Defines the area subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all areas in the working case. SID = a valid area subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem areas to include (input; 1 by default). FLAG = 1 for only subsystem areas with at least one ac bus assigned to them. FLAG = 2 for subsystem areas with any equipment (ac buses, loads, induction machines, and/or dc buses) assigned to them.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the area quantities desired (input; no default allowed): For the following, values are returned in MW or Mvar:

’PDES’	Desired net active power interchange leaving the area.
’PTOL’	Interchange tolerance bandwidth.
’PLOAD’	Total active power load on buses in the area.
’QLOAD’	Total reactive power load on buses in the area.
’PLOADLD’	Total active power load of loads in the area.
’QLOADLD’	Total reactive power load of loads in the area.
’PGEN’	Total active power generation on buses in the area.
’QGEN’	Total reactive power generation on buses in the area.
'PINDMAC'	Total active power of induction machines on buses in the area.
'QINDMAC'	Total reactive power of induction machines on buses in the area.
'PINDMACMC'	Total active power of induction machines in the area.
'QINDMACMC'	Total reactive power of induction machines in the area.
’PLOSS’	Total active power losses assigned to the area.
’QLOSS’	Total reactive power losses assigned to the area.
’PINT’	Total active power net interchange from the area.
’QINT’	Total reactive power net interchange from the area.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

For the following, values are returned in units determined by the power output option setting:

'O_PDES'	Desired net active power interchange leaving the area.
'O_PTOL'	Interchange tolerance bandwidth.
'O_PLOAD'	Total active power load on buses in the area.
'O_QLOAD'	Total reactive power load on buses in the area.
'O_PLOADLD'	Total active power load of loads in the area.
'O_QLOADLD'	Total reactive power load of loads in the area.
'O_PGEN'	Total active power generation on buses in the area.
'O_QGEN'	Total reactive power generation on buses in the area.
'O_PINDMAC'	Total active power of induction machines on buses in the area.
'O_QINDMAC'	Total reactive power of induction machines on buses in the area.
'O_PINDMACMC'	Total active power of induction machines in the area.
'O_QINDMACMC'	Total reactive power of induction machines in the area.
'O_PLOSS'	Total active power losses assigned to the area.
'O_QLOSS'	Total reactive power losses assigned to the area.
'O_PINT'	Total active power net interchange from the area.
'O_QINT'	Total reactive power net interchange from the area.

Integer DIM

Is the number of elements in each column of the array RARRAY (input; no default allowed).

DIM must be as least as large as the value of AREAS returned by the GET_AREA_COUNT API when it is called with the same values specified for SID and FLAG.

Real RARRAY

Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer AREAS

Is the number of elements in each column of RARRAY that are set by this API (output).

AREAS is identical to the value of AREAS returned by the GET_AREA_COUNT API when it is called with the same values specified for SID and FLAG.

Integer IERR

Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid FLAG value.
IERR = 4	Invalid NSTR value.
IERR = 5	DIM, and hence the size of RARRAY, is not large enough.
IERR = 6	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.24.4 aAreaCplx

Use this API to return an array of complex values for subsystem areas.

Python syntax:

```
ierr, xarray = aareacplx(sid, flag, string)
```

Fortran syntax:

```
CALL GET_AREA_COMPLEX(SID,FLAG,NSTR,STRING,DIM,XARRAY,AREAS,IERR)
```

where:

Integer SID	Defines the area subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all areas in the working case. SID = a valid area subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem areas to include (input; 1 by default). FLAG = 1 for only subsystem areas with at least one ac bus assigned to them. FLAG = 2 for subsystem areas with any equipment (ac buses, loads, induction machines, and/or dc buses) assigned to them.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the area quantities desired (input; no default allowed): For the following, values are returned in MW and Mvar: ’PQLOAD’ Total complex load on buses in the area. ’PQLOADLD’ Total complex load of loads in the area. ’PQGEN’ Total complex generation on buses in the area. 'PQINDMAC' Total complex power of induction machines on buses in the area. 'PQINDMACMC' Total complex power of induction machines in the area. ’PQLOSS’ Total complex losses assigned to the area. ’PQINT’ Total complex net interchange from the area. For the following, values are returned in units determined by the power output option setting: ’O_PQLOAD’ Total complex load on buses in the area. ’O_PQLOADLD’ Total complex load of loads in the area. ’O_PQGEN’ Total complex generation on buses in the area. 'O_PQINDMAC' Total complex power of induction machines on buses in the area. 'O_PQINDMACMC' Total complex power of induction machines in the area. ’O_PQLOSS’ Total complex losses assigned to the area. ’O_PQINT’ Total complex net interchange from the area.
Integer DIM	Is the number of elements in each column of the array XARRAY (input; no default allowed). DIM must be as least as large as the value of AREAS returned by the GET_AREA_COUNT API when it is called with the same values specified for SID and FLAG.
Complex XARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer AREAS	Is the number of elements in each column of XARRAY that are set by this API (output). AREAS is identical to the value of AREAS returned by the GET_AREA_COUNT API when it is called with the same values specified for SID and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid FLAG value. IERR = 4 Invalid NSTR value. IERR = 5 DIM, and hence the size of XARRAY, is not large enough. IERR = 6 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.24.5 aAreaChar

Use this API to return an array of character values for subsystem areas.

Python syntax:

```
ierr, carray = aareachar(sid, flag, string)
```

Fortran syntax:

```
CALL GET_AREA_CHAR(SID,FLAG,NSTR,STRING,DIM,CARRAY,AREAS,IERR)
```

where:

Integer SID	Defines the area subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all areas in the working case. SID = a valid area subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem areas to include (input; 1 by default). FLAG = 1 — for only subsystem areas with at least one ac bus assigned to them. FLAG = 2 — for subsystem areas with any equipment (ac buses, loads, induction machines, and/or dc buses) assigned to them.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the area quantities desired (input; no default allowed): 'AREANAME' — Area name (12 characters). 'NAME' — Bus name of the area swing bus (12 characters). 'EXNAME' — Extended bus name of the area swing bus (18 characters).
Integer DIM	Is the number of elements in each column of the array CARRAY (input; no default allowed). DIM must be as least as large as the value of AREAS returned by the GET_AREA_COUNT API when it is called with the same values specified for SID and FLAG.
Character CARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer AREAS	Is the number of elements in each column of CARRAY that are set by this API (output). AREAS is identical to the value of AREAS returned by the GET_AREA_COUNT API when it is called with the same values specified for SID and FLAG.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output):	
	IERR = 0	No error.
	IERR = 1	Working case is empty.
	IERR = 2	Invalid SID value.
	IERR = 3	Invalid FLAG value.
	IERR = 4	Invalid NSTR value.
	IERR = 5	DIM, and hence the size of CARRAY, is not large enough.
	IERR = 6	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.24.6 aAreaTypes

Use this API to return an array of character values indicating the data types corresponding to a set of specified STRING values that are valid input values to any of the data retrieval routines of the area data family (aAreaInt, aAreaReal, aAreaCplx and aAreaChar).

Python syntax:

```
ierr, types = aareatypes(string)
```

Fortran syntax:

```
CALL GET_AREA_TYPES(NSTR,STRING,TYPES,IERR)
```

where:

Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR strings that are valid STRING values for any of the data retrieval routines of the area data family (input; no default allowed):
Character TYPES*1	Is an array of at least NSTR elements, each of which contains one of the following characters indicating the data type of the data associated with the corresponding element of STRING (output): 'I' Integer. 'R' Real. 'X' Complex. 'C' Character. '?' Invalid STRING value.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = n Position of the first element in STRING that contains a value that is not a valid STRING value for any of the data retrieval routines in the area data family (1 ≤ n ≤ NSTR). IERR = 99 Invalid NSTR value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.25 Owner Data

The owner data family of subsystem data retrieval APIs returns arrays of data for owners in a specified owner subsystem. One of the input items to each API allows the user to elect either to retrieve data for owners with any ac buses, loads, machines, induction machines, branches, FACTS devices, VSC dc lines, or dc buses assigned to them, or to restrict the array entries to data for owners with at least one ac bus assigned to them.

Data for subsystem owners is returned from all APIs in the owner data family in the same order: ascending owner number order.

8.25.1 aOwnerCount

Use this API to return the number of array entries required to accommodate the data to be returned by the remaining members of the owner data family.

Python syntax:

```
ierr, owners = aownercount(sid, flag)
```

Fortran syntax:

```
CALL GET_OWNER_COUNT(SID, FLAG, OWNERS, IERR)
```

where:

Integer SID	Defines the owner subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all owners in the working case. SID = a valid owner subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem owners to include (input; 1 by default). FLAG = 1 for only subsystem owners with at least one ac bus assigned to them. FLAG = 2 for subsystem owners with any equipment (ac buses, loads, machines, induction machines, ac branches, FACTS devices, VSC dc lines and/or dc buses) assigned to them.
Integer OWNERS	Is the number of owners in the subsystem indicated by SID that meet the editing criteria indicated by FLAG (output).
Integer IERR	Is the error code (output): IERR = 0 No error, OWNERS returned. IERR = 1 Working case is empty; OWNERS returned as 0. IERR = 2 Invalid SID value; OWNERS returned as 0. IERR = 3 Invalid FLAG value; OWNERS returned as 0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.25.2 aOwnerInt

Use this API to return an array of integer values for subsystem owners.

Python syntax:

```
ierr, iarray = aownerint(sid, flag, string)
```

Fortran syntax:

```
CALL GET_OWNER_INTEGER(SID,FLAG,NSTR,STRING,DIM,IARRAY,OWNERS,IERR)
```

where:

Integer SID	Defines the owner subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all owners in the working case. SID = a valid owner subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem owners to include (input; 1 by default). FLAG = 1 — for only subsystem owners with at least one ac bus assigned to them. FLAG = 2 — for subsystem owners with any equipment (ac buses, loads, machines, induction machines, ac branches, FACTS devices, VSC dc lines and/or dc buses) assigned to them.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the owner quantities desired (input; no default allowed): 'NUMBER' — Owner number. 'BUSES' — Number of ac buses assigned to the owner. 'LOADS' — Number of loads assigned to the owner. 'MACHINES' — Number of machine ownership specifications for the owner. 'INDMACS' — Number of induction machines assigned to the owner. 'BRANCHES' — Number of branch ownership specifications for the owner. 'FACTS' — Number of FACTS devices assigned to the owner. 'VSCDCLINES' — Number of VSC dc line ownership specifications for the owner. 'DCBUSES' — Number of dc buses assigned to the owner.
Integer DIM	Is the number of elements in each column of the array IARRAY (input; no default allowed). DIM must be as least as large as the value of OWNERS returned by the GET_OWNER_COUNT API when it is called with the same values specified for SID and FLAG.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer OWNERS	Is the number of elements in each column of IARRAY that are set by this API (output). OWNERS is identical to the value of OWNERS returned by the GET_OWNER_COUNT API when it is called with the same values specified for SID and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid FLAG value. IERR = 4 Invalid NSTR value. IERR = 5 DIM, and hence the size of IARRAY, is not large enough. IERR = 6 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.25.3 aOwnerReal

Use this API to return an array of real values for subsystem owners.

Python syntax:

```
ierr, rarray = aownerreal(sid, flag, string)
```

Fortran syntax:

```
CALL GET_OWNER_REAL(SID,FLAG,NSTR,STRING,DIM,RARRAY,OWNERS,IERR)
```

where:

Integer SID	Defines the owner subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all owners in the working case. SID = a valid owner subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem owners to include (input; 1 by default). FLAG = 1 — for only subsystem owners with at least one ac bus assigned to them. FLAG = 2 — for subsystem owners with any equipment (ac buses, loads, machines, induction machines, ac branches, FACTS devices, VSC dc lines and/or dc buses) assigned to them.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the owner quantities desired (input; no default allowed): For the following, values are returned in MW or Mvar:

’PLOAD’	Total active power load on the owner’s buses..
’QLOAD’	Total reactive power load on the owner’s buses.
’PLOADLD’	Total active power load of the owner’s loads.
’QLOADLD’	Total reactive power load of the owner’s loads.
’PGEN’	Total active power generation on the owner’s buses.
’QGEN’	Total reactive power generation on the owner’s buses.
’PGENMC’	Total active power generation on the owner’s machines.
’QGENMC’	Total reactive power generation on the owner’s machines.
'PINDMAC'	Total active power of induction machines on the owner's buses.
'QINDMAC'	Total reactive power of induction machines on the owner's buses.
'PINDMACMC'	Total active power of the owner's induction machines.
'QINDMACMC'	Total reactive power of the owner's induction machines.
’PLOSS’	Total active power losses assigned to the owner.
’QLOSS’	Total reactive power losses assigned to the owner.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

For the following, values are returned in units determined by the power output option setting:

’O_PLOAD’	Total active power load on the owner’s buses.
’O_QLOAD’	Total reactive power load on the owner’s buses.
’O_PLOADLD’	Total active power load of the owner’s loads.
’O_QLOADLD’	Total reactive power load of the owner’s loads.
’O_PGEN’	Total active power generation on the owner’s buses.
’O_QGEN’	Total reactive power generation on the owner’s buses.
’O_PGENMC’	Total active power generation on the owner’s machines.
’O_QGENMC’	Total reactive power generation on the owner’s machines.
'O_PINDMAC'	Total active power of induction machines on the owner's buses.
'O_QINDMAC'	Total reactive power of induction machines on the owner's buses.
'O_PINDMACMC'	Total active power of the owner's induction machines.
'O_QINDMACMC'	Total reactive power of the owner's induction machines.
’O_PLOSS’	Total active power losses assigned to the owner.
’O_QLOSS’	Total reactive power losses assigned to the owner.

Integer DIM

Is the number of elements in each column of the array RARRAY (input; no default allowed).

DIM must be as least as large as the value of OWNERS returned by the GET_OWNER_COUNT API when it is called with the same values specified for SID and FLAG.

Real RARRAY

Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer OWNERS

Is the number of elements in each column of RARRAY that are set by this API (output).

OWNERS is identical to the value of OWNERS returned by the GET_OWNER_COUNT API when it is called with the same values specified for SID and FLAG.

Integer IERR

Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid FLAG value.
IERR = 4	Invalid NSTR value.
IERR = 5	DIM, and hence the size of RARRAY, is not large enough.
IERR = 6	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.25.4 aOwnerCplx

Use this API to return an array of complex values for subsystem owners.

Python syntax:

```
ierr, xarray = aownercplx(sid, flag, string)
```

Fortran syntax:

```
CALL GET_OWNER_COMPLEX(SID,FLAG,NSTR,STRING,DIM,XARRAY,OWNERS,IERR)
```

where:

Integer SID	Defines the owner subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all owners in the working case. SID = a valid owner subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem owners to include (input; 1 by default). FLAG = 1 for only subsystem owners with at least one ac bus assigned to them. FLAG = 2 for subsystem owners with any equipment (ac buses, loads, machines, induction machines, ac branches, FACTS devices, VSC dc lines and/or dc buses) assigned to them.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)

Is an array of NSTR elements specifying NSTR of the following strings indicating the owner quantities desired (input; no default allowed):

For the following, values are returned in MW and Mvar:

’PQLOAD’	Total complex load on the owner’s buses.
’PQLOADLD’	Total complex load of the owner’s loads.
’PQGEN’	Total complex generation on the owner’s buses.
’PQGENMC’	Total complex generation on the owner’s machines.
'PQINDMAC'	Total complex power of induction machines on the owner's buses.
'PQINDMACMC'	Total complex power of the owner's induction machines.
’PQLOSS’	Total complex losses assigned to the owner.

For the following, values are returned in units determined by the power output option setting:

’O_PQLOAD’	Total complex load on the owner’s buses.
’O_PQLOADLD’	Total complex load of the owner’s loads.
’O_PQGEN’	Total complex generation on the owner’s buses.
’O_PQGENMC’	Total complex generation on the owner’s machines.
'O_PQINDMAC'	Total complex power of induction machines on the owner's buses.
'O_PQINDMACMC'	Total complex power of the owner's induction machines.
’O_PQLOSS’	Total complex losses assigned to the owner.

Integer DIM

Is the number of elements in each column of the array XARRAY (input; no default allowed).

DIM must be as least as large as the value of OWNERS returned by the GET_OWNER_COUNT API when it is called with the same values specified for SID and FLAG.

Complex XARRAY

Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer OWNERS

Is the number of elements in each column of XARRAY that are set by this API (output).

OWNERS is identical to the value of OWNERS returned by the GET_OWNER_COUNT API when it is called with the same values specified for SID and FLAG.

Integer IERR

Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid FLAG value.
IERR = 4	Invalid NSTR value.
IERR = 5	DIM, and hence the size of XARRAY, is not large enough.
IERR = 6	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.25.5 aOwnerChar

Use this API to return an array of character values for subsystem owners.

Python syntax:

```
ierr, carray = aownerchar(sid, flag, string)
```

Fortran syntax:

```
CALL GET_OWNER_CHAR(SID,FLAG,NSTR,STRING,DIM,CARRAY,OWNERS,IERR)
```

where:

Integer SID	Defines the owner subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all owners in the working case. SID = a valid owner subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem owners to include (input; 1 by default). FLAG = 1 for only subsystem owners with at least one ac bus assigned to them. FLAG = 2 for subsystem owners with any equipment (ac buses, loads, machines, induction machines, ac branches, FACTS devices, VSC dc lines and/or dc buses) assigned to them.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the owner quantities desired (input; no default allowed): 'OWNERNAME' Owner name (12 characters).
Integer DIM	Is the number of elements in each column of the array CARRAY (input; no default allowed). DIM must be as least as large as the value of OWNERS returned by the GET_OWNER_COUNT API when it is called with the same values specified for SID and FLAG.
Character CARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer OWNERS	Is the number of elements in each column of CARRAY that are set by this API (output). OWNERS is identical to the value of OWNERS returned by the GET_OWNER_COUNT API when it is called with the same values specified for SID and FLAG.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output):	
	IERR = 0	No error.
	IERR = 1	Working case is empty.
	IERR = 2	Invalid SID value.
	IERR = 3	Invalid FLAG value.
	IERR = 4	Invalid NSTR value.
	IERR = 5	DIM, and hence the size of CARRAY, is not large enough.
	IERR = 6	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.25.6 aOwnerTypes

Use this API to return an array of character values indicating the data types corresponding to a set of specified STRING values that are valid input values to any of the data retrieval routines of the owner data family (aOwnerInt, aOwnerReal, aOwnerCplx and aOwnerChar).

Python syntax:

```
ierr, types = aownertypes(string)
```

Fortran syntax:

```
CALL GET_OWNER_TYPES(NSTR,STRING,TYPES,IERR)
```

where:

Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR strings that are valid STRING values for any of the data retrieval routines of the owner data family (input; no default allowed):
Character TYPES*1	Is an array of at least NSTR elements, each of which contains one of the following characters indicating the data type of the data associated with the corresponding element of STRING (output): 'I' Integer. 'R' Real. 'X' Complex. 'C' Character. '?' Invalid STRING value.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = n Position of the first element in STRING that contains a value that is not a valid STRING value for any of the data retrieval routines in the owner data family (1 ≤ n ≤ NSTR). IERR = 99 Invalid NSTR value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.26 Zone Data

The zone data family of subsystem data retrieval APIs returns arrays of data for zones in a specified zone subsystem. One of the input items to each API allows the user to elect either to retrieve data for zones with any ac buses, loads, induction machines, or dc buses assigned to them, or to restrict the array entries to data for zones with at least one ac bus assigned to them.

Data for subsystem zones is returned from all APIs in the zone data family in the same order: ascending zone number order.

8.26.1 aZoneCount

Use this API to return the number of array entries required to accommodate the data to be returned by the remaining members of the zone data family.

Python syntax:

```
ierr, zones = azonecount(sid, flag)
```

Fortran syntax:

```
CALL GET_ZONE_COUNT(SID, FLAG, ZONES, IERR)
```

where:

Integer SID	Defines the zone subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all zones in the working case. SID = a valid zone subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem zones to include (input; 1 by default). FLAG = 1 for only subsystem zones with at least one ac bus assigned to them. FLAG = 2 for subsystem zones with any equipment (ac buses, loads, induction machines, and/or dc buses) assigned to them.
Integer ZONES	Is the number of zones in the subsystem indicated by SID that meet the editing criteria indicated by FLAG (output).
Integer IERR	Is the error code (output): IERR = 0 No error, ZONES returned. IERR = 1 Working case is empty; ZONES returned as 0. IERR = 2 Invalid SID value; ZONES returned as 0. IERR = 3 Invalid FLAG value; ZONES returned as 0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.26.2 aZoneInt

Use this API to return an array of integer values for subsystem zones.

Python syntax:

```
ierr, iarray = azoneint(sid, flag, string)
```

Fortran syntax:

```
CALL GET_ZONE_INTEGER(SID,FLAG,NSTR,STRING,DIM,IARRAY,ZONES,IERR)
```

where:

Integer SID	Defines the zone subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all zones in the working case. SID = a valid zone subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem zones to include (input; 1 by default). FLAG = 1 — for only subsystem zones with at least one ac bus assigned to them. FLAG = 2 — for subsystem zones with any equipment (ac buses, loads, induction machines, and/or dc buses) assigned to them.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the zone quantities desired (input; no default allowed): 'NUMBER' — Zone number. 'BUSES' — Number of ac buses assigned to the zone. 'LOADS' — Number of loads assigned to the zone. 'INDMACS' — Number of induction machines assigned to the zone. 'DCBUSES' — Number of dc buses assigned to the zone.
Integer DIM	Is the number of elements in each column of the array IARRAY (input; no default allowed). DIM must be as least as large as the value of ZONES returned by the GET_ZONE_COUNT API when it is called with the same values specified for SID and FLAG.
Integer IARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer ZONES	Is the number of elements in each column of IARRAY that are set by this API (output). ZONES is identical to the value of ZONES returned by the GET_ZONE_COUNT API when it is called with the same values specified for SID and FLAG.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output):	
	IERR = 0	No error.
	IERR = 1	Working case is empty.
	IERR = 2	Invalid SID value.
	IERR = 3	Invalid FLAG value.
	IERR = 4	Invalid NSTR value.
	IERR = 5	DIM, and hence the size of IARRAY, is not large enough.
	IERR = 6	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.26.3 aZoneReal

Use this API to return an array of real values for subsystem zones.

Python syntax:

```
ierr, rarray = azonereal(sid, flag, string)
```

Fortran syntax:

```
CALL GET_ZONE_REAL(SID,FLAG,NSTR,STRING,DIM,RARRAY,ZONES,IERR)
```

where:

Integer SID	Defines the zone subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all zones in the working case. SID = a valid zone subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem zones to include (input; 1 by default). FLAG = 1 for only subsystem zones with at least one ac bus assigned to them. FLAG = 2 for subsystem zones with any equipment (ac buses, loads, induction machines, and/or dc buses) assigned to them.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)

Is an array of NSTR elements specifying NSTR of the following strings indicating the zone quantities desired (input; no default allowed):

For the following, values are returned in MW or Mvar:

’PLOAD’	Total active power load on buses in the zone..
’QLOAD’	Total reactive power load on buses in the zone.
’PLOADLD’	Total active power load of loads in the zone.
’QLOADLD’	Total reactive power load of loads in the zone.
’PGEN’	Total active power generation on buses in the zone.
’QGEN’	Total reactive power generation on buses in the zone.
'PINDMAC'	Total active power of induction machines on buses in the zone.
'QINDMAC'	Total reactive power of induction machines on buses in the zone.
'PINDMACMC'	Total active power of induction machines in the zone.
'QINDMACMC'	Total reactive power of induction machines in the zone.
’PLOSS’	Total active power losses assigned to the zone.
’QLOSS’	Total reactive power losses assigned to the zone.
’PINT’	Total active power net interchange from the zone.
’QINT’	Total reactive power net interchange from the zone.

For the following, values are returned in units determined by the power output option setting:

’O_PLOAD’	Total active power load on buses in the zone.
’O_QLOAD’	Total reactive power load on buses in the zone.
’O_PLOADLD’	Total active power load of loads in the zone.
’O_QLOADLD’	Total reactive power load of loads in the zone.
’O_PGEN’	Total active power generation on buses in the zone.
’O_QGEN’	Total reactive power generation on buses in the zone.
'O_PINDMAC'	Total active power of induction machines on buses in the zone.
'O_QINDMAC'	Total reactive power of induction machines on buses in the zone.
'O_PINDMACMC'	Total active power of induction machines in the zone.
'O_QINDMACMC'	Total reactive power of induction machines in the zone.
’O_PLOSS’	Total active power losses assigned to the zone.
’O_QLOSS’	Total reactive power losses assigned to the zone.
’O_PINT’	Total active power net interchange from the zone.
’O_QINT’	Total reactive power net interchange from the zone.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer DIM	Is the number of elements in each column of the array RARRAY (input; no default allowed). DIM must be as least as large as the value of ZONES returned by the GET_ZONE_COUNT API when it is called with the same values specified for SID and FLAG.
Real RARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer ZONES	Is the number of elements in each column of RARRAY that are set by this API (output). ZONES is identical to the value of ZONES returned by the GET_ZONE_COUNT API when it is called with the same values specified for SID and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid FLAG value. IERR = 4 Invalid NSTR value. IERR = 5 DIM, and hence the size of RARRAY, is not large enough. IERR = 6 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.26.4 aZoneCplx

Use this API to return an array of complex values for subsystem zones.

Python syntax:

```
ierr, xarray = azonecplx(sid, flag, string)
```

Fortran syntax:

```
CALL GET_ZONE_COMPLEX(SID,FLAG,NSTR,STRING,DIM,XARRAY,ZONES,IERR)
```

where:

Integer SID	Defines the zone subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all zones in the working case. SID = a valid zone subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem zones to include (input; 1 by default). FLAG = 1 for only subsystem zones with at least one ac bus assigned to them. FLAG = 2 for subsystem zones with any equipment (ac buses, loads, induction machines, and/or dc buses) assigned to them.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the zone quantities desired (input; no default allowed): For the following, values are returned in MW and Mvar: ’PQLOAD’ Total complex load on buses in the zone. ’PQLOADLD’ Total complex load of loads in the zone. ’PQGEN’ Total complex generation on buses in the zone. 'PQINDMAC' Total complex power of induction machines on buses in the zone. 'PQINDMACMC' Total complex power of induction machines in the zone. ’PQLOSS’ Total complex losses assigned to the zone. ’PQINT’ Total complex net interchange from the zone. For the following, values are returned in units determined by the power output option setting: ’O_PQLOAD’ Total complex load on buses in the zone. ’O_PQLOADLD’ Total complex load of loads in the zone. ’O_PQGEN’ Total complex generation on buses in the zone. 'O_PQINDMAC' Total complex power of induction machines on buses in the zone. 'O_PQINDMACMC'Total complex power of induction machines in the zone. ’O_PQLOSS’ Total complex losses assigned to the zone. ’O_PQINT’ Total complex net interchange from the zone.
Integer DIM	Is the number of elements in each column of the array XARRAY (input; no default allowed). DIM must be as least as large as the value of ZONES returned by the GET_ZONE_COUNT API when it is called with the same values specified for SID and FLAG.
Complex XARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer ZONES	Is the number of elements in each column of XARRAY that are set by this API (output). ZONES is identical to the value of ZONES returned by the GET_ZONE_COUNT API when it is called with the same values specified for SID and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid FLAG value. IERR = 4 Invalid NSTR value. IERR = 5 DIM, and hence the size of XARRAY, is not large enough. IERR = 6 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.26.5 aZoneChar

Use this API to return an array of character values for subsystem zones.

Python syntax:

```
ierr, carray = azonechar(sid, flag, string)
```

Fortran syntax:

```
CALL GET_ZONE_CHAR(SID,FLAG,NSTR,STRING,DIM,CARRAY,ZONES,IERR)
```

where:

Integer SID	Defines the zone subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API to assume a subsystem containing all zones in the working case. SID = a valid zone subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem zones to include (input; 1 by default). FLAG = 1 for only subsystem zones with at least one ac bus assigned to them. FLAG = 2 for subsystem zones with any equipment (ac buses, loads, induction machines, and/or dc buses) assigned to them.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the zone quantities desired (input; no default allowed): 'ZONENAME' Zone name (12 characters).
Integer DIM	Is the number of elements in each column of the array CARRAY (input; no default allowed). DIM must be as least as large as the value of ZONES returned by the GET_ZONE_COUNT API when it is called with the same values specified for SID and FLAG.
Character CARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer ZONES	Is the number of elements in each column of CARRAY that are set by this API (output). ZONES is identical to the value of ZONES returned by the GET_ZONE_COUNT API when it is called with the same values specified for SID and FLAG.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output):	
	IERR = 0	No error.
	IERR = 1	Working case is empty.
	IERR = 2	Invalid SID value.
	IERR = 3	Invalid FLAG value.
	IERR = 4	Invalid NSTR value.
	IERR = 5	DIM, and hence the size of CARRAY, is not large enough.
	IERR = 6	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.26.6 aZoneTypes

Use this API to return an array of character values indicating the data types corresponding to a set of specified STRING values that are valid input values to any of the data retrieval routines of the zone data family (aZoneInt, aZoneReal, aZoneCplx and aZoneChar).

Python syntax:

```
ierr, types = azonetypes(string)
```

Fortran syntax:

```
CALL GET_ZONE_TYPES(NSTR,STRING,TYPES,IERR)
```

where:

Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR strings that are valid STRING values for any of the data retrieval routines of the zone data family (input; no default allowed):
Character TYPES*1	Is an array of at least NSTR elements, each of which contains one of the following characters indicating the data type of the data associated with the corresponding element of STRING (output): 'I' Integer. 'R' Real. 'X' Complex. 'C' Character. '?' Invalid STRING value.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = n Position of the first element in STRING that contains a value that is not a valid STRING value for any of the data retrieval routines in the zone data family (1 ≤ n ≤ NSTR). IERR = 99 Invalid NSTR value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.27 Induction Machine Bus Data

The induction machine bus data family of subsystem data retrieval API routines returns arrays of data for buses in a specified subsystem. One of the input items to each API routine allows the user to elect either to retrieve data for all subsystem buses, or to restrict the array entries to data for a subset of buses according to their bus and/or induction machine configuration and statuses.

Data for subsystem buses is returned from all API routines in the induction machine bus data family in the same order: ascending bus number order or alphabetic extended bus name order, according to the bus output option setting in effect at the time the API routine is referenced.

8.27.1 aIndMacBusCount

Use this API routine to obtain the number of array entries required to accommodate the data to be returned by the remaining members of the induction machine bus data family.

Python syntax:

```
ierr, buses = aindmacbuscount(sid, flag)
```

Fortran syntax:

```
CALL GET_IND_MAC_BUS_COUNT(SID, FLAG, BUSES, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API routine to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem buses to include (input; 1 by default).
	FLAG = 1 — for all in-service subsystem buses (type code 1, 2 or 3) with at least one in-service induction machine.
	FLAG = 2 — for all in-service subsystem buses (type code 1, 2 or 3) with at least one induction machine, even if all induction machines at the bus are out-of-service.
	FLAG = 3 — for all in-service subsystem buses (type code 1, 2 or 3), including those that have no induction machines at the bus.
	FLAG = 4 — for all subsystem buses with at least one induction machine, even if all induction machines at the bus are out-of-service.
	FLAG = 5 — for all subsystem buses, including those that have no induction machines at the bus.
	FLAG = 10 — for all in-service subsystem buses (type code 1, 2 or 3) with at least one in-service induction generator.
	FLAG = 11 — for all in-service subsystem buses (type code 1, 2 or 3) with at least one induction generator, even if all induction generators at the bus are out-of-service.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	FLAG = 12	for all subsystem buses with at least one induction generator, even if all induction generators at the bus are out-of-service.
	FLAG = 20	for all in-service subsystem buses (type code 1, 2 or 3) with at least one in-service induction motor.
	FLAG = 21	for all in-service subsystem buses (type code 1, 2 or 3) with at least one induction motor, even if all induction motors at the bus are out-of-service.
	FLAG = 22	for all subsystem buses with at least one induction motor, even if all induction motors at the bus are out-of-service.
Integer BUSES	Is the number of buses in the subsystem indicated by SID that meet the editing criteria indicated by FLAG (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error.
	IERR = 1	Working case is empty; BUSES returned as 0.
	IERR = 2	Invalid SID value; BUSES returned as 0.
	IERR = 3	Invalid FLAG value; BUSES returned as 0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.27.2 aIndMacBusInt

Use this API routine to return an array of integer values for subsystem buses.

Python syntax:

```
ierr, iarray = aindmacbusint(sid, flag, string)
```

Fortran syntax:

```
CALL GET_IND_MAC_BUS_INTEGER(SID, FLAG, NSTR, STRING, DIM, IARRAY,
BUSES, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API routine to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem buses to include (input; 1 by default).
	FLAG = 1 — for all in-service subsystem buses (type code 1, 2 or 3) with at least one in-service induction machine.
	FLAG = 2 — for all in-service subsystem buses (type code 1, 2 or 3) with at least one induction machine, even if all induction machines at the bus are out-of-service.
	FLAG = 3 — for all in-service subsystem buses (type code 1, 2 or 3), including those that have no induction machines at the bus.
	FLAG = 4 — for all subsystem buses with at least one induction machine, even if all induction machines at the bus are out-of-service.
	FLAG = 5 — for all subsystem buses, including those that have no induction machines at the bus.
	FLAG = 10 — for all in-service subsystem buses (type code 1, 2 or 3) with at least one in-service induction generator.
	FLAG = 11 — for all in-service subsystem buses (type code 1, 2 or 3) with at least one induction generator, even if all induction generators at the bus are out-of-service.
	FLAG = 12 — for all subsystem buses with at least one induction generator, even if all induction generators at the bus are out-of-service.
	FLAG = 20 — for all in-service subsystem buses (type code 1, 2 or 3) with at least one in-service induction motor.
	FLAG = 21 — for all in-service subsystem buses (type code 1, 2 or 3) with at least one induction motor, even if all induction motors at the bus are out-of-service.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	FLAG = 22	for all subsystem buses with at least one induction motor, even if all induction motors at the bus are out-of-service.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).	
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or induction machine quantities desired (input; no default allowed):	
	'NUMBER'	Bus number.
	'TYPE'	Bus type code.
	'AREA'	Bus area number.
	'ZONE'	Bus zone number.
	'OWNER'	Bus owner number.
	'DUMMY'	Returns 1 if the bus is a dummy bus of a multisection line, or 0 if it is not.
	'STATUS'	Inferred induction machine status. Returns 1 if the bus type code is 1, 2 or 3, and at least one in-service induction machine is present, or 0 otherwise.
Integer DIM	Is the number of elements in each column of the array IARRAY (input; no default allowed). DIM must be as least as large as the value of BUSES returned by the aIndMacBusCount API routine when it is called with the same values specified for SID and FLAG.	
Integer IARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).	
Integer BUSES	Is the number of elements in each column of IARRAY that are set by this API routine (output). BUSES is identical to the value of BUSES returned by the aIndMacBusCount API routine when it is called with the same values specified for SID and FLAG.	
Integer IERR	Is the error code (output):	
	IERR = 0	No error.
	IERR = 1	Working case is empty.
	IERR = 2	Invalid SID value.
	IERR = 3	Invalid FLAG value.
	IERR = 4	Invalid NSTR value.
	IERR = 5	DIM, and hence the size of IARRAY, is not large enough.
	IERR = 6	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.27.3 aIndMacBusReal

Use this API routine to return an array of real values for subsystem buses.

Python syntax:

```
ierr, rarray = aindmacbusreal(sid, flag, string)
```

Fortran syntax:

```
CALL GET_IND_MAC_BUS_REAL(SID,FLAG,NSTR,STRING,DIM,RARRAY,BUSES,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API routine to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem buses to include (input; 1 by default).
	FLAG = 1 — for all in-service subsystem buses (type code 1, 2 or 3) with at least one in-service induction machine.
	FLAG = 2 — for all in-service subsystem buses (type code 1, 2 or 3) with at least one induction machine, even if all induction machines at the bus are out-of-service.
	FLAG = 3 — for all in-service subsystem buses (type code 1, 2 or 3), including those that have no induction machines at the bus.
	FLAG = 4 — for all subsystem buses with at least one induction machine, even if all induction machines at the bus are out-of-service.
	FLAG = 5 — for all subsystem buses, including those that have no induction machines at the bus.
	FLAG = 10 — for all in-service subsystem buses (type code 1, 2 or 3) with at least one in-service induction generator.
	FLAG = 11 — for all in-service subsystem buses (type code 1, 2 or 3) with at least one induction generator, even if all induction generators at the bus are out-of-service.
	FLAG = 12 — for all subsystem buses with at least one induction generator, even if all induction generators at the bus are out-of-service.
	FLAG = 20 — for all in-service subsystem buses (type code 1, 2 or 3) with at least one in-service induction motor.
	FLAG = 21 — for all in-service subsystem buses (type code 1, 2 or 3) with at least one induction motor, even if all induction motors at the bus are out-of-service.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	FLAG = 22 — for all subsystem buses with at least one induction motor, even if all induction motors at the bus are out-of-service.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or induction machine quantities desired (input; no default allowed): 'BASE' — Bus base voltage, in kV. 'PU' — Actual bus voltage magnitude, in pu. 'KV' — Actual bus voltage magnitude, in kV (in pu if base voltage is 0.0). 'ANGLE' — Bus voltage phase angle, in radians. 'ANGLED' — Bus voltage phase angle, in degrees. For the following, values are returned in MVA: 'TOTAL' — Flow from the bus to the in-service induction machines at the bus. 'MISMATCH' — Bus mismatch (0.0 if bus type code > 3). For the following, values are returned in units determined by the power output option setting: 'O_TOTAL' — Flow from the bus to the in-service induction machines at the bus. 'O_MISMATCH' — Bus mismatch (0.0 if bus type code > 3).
Integer DIM	Is the number of elements in each column of the array RARRAY (input; no default allowed). DIM must be as least as large as the value of BUSES returned by the aIndMacBusCount API routine when it is called with the same values specified for SID and FLAG.
Real RARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BUSES	Is the number of elements in each column of RARRAY that are set by this API routine (output). BUSES is identical to the value of BUSES returned by the aIndMacBusCount API routine when it is called with the same values specified for SID and FLAG.
Integer IERR	Is the error code (output): IERR = 0 — No error. IERR = 1 — Working case is empty. IERR = 2 — Invalid SID value. IERR = 3 — Invalid FLAG value. IERR = 4 — Invalid NSTR value. IERR = 5 — DIM, and hence the size of RARRAY, is not large enough. IERR = 6 — Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.27.4 aIndMacBusCplx

Use this API routine to return an array of complex values for subsystem buses.

Python syntax:

```
ierr, xarray = aindmacbuscplx(sid, flag, string)
```

Fortran syntax:

```
CALL GET_IND_MAC_BUS_COMPLEX(SID, FLAG, NSTR, STRING, DIM, XARRAY,
BUSES, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API routine to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.	
Integer FLAG	Is a flag indicating which subsystem buses to include (input; 1 by default).	
	FLAG = 1	for all in-service subsystem buses (type code 1, 2 or 3) with at least one in-service induction machine.
	FLAG = 2	for all in-service subsystem buses (type code 1, 2 or 3) with at least one induction machine, even if all induction machines at the bus are out-of-service.
	FLAG = 3	for all in-service subsystem buses (type code 1, 2 or 3), including those that have no induction machines at the bus.
	FLAG = 4	for all subsystem buses with at least one induction machine, even if all induction machines at the bus are out-of-service.
	FLAG = 5	for all subsystem buses, including those that have no induction machines at the bus.
	FLAG = 10	for all in-service subsystem buses (type code 1, 2 or 3) with at least one in-service induction generator.
	FLAG = 11	for all in-service subsystem buses (type code 1, 2 or 3) with at least one induction generator, even if all induction generators at the bus are out-of-service.
	FLAG = 12	for all subsystem buses with at least one induction generator, even if all induction generators at the bus are out-of-service.
	FLAG = 20	for all in-service subsystem buses (type code 1, 2 or 3) with at least one in-service induction motor.
	FLAG = 21	for all in-service subsystem buses (type code 1, 2 or 3) with at least one induction motor, even if all induction motors at the bus are out-of-service.
	FLAG = 22	for all subsystem buses with at least one induction motor, even if all induction motors at the bus are out-of-service.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or induction machine quantities desired (input; no default allowed): 'VOLTAGE' Actual bus voltage, in pu, rectangular coordinates. For the following, values are returned in MW and Mvar: 'TOTAL' Flow from the bus to the in-service induction machines at the bus. 'MISMATCH' Bus mismatch (0.0 if bus type code > 3). For the following, values are returned in units determined by the power output option setting: 'O_TOTAL' Flow from the bus to the in-service induction machines at the bus. 'O_MISMATCH' Bus mismatch (0.0 if bus type code > 3).
Integer DIM	Is the number of elements in each column of the array XARRAY (input; no default allowed). DIM must be as least as large as the value of BUSES returned by the aIndMacBusCount API routine when it is called with the same values specified for SID and FLAG.
Complex XARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer BUSES	Is the number of elements in each column of XARRAY that are set by this API routine (output). BUSES is identical to the value of BUSES returned by the aIndMacBusCount API routine when it is called with the same values specified for SID and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid FLAG value. IERR = 4 Invalid NSTR value. IERR = 5 DIM, and hence the size of XARRAY, is not large enough. IERR = 6 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.27.5 aIndMacBusChar

Use this API routine to return an array of character values for subsystem buses.

Python syntax:

```
ierr, carray = aindmacbuschar(sid, flag, string)
```

Fortran syntax:

```
CALL GET_IND_MAC_BUS_CHAR(SID,FLAG,NSTR,STRING,DIM,CARRAY,BUSES,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API routine to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem buses to include (input; 1 by default).
	FLAG = 1 — for all in-service subsystem buses (type code 1, 2 or 3) with at least one in-service induction machine.
	FLAG = 2 — for all in-service subsystem buses (type code 1, 2 or 3) with at least one induction machine, even if all induction machines at the bus are out-of-service.
	FLAG = 3 — for all in-service subsystem buses (type code 1, 2 or 3), including those that have no induction machines at the bus.
	FLAG = 4 — for all subsystem buses with at least one induction machine, even if all induction machines at the bus are out-of-service.
	FLAG = 5 — for all subsystem buses, including those that have no induction machines at the bus.
	FLAG = 10 — for all in-service subsystem buses (type code 1, 2 or 3) with at least one in-service induction generator.
	FLAG = 11 — for all in-service subsystem buses (type code 1, 2 or 3) with at least one induction generator, even if all induction generators at the bus are out-of-service.
	FLAG = 12 — for all subsystem buses with at least one induction generator, even if all induction generators at the bus are out-of-service.
	FLAG = 20 — for all in-service subsystem buses (type code 1, 2 or 3) with at least one in-service induction motor.
	FLAG = 21 — for all in-service subsystem buses (type code 1, 2 or 3) with at least one induction motor, even if all induction motors at the bus are out-of-service.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	FLAG = 22	for all subsystem buses with at least one induction motor, even if all induction motors at the bus are out-of-service.
Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).	
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or induction machine quantities desired (input; no default allowed):	
	'NAME'	Bus name (12 characters).
	'EXNAME'	Extended bus name (18 characters).
Integer DIM	Is the number of elements in each column of the array CARRAY (input; no default allowed). DIM must be as least as large as the value of BUSES returned by the aIndMacBusCount API routine when it is called with the same values specified for SID and FLAG.	
Character CARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).	
Integer BUSES	Is the number of elements in each column of CARRAY that are set by this API routine (output). BUSES is identical to the value of BUSES returned by the aIndMacBusCount API routine when it is called with the same values specified for SID and FLAG.	
Integer IERR	Is the error code (output):	
	IERR = 0	No error.
	IERR = 1	Working case is empty.
	IERR = 2	Invalid SID value.
	IERR = 3	Invalid FLAG value.
	IERR = 4	Invalid NSTR value.
	IERR = 5	DIM, and hence the size of CARRAY, is not large enough.
	IERR = 6	Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.27.6 aIndMacBusTypes

Use this API routine to return an array of character values indicating the data types corresponding to a set of specified STRING values that are valid input values to any of the data retrieval routines of the induction machine bus data family (aIndMacBusInt, aIndMacBusReal, aIndMacBusCplx and aIndMacBusChar).

Python syntax:

```
ierr, types = aindmacbustypes(string)
```

Fortran syntax:

```
CALL GET_IND_MAC_BUS_TYPES(NSTR,STRING,TYPES,IERR)
```

where:

Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR strings that are valid STRING values for any of the data retrieval routines of the induction machine bus data family (input; no default allowed):
Character TYPES*1	Is an array of at least NSTR elements, each of which contains one of the following characters indicating the data type of the data associated with the corresponding element of STRING (output): 'I' Integer. 'R' Real. 'X' Complex. 'C' Character. '?' Invalid STRING value.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = n Position of the first element in STRING that contains a value that is not a valid STRING value for any of the data retrieval routines in the induction machine bus data family (1 ≤ n ≤ NSTR). IERR = 99 Invalid NSTR value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.28 Induction Machine Data

The induction machine data family of subsystem data retrieval API routines returns arrays of data for induction machines in a specified subsystem. One of the input items to each API routine allows the user to elect either to retrieve data for all subsystem induction machines, or to restrict the array entries to data for a subset of induction machines according to their bus and/or induction machine configuration and statuses.

Data for subsystem induction machines is returned from all API routines in the induction machine data family in the same order: ascending bus number order or alphabetic extended bus name order, according to the bus output option setting in effect at the time the API routine is referenced; multiple induction machines at the same bus are in ascending induction machine identifier order.

8.28.1 aIndMacCount

Use this API routine to obtain the number of array entries required to accommodate the data to be returned by the remaining members of the induction machine data family.

Python syntax:

```
ierr, machs = aindmaccount(sid, flag)
```

Fortran syntax:

```
CALL GET_IND_MAC_COUNT(SID, FLAG, MACHS, IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API routine to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem induction machines to include (input; 1 by default).
	FLAG = 1 — for all in-service induction machines at in-service subsystem buses (type code 1, 2 or 3).
	FLAG = 2 — for all induction machines at in-service subsystem buses (type code 1, 2 or 3).
	FLAG = 3 — for all in-service induction machines at subsystem buses, even if the bus to which it is connected is out-of-service (type code 4).
	FLAG = 4 — for all induction machines at all subsystem buses.
	FLAG = 11 — for all in-service induction generators at in-service subsystem buses (type code 1, 2 or 3).
	FLAG = 12 — all induction generators at in-service subsystem buses (type code 1, 2 or 3).
	FLAG = 13 — for all in-service induction generators at subsystem buses, even if the bus to which it is connected is out-of-service (type code 4).
	FLAG = 14 — for all induction generators at all subsystem buses.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

	FLAG = 21	for all in-service induction motors at in-service subsystem buses (type code 1, 2 or 3).
	FLAG = 22	all induction motors at in-service subsystem buses (type code 1, 2 or 3).
	FLAG = 23	for all in-service induction motors at subsystem buses, even if the bus to which it is connected is out-of-service (type code 4).
	FLAG = 24	for all induction motors at all subsystem buses.
Integer MACHS	Is the number of induction machines in the subsystem indicated by SID that meet the editing criteria indicated by FLAG (output).	
Integer IERR	Is the error code (output):	
	IERR = 0	No error.
	IERR = 1	Working case is empty; MACHS returned as 0.
	IERR = 2	Invalid SID value; MACHS returned as 0.
	IERR = 3	Invalid FLAG value; MACHS returned as 0.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.28.2 aIndMacInt

Use this API routine to return an array of integer values for subsystem induction machines.

Python syntax:

```
ierr, iarray = aindmacint(sid, flag, string)
```

Fortran syntax:

```
CALL GET_IND_MAC_INTEGER(SID,FLAG,NSTR,STRING,DIM,IARRAY,MACHS,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API routine to assume a sub-system containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem induction machines to include (input; 1 by default).
	FLAG = 1 — for all in-service induction machines at in-service subsystem buses (type code 1, 2 or 3).
	FLAG = 2 — for all induction machines at in-service subsystem buses (type code 1, 2 or 3).
	FLAG = 3 — for all in-service induction machines at subsystem buses, even if the bus to which it is connected is out-of-service (type code 4).
	FLAG = 4 — for all induction machines at all subsystem buses.
	FLAG = 11 — for all in-service induction generators at in-service subsystem buses (type code 1, 2 or 3).
	FLAG = 12 — all induction generators at in-service subsystem buses (type code 1, 2 or 3).
	FLAG = 13 — for all in-service induction generators at subsystem buses, even if the bus to which it is connected is out-of-service (type code 4).
	FLAG = 14 — for all induction generators at all subsystem buses.
	FLAG = 21 — for all in-service induction motors at in-service subsystem buses (type code 1, 2 or 3).
	FLAG = 22 — all induction motors at in-service subsystem buses (type code 1, 2 or 3).
	FLAG = 23 — for all in-service induction motors at subsystem buses, even if the bus to which it is connected is out-of-service (type code 4).
	FLAG = 24 — for all induction motors at all subsystem buses.
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)

Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or induction machine quantities desired (input; no default allowed):

'NUMBER'	Bus number.
'AREA'	Machine area assignment.
'ZONE'	Machine zone assignment.
'OWNER'	Machine owner assignment.
'STATUS'	Machine status.
'STANDARDCODE'	Machine standard code (1 for NEMA, 2 for IEC).
'DESIGNCODE'	Machine design code (0 through 5).
'PSETCODE'	Power setpoint code (1 for mechanical 2 for electrical).
'BASECODE'	Machine base power code (1 for mechanical 2 for electrical).
'TRQVARCODE'	Mechanical torque variation code (1 for simple power law, 2 for WECC model).
'CZG'	Grounding impedance data input/outout (I/O) code (1 for per unit, 2 for ohms).
'CGR'	Grounding code (0 for an ungrounded machine, 1 for a grounded machine).

Integer DIM

Is the number of elements in each column of the array IARRAY (input; no default allowed).

DIM must be as least as large as the value of MACHS returned by the aIndMacCount API routine when it is called with the same values specified for SID and FLAG.

Integer IARRAY

Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer MACHS

Is the number of elements in each column of IARRAY that are set by this API routine (output).

MACHS is identical to the value of MACHS returned by the aIndMacCount API routine when it is called with the same values specified for SID and FLAG.

Integer IERR

Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid FLAG value.
IERR = 4	Invalid NSTR value.
IERR = 5	DIM, and hence the size of IARRAY, is not large enough.
IERR = 6	Invalid STRING value.
IERR = 7	Sequence data not in case (when STRING = 'CZG' or 'CGR').

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.28.3 aIndMacReal

Use this API routine to return an array of real values for subsystem induction machines.

Python syntax:

```
ierr, rarray = aindmacreal(sid, flag, string)
```

Fortran syntax:

```
CALL GET_IND_MAC_REAL(SID,FLAG,NSTR,STRING,DIM,RARRAY,MACHS,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API routine to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem induction machines to include (input; 1 by default). FLAG = 1 — for all in-service induction machines at in-service subsystem buses (type code 1, 2 or 3). FLAG = 2 — for all induction machines at in-service subsystem buses (type code 1, 2 or 3). FLAG = 3 — for all in-service induction machines at subsystem buses, even if the bus to which it is connected is out-of-service (type code 4). FLAG = 4 — for all induction machines at all subsystem buses. FLAG = 11 — for all in-service induction generators at in-service subsystem buses (type code 1, 2 or 3). FLAG = 12 — all induction generators at in-service subsystem buses (type code 1, 2 or 3). FLAG = 13 — for all in-service induction generators at subsystem buses, even if the bus to which it is connected is out-of-service (type code 4). FLAG = 14 — for all induction generators at all subsystem buses. FLAG = 21 — for all in-service induction motors at in-service subsystem buses (type code 1, 2 or 3). FLAG = 22 — all induction motors at in-service subsystem buses (type code 1, 2 or 3). FLAG = 23 — for all in-service induction motors at subsystem buses, even if the bus to which it is connected is out-of-service (type code 4). FLAG = 24 — for all induction motors at all subsystem buses.
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)

Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or induction machine quantities desired (input; no default allowed):

'MBASE'	Machine base power.
'RATEKV'	Machine rated voltage.
'PSETPOINT'	Power setpoint (positive for a motor).
'RA'	Armature resistance.
'XA'	Armature leakage reactance.
'XM'	Magnetizing reactance.
'R1'	First cage resistance.
'X1'	First cage reactance.
'R2'	Second cage resistance.
'X2'	Second cage reactance.
'X3'	Third reactance.
'E1'	First saturation voltage.
'SE1'	SE(E1).
'E2'	Second saturation voltage.
'SE2'	SE(E2).
'IA1'	Starting stator current for applying stator leakage reactance saturation.
'IA2'	Stator current at which the stator leakage reactance is fully saturated.
'XAMULT'	Xa multiplier at full leakage reactance saturation.
'TRQA'	Load torque variation constant A for WECC model.
'TRQB'	Load torque variation constant B for WECC model.
'TRQD'	Load torque variation constant D for both models.
'TRQE'	Load torque variation constant E for WECC model.
'SLIP'	Slip.
'H'	Machine inertia.
'ILKDOVERIRAT'	Ratio of locked rotor current to rated current.
'ROVERX'	Ratio of Rpos to Xpos.
'RZERO'	Machine zero sequence fault analysis resistance.
'XZERO'	Machine zero sequence fault analysis reactance.
'RGROUND'	Machine zero sequence grounding resistance.
'XGROUND'	Machine zero sequence grounding reactance.
For the following, values are returned in MVA:	
'P'	Active power flow from the bus to the machine.
'Q'	Reactive power flow from the bus to the machine.
'MVA'	MVA flow from the bus to the machine.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

For the following, values are returned in units determined by the power output option setting:

'O_P'	Active power flow from the bus to the machine.
'O_Q'	Reactive power flow from the bus to the machine.
'O_MVA'	MVA flow from the bus to the machine.

Integer DIM

Is the number of elements in each column of the array RARRAY (input; no default allowed).

DIM must be as least as large as the value of MACHS returned by the aIndMacCount API routine when it is called with the same values specified for SID and FLAG.

Real RARRAY

Is an array of at least NSTR columns, each of which contains DIM elements (output).

Integer MACHS

Is the number of elements in each column of RARRAY that are set by this API routine (output).

MACHS is identical to the value of MACHS returned by the aIndMacCount API routine when it is called with the same values specified for SID and FLAG.

Integer IERR

Is the error code (output):

IERR = 0	No error.
IERR = 1	Working case is empty.
IERR = 2	Invalid SID value.
IERR = 3	Invalid FLAG value.
IERR = 4	Invalid NSTR value.
IERR = 5	DIM, and hence the size of RARRAY, is not large enough.
IERR = 6	Invalid STRING value.
IERR = 7	Sequence data not in case (when STRING = 'ILKDOVERIRAT', 'ROVERX', 'RZERO', 'XZERO', 'XZERO', 'RGROUND' or 'XGROUND').

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.28.4 aIndMacCplx

Use this API routine to return an array of complex values for subsystem induction machines.

Python syntax:

```
ierr, xarray = aindmaccplx(sid, flag, string)
```

Fortran syntax:

```
CALL GET_IND_MAC_COMPLEX(SID,FLAG,NSTR,STRING,DIM,XARRAY,MACHS,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API routine to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem induction machines to include (input; 1 by default).
	FLAG = 1 — for all in-service induction machines at in-service subsystem buses (type code 1, 2 or 3).
	FLAG = 2 — for all induction machines at in-service subsystem buses (type code 1, 2 or 3).
	FLAG = 3 — for all in-service induction machines at subsystem buses, even if the bus to which it is connected is out-of-service (type code 4).
	FLAG = 4 — for all induction machines at all subsystem buses.
	FLAG = 11 — for all in-service induction generators at in-service subsystem buses (type code 1, 2 or 3).
	FLAG = 12 — all induction generators at in-service subsystem buses (type code 1, 2 or 3).
	FLAG = 13 — for all in-service induction generators at subsystem buses, even if the bus to which it is connected is out-of-service (type code 4).
	FLAG = 14 — for all induction generators at all subsystem buses.
	FLAG = 21 — for all in-service induction motors at in-service subsystem buses (type code 1, 2 or 3).
	FLAG = 22 — all induction motors at in-service subsystem buses (type code 1, 2 or 3).
	FLAG = 23 — for all in-service induction motors at subsystem buses, even if the bus to which it is connected is out-of-service (type code 4).
	FLAG = 24 — for all induction motors at all subsystem buses.
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or induction machine quantities desired (input; no default allowed): 'ZA' (RA + j XA). 'Z1' (R1 + j X1). 'Z2' (R2 + j X2). 'ZZERO' Machine zero sequence fault analysis impedance. 'ZGROUND' Machine zero sequence grounding impedance. For the following, values are returned in MW and Mvar: 'PQ' Complex power flow from the bus to the machine. For the following, values are returned in units determined by the power output option setting: 'O_PQ' Complex power flow from the bus to the machine.
Integer DIM	Is the number of elements in each column of the array XARRAY (input; no default allowed). DIM must be as least as large as the value of MACHS returned by the aIndMacCount API routine when it is called with the same values specified for SID and FLAG.
Complex XARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer MACHS	Is the number of elements in each column of XARRAY that are set by this API routine (output). MACHS is identical to the value of MACHS returned by the aIndMacCount API routine when it is called with the same values specified for SID and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid FLAG value. IERR = 4 Invalid NSTR value. IERR = 5 DIM, and hence the size of XARRAY, is not large enough. IERR = 6 Invalid STRING value. IERR = 7 Sequence data not in case (when STRING = 'ZZERO' or 'ZGROUND').

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.28.5 aIndMacChar

Use this API routine to return an array of character values for subsystem induction machines.

Python syntax:

```
ierr, carray = aindmacchar(sid, flag, string)
```

Fortran syntax:

```
CALL GET_IND_MAC_CHAR(SID,FLAG,NSTR,STRING,DIM,CARRAY,MACHS,IERR)
```

where:

Integer SID	Defines the bus subsystem to be used (input; -1 by default). SID = a negative value, to instruct the API routine to assume a subsystem containing all buses in the working case. SID = a valid bus subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined.
Integer FLAG	Is a flag indicating which subsystem induction machines to include (input; 1 by default).
	FLAG = 1 — for all in-service induction machines at in-service subsystem buses (type code 1, 2 or 3).
	FLAG = 2 — for all induction machines at in-service subsystem buses (type code 1, 2 or 3).
	FLAG = 3 — for all in-service induction machines at subsystem buses, even if the bus to which it is connected is out-of-service (type code 4).
	FLAG = 4 — for all induction machines at all subsystem buses.
	FLAG = 11 — for all in-service induction generators at in-service subsystem buses (type code 1, 2 or 3).
	FLAG = 12 — all induction generators at in-service subsystem buses (type code 1, 2 or 3).
	FLAG = 13 — for all in-service induction generators at subsystem buses, even if the bus to which it is connected is out-of-service (type code 4).
	FLAG = 14 — for all induction generators at all subsystem buses.
	FLAG = 21 — for all in-service induction motors at in-service subsystem buses (type code 1, 2 or 3).
	FLAG = 22 — all induction motors at in-service subsystem buses (type code 1, 2 or 3).
	FLAG = 23 — for all in-service induction motors at subsystem buses, even if the bus to which it is connected is out-of-service (type code 4).
	FLAG = 24 — for all induction motors at all subsystem buses.
Integer NSTR	Is the number of elements in STRING ($1 \leq NSTR \leq 50$) (input; no default allowed).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR of the following strings indicating the bus and/or induction machine quantities desired (input; no default allowed): 'ID' Induction machine identifier (2 characters). 'NAME' Bus name (12 characters). 'EXNAME' Extended bus name (18 characters).
Integer DIM	Is the number of elements in each column of the array CARRAY (input; no default allowed). DIM must be as least as large as the value of MACHS returned by the aIndMacCount API routine when it is called with the same values specified for SID and FLAG.
Character CARRAY	Is an array of at least NSTR columns, each of which contains DIM elements (output).
Integer MACHS	Is the number of elements in each column of CARRAY that are set by this API routine (output). MACHS is identical to the value of MACHS returned by the aIndMacCount API routine when it is called with the same values specified for SID and FLAG.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = 1 Working case is empty. IERR = 2 Invalid SID value. IERR = 3 Invalid FLAG value. IERR = 4 Invalid NSTR value. IERR = 5 DIM, and hence the size of CARRAY, is not large enough. IERR = 6 Invalid STRING value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

8.28.6 aIndMacTypes

Use this API routine to return an array of character values indicating the data types corresponding to a set of specified STRING values that are valid input values to any of the data retrieval routines of the induction machine data family (aIndMacInt, aIndMacReal, aIndMacCplx and aIndMacChar).

Python syntax:

```
ierr, types = aindmactypes(string)
```

Fortran syntax:

```
CALL GET_IND_MAC_TYPES(NSTR,STRING,TYPES,IERR)
```

where:

Integer NSTR	Is the number of elements in STRING (1 ≤ NSTR ≤ 50) (input; no default allowed).
Character STRING(NSTR)	Is an array of NSTR elements specifying NSTR strings that are valid STRING values for any of the data retrieval routines of the induction machine data family (input; no default allowed):
Character TYPES*1	Is an array of at least NSTR elements, each of which contains one of the following characters indicating the data type of the data associated with the corresponding element of STRING (output): 'I' Integer. 'R' Real. 'X' Complex. 'C' Character. '?' Invalid STRING value.
Integer IERR	Is the error code (output): IERR = 0 No error. IERR = n Position of the first element in STRING that contains a value that is not a valid STRING value for any of the data retrieval routines in the induction machine data family (1 ≤ n ≤ NSTR). IERR = 99 Invalid NSTR value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

SIEMENS

Chapter 9

Results Data Retrieval Modules

This chapter defines API routines that are used to retrieve results from the PSS®E solutions/calculations in the form of arrays. These errors can be used to produce customized reports or can be exported to a text file or an Excel spreadsheet.

These APIs do not record themselves when called with recording enabled.

Routines described in this chapter are not members of the psspy Python extension module.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

9.1 PSSARRAYS module

The PSSARRAYS module provides Python functions to retrieve PSS®E solution results in arrays. These arrays can then be used in a Python script for post-processing, to create customized reports for export to a text file or an Excel spreadsheet, etc.

These functions do not record themselves when called with recording enabled.

The syntax on the following pages assumes that the module has been imported by executing:

```
import pssarrays
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

9.1.1 ACCC_SUMMARY

This function returns AC Contingency monitored element labels, contingency labels etc.

Python syntax:

```
rlst = pssarrays.accc_summary(accfile)
```

where:

string	accfile	Is the name of Contingency Solution Output file (input, no default allowed), maximum string length=260

Returned object 'rlst' contains the following attributes:

integer	rlst.ierr	Is the error code =0 no error occurred =4 error opening accfile =5 error reading accfile =6 prerequisite requirements for function not met
integer	rlst.acccsize.nmline	Is the number of monitored branches
integer	rlst.acccsize.ninter	Is the number of monitored interfaces
integer	rlst.acccsize.ncase	Is the number of contingencies + 1 (for base case)
integer	rlst.acccsize.nmvbus	Is the number of voltage monitored buses
integer	rlst.acccsize.nmvrec	Is the number of voltage monitored records
integer	rlst.acccsize.nmvbusrec	Is the number of voltage monitored bus records
integer	rlst.acccsize.nbus	Is the number of buses in the case
integer	rlst.acccsize. ncntlshed	Is the number of load sheds due to dispatch and contingency
integer	rlst.acccsize. ntrplshed	Is the number of load sheds due to tripping
integer	rlst.acccsize. ncactlshed	Is the number of load sheds due to corrective actions
integer	rlst.acccsize. ncactgdisp	Is the number of generation dispatched due to corrective actions
integer	rlst.acccsize. ncactphsftr	Is the number of phase shifter changed due to corrective actions
string	rlst.casetitle.line1	Is the case short title line 1, maximum string length=60
string	rlst.casetitle.line2	Is the case short title line 2, maximum string length=60
string	rlst.file.acc	Is the name of contingency output (.acc) file, maximum string length=260
string	rlst.file.sav	Is the name of saved case (.sav) file, maximum string length=260
string	rlst.file.dfx	Is the name of distribution factor data (.dfx) file, maximum string length=260
string	rlst.file.sub	Is the name of subsystem definition data (.sub) file, maximum string length=260

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

string	rlst.file.mon	Is the name of monitored element data (.mon) file, maximum string length=260
string	rlst.file.con	Is the name of contingency description data (.con) file, maximum string length=260
string	rlst.file.thr	Is the name of load throwover data (.thr) file, maximum string length=260
string	rlst.file.inl	Is the name of unit inertia and governor data (.inl) file, maximum string length=260
string	rlst.file.trp	Is the name of tripping element data (.trp) file, maximum string length=260
string	rlst.melement	Is a list of length (nmline+ninter) of monitored branch and interface names, maximum string length=54
float	rlst.rating.a	Is a list of length (nmline+ninter) of monitored element rating A
float	rlst.rating.b	Is a list of length (nmline+ninter) of monitored element rating B
float	rlst.rating.c	Is a list of length (nmline+ninter) of monitored element rating C
string	rlst.mvbuslabel	Is a list of length (nmvbusrec) of monitored voltage bus label, maximum string length=25
string	rlst.mvreclabel	Is a list of length (nmvbusrec) of monitored voltage record label, maximum string length=20
float	rlst.mvrecmax	Is a list of length (nmvbusrec) of monitored voltage bus maximum
float	rlst.mvrecmin	Is a list of length (nmvbusrec) of monitored voltage bus minimum
string	rlst.mvrectype	Is a list of length (nmvbusrec) of monitored voltage record type (RANGE / DEVIATION), maximum string length=9
string	rlst.colabel	Is a list of length (ncase) of contingency labels, maximum string length=12
string	rlst.busname	Is a list of length (nbus) of extended bus names, maximum string length=25

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

9.1.2 ACCC_SOLUTION

This function returns ACCC post-contingency, post-tripping or post-corrective action solution monitored MVA flows, ampere flows and bus voltages for one contingency.

Python syntax:

```
rlst = pssarrays.accc_solution(accfile, colabel, stype, busmsm,
sysmsm)
```

where:

string	accfile	Is the name of Contingency Solution Output file (input, no default allowed), maximum string length=260
string	colabel	Is the name of contingency label to get ACCC solution for, only one label allowed (input, no default allowed), maximum string length=12
string	stype	Is the name of solution type to get (input, *contingency* by default), allowed values: *contingency*, *con*, *tripping*, *trp*, *caction*, *contingency action*, *cor*
float	busmsm	Is the value of bus mismatch tolerance in MVA (input, 0.5 MVA by default)
float	sysmsm	Is the value of system mismatch tolerance in MVA (input, 5.0 MVA by default)

Returned object 'rlst' contains the following attributes:

integer	rlst.ierr	Is the error code =0 no error occurred =4 error opening accfile =5 error reading accfile =6 prerequisite requirements for function not met
string	rlst.codesc	Is the list of length (up to 16) of contingency events description, maximum string length=152
boolean	rlst.cnvflag	Is the convergence status (True when converged)
string	rlst.cnvcond	Is the convergence condition description, maximum string length=34
integer	rlst.island	Is the number of islands
float	rlst.mvaworst	Is the worst MVA mismatch
float	rlst.mvatotal	Is the total MVA mismatch
float	rlst.volts	Is a list of length (nmvbusrec) of monitored bus voltage in pu
float	rlst.mvaflow	Is a list of length (ncase) of monitored branch and interface MVA flow
float	rlst.ampflow	Is a list of length (ncase) of monitored branch ampere flow
string	rlst.lshedbus	Is a list of extended bus names of the load shed buses, maximum string length=25, list length: = ncntlshed, for post-contingency solution = ntrplshed, for post-tripping solution = ncactlshed, for post-corrective action solution

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

float	rlst.loadshed	Is a list of loads shed in MW, list length: = ncntlshed, for post-contingency solution = ntrplshed, for post-tripping solution = 2 columns of ncactlshed, for post-corrective action solution, 1st column gives initial load in MW, and 2nd column gives load shed in MW.
string	rlst.gdispbus	Is a list of length (ncactgdisp) of extended bus names of the generation dispatch buses, maximum string length=25, available for post-corrective action solution only.
float	rlst.gendisp	Is a list of generation dispatches in MW, available for post-corrective action solution only, list length = 2 columns of ncactgdisp, 1st column gives initial generation in MW, and 2nd column gives generation dispatch in MW.
string	rlst.phsftr	Is a list of length (ncactphsftr) of phase shifter names, maximum string length=54, available for post-corrective action solution only.
float	rlst.phsftrang	Is a list of phase shifter angles in degrees, available for post-corrective action solution only, list length = 2 columns of ncactphsftr, 1st column gives initial angle in degrees, and 2nd column gives new angle in degrees.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

9.1.3 ACCC_SUMMARY_REPORT

This function generates text report of lists returned by accc_summary.

Python syntax:

```
ierr = pssarrays.accc_summary_report(accfile, rptfile)
```

where:

string	accfile	Is the name of Contingency Solution Output file (input, no default allowed), maximum string length=260
string	rptfile	Is the name of report file (input, PSS®E Report by default), maximum string length=260

returns:

integer	rlst.ierr	Is the error code	
		=0	no error occurred
		=4	error opening accfile
		=5	error reading accfile
		=6	prerequisite requirements for function not met

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

9.1.4 SENSITIVITY_FLOW_TO_MW

This API is used to calculate sensitivity factors of a branch flow to MW power at generator buses and MW power at load buses and return the results.

Python syntax:

```
rlist =pssarrays.sensitivity_flow_to_mw(options,value,brnckt,labels,
dfxfile)
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

where:

Integer OPTIONS

is an array of 8 integer elements (input), the value of each element is as follows:

(1) Is the from bus of the branch for which sensitivity factors are calculated (no default is allowed)

(2) Is the to bus of the branch (no default is allowed)

(3) Is the third bus connected by a three-winding transformer, or zero for a non-transformer branch or a two-winding transformer (0 by default)

(4) Is network model on which sensitivity analysis is based

0 => DC network model

1 => AC network model

(5) Is the branch flow type to which the sensitivity factors are calculated (If DC network is used, only the branch MW flow is valid for sensitivity analysis. 0 by default)

0 => MW flow

1 => MVar flow

2 => MVA flow

3 => Amps flow

(6) Is the flag to specify the transfer from the components to which sensitivity factors of the branch flow are calculated to the opposing subsystem (the flag is ignored if none of bus, generator and load sensitivity calculation flags is selected, 0 by default)

0 => Import, the positive transfer is from the components to the opposing subsystem

1 => Export, the positive transfer is from the opposing subsystem to the components

(7) Is the opposing subsystem type (0 by default)

0 =>Slack bus

1 =>Single bus

2 =>Subsystem

(8) Is ignored if opposing subsystem type is slack bus. It is a bus number when opposing subsystem type is a single bus; or one of the following seven dispatch modes if the opposing subsystem type is a subsystem (1 by default):

1 => Buses & participating factors from DFAX file for buses with positive MW machines

2 => Buses & participating factors from DFAX file for buses with positive MW constant MVA load

3 => Buses & participating factors from DFAX file for buses with either positive MW machines or positive MW constant MVA load

4 => Subsystem buses with positive MW constant MVA load in proportion to their MW load

5 => Subsystem buses with positive MW machines in proportion to their MW output

6 => Subsystem buses with positive MW machines in proportion to their MBASE

7 => Subsystem buses with positive MW machines in proportion to their reserves (Pmaxmc-Pgenmc)

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Real VALUE	Is mismatch tolerance for continuing sensitivity analysis in AC mode, ignored in DC mode (TOLN by default)
Character BRNCKT*2	Is the branch circuit ID (input; blank by default)
Character LABELS*32	Is an array of two elements specifying subsystem labels (input) LABELS (1) Is the label of the subsystem containing components to which the sensitivity factors of the branch flow are calculated (no default allowed) LABELS (2) Is the opposing subsystem label when the opposing subsystem type is a subsystem (blank by default)
Character DFXFILE*260	Object rlist Is returned object contains the following attributes (output): rlist.ngenbuses Is the number of generator buses rlist.genvalues.nam Is the list of generator bus names of buses to which sensitivity factors are calculated rlist.genvalues.pgen Is the list of Pgen of the generator buses rlist.genvalues.Pmax Is the list of Pmax of the generator buses rlist.genvalues.Pmin Is the list of Pmin of the generator buses rlist.genvalues.factor Is the list of the sensitivity factors of the branch flow to MW power at generator buses rlist.nloadbuses Is the number of load buses rlist.loadvalues.nam Is the list of load bus names of buses to which sensitivity factors are calculated rlist.loadvalues.factor Is the list of the sensitivity factors of the branch flow to MW power at load buses rlist.ierr Is the error return code 0 => No error 1 => Invalid OPTIONS value 2 => Invalid VALUE value 3 => The branch can not be found 4 => Generators are converted 5 => Buses in island(s) without a swing bus--use activity TREE 6 => Dispatch bus does not exist 7 => No Distribution Factor Data input file specified 8 => Subsystem containing components to which sensitivity factors are calculated is not specified 9=> Opposing subsystem is not specified 10 => Largest mismatch exceeds mismatch tolerance 11 => The branch bus does not exist 12 => The branch is out-of-service 13 => The branch is a zero impedance line 14 => The branch is a multi-section line but the Report Multi-Section line option is disabled 21=> file DFXFILE is not in the form of a PSS™E-25 or later DFAX file, run DFAX 22=> Error in linearizing Jacobian matrix with AC network model 51=> error opening Distribution Factor Data File 52=> prerequisite requirements for function not met

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

9.1.5 ACCC_SOLUTION_REPORT

This function generates text report of lists returned by accc_solution.

Python syntax:

```
ierr = pssarrays.accc_solution_report(accfile, colabels, stype,
busmsm, sysmsm, rating, rptfile)
```

where:

string	accfile	Is the name of Contingency Solution Output file (input, no default allowed), maximum string length=260
string	colabels	Is the list of contingency label names (input, all contingencies in accfile by default), maximum string length=12
string	stype	Is the name of solution type to get (input, *contingency* by default allowed), allowed values: *contingency*, *con*, *tripping*, *trp*, *caction*, *contingency action*, *cor*
float	busmsm	Is the value of bus mismatch tolerance in MVA (input, 0.5 MVA by default)
float	sysmsm	Is the value of system mismatch tolerance in MVA (input, 5.0 MVA by default)
string	rating	Is the name of rating ('a' or 'b' or 'c') to use for calculating percentage loading (input, 'a' by default), maximum string length=1
string	rptfile	Is the name of report file (input, PSS®E Report by default), maximum string length=260

returns:

integer	rlst.ierr	Is the error code	
		=0	no error occurred
		=4	error opening accfile
		=5	error reading accfile
		=6	prerequisite requirements for function not met

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

9.1.6 ACCC_VIOLATIONS_REPORT

This function generates text report of monitored flow and bus voltage violations of ACCC post-contingency, post-tripping or post-corrective action solution. For each monitored element flow and bus voltage, only the worst violation and contingency which results that violation are reported. That is each element violations is reported once and that violation is the worst violation.

Python syntax:

```
ierr = pssarrays.accc_violations_report
(accfile,stype,busmsm,sysmsm,rating,flowlimit,rptfile)
```

where:

string	accfile	Is the name of Contingency Solution Output file (input, no default allowed), maximum string length=260
string	stype	Is the name of solution type (input, *contingency* by default allowed), allowed values: *contingency*, *con*, *tripping*, *trp*, *caction*, *contingency action*, *cor*
float	busmsm	Is the value of bus mismatch tolerance in MVA (input, 0.5 MVA by default)
float	sysmsm	Is the value of system mismatch tolerance in MVA (input, 5.0 MVA by default)
string	rating	Is the name of rating ('a' or 'b' or 'c') to use for calculating percentage loading (input, 'a' by default), maximum string length=1
float	flowlimit	Is the value of percent of flow rating (input, 100% by default)
string	rptfile	Is the name of report file (input, PSS®E Report by default), maximum string length=260

returns:

integer	rlst.ierr	Is the error code	
		=0	no error occurred
		=4	error opening accfile
		=5	error reading accfile
		=6	prerequisite requirements for function not met

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

9.1.7 ASCC_CURRENTS

This function returns short circuit currents and Thevenin impedance in three-phase ac systems from API. Results are returned for bus faults only. Results for line out and line end faults are not returned.

Python syntax:

```
rlst = pssarrays.ascc_currents
(sid,all,flt3ph,fltlg,fltllg,fltll,linout,linend,voltop,genxop,
tpunty,dcload,zcorec,lnchrg,shntop,loadop,machpq,volts,relfil,fcdfil,
scfile)
```

int	sid	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default).
int	all	Is the *all buses* or selected subsystem flag (input; 1 by default) = 1 process all buses = 0 process only buses in subsystem SID
int	flt3ph	Is to report three phase faults (0 by default). = 0 omit = 1 include
int	fltlg	Is to report line-to-ground faults (0 by default). = 0 omit = 1 include
int	fltllg	Is to report line-line-to-ground faults (0 by default). = 0 omit = 1 include
int	fltll	Is to report line-to-line faults (0 by default). = 0 omit = 1 include
int	linout	Is to report line-out faults (0 by default). = 0 omit = 1 include
int	linend	Is to report line-end faults (0 by default). = 0 omit = 1 include
int	voltop	Is the bus voltage option (0 by default). = 0 use bus voltages from power flow solution = 1 set all bus voltages at specified value and at 0 deg = 2 set faulted bus voltage at specified value and at 0 deg

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

int	genxop	Is the synchronous machine reactance option, ignored if generators are already "converted" (0 by default). = 0 use subtransient reactance = 1 use transient reactance = 2 use synchronous reactance
int	tpunty	Is the transformer tap tatios and phase angle option (0 by default) = 0 leave tap ratios and phase shift angles unchanged = 1 set tap ratios to 1.0 pu and phase shift angles to 0 deg = 2 set tap ratios to 1.0 pu and phase shift angles unchanged = 3 set tap ratios unchanged and phase shift angles to 0 deg
int	dcload	Is the dc line and FACTS device option (0 by default) = 0 blocked = 1 represent as load
int	zcorec	Is the zero sequence transformer impedance correction option (0 by default) = 0 ignore = 1 apply
int	lnchrg	Is the line charging option (0 by default) = 0 leave unchanged = 1 set to 0.0 in the positive and negative sequences = 2 set to 0.0 in all sequences
int	shntop	Is the line shunts, fixed shunts, switched shunts and transformer magnetizing admittance option (0 by default) = 0 leave unchanged = 1 set to 0.0 in the positive and negative sequences = 2 set to 0.0 in all sequences
int	loadop	Is the load option (0 by default) = 0 leave unchanged = 1 set to 0.0 in the positive and negative sequences = 2 set to 0.0 in all sequences
int	machpq	Is the synchronous and asynchronous machines power output option (0 by default) = 0 use real and reactive power outputs from power flow solution = 1 set real and reactive power outputs to 0.0
int	volts	Is the user specified bus voltage value in pu (1.0 by default) This is used only when voltop = 1 or 2.
str	relfil	Is the Relay Output Data (.rel) file; blank for none (input; blank by default); maximum string length=260

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

str	fcdfil	Is the name of fault control (.fcd) file; blank for none (input; blank by default); maximum string length=260.
str	scfile	Is the ASCC_3 Results (.sc) file; blank for none (output; blank by default); maximum string length=260.

Returned object 'rlst' contains the following attributes:

Index i is the ith faulted bus.

int	rlst.ierr	Is the error code = 0, no error occurred = 1, invalid SID value or subsystem SID is not defined = 2, invalid ALL value = 3, invalid STATUS value = 4, line end fault invalid; bus table is full = 5, error opening RELFIL = 6, error opening FCDFIL = 7, prerequisite requirements for function not met = 8, no faults specified = 9, error opening SCFILE = 10, SCFILE error = 11, bus SC results not available. = 12, branch SC results not available. = 13, three winding transformer SC results not available. = 14, incorrect SCFILE. = 15, SCFILE not allowed.
str	rlst.scunit	Is the units of returned fault currents = pu = physical (r.m.s. amperes)
str	rlst.scfmt	Is the coordinates of returned fault currents = rectangular = polar
int	rlst.fltbus	Is a list of faulted bus numbers
complex	rlst.flt3ph[i].ia1	Is three phase fault, positive sequence current = I''k
complex	rlst.flt3ph[i].ia2	Is three phase fault, negative sequence current
complex	rlst.flt3ph[i].ia0	Is three phase fault, zero sequence current
complex	rlst.flt3ph[i].ia	Is three phase fault, Phase A current
complex	rlst.flt3ph[i].ib	Is three phase fault, Phase B current
complex	rlst.flt3ph[i].ic	Is three phase fault, Phase C current
complex	rlst.fltlg[i].ia1	Is line to ground fault, positive sequence current

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

complex	rlst.fltlg[i].ia2	Is line to ground fault, negative sequence current
complex	rlst.fltlg[i].ia0	Is line to ground fault, zero sequence current = I”k/3
complex	rlst.fltlg[i].ia	Is line to ground fault, Phase A current
complex	rlst.fltlg[i].ib	Is line to ground fault, Phase B current
complex	rlst.fltlg[i].ic	Is line to ground fault, Phase C current
complex	rlst.fltllg[i].ia1	Is line line to ground fault, positive sequence current
complex	rlst.fltllg[i].ia2	Is line line to ground fault, negative sequence current
complex	rlst.fltllg[i].ia0	Is line line to ground fault, zero sequence current = I”k/3
complex	rlst.fltllg[i].ia	Is line line to ground fault, Phase A current
complex	rlst.fltllg[i].ib	Is line line to ground fault, Phase B current
complex	rlst.fltllg[i].ic	Is line line to ground fault, Phase C current
complex	rlst.fltll[i].ia1	Is line to line fault, positive sequence current
complex	rlst.fltll[i].ia2	Is line to line fault, negative sequence current
complex	rlst.fltll[i].ia0	Is line to line fault, zero sequence current
complex	rlst.fltll[i].ia	Is line to line fault, Phase A current
complex	rlst.fltll[i].ib	Is line to line fault, Phase B current = I”k
complex	rlst.fltll[i].ic	Is line to line fault, Phase C current = -I”k
complex	rlst.thevz[i].z1	Is Thevenin impedance, positive sequence, in PU or OHM as set by SCUNIT
complex	rlst.thevz[i].z2	Is Thevenin impedance, negative sequence, in PU or OHM as set by SCUNIT
complex	rlst.thevz[i].z0	Is Thevenin impedance, zero sequence, in PU or OHM as set by SCUNIT
complex	rlst.thevzpu[i].z1	Is PU Thevenin impedance, positive sequence
complex	rlst.thevzpu[i].z2	Is PU Thevenin impedance, negative sequence
complex	rlst.thevzpu[i].z0	Is PU Thevenin impedance, zero sequence
complex	rlst.maxflt[i].ia1	Is maximum fault current, positive sequence current
complex	rlst.maxflt[i].ia2	Is maximum fault current, negative sequence current
complex	rlst.maxflt[i].ia0	Is maximum fault current, zero sequence current
complex	rlst.maxflt[i].ia	Is maximum fault current, Phase A current
complex	rlst.maxflt[i].ib	Is maximum fault current, Phase B current
complex	rlst.maxflt[i].ic	Is maximum fault current, Phase C current
str	rlst.maxfltdsc[i]	Is description of fault condition for maximum fault current

Note: Maximum fault current is the largest fault current among analyzed faults at a bus.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

9.1.8 IECS_CURRENTS

This function returns short circuit currents and Thevenin impedance in three-phase ac systems according to IEC 60909 standard (IECS_3 API). Results are returned for bus faults only. Results for line out and line end faults are not returned.

Python syntax:

```
rlst = pssarrays.iecs_currents(
sid,all,flt3ph,fltlg,fltllg,fltll,fltloc,linout,linend,tpunty,lnchrg,
shntop,dcload,zcorec,optnftrc,loadop,genxop,brktime,vfactorc,iecfil,
fcdfil,scfile)
```

where:

int	sid	Is a valid subsystem identifier. Valid subsystem identifiers range from 0 to 11. Subsystem SID must have been previously defined (input; 0 by default)
int	all	Is the *all buses* or specified subsystem flag (input; 1 by default). = 1 process all buses. = 0 process only buses in subsystem SID.
int	flt3ph	Is to report three phase faults, (0 by default). = 0 omit = 1 include
int	fltlg	Is to report line-to-ground faults (0 by default). = 0 omit = 1 include
int	fltllg	Is to report line-line-to-ground faults, (0 by default). = 0 omit = 1 include
int	fltll	Is to report line-to-line faults, (0 by default). = 0 omit = 1 include
int	fltloc	Is the location of fault (1 by default) =0 fault at network bus =1 fault at LV bus of Power Station Unit (PSU) =2 fault at auxiliary transformer (connected to PSU) LV bus
int	linout	Is to report line-out faults (0 by default). = 0 omit = 1 include
int	linend	Is to report line-end faults (0 by default). = 0 omit = 1 include

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

int	tpunty	Is the transformer tap ratios and phase angle option (0 by default) = 0 leave tap ratios and phase shift angels unchanged = 1 set tap ratios to 1.0 pu and phase shift angels to 0 deg = 2 set tap ratios to 1.0 pu and phase shift angels unchanged = 3 set tap ratios unchanged and phase shift angels to 0 deg
int	lnchrg	Is the line charging option (1 by default). = 0 leave unchanged = 1 set to 0.0 in the positive and negative sequences = 2 set to 0.0 in all sequences
int	shntop	Is the line shunts, fixed shunts, switched shunts and transformer magnetizing admittance option (1 by default). = 0 leave unchanged = 1 set to 0.0 in the positive and negative sequences = 2 set to 0.0 in all sequences
int	dcload	Is the dc line and FACTS device option (0 by default) = 0 blocked = 1 represent as load
int	zcorec	Is the zero sequence transformer impedance correction option (0 by default) = 0 ignore = 1 apply
int	optnftrc	Is the option for Voltage Factor C = 0, Voltage Factor C for maximum fault currents (default) = 1, Voltage Factor C for minimum fault currents = 2, Voltage Factor C as specified by 'vfactorc' value
int	loadop	Is the load option (1 by default) = 0 leave unchanged = 1 set to 0.0 in the positive and negative sequences = 2 set to 0.0 in all sequences
int	genxop	Is the synchronous machine reactance option, ignored if generators are already “converted” (0 by default). = 0 use subtransient reactance = 1 use transient reactance = 2 use synchronous reactance
float	brktime	Is the breaker contact parting time in seconds (5/BaseFrequency by default)
float	vfactorc	Is the user specified Voltage Factor C This is used only when optnftrc=3.
str	iecfil	Is the name of IEC data (.iec) file; blank for none (input; blank by default); maximum string length=260
str	fcdfil	Is the name of fault control (.fcd) file; blank for none (input; blank by default); maximum string length=260
str	scfile	Is the IECS_3 Results (.sc) file; blank for none (output; blank by default); maximum string length=260.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Returned object 'rlst' contains the following attributes::

Index i is the ith faulted bus.

int	rlst.ierr	Is the error code	
		= 0	no error occurred.
		= 1	invalid sid value or subsystem sid is not defined.
		= 2	invalid all value.
		= 3	invalid STATUS value.
		= 4	invalid brktime value.
		= 5	no faults specified.
		= 6	error opening iecfile.
		= 7	error opening fcdfile.
		= 8	prerequisite requirements for API are not met.
		= 9	error opening SCFILE
		= 10	SCFILE error
		= 11	bus SC results not available.
		= 12	branch SC results not available.
		= 13	three winding transformer SC results not available.
		= 14	incorrect SCFILE.
		= 15	SCFILE not allowed.
str	rlst.scunit	Is the units of returned fault currents	
		= pu	
		= physical (r.m.s. amperes)	
str	rlst.scfmt	Is the coordinates of returned fault currents	
		= rectangular	
		= polar	
int	rlst.fltbus	Is a list of faulted bus numbers	
complex	rlst.flt3ph[i].ia1	Is three phase fault, positive sequence current = I"k	
complex	rlst.flt3ph[i].ia2	Is three phase fault, negative sequence current	
complex	rlst.flt3ph[i].ia0	Is three phase fault, zero sequence current	
complex	rlst.flt3ph[i].ia	Is three phase fault, Phase A current	
complex	rlst.flt3ph[i].ib	Is three phase fault, Phase B current	
complex	rlst.flt3ph[i].ic	Is three phase fault, Phase C current	
complex	rlst.flt3ph[i].ipb	Is three phase fault, peak current - Method B, ip(B)	
complex	rlst.flt3ph[i].ipc	Is three phase fault, peak current - Method C, ip(C)	
complex	rlst.flt3ph[i].idc	Is three phase fault, dc component of asymmetrical breaking current, idc	
complex	rlst.flt3ph[i].ibsym	Is three phase fault, symmetrical breaking current (r.m.s.), ib(sym)	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

complex	rlst.flt3ph[i].ibuns	Is three phase fault, symmetrical breaking current (r.m.s.), ib(uns)
complex	rlst.fltlg[i].ia1	Is line to ground fault, positive sequence current
complex	rlst.fltlg[i].ia2	Is line to ground fault, negative sequence current
complex	rlst.fltlg[i].ia0	Is line to ground fault, zero sequence current = I''k/3
complex	rlst.fltlg[i].ia	Is line to ground fault, Phase A current
complex	rlst.fltlg[i].ib	Is line to ground fault, Phase B current
complex	rlst.fltlg[i].ic	Is line to ground fault, Phase C current
complex	rlst.fltlg[i].ipb	Is line to ground fault, peak current - Method B, ip(B)
complex	rlst.fltlg[i].ipc	Is line to ground fault, peak current - Method C, ip(C)
complex	rlst.fltlg[i].idc	Is line to ground fault dc component of asymmetrical breaking current
complex	rlst.fltlg[i].ibsym	Is line to ground fault, symmetrical breaking current (r.m.s.)
complex	rlst.fltlg[i].ibuns	Is line to ground fault, asymmetrical breaking current (r.m.s.)
complex	rlst.fltllg[i].ia1	Is line line to ground fault, positive sequence current
complex	rlst.fltllg[i].ia2	Is line line to ground fault, negative sequence current
complex	rlst.fltllg[i].ia0	Is line line to ground fault, zero sequence current = I"k / 3
complex	rlst.fltllg[i].ia	Is line line to ground fault, Phase A Current
complex	rlst.fltllg[i].ib	Is line line to ground fault, Phase B Current
complex	rlst.fltllg[i].ic	Is line line to ground fault, Phase C Current
complex	rlst.fltllg[i].ipb	Is line line to ground fault, peak current - Method B, ip(B)
complex	rlst.fltllg[i].ipc	Is line line to ground fault, peak current - Method C, ip(C)
complex	rlst.fltllg[i].idc	Is line line to ground fault dc component of asymmetrical breaking current, idc
complex	rlst.fltllg[i].ibsym	Is line line to ground fault, symmetrical breaking current (r.m.s.), ib(sym)
complex	rlst.fltllg[i].ibuns	Is line line to ground fault, asymmetrical breaking current (r.m.s.), ib(uns)
complex	rlst.fltll[i].ia1	Is line to line fault, positive sequence current
complex	rlst.fltll[i].ia2	Is line to line fault, negative sequence current
complex	rlst.fltll[i].ia0	Is line to line fault, zero sequence current
complex	rlst.fltll[i].ia	Is line to line fault, Phase A current
complex	rlst.fltll[i].ib	Is line to line fault, Phase B current = I''k
complex	rlst.fltll[i].ic	Is line to line fault, Phase C current = -I''k
complex	rlst.fltll[i].ipb	Is line to line fault, peak current - Method B, ip(B)
complex	rlst.fltll[i].ipc	Is line to line fault, peak current - Method C, ip(C)
complex	rlst.fltll[i].idc	Is line to line fault, dc component of asymmetrical breaking current, idc
complex	rlst.fltll[i].ibsym	Is line to line fault, symmetrical breaking current (r.m.s.), ib(sym)
complex	rlst.fltll[i].ibuns	Is line to line fault, asymmetrical breaking current (r.m.s.), ib(uns)
complex	rlst.thevz[i].z1	Is PU Thevenin impedance, positive sequence

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

complex	rlst.thevzpu[i].z2	Is PU Thevenin impedance, negative sequence
complex	thevzpu[i].z0	Is PU Thevenin impedance, zero sequence
complex	rlst.maxflt[i].ia1	Is maximum fault current, positive sequence current
complex	rlst.maxflt[i].ia2	Is maximum fault current, negative sequence current
complex	rlst.maxflt[i].ia0	Is maximum fault current, zero sequence current
complex	rlst.maxflt[i].ia	Is maximum fault current, Phase A current
complex	rlst.maxflt[i].ib	Is maximum fault current, Phase B current
complex	rlst.maxflt[i].ic	Is maximum fault current, Phase C current
str	rlst.maxfltdsc[i]	Is description of fault condition for maximum fault current

Note: Maximum fault current is the largest fault current among analyzed faults at a bus.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

9.1.9 OTDF_FACTORS

This function returns the OTDF factors array. The PSS®E case must be opened prior to execution of this function. Also the dfxfile provided must correspond to the PSS®E case opened.

Python syntax:

```
rlst = pssarrays.otdf_factors(dfxfile)
```

where:

string	dfxfile	Is the name of distribution factor .dfx file (input, no default allowed), maximum string length=260

Returned object 'rlst' contains the following attributes:

integer	rlst.ierr	Is the error code	
		=0	no error occurred
		=1	no Distribution Factor Data File specified
		=2	generators are converted
		=3	buses in island(s) without a swing bus; use activity TREE
		=4	error opening file DFXFILE
		=5	file DFXFILE is not in the form of a PSS®E-25 or later DFAX file; run DFAX
		=6	file does not contain distribution factors; run DFAX with distribution factor option enabled
		=7	no monitored branches or interfaces defined
		=8	prerequisite requirements for API are not met
		=9	requires opened SAV file, for retrieving OTDF factors
integer	rlst.size.nmline	Is the number of monitored branches	
integer	rlst.size.ninter	Is the number of monitored interfaces	
integer	rlst.size.ncase	Is the number of contingencies + 1 (for base case)	
string	rlst.casetitle.line1	Is the case short title line 1, maximum string length=60	
string	rlst.casetitle.line2	Is the case short title line 2, maximum string length=60	
string	rlst.file.sav	Is the name of saved case (.sav) file, maximum string length=260	
string	rlst.file.dfx	Is the name of distribution factor data (.dfx) file, maximum string length=260	
string	rlst.file.sub	Is the name of subsystem definition data (.sub) file, maximum string length=260	
string	rlst.file.mon	Is the name of monitored element data (.mon) file, maximum string length=260	

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

string	rlst.file.con	Is the name of contingency description data (.con) file, maximum string length=260
string	rlst.melement	Is a list of length (nmline+ninter) of monitored branch and interface names, maximum string length=54
string	rlst.colabel	Is a list of length (ncase) of contingency labels, maximum string length=12
string	rlst.codesc	Is a list of length (ncase) of contingency description, maximum string length=152
float	rlst.factor	Is a list of list OTDF factors, list of ncase elements with each element is a list of nmline+ninter values

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

9.1.10 PV_SUMMARY

This functions returns PV analysis summary.

Python syntax:

```
rlst = pssarrays.pv_summary(pvfile)
```

where:

string	pvfile	PV output filename (.pv), (input, no default allowed), maximum string length=260

Returned object 'rlst' contains the following attributes:

int	rlst.ierr	error code: =0 no error =1 file not PV type =4 requires SAV file
int	rlst.pvsize.ncase	number of contingencies + 1 (for base case)
int	rlst.pvsize.nmline	number of monitored branches
int	rlst.pvsize.ninter	number of monitored interfaces
int	rlst.pvsize.nmvbus	number of voltage monitored buses
int	rlst.pvsize.nmvrec	number of voltage monitored records
int	rlst.pvsize.nmgnbus	number of monitored plant (generator) buses
int	rlst.pvsize.nmldbus	number of monitored load buses
int	rlst.pvsize.nmxtrns	maximum number of MW transfer changes
int	rlst.options	PV solution options (same as in API manual)
float	rlst.values	PV solution values (same as in API manual)
string	rlst.brnflowunits.xfrcur	PV solution transformer branch flow units ('mbrnamp' in pv_solution return list)
string	rlst.brnflowunits.nxfrcr	PV solution non-transformer branch flow units ('mbrnamp' in pv_solution return list)
string	rlst.casetitle.line1	short title line 1
string	rlst.casetitle.line2	short title line 2
string	rlst.file.pv	contingency output (.acc) filename
string	rlst.file.sav	saved case (.sav) filename
string	rlst.file.ecd	economic dispatch data (.ecd) filename
string	rlst.file.thr	load throwover data (.thr) filename
string	rlst.file.dfx	distribution factor data (.dfx) filename
string	rlst.file.sub	subsystem definition data (.sub) filename
string	rlst.file.mon	monitored element data (.mon) filename
string	rlst.file.con	contingency description data (.con) filename
string	rlst.srcsink	source and sink subsystem names
string	rlst.mbranch	monitored branch names

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

float	rlst.mbrnrating.a	rating A
float	rlst.mbrnrating.b	rating B
float	rlst.mbrnrating.c	rating C
string	rlst.minterface	monitored interface names
float	rlst.mitfrating	selected rating of monitored interface
string	rlst.mgenbus	monitored plant (generator) bus label
string	rlst.mloadbus	monitored load bus label
string	rlst.mvbuslabel	monitored voltage bus label
string	rlst.mvreclabel	monitored voltage record label
float	rlst.mvrecmax	monitored voltage bus maximum
float	rlst.mvrecmin	monitored voltage bus minimum
string	rlst.mvrectype	monitored voltage record type (range/deviation)
string	rlst.colabel	contingency labels
string	rlst.codesc	contingency description
float	rlst.maxmw	maximum MW transfer

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

9.1.11 PV_SUMMARY_REPORT

This function creates PV analysis summary text report.

Python syntax:

```
ierr = pv_summary_report(pvfile,rptfile)
```

where:

string	pvfile	PV output filename (.pv), (input, no default allowed), maximum string length=260
string	rptfile	Is the name of report file (input, PSS®E Report by default), maximum string length=260

Returns:

int	rlst.ierr	error code:	
		=0	no error
		=1	file not PV type
		=4	requires SAV file

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

9.1.12 PV_SOLUTION

This functions returns PV analysis monitored flows and bus voltages for one contingency.

Python syntax:

```
rlst = pssarrays.pv_solution(pvfile,colabel)
```

where:

string	pvfile	PV output filename (.pv), (input, no default allowed), maximum string length=260
string	colabel	Is the name of contingency label to get PV solution for, only one label allowed (input, no default allowed), maximum string length=12

Returned object 'rlst' contains the following attributes:

int	rlst.ierr	error code: =0 no error =1 file not PV type =4 requires SAV file
int	rlst.island	number of islands
float	rlst.mwtransfer	MW transactions, list of size [ntrns]
boolean	rlst.cnvflag	convergence flag (True when converged), list of size [ntrns]
string	rlist.cnvcond	convergence condition
float	rlst.mvaworst	largest bus MVA mismatch, list of size [ntrns]
float	rlst.mvatotal	total system MVA mismatch, list of size [ntrns]
float	rlst.volts	monitored bus voltage (pu), list of size [nmvbus][ntrns]
float	rlst.mgenmw	monitored plant MW, list of size [nmgnbus][ntrns]
float	rlst.mgenmvar	monitored plant MVAR, list of size [nmgnbus][ntrns]
float	rlst.mloadmw	monitored load MW, list of size [nmldbus][ntrns]
float	rlst.mloadmvar	monitored load MVAR, list of size [nmldbus][ntrns]
float	rlst.mbrnmva	monitored branch MVA flow (MVA), list of size [nmline][ntrns]
float	rlst.mbrnamp	monitored branch flow, AMPS expressed in MVA or MVA, list of size [nmline][ntrns]. The brnflowunits variable in pv_summary determine the units. Use this value to calculate branch loadings.
float	rlst.mitfmw	monitored interface MW flow (MW), list of size [ninter][ntrns]

where

ntrans =number of MW transfer changes

nmvbus =number of voltage monitored buses

nmgnbus =number of monitored plant (generator) buses

nmldbus =number of monitored load buses

nmline =number of monitored branches

ninter =number of monitored interfaces

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

9.1.13 PV_SOLUTION_REPORT

This function creates PV analysis solution text report.

Python syntax:

```
ierr = pv_solution_report(pvfile,colabels,rptfile)
```

where:

string	pvfile	PV output filename (.pv), (input, no default allowed), maximum string length=260
string	colabels	Is the list of contingency label names (input, all contingencies in accfile by default), maximum string length=12
string	rptfile	Is the name of report file (input, PSS®E Report by default), maximum string length=260

Returns:

int	rlst.ierr	error code:	
		=0	no error
		=1	file not PV type
		=4	requires SAV file

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

9.1.14 QV_SOLUTION

This functions returns QV analysis monitored flows and bus voltages for one contingency.

Python syntax:

```
rlst = pssarrays.qv_solution(qvfile,colabel)
```

where:

string	qvfile	QV output filename (.qv), (input, no default allowed), maximum string length=260
string	colabel	Is the name of contingency label to get QV solution for, only one label allowed (input, no default allowed), maximum string length=12

Returned object 'rlst' contains the following attributes:

int	rlst.ierr	error code: =0 no error =2 file not QV type =5 requires SAV file
int	rlst.island	number of islands
float	rlst.vsetpoint	voltage setpoints, list of size [nvstp]
boolean	rlst.cnvflag	convergence flag (True when converged), list of size [nvstp]
string	rlst.convcond	convergence condtion
float	rlst.mvaworst	largest bus MVA mismatch, list of size [nvstp]
float	rlst.mvatotal	total system MVA mismatch, list of size [nvstp]
float	rlst.volts	monitored bus voltage (pu), list of size [nmvbus][nvstp]
float	rlst.mgenmvar	monitored plant Mvar, list of size [nmgnbus][nvstp]

where

ntrans = number of voltage setpoint changes

nmvbus = number of voltage monitored buses

nmgnbus = number of monitored plant (generator) buses

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

9.1.15 QV_SOLUTION_REPORT

This function creates QV analysis solution text report.

Python syntax:

```
ierr = qv_solution_report(qvfile,colabels,rptfile)
```

where:

string	qvfile	QV output filename (.qv), (input, no default allowed), maximum string length=260
string	colabels	Is the list of contingency label names (input, all contingencies in accfile by default), maximum string length=12
string	rptfile	Is the name of report file (input, PSS®E Report by default), maximum string length=260

Returned list (accessed by attributes as):

int	rlst.ierr	error code:	
		=0	no error
		=2	file not QV type
		=5	requires SAV file

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

9.1.16 QV_SUMMARY

This functions returns QV analysis summary.

Python syntax:

```
rlst = pssarrays.qv_summary(qvfile)
```

where:

string	qvfile	QV output filename (.qv), (input, no default allowed), maximum string length=260

Returned object 'rlst' contains the following attributes:

int	rlst.ierr	error code: =0 no error =2 file not QV type, =5 requires SAV file
int	rlst.qvbus	QV analysis bus name
int	rlst.qvsize.ncase	number of contingencies + 1 (for base case)
int	rlst.qvsize.nmvbus	number of voltage monitored buses
int	rlst.qvsize.nmvrec	number of voltage monitored records
int	rlst.qvsize.nmgnbus	number of monitored plant (generator) buses
int	rlst.qvsize.nmxvstp	maximum number of voltage setpoint changes
int	rlst.options	QV solution options (same as in API manual)
string	rlst.values	QV solution values (same as in API manual)
string	rlst.casetitle.line1	short title line 1
string	rlst.casetitle.line2	short title line 2
string	rlst.file.qv	contingency output (.acc) filename
string	rlst.file.sav	saved case (.sav) filename
string	rlst.file.thr	load throwover data (.thr) filename
string	rlst.file.dfx	distribution factor data (.dfx) filename
string	rlst.file.sub	subsystem definition data (.sub) filename
string	rlst.file.mon	monitored element data (.mon) filename
string	rlst.file.con	contingency description data (.con) filename
string	rlst.mgenbus	monitored plant (generator) bus label
string	rlst.mvbuslabel	monitored voltage bus label
string	rlst.mvreclabel	monitored voltage record label
float	rlst.mvrecmax	monitored voltage bus maximum
float	rlst.mvrecmin	monitored voltage bus minimum
string	rlst.mvrectype	monitored voltage record type (range/deviation)
string	rlst.colabel	contingency labels
string	rlst.codesc	contingency description
float	rlst.minvstp	minimum voltage setpoint

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

float	rlst.maxvstp	maximum voltage setpoint
float	rlst.minmvar	minimum MVAR change
float	rlst.maxmvar	maximum MVAR change
float	rlst.maxmsm	maximum MVAR mismatch

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

9.1.17 QV_SUMMARY_REPORT

This function creates QV analysis summary text report.

Python syntax:

```
ierr = qv_summary_report(qvfile,rptfile)
```

where:

string	qvfile	QV output filename (.qv), (input, no default allowed), maximum string length=260
string	rptfile	Is the name of report file (input, PSS®E Report by default), maximum string length=260

Returned list (accessed by attributes as):

int	rlst.ierr	error code:	
		=0	no error
		=2	file not QV type
		=5	requires SAV file

9.2 PSSEXCEL module

The PSSEXCEL module provides Python functions to export PSS®E data or solution results to Excel spreadsheets.

These functions do not record themselves when called with recording enabled.

The syntax on the following pages assumes that the module has been imported by executing:

```
import pssexcel
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

9.2.1 ACCC

This function exports ACCC analysis results to Excel Spreadsheet.

Python syntax:

```
pssexcel.accc(accfile, string, colabel='', stype='contingency',
busmsm=0.5, sysmsm=5.0, rating='a', namesplit=True, xlsfile='',
sheet='', overwritesheet=True, show=True, ratecon='b',
baseflowvio=True, basevoltvio=True, flowlimit=100.0, flowchange=0.0,
voltchange=0.0)
```

where:

string	accfile	ACCC analysis output filename (.acc), no default allowed

string	string	Name or list of names indicating which results to export, no default allowed. Allowed strings:
		`'s' or 'summary'` `ACCC Analysis Summary`
		`’e’ or 'events'` `Contingency Events Description`
		`'b' or 'branch'` `Monitored Branch Flow (MVA)`
		`'i' or 'interface'` `Monitored Interface Flow (MW)`
		`'v' or 'voltage'` `Monitored Bus Voltage`
		`'l' or 'load'` `Loads Shed (MW)`
		`'g' or 'generator'` `Generator Dispatch (MW)`
		`'p' or 'phase shifter'` `Phase Shifter Angle`
		Example: string = `'v'` or `['s','v','g','l','b','i']`

string	colabel	Contingency label or list of contingency labels for which the solution is exported, default - all contingencies, not required when exporting only summary
string	stype	Is the name of solution type to get (input, contingency by default), allowed values: contingency, con, tripping, trp, caction, contingency action, cor
float	busmsm	Is the value of bus mismatch tolerance in MVA (input, 0.5 MVA by default)
float	sysmsm	Is the value of system mismatch tolerance in MVA (input, 5.0 MVA by default)
string	rating	s the value of rating to use to calculate base case percentage overload, default 'a', allowed values:'a', 'b', or 'c'
string	stype	Is the name of solution type to get (input, *contingency* by default allowed), allowed values: *contingency*, *con*, *tripping*, *trp*, *caction*, *contingency action*, *cor*
float	busmsm	Is the value of bus mismatch tolerance in MVA (input, 0.5 MVA by default)
float	sysmsm	Is the value of system mismatch tolerance in MVA (input, 5.0 MVA by default)
string	rating	Rating to use to calculate percentage overload. Default ’a’, allowed values: ’a’, ’b’, or ’c’.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

boolean	namesplit	Split extended bus names, default True = True — bus names split into three values: numbername and bus voltage (three columns in worksheet) = False — bus names are kept as single value string (one column in worksheet)
string	xlsfile	Excel workbook name, default Book#.xls

string	sheet	Excel worksheet name, default (blank string), depending on string input, worksheets created are:
		`Summary` — `ACCC Analysis Summary`
		`Contingency Events` — `Events Description`
		`Branch Flow` — `Monitored Branch Flow (MVA)`
		`Interface Flow` — `Monitored Interface Flow (MW)`
		`Bus Voltage` — `Monitored Bus Voltage`
		`Load Shed` — `Loads Shed (MW)`
		`Generator Dispatch` — `Generator Dispatch (MW)`
		`Phase Shifter Angle` — `Phase Shifter Angle (degrees)`
		All worksheet names are prefixed with sheet name, when provided.

boolean	overwritesheet	Overwrite worksheets flag, default True = True — existing worksheets are overwritten = False — existing worksheets are copied and their names appended with (#)where # is next sequence number.
boolean	show	Show or Hide Excel Spreadsheet flag, default True = True — open and show Excel Spreadsheet = False — do not show Excel Spreadsheet (create and save)
string	ratecon	Is the value of rating to use to calculate Contingency Case percentage overload default 'b', allowed values:'a', 'b', or 'c'
boolean	baseflowvio	Exclude elements with base case loading violations from contingency reports default True, allowed values: True or False
boolean	basevoltvio	Exclude buses with base case voltage range violations from contingency reports default True, allowed values: True or False
float	flowlimit	Percent of flow rating, default=100 for 100% Show monitored elements with flow above flowlimit value. flowlimit=0 will export flows for all elements.
float	flowchange	Minimum contingency case flow change in MVA for overload reports default 0 MVA, allowed values: Any number
float	voltchange	Minimum contingency case voltage change in pu for voltage range violations default 0 pu, allowed values: Any number

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

9.2.2 iec_data_file

This function exports synchronous machines, asynchronous machines and transformer data from PSS®E Saved Case required to assemble IEC fault data input file as an Excel® spreadsheet. IEC data file (.iec) is used by IECS API. The PSS®E Saved case must be opened prior to execution of this function.

It creates three worksheets 'Machines', ’Transformers' and ’IEC Data’.

This file is used as a convenient way to create an .iec file by copy/paste generator and transformer records to form a power station unit, modify the generator, motor and transformer data as required by IECS API. After appropriate modifications, save IEC Data worksheet as comma separated file with extension .iec. This file then can be used as *Fault data input file* to the IECS API.

Python syntax:

```
pssexcel.iec_data_file(xlsfile, sheet, overwritesheet, show)
```

where:

string	xlsfile	Is the name of excel workbook filename (input, default savefile prefix and extension .iec)

string	sheet	Excel worksheet name, default (blank string), worksheets created are:
		`Machines` — `Saved case machines data`
		`Transformers` — `Saved case two and three windings transformers data`
		`IEC Data` — `Data required by IECS API`
		All worksheet names are prefixed with sheet name, when provided.

boolean	overwritesheet	Overwrite worksheets flag, default True = True — existing worksheets are overwritten = False — existing worksheets are copied and their names appended with (#), where # is next sequence number.
boolean	show	Show or Hide Excel Spreadsheet flag, default True = True — open and show Excel Spreadsheet = False — do not show Excel Spreadsheet (create and save)

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

9.2.3 pv

This function exports PV solution results to Excel Spreadsheet.

Python syntax:

```
pssexcel.pv(pvfile, string, colabel, namesplit, xlsfile, sheet,
overwritesheet, show)
```

where:

string	pvfile	PV analysis output filename (.pv), no default allowed

string	string	Name or list of names indicating which results to export, no default allowed. Allowed strings:
		`'s' or 'summary'` PV Solution Summary
		`'v' or 'voltage'` Monitored Bus Voltage
		`'m' or 'mismatch'` Largest and Total Mismatch
		`'g' or 'generator'` Monitored Plants MW and MVAR
		`'l' or 'load'` Monitored Loads MW and MVAR
		`'b' or 'branch'` Monitored Branch Flow (MVA)
		`'i' or 'interface'` Monitored Interface Flow (MW)
		Example:
		`string = 'v' or`
		`string = ['s','m','v','g','l','b','i']`

string	colabel	Contingency label or list of contingency labels for which the solution is exported, default - all contingencies, not required when exporting only summary
boolean	namesplit	Split extended bus names, default True = True: bus names split into three values: numbername and bus voltage (three columns in worksheet) = False: bus names are kept as single value string (one column in worksheet)
string	xlsfile	Excel workbook name, default Book#.xls
string	sheet	Excel worksheet name, default ’’ (blank string), depending on string input, worksheets created are: `Summary` PV Solution Summary `Bus Voltage` Monitored Bus Voltage `Mismatch` Largest and Total Mismatch `Generator Dispatch` Monitored Plants MW and MVAR `Bus Load` Monitored Loads MW and MVAR `Branch Flow` Monitored Branch Flow (MVA) `Interface Flow (MW)` Monitored Interface Flow All worksheet names are prefixed with sheet name, when provided.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

boolean	overwritesheet	Overwrite worksheets flag, default True	
		= True	existing worksheets are overwritten
		= False	existing worksheets are copied and their names appended with (#)where # is next sequence number.
boolean	show	Show or Hide Excel Spreadsheet flag, default True	
		= True	open and show Excel Spreadsheet
		= False	do not show Excel Spreadsheet (create and save)

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

9.2.4 qv

This function exports QV solution results to Excel Spreadsheet.

Python syntax:

```
pssexcel.qv(qvfile, string, colabel, namesplit, xlsfile, sheet,
overwritesheet, show)
```

where:

string	qvfile	QV analysis output filename (.qv), no default allowed

string	string	Name or list of names indicating which results to export, no default allowed. Allowed strings:
		`'s' or 'summary'` QV Solution Summary
		`'v' or 'voltage'` Monitored Bus Voltage
		`'m' or 'mismatch'` Largest and Total Mismatch
		`'g' or 'generator'` Monitored Plants MW and MVAR
		Example:
		`string = 's' or`
		`string = ['s','m','v','g']`

string	colabel	Contingency label or list of contingency labels for which the solution is exported, default - all contingencies, not required when exporting only summary
boolean	namesplit	Split extended bus names, default True = True: bus names split into three values: numbername and bus voltage (three columns in worksheet) = False: bus names are kept as single value string (one column in worksheet)
string	xlsfile	Excel workbook name, default Book#.xls
string	sheet	Excel worksheet name, default ’’ (blank string), depending on string input, worksheets created are: `Summary` QV Solution Summary `Bus Voltage` Monitored Bus Voltage `Mismatch` Largest and Total Mismatch `Generator Dispatch` Monitored Plants MVAR All worksheet names are prefixed with sheet name, when provided.
boolean	overwritesheet	Overwrite worksheets flag, default True = True: existing worksheets are overwritten = False: existing worksheets are copied and their names appended with (#)where # is next sequence number.
boolean	show	Show or Hide Excel Spreadsheet flag, default True = True: open and show Excel Spreadsheet = False: do not show Excel Spreadsheet (create and save)

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

9.3 EXCELPY module

The EXCELPY module provides Python functions to interface with Excel. Use this module to create, populate and format Excel workbooks from Python.

These functions do not record themselves when called with recording enabled.

The syntax on the following pages assumes that the module has been imported by executing:

```
import excelpy
```

Following are the definition of arguments used by various functions defined in this module.

string or int	address	tuple of integers/strings. Allowed address when specifying cell: 'adrstr' or ('adrstr') or (row,'clnstr') or (row, column) Example: Cell address referencing 2nd row, column A, can be provided as- 'a2' or ('a2') or (2,'a') or (2,1) range: ('adrstr1','adrstr2') or (toprow, 'leftcolumnstr', bottomrow, 'rightcolumnstr') or (toprow, leftcolumn, bottomrow, rightcolumn) or ('adrstr1', (bottomrow, rightcolumn)) or ('adrstr1', (bottomrow, 'rightcolumnstr')) ((toprow, leftcolumn), 'adrstr2') or ((toprow, 'leftcolumnstr'), 'adrstr2') where row, column, toprow, leftcolumn, bottomrow, rightcolumn are integer values, adrstr, adrstr1, adrstr2 are cell address string values, and clnstr, leftcolumnstr, rightcolumnstr are column name string values. Note: Column and address refernce string values are case-insensitive.
string	alignv	alignment value for cell, range, rows or columns. Allowed values when specifying horizontal alignment: 'left','right','h_center' vertical alignment: 'top','bottom','v_center'

string	borderv	border name. Allowed border names when formatting:
		cell:
		`'diagonaldown', 'diagnalup', 'top', 'bottom', 'left', 'right', 'none', 'outline'`
		range:
		`'diagonaldown', 'diagonalup', 'top', 'bottom', 'left', 'right', 'inside', 'hinside', 'vinside', 'none', 'outline'`

string or int	borderColor	string, color name; or integer, color index.
		Allowed color names:
		`'black', 'white', 'red', 'green', 'blue', 'yellow', 'magenta', 'cyan', 'brown', 'dgreen', 'dblue', 'orange'`
		Allowed ColorIndex:
		`1 to 56 as per ColorIndex property in Excel Visual Basic Reference.`

string	borderStyle	border style name. Allowed names: `'continuous', 'dash','dashdot', 'dashdotdot', 'dot', 'double', 'none', 'slantdashdot'`
string	borderWeight	border weight name. Allowed names: `'hairline', 'medium', 'thick', 'thin'`

string	fontStyle	font style name, specified as style name or tuple of style names.
		Allowed font style names:
		`'regular', 'bold', 'italic', 'underline'`
		Example:
		`(Italic,Bold) or`
		`'bold'`

string or int	fontColor	string, color name; or integer, color index. This is same as color in 'borderColor'.
string	numberFormat	number format Example: 0.000 for three decimal places
int	leftCol	left column number
string	newSheet	name of the new worksheet
string	oldSheet	name of the worksheet used as reference for worksheet operations
boolean	overwritesheet	overwrite if worksheet exists, Allowed values: = True — existing worksheet will be overwritten. = False — existing sheet copied and its name appended with (#), where # is next sequence number. Used when creating new worksheets or renaming existing worksheets.
string	sheetspread-sheet	name of the excel worksheet. If sheet is not provided, active work-sheet used.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

int	topRow	top row number
boolean	wrapText	wrap text or not, Allowed values: True or False
string	xlsfile	name of the Excel workbook

Default Workbook Style

The excel workbook is created with following default font style:

```
Black, Regular, Courier New, size 10, no wrapText, General number
format
```

How to Use?

Create xls file object as below and apply various methods defined here.

(1) When used to create new Excel files or add worksheets to existing Excel files:

```
testxls = excelpy.workbook()
testxls = excelpy.workbook(r"c:\working dir\ex1.xls")
testxls = excelpy.workbook(r"c:\working dir\ex1.xls", "MySheet") or
testxls = excelpy.workbook(r"c:\working dir\ex1.xls", "MySheet", False) or
testxls = excelpy.workbook(r"c:\working dir\ex1.xls", "MySheet", False, 'w') or
testxls = excelpy.workbook(xlsfile=r"c:\working dir\ex1.xls", sheet="MySheet",
                    overwritesheet=False, mode='w')
```

(2) When used to read existing Excel files:

```
testxls = excelpy.workbook(r"c:\working dir\ex1.xls", mode='r')
```

Note: When using Excel 2007 and later, create file in either 'xlsx' or 'xls' format.

Excel 2003 and earlier - default save format 'xls'.

Excel 2007 and later - default save format 'xlsx'.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

9.3.1 Methods

The following methods are members of the excelpy class that can be instantiated using the workbook functions defined in Section 9.3.2 Workbook Functions. In all these methods, when sheet=None or not provided, active worksheet is used.

align(address, alignv, sheet=None)

Align Cell or Range.
cell, address = (row,col)
range, address = (topRow,leftCol,bottomRow,rightCol)

align_columns(address, alignv, sheet=None)

Align one or more columns.
One column, address = (row,col)
Multiple columns, address = (topRow,leftCol,topRow,rightCol)

align_rows(address, alignv, sheet=None)

Align one or more rows.
One row, address = (row,col)
Multiple rows, address = (topRow,leftCol,bottomRow,leftCol)

autofit_columns(address, sheet=None)

Autofit column or columns.
One column, address = (row,col)
Multiple columns, address = (topRow,leftCol,topRow,rightCol)

autofit_rows(address, sheet=None)

Autofit row or rows.
One row, address = (row,col)
Multiple rows, address = (topRow,leftCol,bottomRow,leftCol)

border(address, borderv, borderStyle='continuous', borderWeight='hairline', borderColor='black', sheet=None)

Format Cell or Range borders.
cell, address = (row,col)
range, address = (topRow,leftCol,bottomRow,rightCol)

close()

Close active Excel workbook and quit Excel application.
delete_columns(address, howmany=1, sheet=None)
Delete columns to the right of specified row.
row address = (row,col)
Number of columns deleted is provided by integer value 'howmany'.
delete_rows(address, howmany=1, sheet=None)
Delete rows below specified row.
row address = (row,col)
Number of rows deleted is provided by integer value 'howmany'.

font(address, fontStyle=None, fontName=None, fontSize=None, fontColor=None, wrap-

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Text=False, numberFormat=None, sheet=None)

Set cell or range font properties: style, name, size, color, wrap, number format
Cell, address = (row,col)
Range, address = (topRow,leftCol,bottomRow,rightCol)

font_color(address, color, sheet=None)

Set cell or range font color.
Cell, address = (row,col)
Range, address = (topRow,leftCol,bottomRow,rightCol)

font_sheet(fontStyle=('Regular',), fontName='Courier New', fontSize=10, fontColor='black', wrap-Text=False, numberFormat=None, sheet=None)

Set font properties for Excel sheet: style, name, size, color, wrap, number format

freezepanes(address, sheet=None)

Freeze worksheet panes.
address = (row,1) to freeze worksheet rows above 'row'
address = (1, col) to freeze worksheet columns left of 'col'
address = (row,col to freeze worksheet rows above 'row' and columns left of 'col'

height(address, height, sheet=None)

Set height of one row or multiple rows. Height is provided in points.
One row, address = (row,col)
Multiple rows, address = (topRow,leftCol,bottomRow,rightCol)

hide()

Hide opened Excel workbook.

merge(address, sheet=None)

Merge cells in the range.

page_footer(left=None, center=None, right=None, sheet=None)

Set page left, center, and right footers. Footer input value is a string.

page_format(orientation='portrait', left=1.0, right=1.0, top=1.0, bottom=1.0, header=0.5, footer=0.5, sheet=None)

Set page format: orientation, margins (given in inches).

```
Orientation = 'portrait'  or 'p'  or 1
            = 'landscape' or 'l'  or 2
Defaults:
orientation = Portrait
margins: left   = 1.0 inch
         right  = 1.0 inch
         top    = 1.0 inch
         bottom = 1.0 inch
         header = 0.5 inch
         footer = 0.5 inch
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

page_header(left=None, center=None, right=None, sheet=None)

Set page left, center, and right headers. Header input value is a string.

page_landscape(sheet=None)

Set page orientation to landscape.

page_margin(left=1.0, right=1.0, top=1.0, bottom=1.0, header=0.5, footer=0.5, sheet=None)

Set page margins. The margin value is given in inches.

page_portrait(sheet=None)

Set page orientation to portrait.

save(xlsfile=None)

Save active Excel workbook.

xlsfile = save(xlsfile)

Returns name of the saved Excel file.

set_active_sheet(sheet)

Set 'sheet' as active worksheet.

set_cell(address, value, fontStyle=None, fontName=None, fontSize=None, fontColor=None, wrapText=False, numberFormat=None, sheet=None)

Set value to one cell.

set_range(topRow, leftCol, data, transpose=False, fontStyle=None, fontName=None, fontSize=None, fontColor=None, wrapText=False, numberFormat=None, sheet=None)

Set data to cells in the range.

bottomRow, rightCol = set_range(topRow,leftCol, data, ...)
Boolean, transpose: Allowed values: True or False

Excel methods write data row by row. If data in Python list of lists, or tuple of tuples or any combination of these, is not stored row by row, transpose that data. This functions works same as matrix transpose.

Returns:

bottomRow: Bottom row number after writing input data to worksheet.
rightCol : Right column number after writing input data to worksheet.

show( )

Show opened Excel workbook.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

show_alerts(display)

Show or suppresses all POP-UP windows, like File Overwrite Yes/No/Cancel, Merge Overwrite.
Boolean, display: Allowed values: True or False, 1 or 0
= True or 1, show alerts
= False or 0, do not show alerts. Data will be overwritten.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

transpose_data(data, transpose=True)

Transpose data.
Excel methods write data row by row. If data in Python list of lists, or tuple of tuples or any combination of these, is not stored row by row, transpose that data. This functions works same as matrix transpose.

nrows, nclns, datat = set_range(data, transpose=True)

data : any type of data to be tranposed
transpose: boolean, allowed values: True or False, True by default

Returns:
nrows - number of rows in transposed data
nclns - number of columns in transposed data
datat - transposed data

width(address, width, sheet=None)

Set width of one column or multiple columns. Width is provided in points.
One column, address = (row,col)
Multiple columns, address = (topRow,leftCol,topRow,rightCol)

worksheet_add_after(newSheet=None, oldSheet=None, overwritesheet=True)spreadsheet

Add a newSheet after the oldSheet or active sheet. If newSheet name is not provided, Sheet# worksheet is added. If oldSheet name is not provided, worksheet is added at the end. Sheet# is a next available sheet number in the active workbook.

worksheet_add_before(newSheet=None, oldSheet=None, overwritesheet=True)

Add a newSheet before the oldSheet or active sheet. If newSheet name is not provided, Sheet# worksheet is added. If oldSheet name is not provided, worksheet is added at the end. Sheet# is a next available sheet number in the active workbook.spreadsheet

worksheet_add_begin(sheet=None, overwritesheet=True)

Add a sheet at the beginning. If sheet name is not provided, Sheet# worksheet is added. Sheet# is a next available sheet number in the active workbook.spreadsheet

worksheet_add_end(sheet=None, overwritesheet=True)spreadsheet

Add a sheet at the end. If sheet name is not provided, Sheet# worksheet is added. Sheet# is a next available sheet number in the active workbook.

worksheet_delete(sheet)

Delete a sheet from active workbook.

worksheet_left_columns2repeat(columns, sheet=None)spreadsheet

Repeat 'columns' at the left of each page of the worksheet.
'columns' is specified as tuple or list of column numbers,
e.g., columns = (1,3) to repeat columns 1 to 3 at the left or
columns = 1 to repeat column 1 at the left.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

worksheet_rename(newSheet, oldSheet=None, overwritesheet=True)spreadsheet

Rename Excel 'oldSheet' or active sheet with newSheet name.
overwritesheet = True, contents of oldSheet are deleted, so newSheet is blank worksheet
= False, contents of oldSheet are written to newSheet.\

worksheet_size_violation(topRow, leftCol, data=None, transpose=False)

Get worksheet size violations and extents.

Inputs:
data - List or List of Lists values to be written
transpose - Logical
Excel methods write data row by row. If data in Python list of lists, or
tuple of tuples or any combination of these, is not stored that way,
transpose that data. This functions same as matrix transpose.

Returns:
Returned tuple contains:
row_violation = rtple[0], True when top/bottom row exceeds maximum allowable rows
col_violation = rtple[2], True when left/right column exceeds maximum allowable columns
topRow = rtple[3], top row number
leftCol = rtple[4], left column number
bottomRow = rtple[5], bottom row number
rightCol = rtple[6], right column number
wshtmaxrows = rtple[7], = 65536, maximum number of rows allowed in a worksheet
wshtmaxcols = rtple[8], = 256, maximum number of columns allowed in a worksheet

worksheet_top_rows2repeat(rows, sheet=None)spreadsheet

Repeat 'rows' at the top of each page of the worksheet.
'rows' is specified as tuple or list of row numbers,
e.g., rows = (1,3) to repeat rows 1 to 3 at the top or
rows = 1 to repeat row 1 at the top.

wraptext(address, sheet=None)

Wrap Text inside the cell or range.
Cell, address = (row,col)
Range, address = (topRow,leftCol,topRow,rightCol)

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

9.3.2 Workbook Functions

Use this function to create an Excel workbook object that will possess the methods defined in Section 9.3.1 Methods.

Python syntax:

```
excelpy.workbook(xlsfile=None, sheet=None, overwritesheet=True)
```

Examples:

```
testxls = excelpy.workbook()

testxls = excelpy.workbook("c:\working dir\test1.xls")

testxls = excelpy.workbook("c:\working dir\test1.xls", "MySheet")

testxls = excelpy.workbook("c:\working dir\test1.xls", "MySheet",
                           False)

testxls = excelpy.workbook(xlsfile="c:\working dir\test1.xls",
                           sheet="MySheet", overwritesheet=False)
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

9.4 GICDATA module

The 'gicdata' module provides functions to create GIC data file required in GIC analysis.

GIC (Geomagnetically Induced Current) Analysis requires an additional GIC data file (See Section 7.2, GIC Data File Contents in the *Program Operation Manual*) that contains:

- Substation data
- Bus Substation data and
- Transformer data

Functions in this 'gicdata' module help to create GIC data file applicable to a specific Power Flow Network Case for an entire network or just desired study subsystem.

Creating GIC data using 'gicdata' module is a two step process.

Step 1

Create GIC data file template using functions:

- template_excel(..)
- template_py(..)

This will create structure of GIC data that is required.

Step 2

Edit/Update the GIC data file template created in Step (1) to provide:

- Substation data
- Bus Substation data
- Transformer data

For transformers, vector group information must be provided. If power flow Saved Case has vector group information, GIC data file template will have this information as well. If PSSE Saved Case does not have vector group information, this needs to be provided. The GIC data file then can default values for all other transformer record data items.

Once GIC data file template is updated, create GIC data file using functions:

- excel2gicfile(..)
- py2gicfile(..)

This GIC data file can be used in GIC Analysis.

Argument Definitions:

string savfile-	PSSE Saved Case file name
string pyfile-	GIC data file formatted in Python structures
	This module defines Python structures and saves in this file (Step 1 above).
	User only edits/updates values.
string excelfile-	GIC data file formatted in Excel worksheets
	This module creates this Excel file and worksheets in it (Step 1 above).
	User only edits/updates values.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

string gicfile- GIC data file name (output) that can be used in GIC Analysis.

The following optional arguments are used to define subsystem, if GIC study is to be performed for a subsystem.

real basekv- List of two real values: minimum voltage and maximum voltage,

For example [minVolt, maxVolt], default = []

integer areas- List of areas, default = []

integer buses- List of buses, default = []

integer owners- List of owners, default = []

integer zones- List of zones, default = []

integer tielevels- Number of intertie levels, default = 0

Functions:

check_gicfile(gicfile0, gicfile='')

Check GIC data file 'gicfile0' and create new GIC data file 'gicfile'.

Input gicfile0 is read and processed to do following:

- Add quotes (if already not there) around character data items likes "SUBNAM"
- Remove tabs and such non-ASCII characters
- Check if same geographical location is provided for different substations and if found remove that substation from 'substation' record groups and replace that substation in 'bus subsation' record group.

Output 'gicfile' created only if it is specified.

excel2gicfile(excelfile, gicfile)

Create GIC data file (.gic) from GIC data template in Excel.

py2gicfile(pyfile, gicfile)

Create GIC data file (.gic) from GIC data template in Python structures.

template_excel(savfile, excelfile, basekv=[], areas=[], buses=[], owners=[], zones=[], tielevels=0)

Create GIC data file template in Excel Spreadsheet.

template_py(savfile, pyfile, basekv=[], areas=[], buses=[], owners=[], zones=[], tielevels=0)

Create GIC data file template in Python structures.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

SIEMENS

Chapter 10

Graphical Analysis Output

This chapter defines API routines that interact with displays and files associated with the graphical analysis capabilities of the PSS®E GUI. They will elicit no action if a GUI is not present when invoked.

These routines are not recorded when recording is enabled.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

10.1 POLY_PRINT

Use this API to print the results of activity POLY from the file.

Batch command syntax:

```
BAT_POLY_PRINT  NPLOT  PLOTARRAY(1)…PLOTARRAY(NPLOT)  BOUNDVAL
PRTPARAMS  RESFILE
```

Python syntax:

```
poly_print(nplot, plotarray, boundval, prtparams, resfile)
```

Fortran syntax:

```
CALL POLY_PRINT(NPLOT, PLOTARRAY, BOUNDVAL, PRTPARAMS, RESFILE)
```

where:

Integer NPLOT	Number of plots.
Integer PLOTARRAY(NPLOT)	Is an array of the numbers of plots, selected for printing. The base case is marked by 0, then there are all the contingencies, beginning from 1, and the last number means all the cases together. The array consists of the set of these numbers. The size of the array is equal to NPLOT.
Real BOUNDVAL	Boundary value for the plot area. It is equal to Maximum opposing system delta interchange used for calculations.
Integer PRTPARAMS	To print (or not) the set of parameters used for calculations. The values are as follows: PRTPARAMS = 0 no printing of parameters. PRTPARAMS = 1 the set of parameters will be printed.
Character RESFILE*260	Is the name of the existing POLY Results File (*.pol) file.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

10.2 PV_PRINT

Use this API to print the results of PV analysis from the file.

Batch command syntax:

```
BAT_PV_PRINT  RESFILE  RESTYPE  NCONTING  CONTINGARRAY(1)…
CONTINGARRAY(NCONTING)  NOUTPUTS  OUTPUTSARRAY(1)…
OUTPUTSARRAY(NOUTPUTS)  INTERFACE  ENFORCESCALE  USELINESTYLES
NUMBERLINES  PRTPARAMS
```

Python syntax:

```
pv_print(resfile, restype, nconting, contingarray, noutputs,
outputsarray, interface, enforcescale, uselinestyles, numberlines,
prtparams)
```

Fortran syntax:

```
CALL PV_PRINT(resfile, restype, nconting, contingarray, noutputs,
outputsarray, interface, enforcescale, uselinestyles, numberlines,
prtparams)
```

where:

Character RESFILE*260	Is the name of the existing *.pv file.
Integer RESTYPE	Type of the results. The values are as follows: RESTYPE = 0 bus voltage RESTYPE = 1 generator MW RESTYPE = 2 branch MVA flow RESTYPE = 3 interface MW flow RESTYPE = 4 bus voltage against interface flow.
Integer NCONTING	Number of specified contingencies.
Integer CONTINGARRAY(NCONTING)	Is an array of the numbers of specified contingencies. The base case is marked by 0. The size of the array is equal to NCONTING.
Integer NOUTPUTS	Number of specified outputs.
Integer OUTPUTSARRAY(NOUTPUTS)	Is an array of the numbers of specified outputs. The count starts from 0. The size of the array is equal to NOUTPUTS.
Integer INTERFACE	Number of specified interface. Valid only in case of bus voltage against interface flow (RESTYPE = 4).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer ENFORCESCALE

Allows to print the results with the different scales. The values are as follows:

ENFORCESCALE = 0
the scales will be not changed according with the results; this allows to print the results from different files in the same scales.

ENFORCESCALE = 1
the maximum and minimum values for the used scales will be taken from the file of results, and the printing will be done according with these values.

Integer USELINESTYLES

To use (or not) different line styles for plotting. The values are as follows:

USELINESTYLES = 0
all the lines will be solid.

USELINESTYLES = 1
five different (standard) styles of lines will be used: Solid, Dash-Dash, Dot-Dot, Dash-Dot, Dash-Dot-Dot.

Integer NUMBERLINES

To put the numbers (or not) on the graphs. The values are as follows:

NUMBERLINES = 0
no numbering of the graphs.

NUMBERLINES = 1
each graph will have its number for better recognition.

Integer PRTPARAMS

To print (or not) the set of parameters used for calculations. The values are as follows:

PRTPARAMS = 0
no printing of parameters.

PRTPARAMS = 1
the set of parameters will be printed.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

10.3 QV_PRINT

Use this API to print the results of QV analysis from the file.

Batch command syntax:

```
BAT_QV_PRINT  RESFILE  NCONTING  CONTINGARRAY(1)…
CONTINGARRAY(NCONTING)  ENFORCESCALE  USELINESTYLES  NUMBERLINES
PRTPARAMS
```

Python syntax:

```
qv_print(resfile, nconting, contingarray, enforcescale,
uselinestyles, numberlines, prtparams)
```

Fortran syntax:

```
CALL QV_PRINT(resfile, nconting, contingarray, enforcescale,
uselinestyles, numberlines, prtparams)
```

where:

Character RESFILE*260	Is the name of the existing *.qv file.
Integer NCONTING	Number of specified contingencies.
Integer CONTINGARRAY(NCONTING)	Is an array of the numbers of specified contingencies. The base case is marked by 0. The size of the array is equal to NCONTING.
Integer ENFORCESCALE	Allows to print the results with the different scales. The values are as follows: ENFORCESCALE = 0 the scales will be not changed according with the results; this allows to print the results from different files in the same scales. ENFORCESCALE = 1 the maximum and minimum values for the used scales will be taken from the file of results, and the printing will be done according with these values.
Integer USELINESTYLES	To use (or not) different line styles for plotting. The values are as follows: USELINESTYLES = 0 all the lines will be solid. USELINESTYLES = 1 five different (standard) styles of lines will be used: Solid, Dash-Dash, Dot-Dot, Dash-Dot, Dash-Dot-Dot.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer NUMBERLINES	To put the numbers (or not) on the graphs. The values are as follows: NUMBERLINES = 0 no numbering of the graphs. NUMBERLINES = 1 each graph will have its number for better recognition.
Integer PRTPARAMS	To print (or not) the set of parameters used for calculations. The values are as follows: PRTPARAMS = 0 no printing of parameters. PRTPARAMS = 1 the set of parameters will be printed.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

10.4 SETFULLVIEWSCALE

Use this API to specify the parameters of the scales used for plotting the results of PV and QV analysis.

Batch command syntax:

```
BAT_SETFULLVIEWSCALE  SCALE  LTVAL  RBVAL  ADJUST(1)  ADJUST(2)
AXISPLACE AXISVALUE  GRIDCALCTYPE  GRIDSTEP  GRIDMINLINES  SHOWNUM
(1)… SHOWNUM(3) LTPOS  INPOS  RBPOS  LTPLACES  INPLACES  RBPLACES
```

Python syntax:

```
setfullviewscale(scale, ltval, rbval, adjust, axisplace, axisvalue,
gridcalctype, gridstep, gridminlines, shownum, ltpos, inpos, rbpos,
ltplaces, inplaces, rbplaces)
```

Fortran syntax:

```
CALL SET_FULLVIEW_SCALE(scale, ltval, rbval, adjust, axisplace,
axisvalue, gridcalctype, gridstep, gridminlines, shownum, ltpos,
inpos, rbpos, ltplaces, inplaces, rbplaces)
```

where:

Integer SCALE	Is the number of the scale. The values are as follows:	
	SCALE = 0	QV analysis, horizontal scale for Q(V) plot.
	SCALE = 1	QV analysis, vertical scale for Q(V) plot.
	SCALE = 2	QV analysis, vertical scale for mismatch plot.
	SCALE = 3	PV analysis, horizontal scale for incremental transfer.
	SCALE = 4	PV analysis, horizontal scale for interface MW flow (bus voltage against interface flow).
	SCALE = 5	PV analysis, vertical scale for bus voltage.
	SCALE = 6	PV analysis, vertical scale for generator MW.
	SCALE = 7	PV analysis, vertical scale for branch MVA flow.
	SCALE = 8	PV analysis, vertical scale for interface MW flow.
	SCALE = 9	PV analysis, vertical scale for bus voltage (bus voltage against interface flow).
Real LTVAL	Left value (for horizontal scale) or top value (for vertical scale).	
Real RBVAL	Right value (for horizontal scale) or bottom value (for vertical scale).	
Integer ADJUST(2)	Is an array of two elements. The values are as follows:	
	ADJUST(1)	to adjust (or not) the left (top) value of the scale.
	ADJUST(2)	to adjust (or not) the right (bottom) value of the scale.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer AXISPLACE	Is the place of the axis. The values are as follows: AXISPLACE = 0 on the left (top) border. AXISPLACE = 1 the axis place is set in AXISVALUE. AXISPLACE = 2 on the right (bottom) border. AXISPLACE = 3 undefined, the exact value will be calculated.
Real AXISVALUE	Is the place of the axis. Valid only in case AXISPLACE = 1.
Integer GRIDCALCTYPE	The way the grid is calculated. The values are as follows: GRIDCALCTYPE = 0 grid step is set in GRIDSTEP. GRIDCALCTYPE = 1 grid is defined by minimum grid lines set in GRIDMINLINES.
Real GRIDSTEP	Step of the grid. Valid only in case GRIDCALCTYPE = 0.
Integer GRIDMINLINES	Minimum number of the grid lines. Valid only in case GRIDCALCTYPE = 1.
Integer SHOWNUM(3)	Is an array of three elements. The values are as follows: SHOWNUM (1) to show (or not) the left (top) value of the scale. SHOWNUM (2) to show (or not) all the intermediate values for the grid. SHOWNUM (3) to show (or not) the right (bottom) value of the scale.
Integer LTPOS	Is the place of the left (top) value relative to the left (top) border. The values are as follows: LTPOS = 0 to the left of the border (above the border). LTPOS = 1 centered to the border. LTPOS = 2 to the right of the border (below the border).
Integer INPOS	Is the place of the intermediate values relative to the intermediate grid lines. The values are as follows: INPOS = 0 to the left of the line (above the line). INPOS = 1 centered to the line. INPOS = 2 to the right of the line (below the line).
Integer RBPOS	Is the place of the right (bottom) value relative to the right (bottom) border. The values are as follows: RBPOS = 0 to the left of the border (above the border). RBPOS = 1 centered to the border. RBPOS = 2 to the right of the border (below the border).
Integer LTPLACES	Number of digits to the right of the decimal point for the left (top) value.
Integer INPLACES	Number of digits to the right of the decimal point for intermediate values.
Integer RBPLACES	Number of digits to the right of the decimal point for the right (bottom) value.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

10.5 SETFULLVIEWGRAPHAREA

Use this API to specify the parameters of the scales used for plotting the results of PV and QV analysis.

Batch command syntax:

```
BAT_SETFULLVIEWGRAPHAREA  AREA  VAXISSHOW  VAXISSTYLE  VAXISCLR(1)…
VAXISCLR(3)  VGRIDSHOW  VGRIDSTYLE  VGRIDCLR(1)… VGRIDCLR(3)
HAXISSHOW  HAXISSTYLE  HAXISCLR(1)… HAXISCLR(3)  HGRIDSHOW
HGRIDSTYLE  HGRIDCLR(1)… HGRIDCLR(3)  BORDERSHOW  BORDERSTYLE
BORDERCLR(1)… BORDERCLR(3)  SETAREACLR  AREACLR(1)… AREACLR(3)
```

Python syntax:

```
setfullviewgrapharea(area, vaxisshow, vaxisstyle, vaxisclr,
vgridshow, vgridstyle, vgridclr, haxisshow, haxisstyle, haxisclr,
hgridshow, hgridstyle, hgridclr, bordershow, borderstyle, borderclr,
setareaclr, areaclr)
```

Fortran syntax:

```
CALL SET_FULLVIEW_GRAPH_AREA(area, vaxisshow, vaxisstyle, vaxisclr,
vgridshow, vgridstyle, vgridclr, haxisshow, haxisstyle, haxisclr,
hgridshow, hgridstyle, hgridclr, bordershow, borderstyle, borderclr,
setareaclr, areaclr)
```

where:

Integer AREA	Is the number of the plotting area. The values are as follows: AREA = 0 QV analysis, area for Q(V) plot. AREA = 1 QV analysis, area for mismatch plot. AREA = 2 PV analysis, area for bus voltage. AREA = 3 PV analysis, area for generator MW. AREA = 4 PV analysis, area for branch MVA flow. AREA = 5 PV analysis, area for interface MW flow. AREA = 6 PV analysis, area for bus voltage against interface flow.
Integer VAXISSHOW	To show (or not) the vertical axis.
Integer VAXISSTYLE	The style of the vertical axis. Valid only if VAXISSHOW=1. The values are as follows: VAXISSTYLE = 0 solid line. VAXISSTYLE = 1 dash - dash - dash line. VAXISSTYLE = 2 dot - dot - dot line. VAXISSTYLE = 3 dash - dot line. VAXISSTYLE = 4 dash - dot - dot line.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer VAXISCLR(3)	Is an array of three elements to set the color of the vertical axis. Valid only if VAXISSHOW=1. The values are as follows: VAXISCLR (1) Red component; the value between 0 and 255. VAXISCLR (2) Green component; the value between 0 and 255. VAXISCLR (3) Blue component; the value between 0 and 255.
Integer VGRIDSHOW	To show (or not) the vertical grid.
Integer VGRIDSTYLE	The style of the vertical grid. Valid only if VGRIDSHOW =1. The values are as follows: VGRIDSTYLE = 0 solid line. VGRIDSTYLE = 1 dash - dash - dash line. VGRIDSTYLE = 2 dot - dot - dot line. VGRIDSTYLE = 3 dash - dot line. VGRIDSTYLE = 4 dash - dot - dot line.
Integer VGRIDCLR(3)	Is an array of three elements to set the color of the vertical grid. Valid only if VGRIDSHOW =1. The values are as follows: VGRIDCLR (1) Red component; the value between 0 and 255. VGRIDCLR (2) Green component; the value between 0 and 255. VGRIDCLR (3) Blue component; the value between 0 and 255.
Integer HAXISSHOW	To show (or not) the horizontal axis.
Integer HAXISSTYLE	The style of the horizontal axis. Valid only if HAXISSHOW=1. The values are as follows: HAXISSTYLE = 0 solid line. HAXISSTYLE = 1 dash - dash - dash line. HAXISSTYLE = 2 dot - dot - dot line. HAXISSTYLE = 3 dash - dot line. HAXISSTYLE = 4 dash - dot - dot line.
Integer HAXISCLR(3)	Is an array of three elements to set the color of the horizontal axis. Valid only if HAXISSHOW=1. The values are as follows: HAXISCLR (1) Red component; the value between 0 and 255. HAXISCLR (2) Green component; the value between 0 and 255. HAXISCLR (3) Blue component; the value between 0 and 255.
Integer HGRIDSHOW	To show (or not) the horizontal grid.
Integer HGRIDSTYLE	The style of the horizontal grid. Valid only if HGRIDSHOW =1. The values are as follows: HGRIDSTYLE = 0 solid line. HGRIDSTYLE = 1 dash - dash - dash line. HGRIDSTYLE = 2 dot - dot - dot line. HGRIDSTYLE = 3 dash - dot line. HGRIDSTYLE = 4 dash - dot - dot line.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer HGRIDCLR(3)	Is an array of three elements to set the color of the horizontal grid. Valid only if HGRIDSHOW =1. The values are as follows: HGRIDCLR (1) Red component; the value between 0 and 255. HGRIDCLR (2) Green component; the value between 0 and 255. HGRIDCLR (3) Blue component; the value between 0 and 255.
Integer BORDERSHOW	To show (or not) the border around the plot.
Integer BORDERSTYLE	The style of the border. Valid only if BORDERSHOW =1. The values are as follows: BORDERSTYLE = 0 solid line. BORDERSTYLE = 1 dash - dash - dash line. BORDERSTYLE = 2 dot - dot - dot line. BORDERSTYLE = 3 dash - dot line. BORDERSTYLE = 4 dash - dot - dot line.
Integer BORDERCLR(3)	Is an array of three elements to set the color of the border. Valid only if BORDERSHOW =1. The values are as follows: BORDERCLR (1) Red component; the value between 0 and 255. BORDERCLR (2) Green component; the value between 0 and 255. BORDERCLR (3) Blue component; the value between 0 and 255.
Integer SETAREACLR	To set (or not) the back color of the plotting area.
Integer AREACLR(3)	Is an array of three elements to set the color of the plotting area. Valid only if SETAREACLR =1. The values are as follows: AREACLR (1) Red component; the value between 0 and 255. AREACLR (2) Green component; the value between 0 and 255. AREACLR (3) Blue component; the value between 0 and 255.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

This page intentionally left blank.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

SIEMENS

Chapter 11

PSSPLOT Module

This chapter defines API routines that are used to manipulate plots in the interface. They will elicit no action if the GUI is not present when invoked.

Please note that these routines are members of the pssplot Python extension module, not psspy.

These routines will be recorded when recording is enabled, and the Python recording will reference each of these functions within the pssplot namespace.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

11.1 INSERTPAGE

Use this API to insert a page into the plotting book.

Batch Command Syntax:

```
BAT_INSERTPAGE
```

Python Syntax:

```
insertpage()
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

11.2 INSERTPLOT

Use this API to add an empty plot on the active page in the plotting book.

Batch Command Syntax:

```
BAT_INSERTPLOT
```

Python Syntax:

```
insertplot()
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

11.3 DELETESELPLOT

Use this API to delete the selected plot on the active page in the plotting book.

Batch Command Syntax:

```
BAT_DELETESELPLOT
```

Python Syntax:

```
deleteselplot()
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

11.4 EXPORTPLOT

Use this API to export the selected plot on the active page in the plotting book.

Batch Command Syntax:

```
BAT_EXPORTPLOT  NTYPE SFILENAME
```

Python Syntax:

```
exportplot(ntype, sfilename)
```

where:

Integer NTYPE	is the graphics format for the exported file 1 - Metafile export 2 - Bitmap export 3 - JPEG export 4 - PDF export
CHARACTER*260 SFILENAME	is the filename

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

11.5 PRINTSELPLOT

Use this API to print the selected plot on the active page in the plotting book.

Batch Command Syntax:

```
BAT_PRINTSELPLOT  NINDEX  NORIENT  SFILENAME
```

Python Syntax:

```
printselplot(nindex, norient, sfilename)
```

where:

Integer NINDEX	is the index for the printer you wish to use This index refers to the list of printers defined on your system; default = 1.
Integer NORIENT	is the orientation of the paper 0 - Portrait 1 - Landscape
CHARACTER*260 SFILENAME	is the name of the file to print

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

11.6 SHOWPLOTEDITOR

Use this API to display the editor for the selected plot in the plotting book.

Batch Command Syntax:

```
BAT_SHOWPLOTEDITOR
```

Python Syntax:

```
showploteditor()
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

11.7 NEWPLOTBOOK

Use this API to create a new plotting book.

Batch Command Syntax:

```
BAT_NEWPLOTBOOK
```

Python Syntax:

```
newplotbook()
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

11.8 OPENCHANDATAFILE

Use this API to load a Channel Output File into the plotting book.

Batch Command Syntax:

```
BAT_OPENCHANDATAFILE  SFILENAME
```

Python Syntax:

```
openchandatafile(sfilename)
```

where:

CHARACTER*260 SFILENAME	is the name of the channel file to be opened

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

11.9 DRAGDROPPLOTDATA

Use this API to load a Channel Output File into the plotting book.

Batch Command Syntax:

```
BAT_DRAGDROPPLOTDATA  SFILENAME SCHANNELNAME
```

Python Syntax:

```
dragdropplotdata(sfilename, schannelname)
```

where:

CHARACTER*260 SFILENAME	is the Channel File name
CHARACTER*260 SCHANNELNAME	is the Channel to be dragged to the plot

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

11.10 SETSELECTEDPAGE

Use this API to set the active page in the plotting book

Batch Command Syntax:

```
BAT_SETSELECTEDPAGE   NPAGENUM
```

Python Syntax:

```
setselectedpage(npagenum)
```

where:

INTEGER NPAGENUM	is the page number in the plotting book

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

11.11 SETSELECTEDPLOT

Use this API to set the selected plot on the active page in the plotting book.

Batch Command Syntax:

```
BAT_SETSELECTEDPLOT   NPLOTNUM
```

Python Syntax:

```
setselectedplot (nplotnum)
```

where:

INTEGER NPLOTNUM	is the plot number on the selected page

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

SIEMENS

Chapter 12

Non-Engineering Functions

This chapter defines API routines that provide functions not associated with PSS®E data, analysis, or graphics capabilities.

These routines do not record themselves when called with recording enabled.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

12.1 ACTIVATE_LICENSE

When running Python scripts outside of the PSS®E GUI that uses PSS®E functions, the python executable being used need to be activated with the security code corresponding to PSS®E version. For example, if the *python.exe* program is used to run PSS®E functions in PSS®E version xx, then executable name *python* needs to be activated with the security code corresponding to version xx, where xx is the PSS®E version in use.

By default, psspy module, sets the appropriate security code for commonly used Python executables python.exe, pythonw.exe, and pythonwin.exe. If using any other Python executable, activate the security code for that executable using this function.

Python syntax:

```
ierr = activate_license(names)
```

where:

Character names	Is a string, or a tuple of strings, specifiyng one or more names of Python executables (input; no default allowed). Provide each name separated by a comma and without its ".exe" extension. The name is case-insensitive.
Integer IERR	Is the error code (output).
	IERR = 0 — No error occurred.
	IERR = 1 — Could not get PSS®E version in use.
	IERR = 2 — Could not open PSS®E license key.
	IERR = 3 — Could not get PSS®E license value.
	IERR = 4 — Could not set/update license value for Python executable.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

12.2 ALERT

Use this API to send output to the alert device.

Python syntax:

```
alert(string)
```

Fortran syntax:

```
CALL ALERTS(string)
```

or

```
CALL ALERTS(string-array)
```

where:

Character string*220	Is the string to be written (input).
Character string-array*220	Is the array of strings to be written (input).

Note: A Fortran program that calls ALERTS must also include a USE statement for KEYPTI.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

12.3 BEGINREPORT

Use this API to create a new report tab in the GUI.

Python syntax:

```
beginreport()
```

Fortran syntax:

```
CALL RPTBEG()
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

12.4 FILEIN

Get input from file (Fortran unit number).

Python syntax:

```
ierr, string =filein(lunit)
```

where:

Integer lunit	Is the Fortran logical unit number of the file (input).
Character string*260	Is the buffer input by user (output).
Integer ierr	Is the error code (output). ierr = 0 no error occurred.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

12.5 IOFLUSH

Force all files in PSS®E to flush buffers to disk, if possible.

Python syntax:

```
ioflush()
```

Fortran syntax:

```
CALL IOFLSH()
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

12.6 GETBATDEFAULTS

This API is obsolete. It has been replaced by the APIs GETDEFAULTINT and GETDEFAULTREAL.

Python syntax:

```
_i,_f = getbatdefaults()
```

Fortran syntax:

```
CALL GETBATDEFAULTS(_i, _f)
```

where:

Integer _i	Is the default integer value used in the batch command processor (output).
Real _f	Is the default real value used in the batch command processor (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

12.7 GETDEFAULTCHAR

Retrieve “no input” value for characters (i.e., strings) other than filenames. Specify this value to indicate that “no input” is intended.

Python syntax:

```
_c = getdefaultchar()
```

Fortran syntax:

```
CALL GETDEFAULTCHAR(_c)
```

where:

Character _c*1	Is the “no input” character value. (output)

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

12.8 GETDEFAULTINT

Retrieve “no input” value for integers. Specify this value to indicate that “no input” is intended.

Python syntax:

```
_i = getdefaultint()
```

Fortran syntax:

```
CALL GETDEFAULTINT(_i)
```

where:

Integer _i	Is the “no input” integer value. (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

12.9 GETDEFAULTREAL

Retrieve “no input” value for reals (floats). Specify this value to indicate that “no input” is intended.

Python syntax:

```
_f = getdefaultreal()
```

Fortran syntax:

```
CALL GETDEFAULTREAL(_f)
```

where:

Real _f	Is the “no input” real value. (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

12.10 PAGEREPORT

Use this API to place a report separator on the active report tab of the GUI.

Python syntax:

```
pagereport()
```

Fortran syntax:

```
CALL RPTPGBK()
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

12.11 PROGRESS

Send output to progress device.

Python syntax:

```
progress(string)
```

Fortran syntax:

```
CALL PROGRESS(string)
```

or

```
CALL PROGRESS(string-array)
```

where:

Character string*220	Is the string to be written (input).
Character string-array*220	Is the array of strings to be written (input).

Note: A Fortran program that calls PROGRESS must also include a USE statement for KEYPTI.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

12.12 PROMPT

Send output to prompt device. Prompts are stored until the next terminal input request. Up to 36 buffers of up to 220 characters can be stored.

Python syntax:

```
prompt(string)
```

Fortran syntax:

```
CALL PROMPTS(string)
```

or

```
CALL PROMPTS(string-array)
```

where:

Character string*220	Is the string to be written (input).
Character string-array*220	Is the array of strings to be written (input).

Note: A Fortran program that calls PROMPTS must also include a USE statement for KEYPTI.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

12.13 PSSEINIT

Initialize PSS®E.

Python syntax:

```
ierr = psseinit(buses)
```

Fortran syntax:

```
CALL PSSEINIT(BUSES, IERR)
```

where:

Integer buses	Is the requested bus size; zero for default size level (input).
Integer ierr	Is the error code (output). This is future use. No errors are currently defined.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

12.14 PSSEVERSION

Get PSS®E version information.

Python syntax:

```
name,major,minor,modlvl,date,stat = psseversion()
```

Fortran syntax:

```
CALL PSSEVERSION(NAME, MAJOR, MINOR, MODLVL, DATA, STAT)
```

where:

Character name*32	Is the product name (output).
Integer major	Is the major release number (output).
Integer minor	Is the minor release number (output).
Integer modlvl	Is the modification release number (output).
Character date*30	Is the release date. (output).
Character stat*132	(reserved for future use) (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

12.15 REFRESHGUI

Refresh the graphical user interface.

Python syntax:

```
refreshgui()
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

12.16 REPORT

Send output to report device.

Python syntax:

```
report(string)
```

Fortran syntax:

```
CALL REPORTS(string)
```

or

```
CALL REPORTS(string-array)
```

where:

Character string*220	Is the string to be written (input).
Character string-array*220	Is the array of strings to be written (input).

Note: A Fortran program that calls REPORTS must also include a USE statement for KEYPTI.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

12.17 RUNIPLANFILE

Execute an IPLAN program.

Python syntax:

```
runiplanfile(iplname)
```

where:

Character iplname*260	Is the filename (input). The filename can be followed by arguments as part of a single string, in which case the filename must be delimited (e.g. enclosed in quotes). The -arg option can be specified this way as well.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

12.18 RUNRSPNSFILE

Run a response file. Replace standard input with a file. Continue reading from terminal until standard input is no longer redirected using PSS®E command line techniques (allowing interaction with IPLAN programs, for example). Pass any results to CLI.

Python syntax:

```
runrspnsfile(rspname)
```

where:

Character rspname*260	Is the filename (input). The filename can be followed by arguments as part of a single string, in which case the filename must be delimited (e.g. enclosed in quotes).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

12.19 STARTRECORDING

Start recording program operation.

Python syntax:

```
ierr = startrecording(iopt,string)
```

where:

Integer iopt	Is the type of format to be recorded (input). iopt = 0 record in batch (BAT_) format. iopt = 1 record in python format.
Character string*260	Is the filename (input).
Integer ierr	Is the error code (output). ierr = 0 no error occurred. ierr = 1 error closing file. ierr = 2 not recording.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

12.20 STOPRECORDING

Stop recording program operation.

Python syntax:

```
ierr = stoprecording()
```

where:

Integer ierr*256	Is the error code (output).	
	ierr = 0	no error occurred.
	ierr = 1	error closing file.
	ierr = 2	not recording.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

12.21 USERIN

Get input from interactive device.

Python syntax:

```
ierr, string = userin()
```

where:

Character string	Is the buffer input by user (output).
Integer ierr	Is the error code (output). ierr = 0 no error occurred.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

SIEMENS

Chapter 13

Scenario and Event Study APIs

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.1 OPENEVENTSTUDYFILE

Use this API to open a new or existing Event Study file in the GUI.

Batch command syntax:

```
BAT_OPENEVENTSTUDYFILE SFILE
```

Python syntax:

```
openeventstudyfile(sfile)
```

where:

Character SFILE*260	Is the name of the new or existing event study file (input).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.2 SAVEEVENTSTUDYFILE

Use this API to save existing Event Studies in an Event Study file.

Batch command syntax:

```
BAT_SAVEEVENTSTUDYFILE SFILE
```

Python syntax:

```
saveeventstudyfile(sfile)
```

where:

Character SFILE*260	Is the name of the destination event study file (input).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.3 CLOSEEVENTSTUDYFILE

Use this API to close an open Event Study file.

Batch command syntax:

```
BAT_CLOSEEVENTSTUDYFILE
```

Python syntax:

```
closeeventstudyfile()
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.4 SETACTIVEEVENTSTUDY

Use this API to select an Event Study in an Event file as the active Event Study for running dynamic or power flow event studies.

Batch command syntax:

```
BAT_SETACTIVEEVENTSTUDY SNAME
```

Python syntax:

```
setactiveeventstudy(sname)
```

where:

Character SNAME*260	Is the name of an Event Study in the open Event Study file (input).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.5 RUNPOWERFLOWEVENTSTUDY

Use this API to run a power flow study using the active Event Study in the open Event Study file.

Batch command syntax:

```
BAT_RUNPOWERFLOWEVENTSTUDY
```

Python syntax:

```
runpowerfloweventstudy()
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.6 RUNDYNAMICEVENTSTUDY

Use this API to run a dynamic study using the active Event Study in the open Event Study file.

Batch command syntax:

```
BAT_RUNDYNAMICEVENTSTUDY
```

Python syntax:

```
rundynamiceventstudy()
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.7 NEWSCENARIOFILE

This API is obsolete. It has been replaced by the API SCENARIONEW, and is implemented by a call to the SCENARIONEW.

Use this API to create a new Scenario file. For more information see SCENARIONEW.

Batch command syntax:

```
BAT_NEWSCENARIOFILE SFILE
```

Python syntax:

```
newscenariofile(sfile)
```

where:

Character SFILE*260	Is the name of the new scenario file (input).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.8 OPENSCENARIOFILE

This API is obsolete. It has been replaced by the API SCENARIOOPEN, and is implemented by a call to the SCENARIOOPEN.

Use this API to open an existing Scenario. For more information see SCENARIOOPEN.

Batch command syntax:

```
BAT_OPENSCENARIOFILE SFILE
```

Python syntax:

```
openscenariofile(sfile)
```

where:

Character SFILE*260	Is the name of an existing scenario file (input).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.9 SAVESCENARIOFILE

This API is obsolete. It has been replaced by the API SCENARIOSAVE, and is implemented by a call to the SCENARIOSAVE.

Use this API to save the current Scenario. For more information see SCENARIOSAVE.

Batch command syntax:

```
BAT_SAVESCENARIOFILE SFILE
```

Python syntax:

```
savescenariofile(sfile)
```

where:

Character SFILE*260	Is the name of the destination scenario file (input).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.10 CLOSESCENARIOFILE

This API is obsolete. It has been replaced by the API SCENARIOCLOSE, and is implemented by a call to the SCENARIOCLOSE.

Use this API to close the current Scenario. For more information see SCENARIOCLOSE..

Batch command syntax:

```
BAT_CLOSESCENARIOFILE
```

Python syntax:

```
closescenariofile()
```

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.11 SCENARIONEW

Use this API to create a new Scenario. At least one group must be added to the Scenario through SCENARIOGROUPADD before files may be added to it. If adding files through file tracking is also desired, a group must become the active group which can be done by calling SCENARIOGROUPOPEN. <<<Greg, index marker>>>>

Batch command syntax:

```
BAT_SCENARIONEW SFILE
```

Python syntax:

```
ierr = scenarionew(sfile)
```

where:

Character SFILE*260	Is the full file path of the new scenario file (input; blank by default). The name of the Scenario will be determined by the file name portion of the full file path. When the default is used no name is given to the Scenario and a full file name will be required upon saving. The *root path* of the Scenario will be determined by the path portion of the full file path. When the default is used the *root path* will default to the current working directory.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.12 SCENARIOOPEN

Use this API to open a previously created Scenario.<<<<Greg Index Marker>>>

Batch command syntax:

```
BAT_SCENARIOOPEN SFILE SGROUP
```

Python syntax:

```
ierr = scenarioopen(sfile, sgroup)
```

Fortran syntax:

```
CALL SCENARIOOPEN(SFILE, SGROUP, IERR)
```

where:

Character SFILE*260	Is the full file path of the scenario file to open (input; no default).
Character SGROUP*72	Is the name of the group to open (input; ignored by default). When the default is used the *startup group* will be opened if one is defined in the Scenario. If the name of a group in the Scenario is given, that group will be opened. If the string is blank or does not match any of the group names in the Scenario, no group is opened.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred. IERR = 2 file path is invalid.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.13 SCENARIOSAVE

Use this API to save the currently open Scenario.

Batch command syntax:

```
BAT_SCENARIOSAVE SFILE
```

Python syntax:

```
ierr = scenariosave(sfile)
```

Fortran syntax:

```
CALL SCENARIOSAVE (SFILE, IERR)
```

where:

Character SFILE*260	Is the full file path to save the scenario file to (input; unchanged by default). When the default is used the Scenario will be saved to the last filename given in a SCENARIONEW, SCENARIOOPEN, or SCENARIOSAVE operation, if a call to SCENARIONEW with default arguments was the previous call, the Scenario will not be saved due to no previous filename being available.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred. IERR = 1 no Scenario in memory. IERR = 2 file path is invalid.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.14 SCENARIOCLOSE

Use this API to close the currently open Scenario. If a file in the active group was specified as "Load Last", this file will be opened prior to the closing of the Scenario.

Batch command syntax:

```
BAT_SCENARIOCLOSE
```

Python syntax:

```
ierr = scenarioclose()
```

Fortran syntax:

```
CALL SCENARIOCLOSE (IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR > 0	fatal error occurred.
	IERR = 1	no Scenario in memory.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.15 SCENARIOISVALID

Use this API to determine if there is a valid Scenario in memory.

Python syntax:

```
isvalid = scenarioisvalid()
```

Fortran syntax:

```
CALL SCENARIOISVALID (ISVALID)
```

where:

Logical ISVALID	TRUE when a valid Scenario is in memory, currently returns FALSE when in Zip Mode (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.16 SCENARIOISDIRTY

Use this API to determine if there are any unsaved changes in the current Scenario.

Python syntax:

```
isdirty = scenarioisdirty()
```

Fortran syntax:

```
CALL SCENARIOISDIRTY (ISDIRTY)
```

where:

Logical ISDIRTY	TRUE when the Scenario in memory currently contains unsaved changes (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.17 SCENARIOZIPACTIVE

Use this API to determine if the Scenario is currently in Zip Mode.

Python syntax:

```
active = scenariozipactive()
```

Fortran syntax:

```
CALL SCENARIOZIPACTIVE (ACTIVE)
```

where:

Logical ACTIVE	TRUE when the Scenario is in Zip Mode (output).

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.18 SCENARIOGETATTRS

Use this API to retrieve the attributes of the Scenario.

Python syntax:

```
ierr, rootpath, startupgroup, creationdate, createdby, changedate,
changedby, name, version, versionpsse = scenariogetattrs()
```

Fortran syntax:

```
CALL SCENARIOGETATTRS (ROOTPATH, STARTUPGROUP, CREATIONDATE,
CREATEDBY, CHANGEDATE, CHANGEDBY, NAME, VERSION, VERSIONPSSE, IERR)
```

where:

Character ROOTPATH *260	Is the root path of the Scenario (output).
Character STARTUPNAME*72	Is the name of the startup group (output).
Character CREATIONDATE*24	Is the date of creation (output).
Character CREATEDBY*256	Is the username of the creator (output).
Character CHANGEDATE *24	Is the last date of modification (output).
Character CHANGEDBY *256	Is the username of the last modifier (output).
Character NAME*260	Is the name of the Scenario (output).
Character VERSION*8	Is the version of Scenario the file was saved under (output).
Character VERSIONPSSE*8	Is the version of PSSE the file was saved under (output).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR > 0 fatal error occurred.
	IERR = 1 no Scenario in memory.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.19 SCENARIOSETATTRS

Use this API to modify the *root path* and *startup group* attributes of the Scenario.

Batch command syntax:

```
BAT_SCENARIOSETATTRS ROOTPATH STARTUPGROUP
```

Python syntax:

```
ierr = scenariosetattrs (rootpath, startupgroup)
```

Fortran syntax:

```
CALL SCENARIOSETATTRS (ROOTPATH, STARTUPGROUP, IERR)
```

where:

Character ROOTPATH *260	Is the directory to be used as the *root path* of the Scenario (input; unchanged by default).
Character STARTUPNAME*72	Is the name of the group to be designated as the startup *group* (input; unchanged by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred. IERR = 1 no Scenario in memory. IERR = 3 specified group not found.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.20 SCENARIOLENGTHCOMMENTS

Use this API to determine the buffer size needed to hold the current Scenario comments.

Python syntax:

```
ierr, length = scenariolengthcomments()
```

Fortran syntax:

```
CALL SCENARIOLENGTHCOMMENTS (LENGTH, IERR)
```

where:

Integer LENGTH	Is the length of the Scenario comment string (output).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred. IERR = 1 no Scenario in memory.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.21 SCENARIOGETCOMMENTS

Use this API to retrieve the Scenario comments. For all non-python calls, the array COMMENTS must be allocated a length of at least LENGTH from the SCENARIOLENGTHCOMMENTS API.

Python syntax:

```
ierr, comments = scenariogetcomments()
```

Fortran syntax:

```
CALL SCENARIOGETCOMMENTS (COMMENTS, IERR)
```

where:

Character COMMENTS	Is the buffer to hold the comment string for the Scenario (output).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred. IERR = 1 no Scenario in memory.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.22 SCENARIOSETCOMMENTS

Use this API to define or modify the Scenario comments.

Batch command syntax:

```
BAT_SCENARIOSETCOMMENTS NCOMMENTBLOCKS COMMENTS APPEND
```

Python syntax:

```
ierr = scenariosetcomments (ncommentblocks, comments, append)
```

Fortran syntax:

```
CALL SCENARIOSETCOMMENTS (NCOMMENTBLOCKS , COMMENTS, APPEND, IERR)
```

where:

Integer NCOMMENTBLOCKS	Is the number of 72 character strings that make up the comment. In python, this number can be set to 1 and the full comment string given as the first comment block (input; 0 by default).
Character COMMENTS(NCOMMENTBLOCKS)*72	Is the comment string to be defined for or added to the comments for the Scenario (input; unchanged by default).
Logical APPEND	Is the flag for appending comments. If TRUE, COMMENTS is appended to the current Scenario comments. If FALSE, the current Scenario comments are overwritten by COMMENTS (input; FALSE by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred. IERR = 1 no Scenario in memory.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.23 SCENARIOGROUPOPEN

Use this API to add a group to the Scenario.

Batch command syntax:

```
BAT_SCENARIOGROUPOPEN SGROUP
```

Python syntax:

```
ierr = scenariogroupopen (sgroup)
```

Fortran syntax:

```
CALL SCENARIOGROUPOPEN (SGROUP, IERR)
```

where:

Character SGROUP*72	Is the name of the group to open (input; ignored by default). When default is used the startup group will be opened if one is defined in the Scenario. If the name of a group in the Scenario is given, that group will be opened. If the string is blank or does not match any of the group names in the Scenario no group is opened.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred. IERR = 1 no Scenario in memory. IERR = 3 specified group not found

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.24 SCENARIOGROUPRENAME

Use this API to rename one of the groups in the Scenario.

Batch command syntax:

```
BAT_SCENARIOGROUPRENAME SGROUP SNEWNAME
```

Python syntax:

```
ierr = scenariogrouprename (sgroup, snewname)
```

Fortran syntax:

```
CALL SCENARIOGROUPRENAME (SGROUP, SNEWNAME, IERR)
```

where:

Character SGROUP*72	Is the name of the group to be renamed (input; no default allowed).
Character SNEWNAME*72	Is the new name to be given to the specified group (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no fatal error occurred. IERR > 0 fatal error occurred. IERR = 1 no Scenario in memory. IERR = 2 invalid group name. IERR = 3 specified group not found.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.25 SCENARIOGROUPLENGTHCOMMENTS

Use this API to determine the buffer size needed to hold the comments for the specified group.

Python syntax:

```
ierr, length = scenariogrouplengthcomments (sgroup)
```

Fortran syntax:

```
CALL SCENARIOLENGTHCOMMENTS (SGROUP, LENGTH, IERR)
```

where:

Character SGROUP*72	Is the name of the group whose comment length is to be returned (input; no default allowed).
Integer LENGTH	Is the length of the comment string for the specified group (output).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred. IERR = 1 no Scenario in memory. IERR = 3 specified group not found.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.26 SCENARIOGROUPSETCOMMENTS

Use this API to define or modify the comments for the specified group.

Batch command syntax:

```
BAT_SCENARIOGROUPSETCOMMENTS SGROUP NCOMMENTBLOCKS COMMENTS APPEND
```

Python syntax:

```
ierr = scenariogroupsetcomments (sgroup, ncommentblocks, comments,
append)
```

Fortran syntax:

```
CALL SCENARIOGROUPSETCOMMENTS (SGROUP, NCOMMENTBLOCKS, COMMENTS,
APPEND, IERR)
```

where:

Character SGROUP*72	Is the name of the group whose comments are to be modified (input; no default allowed).
Integer NCOMMENTBLOCKS	Is the number of 72 character strings that make up the comment. In python, this number can be set to 1 and the full comment string given as the first comment block (input; 0 by default).
Character COMMENTS (NCOMMENTBLOCKS)*72	Is the comment string to be defined for or added to the comments for the specified group (input; unchanged by default).
Logical APPEND	Is the flag for appending comments. If TRUE, COMMENTS is appended to the current comments. If FALSE, the current comments are overwritten by COMMENTS (input; FALSE by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred. IERR = 1 no Scenario in memory. IERR = 3 specified group not found.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.27 SCENARIOGROUPCOUNT

Use this API to retrieve the number of groups in the Scenario.

Batch command syntax:

```
BAT_SCENARIOGROUPADD SGROUP
```

Python syntax:

```
ierr, ngroups = scenariogroupcount()
```

Fortran syntax:

```
CALL SCENARIOGROUPCOUNT (NGROUPS, IERR)
```

where:

Integer NGROUPS	Is the number of the group in the Scenario (output).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred. IERR = 1 no Scenario in memory.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.28 SCENARIOGROUPLIST

Use this API to retrieve the list of groups in the Scenario.

Python syntax:

```
ierr, sgroups = scenariogrouplist()
```

Fortran syntax:

```
CALL SCENARIOGROUPLIST (NGROUPS, SGROUPS, IERR)
```

where:

Integer NGROUPS	Is the max number of groups to list (input).
Character SGROUPS(NGROUPS)*72	Is the list of groups in the Scenario (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred. IERR = 1 no Scenario in memory.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.29 SCENARIOGROUPADD

Use this API to add a group to the Scenario.

Batch command syntax:

```
BAT_SCENARIOGROUPADD SGROUP
```

Python syntax:

```
ierr = scenariogroupadd (sgroup)
```

Fortran syntax:

```
CALL SCENARIOGROUPADD (SGROUP, IERR)
```

where:

Character SGROUP*72	Is the name of the new group to add to the Scenario (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred. IERR = 1 no Scenario in memory. IERR = 2 invalid group name.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.30 SCENARIOGROUPREMOVE

Use this API to remove a group from the Scenario.

Batch command syntax:

```
BAT_SCENARIOGROUPREMOVE SGROUP
```

Python syntax:

```
ierr = scenariogroupremove (sgroup)
```

Fortran syntax:

```
CALL SCENARIOGROUPREMOVE (SGROUP, IERR)
```

where:

Character SGROUP*72	Is the name of the group to be removed from the Scenario (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred. IERR = 1 no Scenario in memory. IERR = 3 specified group not found.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.31 SCENARIOFILEGETATTRS

Use this API to retrieve the attributes of the specified file.

Python syntax:

```
ierr, stype, load, sargs = scenariofilegetattrs (sgroup, sfile)
```

Fortran syntax:

```
CALL SCENARIOFILEGETATTRS (SGROUP, SFILE, STYPE, LOAD, SARGS, IERR)
```

where:

Character SGROUP*72	Is the group that contains the specified file (input; no default allowed).
Character SFILE*260	Is the name of the file whose attributes are to be retrieved (input; no default allowed).
Character STYPE*8	Is the file type identifier for the specified file (output).
Integer LOAD	Denotes the file's position in the load order of its group. A load order of 0 specifies that the file is not loaded (output).
Character SARGS*260	Is the argument string to be used for the file when automatically opening the file (output).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred. IERR = 1 no Scenario in memory. IERR = 3 specified group not found. IERR = 4 specified file not found.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.32 SCENARIOFILELENGTHCOMMENTS

Use this API to determine the buffer size needed to hold the comments for the specified file.

Python syntax:

```
ierr, length = scenariofilelengthcomments (sgroup, sfile)
```

Fortran syntax:

```
CALL SCENARIOFILELENGTHCOMMENTS (SGROUP, SFILE, LENGTH, IERR)
```

where:

Character SGROUP*72	Is the group that contains the specified file (input; no default allowed).
Character SFILE*260	Is the name of the file whose comment length is to be returned (input; no default allowed).
Integer LENGTH	Is the length of the Scenario comment string (output).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred. IERR = 1 no Scenario in memory. IERR = 3 specified group not found. IERR = 4 specified file not found.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.33 SCENARIOFILEGETCOMMENTS

Use this API to retrieve the comments for the specified file. For all non-python calls, the array COMMENTS must be allocated a length of at least LENGTH for the SCENARIOFILELENGTH-COMMENTS API.

Python syntax:

```
ierr, comments = scenariofilegetcomments (sgroup, sfile)
```

Fortran syntax:

```
CALL SCENARIOFILEGETCOMMENTS (SGROUP, SFILE, COMMENTS, IERR)
```

where:

Character SGROUP*72	Is the group that contains the specified file (input; no default allowed).
Character SFILE*260	Is the name of the file whose attributes are to be retrieved (input; no default allowed).
Character COMMENTS	Is the buffer to hold the comment string for the specified file (output).
Integer LOAD	Denotes the file's position in the load order of its group. A load order of 0 specifies that the file is not loaded (output).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred. IERR = 1 no Scenario in memory. IERR = 3 specified group not found. IERR = 4 specified file not found.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.34 SCENARIOFILESETCOMMENTS

Use this API to define or modify the comments for the specified file.

Batch command syntax:

```
BAT_SCENARIOFILESETCOMMENTS SGROUP SFILE NCOMMENTBLOCKS COMMENTS
APPEND
```

Python syntax:

```
ierr = scenariofilesetcomments (sgroup, sfile, ncommentblocks,
comments, append)
```

Fortran syntax:

```
CALL SCENARIOFILESETCOMMENTS (SGROUP, SFILE, NCOMMENTBLOCKS,
COMMENTS, APPEND, IERR)
```

where:

Character SGROUP*72	Is the group that contains the specified file (input; no default allowed).
Character SFILE*260	Is the name of the file whose attributes are to be modified (input; no default allowed).
Integer NCOMMENTBLOCKS	Is the number of 72 character strings that make up the comment. In python, this number can be set to 1 and the full comment string given as the first comment block (input; 0 by default).
Character COMMENTS (NCOMMENTBLOCKS)*72	Is the comment string to be defined for or added to the comments for the specified file (input; unchanged by default).
Logical APPEND	Is the flag for appending comments. If TRUE, COMMENTS is appended to the current comments. If FALSE, the current comments are overwritten by COMMENTS (input; FALSE by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred. IERR = 1 no Scenario in memory. IERR = 3 specified group not found. IERR = 4 specified file not found.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.35 SCENARIOFILECOUNT

Use this API to retrieve the number of files in a specified group in the Scenario.

Python syntax:

```
ierr, nfiles = scenariofilecount (sgroup)
```

Fortran syntax:

```
CALL SCENARIOFILECOUNT (SGROUP, NFILES, IERR)
```

where:

Character SGROUP*72	Is the group to retrieve the file count of input; no default allowed.
Integer NFILES	Is the number of the files in the specified group (output).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR > 0 fatal error occurred.
	IERR = 1 no Scenario in memory.
	IERR = 3 specified group not found.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.36 SCENARIOFILELIST

Use this API to retrieve the list of files in the specified group.

Python syntax:

```
ierr, sfiles = scenariofilelist (sgroup)
```

Fortran syntax:

```
CALL SCENARIOFILELIST (SGROUP, NFILES, SFILES, IERR)
```

where:

Character SGROUP*72	Is the group to retrieve the file list of (input; no default allowed).
Integer NFILES	Is the max number of files to list (input; no default allowed).
Character SFILES(NFILES)*260	Is the list of files in the specified group (output).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred. IERR = 1 no Scenario in memory. IERR = 3 specified group not found.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.37 SCENARIOFILEADD

Use this API to add a file to the specified group.

Batch command syntax:

```
BAT_SCENARIOFILEADD SGROUP SFILE STYPE LOAD SARGS
```

Python syntax:

```
ierr = scenariofileadd (sgroup, sfile, stype, load, sargs)
```

Fortran syntax:

```
CALL SCENARIOFILEADD (SGROUP, SFILE, STYPE, LOAD, SARGS, IERR)
```

where:

Character SGROUP*72	Is the name of the group to add a file to (input; no default allowed).
Character SFILE*260	Is the name of the file to add (input; no default allowed).
Character STYPE*8	Is the file type identifier for the specified file (input; blank by default).
Integer LOAD	Is the file's position in the load order of its group. A load order of 0 specifies that the file is not loaded (input; 0 by default).
Character SARGS*260	Is the argument string to be used for the file when automatically opening the file (input; blank by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred. IERR = 1 no Scenario in memory. IERR = 2 invalid file name. IERR = 3 specified group not found.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.38 SCENARIOFILEREMOVE

Use this API to remove a file from the specified group.

Batch command syntax:

```
BAT_SCENARIOFILEREMOVE SGROUP SFILE
```

Python syntax:

```
ierr = scenariofileremove (sgroup, sfile)
```

Fortran syntax:

```
CALL SCENARIOFILEREMOVE (SGROUP, SFILE, IERR)
```

where:

Character SGROUP*72	Is the name of the group to remove a file from (input; no default allowed).
Character SFILE*260	Is the name of the file to be removed (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred. IERR = 1 no Scenario in memory. IERR = 2 invalid file name. IERR = 3 specified group not found. IERR = 4 specified file not found.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.39 SCENARIOMODELGROUP

Use this API to create a new group that is identical to another group.

Batch command syntax:

```
BAT_SCENARIOMODELGROUP SGROUP SNEWNAME
```

Python syntax:

```
ierr = scenariomodelgroup (sgroup, snewname)
```

Fortran syntax:

```
CALL SCENARIOMODELGROUP (SGROUP, SNEWNAME, IERR)
```

where:

Character SGROUP*72	Is the name of the group to be copied (input; no default allowed).
Character SNEWNAME*72	Is the name of the new group (input; no default allowed).
Integer IERR	Is the error code (output).
	IERR = 0 no error occurred.
	IERR > 0 fatal error occurred.
	IERR = 1 no Scenario in memory.
	IERR = 2 invalid file name.
	IERR = 3 specified group not found.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.40 SCENARIOMOVEFILE

Use this API to move a file from one group to another group.

Batch command syntax:

```
BAT_SCENARIOMOVEFILE SRCGROUP SFILE DSTGROUP
```

Python syntax:

```
ierr = scenariomovefile(srcgroup, sfile, dstgroup)
```

Fortran syntax:

```
CALL SCENARIOMOVEFILE(SRCGROUP, SFILE, DSTGROUP, IERR)
```

where:

Character SRCGROUP*72	Is the name of the group containing the file to be moved (input; no default allowed).
Character SFILE*260	Is the name of the file to be moved (input; no default allowed).
Character DSTGROUP*72	Is the name of the group the file will be moved to (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred. IERR = 1 no Scenario in memory. IERR = 2 invalid file name. IERR = 3 specified group not found. IERR = 4 specified file not found.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.41 SCENARIOCOPYFILE

Use this API to copy a file from one group to another group.

Batch command syntax:

```
BAT_SCENARIOCOPYFILE SRCGROUP SFILE DSTGROUP
```

Python syntax:

```
ierr = scenariocopyfile(srcgroup, sfile, dstgroup)
```

Fortran syntax:

```
CALL SCENARIOCOPYFILE(SRCGROUP, SFILE, DSTGROUP, IERR)
```

where:

Character SRCGROUP*72	Is the name of the group containing the file to be copied (input; no default allowed).
Character SFILE*260	Is the name of the file to be copied (input; no default allowed).
Character DSTGROUP*72	Is the name of the group the file will be copied to (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred. IERR = 1 no Scenario in memory. IERR = 2 invalid file name. IERR = 3 specified group not found. IERR = 4 specified file not found.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.42 SCENARIOSETOPT

Use this API to modify the Scenario options. Note: Scenario settings are not attached to the Scenario file. They are user preferences that are saved on the user's machine.

Batch command syntax:

```
BAT_SCENARIOSETOPT TRACKOPT UZIPROOTOPT UZIPLCLPLCOPT UZIPLCLOPT
UZIPRMTEPLCOPT UZIPRMTEOPT
```

Python syntax:

```
ierr = scenariosetopt(trackopt, uziprootopt, uziplclplcopt,
uziplclopt, uziprmteplcopt, uziprmteopt)
```

Fortran syntax:

```
CALL SCENARIOSETOPT(TRACKOPT, UZIPROOTOPT, UZIPLCLPLCOPT, UZIPLCLOPT,
UZIPRMTEPCOPT, UZIPRMTEOPT, IERR)
```

where:

Integer TRACKOPT	Is the setting for file tracking (input; unchanged by default). TRACKOPT=0 means the user will not be prompted about tracked files. If there are files that are currently being tracked the user will be prompted as soon as possible for those files. TRACKOPT=1 means the user will be prompted as soon as possible when a new file is being tracked TRACKOPT=2 means the user will be prompted about tracked files when the current active group is being closed TRACKOPT=3 means tracked files will be added to the current active group immediately. If there are files that are currently being tracked the user will be prompted as soon as possible for those files.
Integer UZIPROOTOPT	Is the setting for conflict resolution for root files (input; unchanged by default). UZIPROOTOPT=0 means skip the file. UZIPROOTOPT=1 means prompt the user for action. UZIPROOTOPT=2 means overwrite the file on disk.
Integer UZIPLCLPLCOPT	Is the setting for placement of local files (input; unchanged by default). UZIPLCLPLCOPT=0 means do not unzip. UZIPLCLPLCOPT=1 means unzip to the root path. UZIPLCLPLCOPT=2 means unzip to the same path the file was at before the file was zipped. UZIPLCLPLCOPT=3 means unzip to the same path the file was at before, relative to the root path.
Integer UZIPLCLOPT	Is the setting for conflict resolution for local files (input; unchanged by default). Same values as UZIPROOTOPT.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer UZIPRMTEPLCOPT	Is the setting for placement of remote files (input; unchanged by default). UZIPRMTEPLCOPT=0 means do not unzip. UZIPRMTEPLCOPT=1 means unzip to the root path. UZIPRMTEPLCOPT=2 means unzip to the same path the file was at before the file was zipped. Note: Cannot choose to unzip remote files by relative path
Integer UZIPRMTEOPT	Is the setting for conflict resolution for remote files (input; unchanged by default). Same values as UZIPROOTOPT.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.43 SCENARIOGETOPT

Use this API to retrieve the Scenario options. Note: Scenario settings are not attached to the Scenario file. They are user preferences that are saved on the user's machine.

Python syntax:

```
ierr, trackopt, uziprootopt, uziplclplcopt, uziplclopt,
uziprmteplcopt, uziprmteopt = scenariogetopt ()
```

Fortran syntax:

```
CALL SCENARIOGETOPT (TRACKOPT, UZIPROOTOPT, UZIPLCLPLCOPT,
UZIPLCLOPT, UZIPRMTEPLCOPT, UZIPRMTEOPT, IERR)
```

where:

Integer TRACKOPT	Is the setting for file tracking (output). TRACKOPT=0 means tracking is turned off. TRACKOPT=1 means the user will be prompted as soon as possible when a new file is being tracked TRACKOPT=2 means the user will be prompted about tracked files when the current active group is being closed TRACKOPT=3 means tracked files will be added to the current active group immediately.
Integer UZIPROOTOPT	Is the setting for conflict resolution for root files (output). UZIPROOTOPT=0 means skip the file. UZIPROOTOPT=1 means prompt the user for action. UZIPROOTOPT=2 means overwrite the file on disk.
Integer UZIPLCLPLCOPT	Is the setting for placement of local files (output). UZIPLCLPLCOPT=0 means do not unzip. UZIPLCLPLCOPT=1 means unzip to the root path. UZIPLCLPLCOPT=2 means unzip to the same path the file was at before the file was zipped. UZIPLCLPLCOPT=3 means unzip to the same path the file was at before, relative to the root path.
Integer UZIPLCLOPT	Is the setting for conflict resolution for local files (output). Same values as UZIPROOTOPT.
Integer UZIPRMTEPCOPT	Is the setting for placement of remote files (output). UZIPRMTEPCOPT=0 means do not unzip. UZIPRMTEPCOPT=1 means unzip to the root path. UZIPRMTEPCOPT=2 means unzip to the same path the file was at before the file was zipped. Note: Cannot choose to unzip remote files by relative path
Integer UZIPRMTEOPT	Is the setting for conflict resolution for remote files (output). Same values as UZIPROOTOPT.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Integer IERR	Is the error code (output).	
	IERR = 0	no error occurred.
	IERR > 0	fatal error occurred.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.44 SCENARIOTRACKFILENUM

Use this API to retrieve the current number of files being tracked.

Python syntax:

```
ierr, nfiles = scenariotrackfilenum()
```

Fortran syntax:

```
CALL SCENARIOTRACKFILENUM (NFILES, IERR)
```

where:

Integer NFILES	Is the number of files currently being tracked (output).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.45 SCENARIOTRACKFILELIST

Use this API to retrieve the list of files being tracked.

Python syntax:

```
ierr, sfiles, stypes = scenariotrackfilelist()
```

Fortran syntax:

```
CALL SCENARIOTRACKFILELIST (NFILES, SFILES, STYPES, IERR)
```

where:

Integer NFILES	Is the max number of files currently being tracked to return (input; no default allowed).
Character SFILES(NFILES)*260	Is the list of files currently being tracked (output).
Character STYPES(NFILES)*8	Is the list of file type identifiers for each file being tracked (output).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.46 SCENARIOTRACKFILEADD

Use this API to add the files currently being tracked.

Batch command syntax:

```
BAT_ SCENARIOTRACKFILEADD NFILES INDEXES DISCARD
```

Python syntax:

```
ierr = scenariotrackfileadd (nfiles, indexes, discard)
```

Fortran syntax:

```
CALL SCENARIOTRACKFILEADD (NFILES, INDEXES, DISCARD, IERR)
```

where:

Integer NFILES	Is the number of files currently being tracked to add (input; no default allowed).
Integer INDEXES(NFILES)	Is the array of indexes of the files to add based on the list from SCENARIOTRACKFILEADD (input; no default allowed).
Logical DISCARD	Is a flag to specify whether or not to discard the files being tracked that were not added (input; FALSE by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.47 SCENARIOZIP

Use this API to zip up the current Scenario.

Batch command syntax:

```
BAT_ SCENARIOZIP SFILE
```

Python syntax:

```
ierr = scenariozip (sfile)
```

Fortran syntax:

```
CALL SCENARIOZIP (SFILE, IERR)
```

where:

Character SFILE*260	Is the full file path of the file to zip the scenario to (input; unchanged by default). When the default is used the name of the zip will be the same name as the Scenario but with the ZIP extension instead of the PSSXML extension. If a different name is given, the Scenario's name and the Scenario PSSXML file name will be changed to reflect the new name.
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred. IERR = 1 no Scenario in memory.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.48 SCENARIOZIPOPEN

Use this API to open a zipped Scenario.

Batch command syntax:

```
BAT_ SCENARIOZIPOPEN SFILE SGROUP
```

Python syntax:

```
ierr = scenariozipopen (sfile, sgroup)
```

Fortran syntax:

```
CALL SCENARIOZIPOPEN (SFILE, SGROUP, IERR)
```

where:

Character SFILE*260	Is the full file path of the scenario file to open (input; no default).
Character SGROUP*72	Reserved for future use (input; ignored by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.49 SCENARIOZIPSAVE

Use this API to save a zipped Scenario.

Batch command syntax:

```
BAT_SCENARIOZIPSAVE FILENAME
```

Python syntax:

```
ierr = scenariozipsave(filename)
```

Fortran syntax:

```
CALL SCENARIOZIPSAVE (FILENAME, IERR)
```

where:

Character FILENAME*260	Is the full file path of the scenario file to save to (input; no default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.50 SCENARIOUNZIP

Use this API to unzip a zipped Scenario.

Batch command syntax:

```
BAT_ SCENARIOUNZIP
```

Python syntax:

```
ierr = scenariounzip()
```

Fortran syntax:

```
CALL SCENARIOUNZIP (IERR)
```

where:

Integer IERR	Is the error code (output).	
	IERR = 0	no conflicts occurred.
	IERR > 0	conflicts occurred.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.51 SCENARIOUNZIPPROBLEMS

Use this API to retrieve the problems that occurred during the last unzip attempt. For all non-python calls, the arrays INDEXES, PROBLEMS, and TYPES must be allocated a length of at least ERRCNT, the return value from SCENARIOUNZIP or SCENARIOUNZIPSOLUTIONS API. The array PATHS must be allowed a length of at least ERRCNT*260.

Python syntax:

```
ierr, indexes, problems, paths, types = scenariounzipproblems()
```

Fortran syntax:

```
CALL SCENARIOUNZIPPROBLEMS (INDEXES, PROBLEMS, PATHS, TYPES, IERR)
```

where:

Integer INDEXES(ERRCNT)	Is the internal problem index which will be used in SCENARIOUN-ZIPSOLUTIONS (output).
Integer PROBLEMS(ERRCNT)	Is the array of problems that were encountered (output). PROBLEMS(I)=0 file already exists. PROBLEMS(I)=1 path does not exist. PROBLEMS(I)=2 a file with the specified name does not exist in the archive. PROBLEMS(I)=3 a directory with the specified file name already exists.
Character PATHS(ERRCNT)*260	Is the array of full file paths of all files that had problems being extracted (output).
Integer TYPES(ERRCNT)	Is the array of file types associated with each file (output).
Integer IERR	Is the error code (output). IERR = 0 no conflicts occurred. IERR > 0 conflicts occurred.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.52 SCENARIOUNZIPSOLUTIONS

Use this API to submit potential solutions to the problems that occurred during the last unzip attempt.

Batch command syntax:

```
BAT_ SCENARIOUNZIPSOLUTIONS NSOLUTIONS INDEXES SOLUTIONS PATHS
```

Python syntax:

```
ierr = scenariounzipsolutions (nsolutions, indexes, solutions, paths)
```

Fortran syntax:

```
CALL SCENARIOUNZIPSOLUTIONS (NSOLUTIONS, INDEXES, SOLUTIONS, PATHS,
IERR)
```

where:

Integer NSOLUTIONS	Is the number of solutions being submitted
Integer INDEXES(NSOLUTIONS)	Is the internal problem indexes which were obtained from SCENARIOUNZIPPROBLEMS (input; no default allowed).
Integer SOLUTIONS(NSOLUTIONS)	Is the array of solutions to attempt (input; no default allowed). PROBLEMS(I)=0 skip the file. PROBLEMS(I)=1 retry to unzip, possibly with a different file name. PROBLEMS(I)=2 to overwrite the current item on disk. Note: These solutions do not apply to all problems. For example if the file is missing from the archive only a solution of 0 will solve the problem.
Character PATHS(NSOLUTIONS)*260	Is used to attempt to unzip a file to a new path or under a new name (input). All name changes will be reflected in the Scenario. If no name is specified for a particular problem the name will not change.
Integer IERR	Is the number of conflicts that occurred (output). IERR = 0 no conflicts occurred. IERR > 0 conflicts occurred.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.53 SCENARIOFILESETATTRS

Use this API to modify the attributes of the specified file.

Batch command syntax:

```
BAT_SCENARIOFILESETATTRS SGROUP SFILE STYPE LOAD SARGS IERR
```

Python syntax:

```
ierr = scenariofilesetattrs (sgroup, sfile, stype, load, sargs, ierr)
```

Fortran syntax:

```
CALL SCENARIOFILESETATTRS (SGROUP, SFILE, STYPE, LOAD, SARGS, IERR)
```

where:

Character SGROUP*72	Is the group that contains the specified file (input; no default allowed).
Character SFILE*260	Is the name of the file whose attributes are to be modified (input; no default allowed).
Character STYPE*8	Is the new file type identifier for the specified file (input; ignored by default).
Integer LOAD	Is the file's new position in the load order of its group. A load order of 0 specifies that the file is not loaded (input; ignored by default).
Character SARGS*260	Is the argument string to be used for the file when automatically opening the file (input; ignored by default).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred. IERR = 1 no Scenario in memory. IERR = 3 specified group not found. IERR = 4 specified file not found.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

13.54 SCENARIOFILERENAME

Use this API to rename one of the files in the Scenario.

Batch command syntax:

```
BAT_SCENARIOFILERENAME SGROUP SFILE SNEWNAME
```

Python syntax:

```
ierr = scenariofilerename (sgroup, sfile, snewname)
```

Fortran syntax:

```
CALL SCENARIOFILERENAME (SGROUP, SFILE, SNEWNAME, IERR)
```

where:

Character SGROUP*72	Is the group that contains the specified file (input; no default allowed).
Character SFILE*260	Is the name of the file whose name is to be changed (input; no default allowed).
Character SNEWNAME*260	Is the new name to be given to the specified file (input; no default allowed).
Integer IERR	Is the error code (output). IERR = 0 no error occurred. IERR > 0 fatal error occurred. IERR = 1 no Scenario in memory. IERR = 2 invalid file name. IERR = 3 specified group not found. IERR = 4 specified file not found.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

SIEMENS

Chapter 14

Other Python Modules

This chapter identifies other Python modules that are delivered as part of the PSS®E API.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

14.1 Modules Supplied with a GUI

These modules can be imported or run by themselves. They display a GUI that presents their functionality.

createusrdllw – Create a user dll for dynamics simulation. Combines the traditional compile and cload4 processes in a more user-friendly way.

excelexportw – Provides tools to export psse results to Excel, e.g., pv, qv, and acc files.

n11analysisw – A prototype implementation of an N-1-1 analysis.

plot2wordw - Provides tools to create Word documents from plots.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

14.2 Modules that Define Functions

Each of these modules contains help information about the functions that they define. Import the module and run the help function on them for more information.

bsntools – Bus renaming tools. Functions to modify bus names by area. Requires psspy.

caspy – Create an object to directly returns arrays of data from a saved case file. Encapsulates the functionality of the Saved Case Data Extraction routines (USRCAS).

dyntools – Tools for processing channel output files, including export to Excel.

pssgrpg - Provides functions intended to be used to supply values to report labels on I-line diagrams.

redirect – Tools for directing PSS®E output to Python, and vice versa.

sliderPy - Provides functions to directly manipulate diagram elements.

wordpy - Provides Python functions to interface with Microsoft Word.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

This page intentionally left blank.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

SIEMENS

Index

A

A phase
- *fault* 1-102, 1-104
- *fault current* 7-158
- *short circuit current* 7-156, 7-162
- *shunt current* 7-158
- *voltage* 7-158

ac base case
- *dc contingency checking* 1-146
- *interchange limits* 1-393
- *transmission interchange limits* 1-600, 1-620

ac bus converter 1-448

ac bus count
- *area* 8-255
- *owner* 8-266
- *zone* 8-277

ac contingency analysis 1-2, 1-94, 9-5
- *corrective actions* 1-30, 1-34, 1-43, 1-47, 1-75–1-76, 1-78, 1-82
- *export to Excel spreadsheet* 9-35
- *generation dispatch* 1-56, 1-58, 1-61, 1-188
- *post-contingency tripping* 1-64
- *tripping simulation* 1-30, 1-34

AC Contingency Multiple Run Report 1-4, 1-8

AC Contingency Report 1-21, 1-24, 1-27

ac control 2-158
- *converter* 8-220

ac current flow 7-34, 7-46, 7-196
- *2-terminal dc line converter from bus* 8-176
- *converter bus, multi-terminal dc line converter* 8-199
- *inverter, 2-terminal dc line* 8-162–8-163
- *rectifier, 2-terminal dc line* 8-162–8-163
- *VSC dc line converter from bus* 8-222

ac island 1-629

ac line, out-of-service 1-424

ac loading percent of limit 7-196

ac power flow 7-34, 7-46, 7-196
- *2-terminal dc line converter dc line from bus* 8-176, 8-178
- *converter bus dc line* 8-199
- *inverter dc line* 8-164, 8-166

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

multi-terminal dc line converter dc line from bus 8-201
rectifier dc line 8-164, 8-166
VSC dc line converter dc line from bus 8-222, 8-225
ac power losses 8-166, 8-178, 8-189, 8-212, 8-225
dc line total 8-201
ac setpoint 2-158, 7-196
VSC dc line converter from bus 8-222
ac voltage, primary base 8-199
ACCC 1-2, 1-4, 1-8, 1-21, 1-24, 1-27, 1-30, 1-34, 1-43, 1-47, 1-56, 1-58, 1-61, 1-64, 1-75, 1-78, 1-82, ... 1-94, 1-188, 1-237, 1-252, 1-258, 1-271, 1-277, 1-289, 9-35
acceleration factor
bus type 1 . . . . . 2-101, 2-103
capacitor commutated converter 8-163
imaginary component 2-101, 2-103
network dynamic solution 4-29
Newton solution 2-101, 2-103, 2-141
power flow solutions 2-101, 2-103
switching study 2-101, 2-103
active bridge power exchange 8-234
active models display
auxiliary signal 4-6
dc line 4-16
FACTS device 4-37
line relay 4-50
load 4-43
plant 4-44
switched shunt 4-76
wind machine 4-78
active power constant MVA load 1-211, 1-214, 1-453, 1-458, 1-468, 1-473, 1-478, 1-484
active power dispatch 3-13, 3-154
coefficient 3-113
active power dispatch table 3-76, 3-98–3-99, 3-113
data purged 3-131
active power flow 4-101, 4-109
terminal bus 8-234
active power generation 3-76
active power limits 1-217, 1-219
active power load 4-112
output 4-80
active power losses 3-18, 3-52, 8-164
cost coefficient 3-52
active power slack generation 3-19
active/bypassed status
branch relay model 7-153
active/reactive load 2-36, 2-39, 2-41, 2-77
active/reactive power command output, wind control 4-80, 4-92
active/reactive power flow 4-99, 4-107, 7-46, 8-93
2-terminal dc line 7-34
2-terminal dc line converter dc line from bus 8-176
2-winding transformer from bus 8-121
branch from bus 8-107
converter bus dc line 8-199
data returned 2-57

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

inverter dc line 8-164
rectifier dc line 8-164
sending bus 8-234
series element 8-234
from bus end, FACTS device 8-247
sending end 7-60
terminal end 7-61
shunt element, FACTS device 8-236, 8-250
terminal bus 8-234
VSC dc line converter dc line from bus 8-222
winding bus 8-150
active/reactive power generation 1-165, 1-642
owner total 8-269
zone total 8-280
active/reactive power limits 8-16, 8-26
machine 2-33, 2-49
active/reactive power load
area total 8-258
owner total 8-269
percent constant admittance 1-142
percent constant current 1-142
scaled 1-549, 1-552
zone total 8-280
active/reactive power losses 8-93, 8-107, 8-121, 8-134, 8-150, 8-176, 8-187, 8-210
area total 8-258
dc line total 8-199
owner total 8-269
total 8-222
zone total 8-280
active/reactive power net interchange
area total 8-258
zone total 8-280
active/reactive power output 8-16, 8-26
machine 2-49
adjustable branch reactance 3-13, 3-23, 3-64
data purged 3-126
status 3-64
adjustable bus load 3-17, 3-108
status 3-69
table 3-13, 3-69, 3-120, 3-150
data purged 3-128
adjustable bus shunt 3-13–3-14, 3-71, 3-73, 3-109, 3-151
data purged 3-129–3-130
adjustment control mode 2-129, 2-149, 2-153
transformer 1-615
admittance 4-31, 4-357, 4-364
magnetizing 8-124, 8-136
network 1-170, 1-381, 4-84
per step 2-111, 7-173
per step returned 7-174–7-175
reactors off, capacitors on 8-81
reactors on, capacitors off 8-81
admittance matrix, factorized 4-31
aerodynamic model, wind 4-20, 4-25, 4-78, 4-134, 4-142–4-143, 7-201, 7-203
aerodynamic torque output 4-80, 4-91

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

aIndMacBusChar 8-295
aIndMacBusCount 8-287
aIndMacBusCplx 8-293
aIndMacBusInt 8-289
aIndMacBusReal 8-291
aIndMacBusTypes 8-297
aIndMacChar 8-307
aIndMacCount 8-298
aIndMacCplx 8-305
aIndMacInt 8-300
aIndMacReal 8-302
aIndMacTypes 8-309
alert device setting 1-86, 9-47, 12-3
ALOC 1-336, 1-341–1-342, 1-632–1-633, 1-637
ALPH 1-87
ALPHA angle 7-46
alphabetical order return
2-terminal dc line name 7-78, 7-129
branch name 7-130–7-131
bus name 7-95–7-96, 7-132, 7-145
area specified 7-170
base voltage specified 7-171
zone specified 7-172
FACTS device name 7-88, 7-133
multi-terminal dc line name 7-93, 7-138
VSC dc line name 7-99, 7-142
angle deviation threshold 1-160, 4-54–4-55
angle table, transformer impedance correction 2-30
angle, case comparison 1-159
ANSI 1-88, 1-91
ANSI Fault Specification Data File (*.ans) 1-89, 1-92, 1-536
API routines
data retrieval
single element 7-1
solutions 9-1
subsystem 8-1
dynamic simulation 4-1
graphical analysis 10-1
non-engineering functions 12-1
open access option 1-1
optimal power flow 3-1
data addition/modification 3-63
solutions 3-2
plot manipulation 11-1
power flow and fault analysis 1-1
data addition/modification 2-1
solutions 1-2
slider diagram manipulation 6-1
subsystem control 5-1
superseded 1-18, 1-96, 1-194, 1-369, 1-383, 1-385, 1-399, 1-401, 1-422, 1-465, 1-503, 1-507, 1-555, 1-609, 1-643, 2-10, 2-12, 2-20, 2-37, 2-39, 2-46, 2-68–2-69, 2-73, 2-79, 2-86, 2-90, 2-96, 2-98, 2-100, 2-107, 2-114, 2-123, 2-126, 2-145, 2-149, 3-46, 4-27, 4-122, 4-150, 4-164, 4-180, 4-186, 4-192, 4-209, 4-245, 4-252, 4-258, 4-264, 6-63, 7-10, 7-17–7-18, 7-32, 7-36, 7-42, 7-44, 7-48, 7-57, 7-63, ... 7-105, 7-192, 12-7

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

apparent admittances 1-563, 1-583
apparent impedances 1-563, 1-583, 6-76
dynamic output channel 4-89
output 4-80
AREA 1-97–1-98, 1-427, 1-440, 1-446
area
bus assignment 1-99–1-100, 1-647
count 8-255, 8-259, 8-261–8-262
data
array requirement 8-254
changed 2-2
returned 7-4–7-5, 8-255, 8-257, 8-260, 8-262
equipment assignment returned 7-8
in-service 7-8
name returned 7-6
number returned 7-7
status returned 7-8
subsystem
definition 5-2–5-3
initialization 5-4
subsystem selection 7-165, 7-167
totals 1-97–1-98, 1-135, 1-427, 1-440, 1-446
data returned 7-2
power 4-79
area boundary bus
equivalencing suppressed 1-166, 1-513
retained, electrical equivalent 1-165, 1-642
area bus load 8-261
area generation 7-2, 8-261
area interchange 1-130, 1-132, 1-229, 2-2, 7-2, 7-9, 8-258, 8-261
adjustment
ac contingency analysis 1-2, 1-43, 1-47, 1-56, 1-78, 1-82
ac contingency analysis, corrective actions 1-30, 1-34
ac contingency analysis, post-contingency tripping 1-64
ac contingency analysis, tripping simulation 1-30, 1-34
corrective actions 1-75, 1-271, 1-277
decoupled Newton-Raphson power flow analysis 1-375
fixed slope decoupled Newton-Raphson power flow analysis 1-171
full Newton-Raphson power flow analysis 1-179
Gauss-Seidel power flow analysis 1-334, 1-596
multi-level ac contingency analysis 1-237, 1-252, 1-258
multi-level ac contingency analysis, tripping simulation 1-289
PV analysis 1-450, 1-455, 1-465, 1-468, 1-470, 1-473, 1-475, 1-478, 1-481, 1-485
QV analysis 1-488, 1-490, 1-492, 1-495
contingency case 1-468, 1-473, 1-478, 1-485
control 1-144, 8-255
data returned 7-9
PSS®E activity 1-97, 1-427, 1-440, 1-446
regulation 3-40
area load 7-2, 8-261
assignment 1-99–1-100, 1-647
count 8-255
area model 4-146, 4-155–4-156
area power losses 7-2, 8-261
area slack bus 2-2
number returned 7-5
area swing bus 8-255, 8-262

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

area tie monitoring 4-79
area transfer 1-309
ARNM 1-99–1-100, 1-647
array
allocation table
auxiliary signal model 4-239
dc line model 4-196
FACTS device model 4-213
line relay model 4-168
load model 4-154
plant model 4-129, 4-282
switched shunt model 4-226
wind model 4-141
character ICON (CHRICN) 4-10
constant, real (CON) changed 4-8
data retrieval 9-2
dynamics data storage 4-18, 4-72
governor voltage setpoint (GREF) 4-11, 4-40
index returned
2-terminal dc line auxiliary signal model 7-30
2-terminal dc line model 7-38
branch relay model 7-153
FACTS device auxiliary signal model 7-55
FACTS device model 7-65
load model 7-102, 7-165
N-terminal dc line auxiliary signal model 7-40
N-terminal dc line model 7-50
plant model 7-122
switched shunt model 7-179
VSC dc line auxiliary signal model 7-194
VSC dc line model 7-199
wind model 7-201
integer ICON (ICON) 4-9
state variable (STATE) 4-12
type, dynamic simulation 4-18
variable, algebraic (VAR) changed 4-14
voltage regulator voltage setpoint (VREF) 4-13, 4-15, 4-41–4-42
arrow, reactive power animation 6-29
ASCC 1-101, 1-103, 1-105, 1-108, 9-13
ASTR 4-4, 7-74
asymmetrical breaking current 9-19–9-20
ATC Updates File (*.dat) 1-209
automatic adjustment threshold, power flow solution 2-101, 2-104
automation 1-ii
automation file languages 1-i
auxiliary control model, wind 4-20, 4-78, 4-134, 4-142–4-143
auxiliary signal index
2-terminal dc line 7-31
FACTS device 7-56
N-terminal dc line 7-41
VSC dc line 7-195
auxiliary signal model 4-6, 4-20, 4-25, 4-239, 4-268–4-275
array allocation table 4-239
connection table 4-232, 4-234, 4-236, 4-238–4-239
injection point/index
2-terminal dc line 4-235, 4-256–4-257, 4-260–4-261, 4-272–4-273

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

FACTS device 4-231, 4-242, 4-244, 4-247, 4-249, 4-268–4-269
multi-terminal dc line 4-233, 4-250–4-251, 4-254–4-255, 4-270
VSC dc line 4-237, 4-262–4-263, 4-266–4-267, 4-274–4-275
removed from 2-terminal dc line 4-272
removed from FACTS device 4-268
removed from multi-terminal dc line 4-270
removed from VSC dc lline 4-274
status changed
injection point, 2-terminal dc line 4-273
injection point, FACTS device 4-269
injection point, multi-terminal dc line 4-271
injection point, VSC dc line 4-275
unconnected 4-240
user-written 4-241
wind 7-201, 7-203
auxiliary signal model, 2-terminal dc line
character ICON (CHRICN) changed 4-256
constant, real (CON) changed 4-257
integer ICON changed 4-260
variable, algebraic (VAR) changed 4-261
auxiliary signal model, FACTS device
character ICON (CHRICN) changed 4-242
constant, real (CON) changed 4-244
integer ICON changed 4-247
variable, algebraic (VAR) changed 4-249
auxiliary signal model, multi-terminal dc line
character ICON (CHRICN) changed 4-250
constant, real (CON) changed 4-251
integer ICON changed 4-254
variable, algebraic (VAR) changed 4-255
auxiliary signal model, VSC dc line
character ICON (CHRICN) changed 4-262
constant, real (CON) changed 4-263
integer ICON changed 4-266
variable, algebraic (VAR) changed 4-267
auxiliary transformer fault 1-197, 1-200, 1-203
available capacity table 1-21–1-24, 1-27

B

B factor, branch 2-82
back-swing, voltage 4-67, 4-77
bad iteration limit 3-5–3-6
barrier coefficient 3-9, 3-11
barrier function 3-26
step length 3-26
step length tolerance 3-44
base ac voltage 2-57, 2-136, 7-35, 7-46
base case
line flow 1-146, 1-600, 1-620
constraints 1-393, 1-620
transfer increment 1-466, 1-471, 1-476, 1-482
base frequency option setting 1-113
base voltage 4-357, 4-361, 4-364, 7-171
limits 5-5, 5-7
subsystem 1-517, 1-520

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

BAT_format recording 12-20
batch command syntax 1-i
BGEN 1-114
bitmap image 6-8
BKDY 1-115
block count 8-78
blow-up threshold 2-101, 2-104
BOSL model 2-26, 2-28
boundary bus 1-165, 1-169, 1-517, 1-519
mismatch conversion 1-114
boundary condition options 1-114, 1-175, 1-177
boundary value, plot 10-2
bounded reactive power generation 3-58
bounded switched shunts 3-58
branch 1-130, 1-132, 1-229, 3-104
annotation, Slider Diagram 6-41–6-42, 6-55, 6-57
apparent admittance 1-563, 1-583
B factor 2-82
charging 7-11
count 8-104, 8-107, 8-110, 8-112
owner 8-266
data
array requirement 8-101
deleted 1-169
returned 7-11–7-12, 7-14, 8-103, 8-105, 8-109, 8-111
from bus
active power flow dynamic output channel 4-101
active/reactive power flow dynamic output channel 4-99
apparent impedances dynamic output channel 4-95
MVA flow dynamic output channel 4-97
IBUS 7-84–7-85
impedance 8-110
apparent 1-563, 1-583, 6-76
threshold 1-165, 1-554, 1-642
zero sequence 8-110, 8-124, 8-152
in-service 7-28
interior subsystem 8-101
monitoring 4-79
name returned 7-130–7-131
next 7-130–7-131
number returned 7-130–7-131
outage severity 1-499
out-of-service 1-509, 1-542, 1-544, 3-57
display setting 1-404
owner 7-14
reassignment 1-385, 1-387
total losses 7-146
parameter check 1-116, 1-118
rating set 7-11
reactance 2-6, 3-66, 7-12
data purged 3-127
resistance 2-6, 7-12
returned 1-323
sequence data 1-608
short circuit current returned 7-157
subsystem tie 8-101

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

trip in-service 7-100
branch check 1-116, 1-118, 1-504
branch contingencies
overload ranking 1-239, 1-254, 1-273, 1-291, 1-499, 1-501
voltage depression ranking 1-239, 1-254, 1-273
branch current 1-563, 1-583, 7-16, 8-93, 8-106
loadings 1-224
branch fault 1-561, 1-563, 1-583
branch flow 3-110, 3-152, 7-16
constraint 3-13, 3-78, 3-80
data
array requirement 8-89
flow constraint data purged 3-132–3-133
purged 3-134, 3-147
returned 7-13, 7-15, 8-90, 8-92, 8-95, 8-97
IBUS 7-13
limits 3-78, 3-80, 3-83
loading, Slider Diagram 6-59
overload report 1-524, 1-527
overload tolerance 1-32, 1-36, 1-45, 1-49, 1-76, 1-80, 1-84, 1-149, 1-256, 1-262, 1-275, 1-281
overload violations 1-43, 6-60–6-61
PV analysis 10-3, 10-7, 10-9
record 3-82
type 3-78, 3-80
branch loading 1-504
check 1-453, 1-458, 1-468, 1-473, 1-478, 1-484
branch overload problem indices report 1-524, 1-527
branch rating set 1-323
Slider Diagram 6-51, 6-53
branch relay model 7-153
name returned 7-155
branch relay scanning 4-64
branch report, detailed 1-339–1-340
branch status 2-5, 2-53, 2-120, 2-123, 2-149, 2-153, 7-14, 8-91, 8-104
case comparison 1-157, 1-159
changed 1-152
branch unbalance 1-558, 1-598
calculation 4-370, 4-372
BRCH 1-116, 1-118
breaker contact parting time 1-197, 9-18
Breaker Duty Data File (*.bkd) 1-115, 1-536, 4-53
breaker failure rate 1-606
breaker open 1-598, 4-370, 4-372
bridge
power exchange 7-61
power exchange, active 8-247
power flow 8-236, 8-250
power load 7-61
power transfer 2-24, 7-60, 8-234
limits 7-64, 8-232, 8-245
series 2-56, 2-135, 8-197
rectifier/inverter 8-160
BSNM 1-120
buffer 12-5–12-6, 12-22

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

bus 1-130, 1-132–1-133, 1-229
area assignment 7-2
attribute 3-87
data purged 3-136
report 3-13
base voltage 1-174, 2-12, 2-14, 7-20, 8-6, 8-15, 8-38, 8-59, 8-80, 8-292
bus number option 1-125
compensation limits 1-116, 1-118
count 8-4, 8-6, 8-8, 8-10, 9-3
returned 1-629, 7-185
current returned 7-158
data
array requirement 8-3
changed 2-8, 2-12, 2-14
deleted 1-169
returned 7-20–7-21, 7-23, 7-26, 8-4, 8-6, 8-8, 8-10–8-11, 8-21, 8-32, 8-45, 8-55, 8-65, 8-75, 8-87, 8-99, 8-113, 8-128, 8-140, 8-157, 8-170, 8-182, 8-193, 8-205, 8-216, 8-228, 8-240, 8-253, 8-264, ... 8-275, 8-286, 8-297, 8-309
returned, alphabetical order 7-95–7-96
returned, numerical order 7-86–7-87
disconnected 1-161, 4-365, 7-100
display, Slider Diagram 6-9
extended bus name option 1-125
identifier annotation, Slider Diagram 6-40, 6-42, 6-44
in-service check, ac islands 1-629
list, alphabetical order 1-87
location, Slider Diagram 6-77
monitoring 4-79
next 7-132
specified area 7-170
specified base voltage 7-171
specified zone 7-172
order, optimal 1-379
owner reassignment 1-385, 1-387
power total 4-79
reconnected 1-521
removed from transaction event 1-391
resistance/reactance returned 2-121, 2-124
sequence data changed 2-73
size level 1-127
specified bus exists 7-25
split 1-604
zone assignment 1-646
total zone generation 7-213
total zone load 7-213
bus based load model, variable, algebraic (VAR) changed 4-153
bus fault 1-559–1-560, 4-364, 4-368
bus frequency deviation 4-29
dynamic output channel 4-111
output 4-80
bus load 3-120, 7-22, 7-24
case comparison 1-159
changed 1-550, 1-553
curtailments, Slider Diagram 6-59, 6-61
data
megawatt data returned 1-548, 1-551
Mvar data returned 1-548, 1-551
purged 3-160–3-161

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

returned 7-108, 7-111
returned in MVA 7-106
sequence data 1-608
disconnect/reconnect 1-327–1-328
identifier returned 7-136
megawatt 7-24
megawatt scaling 1-549, 1-553
multiplier 3-69
MVA 7-22
Mvar 7-24
next 7-136
participating bus 2-65
report 3-13
loss of load 1-524, 1-527
scaling 1-549, 1-552
table 3-121
total in-service 7-21, 7-23
zone assignment 1-646
bus load curtailment report 1-606
Bus Location Data File (*.loc) 6-7, 6-18, 6-77
bus mismatch 8-6, 8-8, 8-16–8-17, 8-38, 8-41, 8-61, 8-81, 8-83, 8-292, 8-294, 9-27, 9-29, 9-38, 9-40
converged tolerance 1-6, 1-10, 1-23, 1-26, 1-29, 1-403
data returned 7-27, 7-119
tolerance 9-5, 9-11–9-12, 9-35
bus model 4-146, 4-155–4-156
bus number
changed 2-16
number/name display option 1-126
offset 1-121
range 1-121, 1-128
returned 1-222, 7-127, 7-132, 7-145
setting 1-126
unused 1-128
Bus Number Translation File (*.trn) 1-122, 1-536, 6-23
bus phase angle, dynamic output channel 4-116
bus shunt
case comparison 1-159
record 3-73
scaled 1-549, 1-552
susceptance 3-71, 3-74
bus subsystem
definition 5-5, 5-8, 7-19
by single bus 5-11
initialization 5-9
modified 5-7
selection 7-165, 7-167
Bus Subsystem File (*.sbs) 5-10, 5-12
bus type 2-12, 2-14, 3-74, 7-26, 8-4, 8-13, 8-36, 8-57, 8-78, 8-290
bus type 1 . . . . . 1-141, 1-513, 1-549, 1-552, 3-88, 3-121
acceleration factor 2-101, 2-103
bus type 3 (swing bus) 7-28
bus type 4 . . . . . 1-509, 1-542, 1-544, 7-28, 7-100
bus types 2 and 3 . . . . . 1-141, 1-187, 1-234, 1-549, 1-552, 3-88, 3-121
case comparison 1-159
bus voltage

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

angle 3-118, 3-158, 7-20, 7-186
annotation, Slider Diagram 6-41, 6-45, 6-48, 6-55, 6-57
check 1-453, 1-458, 1-468, 1-473, 1-478, 1-484, 6-45
Slider Diagram 6-50, 6-52
controlled 1-523
data returned 7-158
limits 3-25, 3-85, 3-87
emergency 3-61
magnitude 1-176, 1-178, 2-12, 2-14, 3-85, 3-119, 3-159, 7-20, 7-60, 7-186, 8-6, 8-15, 8-38, 8-59, 8-80, ... 8-292
actual 8-59
data purged 3-135
dynamic output channel 4-115–4-116
output 4-80
phase angle 2-12, 2-14, 8-6, 8-15, 8-38, 8-59, 8-80, 8-292
star point bus 8-133
problem indices report 1-524, 1-527
PV analysis 9-27, 10-3, 10-7, 10-9
QV analysis 9-29
rectangular coordinates 8-8, 8-17, 8-41, 8-61, 8-83, 8-294
setpoint 7-60
Slider Diagram 6-59
Slider Diagram contour 6-45
violation 6-60, 6-62
report 1-524, 1-527
tolerance 1-32, 1-36, 1-45, 1-49, 1-76, 1-80, 1-84, 1-256, 1-262, 1-275, 1-281
violation, Slider Diagram 6-59, 6-61
bus-breaker configuration 1-606
buses not in swing bus tree 1-629
BUSN 1-128
bypassed models display
auxiliary signal model 4-6
dc line model 4-16
FACTS device model 4-37
line relay model 4-50
load model 4-43
plant model 4-44
switched shunt model 4-76
wind model 4-78

C

call
FLOW1 monitoring model 4-97, 4-99, 4-101
FLOW3 monitoring model 4-105, 4-107, 4-109
GENTMC monitoring model 4-93
GENTMZ monitoring model 4-89
RELAY2 monitoring model 4-95
RELAY3 monitoring model 4-103
VOLMAG monitoring model 4-115–4-116
capability limit status, reactive 3-93, 3-95
capacitance, charging 8-107
capacitive reactance 2-136
capacitor
Mvar scaling 1-549, 1-553
Mvar total returned 1-548, 1-551
capacitor commutated converter

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

acceleration factor 8-163
iteration limit 8-161
capacity, dispatched units 1-163
CASE 1-129
case comparison 1-135–1-136, 1-156, 1-158, 1-182
display, Slider Diagram 6-72
power flow 1-158
tie lines 1-156
totals 1-135
voltage, Slider Diagram 6-43
case size, bus count 12-14
case summary 1-130, 1-132–1-133, 1-229
OPF report 3-13
case title 1-370, 9-22, 9-24
changed 2-17, 2-43
long 1-408
returned 7-184
CCC converter 2-136
center tapped 2-phase fault analysis 1-591
channel index 4-59
channel monitoring model 4-87
Channel Output File (*.out) 4-7, 4-35–4-36, 4-38–4-39, 4-46–4-47, 4-75
open in plotting book 11-9–11-10
channel output filename changed, dynamic simulation 4-7
channel, dynamic simulation 4-85
channel monitoring model 4-20, 4-87–4-88
table 4-88
index 4-79, 4-112
monitoring 4-112
character data, array requirement 8-2
character ICON (CHRICN) changed 4-10
2-terminal dc line model 4-184
auxiliary signal model, 2-terminal dc line 4-256
auxiliary signal model, FACTS device 4-242
auxiliary signal model, multi-terminal dc line 4-250
auxiliary signal model, VSC dc line 4-262
FACTS device model 4-207
line relay model 4-162
load model 4-148
multi-terminal dc line model 4-178, 4-303–4-311, 4-313, 4-315, 4-317, 4-319–4-338
plant model 4-120, 4-278
switched shunt model 4-221
VSC dc line model 4-190
wind model 4-136
charging
branch 7-11
branch check 1-116, 1-118
capacitance, zero sequence 8-107
treatment 1-175, 1-177
CHRICN array 4-10
CHSB 4-79
circuit identifier returned 7-140–7-141
clamp tolerance 3-28, 3-34

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

clamped constraint equation . . . 3-43
decay factor . . . 3-27
non-optimized generators . . . 3-7
classical fault analysis conditions . . . 1-175, 1-177
CLOS . . . 1-134
close all PSS®E files and exit . . . 1-610
closed end bus . . . 1-561
CMPR . . . 1-135
CNTB . . . 1-137
color, Slider Diagram . . . 6-28, 6-38, 6-40, 6-44, 6-47, 6-49
command line, close report . . . 1-134
commitment profile . . . 1-162
commutating capacitor reactance magnitude . . . 7-35
commutating resistance/reactance . . . 2-57, 2-136, 7-35, 7-46, 8-163, 8-199
compensating impedance . . . 7-206
load drop . . . 8-124, 8-152
compensator model . . . 4-20, 4-25, 4-44, 4-118, 4-130–4-131, 7-122, 7-126
complex current . . . 7-156
data returned . . . 7-157, 7-162
complex fault admittance/impedance . . . 4-357, 4-361, 4-364
complex generation
area total . . . 8-261
owner total . . . 8-272
zone total . . . 8-283
complex load
area buses/loads . . . 8-261
owner buses/loads . . . 8-272
zone buses/loads . . . 8-283
complex net interchange
area . . . 8-261
zone . . . 8-283
complex power flow
3-winding transformer . . . 7-206
data returned . . . 7-2
real/reactive component returned . . . 7-146, 7-183, 7-187, 7-213
series element, from bus end, FACTS device . . . 8-250
series element, sending end bus . . . 8-236
series element, terminal end bus . . . 8-236
terminal bus . . . 8-236
complex power losses
area . . . 8-261
owner . . . 8-272
zone . . . 8-283
compounded dc voltage . . . 7-34, 8-162
compounding resistance . . . 2-141, 7-34, 8-162
CON index . . . 4-18
2-terminal dc line auxiliary signal model . . . 7-30
2-terminal dc line model . . . 7-38
branch relay model . . . 7-153
FACTS device auxiliary signal model . . . 7-55
FACTS device model . . . 7-65
load model . . . 7-102, 7-165
N-terminal dc line auxiliary signal model . . . 7-40

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

N-terminal dc line model 7-50
plant model 7-122
switched shunt model 7-179
VSC dc line auxiliary signal model 7-194
VSC dc line model 7-199
wind model 7-201
conductance 2-74, 2-95, 2-121, 2-124, 2-150, 2-154, 7-208
CONEC model 4-20, 4-24
CONEC subroutine 4-74
compile 4-33
file 4-32–4-33, 4-74
CONET model 4-20, 4-24
CONET subroutine 4-74
compile 4-33
file 4-32–4-33, 4-74
CONG 1-140
CONL 1-141
connection code 2-68, 2-88, 2-90, 2-92, 7-210, 8-117, 8-132, 8-145
case comparison 1-157, 1-159
connectivity check 1-143
consistency check
plant model 4-128
wind model 4-140
constant admittance 1-159
active/reactive load 2-39, 2-41
load 1-91, 1-103, 1-141, 1-557–1-558, 1-585–1-586, 4-373, 7-21, 7-23, 7-106, 7-108
total 8-38, 8-41, 8-49, 8-51, 8-303
constant current 1-159
active/reactive load 2-39, 2-41
load 1-141, 7-21, 7-23, 7-106, 7-108, 8-38, 8-41, 8-49, 8-51, 8-303
constant MVA 1-159
load 7-21, 7-23, 7-106, 7-108, 8-38, 8-41, 8-49, 8-51, 8-303
conversion 1-141
constant power
active/reactive load 2-39, 2-41
load characteristic breakpoint 2-101, 2-104
constant, real (CON) changed 4-8
2-terminal dc line model 4-185
auxiliary signal model, 2-terminal dc line 4-257
auxiliary signal model, FACTS device 4-244
auxiliary signal model, multi-terminal dc line 4-251
auxiliary signal model, VSC dc line 4-263
FACTS device model 4-208
line relay model 4-163
load model 4-149
multi-terminal dc line model 4-179
plant model 4-121, 4-279
switched shunt model 4-222
VSC dc line model 4-191
wind model 4-137
contact parting times 1-89, 1-92
contingency
analysis results, Slider Diagram 6-59
appended to file 1-94

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

count 9-3
description 9-23, 9-25
description summary table 1-601, 1-621
evaluation level 1-238, 1-253, 1-259, 1-272, 1-278, 1-290
event 1-601, 1-621
description 9-35
label 1-184, 1-601, 1-621
legend 1-5, 1-9
level 1-155
load shed count 9-3
monitored bus voltage 9-27, 9-29
monitored flow 9-27, 9-29
subsystem 1-240, 1-256, 1-262, 1-275, 1-281, 1-292
subsystem ties 1-239, 1-254, 1-260, 1-273, 1-279, 1-291, 1-499, 1-501
summary report 1-524, 1-527
violations 1-5, 1-9, 6-60
contingency case
power flow 1-6, 1-10, 1-147, 1-403, 1-451, 1-456, 1-466, 1-471, 1-476, 1-482
report 1-5, 1-10
voltage 1-403
Contingency Description Data File (*.con) 1-153, 1-155, 1-502, 9-4, 9-23–9-24, 9-31
contingency loading violation report 1-6, 1-10
contingency rating 1-239, 1-254, 1-260, 1-273, 1-279, 1-291
contingency report 1-606
Contingency Solution Output File (*.acc) 1-3–1-4, 1-6, 1-8, 1-10, 1-23, 1-26, 1-29, 1-32, 1-37, 1-45, 1-49, 1-57, 1-65, 1-94, 1-240, 1-256, 1-262, 1-275, 1-281, 1-292, 1-400, 1-403, 1-499–1-500, 1-525, 1-528, ... 9-3, 9-5, 9-7, 9-11–9-12, 9-24, 9-31, 9-35
appended 1-94
continuous mode switched shunt 1-612
contour
dynamic simulation quantities 6-45
parameter change, Slider Diagram 6-2
parameter definition, Slider Diagram 6-6
refresh, Slider Diagram 6-21
remove, Slider Diagram 6-5
contributed reactive power 7-61
percent 2-24, 2-67, 2-158
converter 8-222
FACTS device 8-233
plant 8-15
switched shunt 8-80
control mode 2-23, 2-60, 2-110, 2-140, 2-160, 7-64
dc line 7-37
FACTS device 8-232, 8-245
multi-terminal dc line 8-185
switched shunt 8-78
transformer 8-117, 8-145
two-terminal dc line 8-160
two-terminal dc line converter 8-174
VSC dc line 8-208
controlled bus 1-137–1-138, 1-165, 1-642, 2-129, 2-149, 2-153, 8-78, 8-117, 8-127, 8-145
data returned 1-137–1-138
number 7-207, 7-210
scheduled voltages, check/change 1-137
suppress equivalencing 1-166, 1-513

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

controlling converter 7-196
controlling element 1-138
controlling transformer 1-619
convergence 9-27, 9-29
status 9-5
convergence monitor display 4-4, 4-46–4-47, 4-52, 4-75
convergence tolerance 7-168
network dynamic solution 4-29
reactive power mismatch 2-101, 2-104
voltage error 2-101, 2-104
conversion
PSAS Command File 4-49
PSEB Command File 1-421
converter
bus number 7-198
returned 2-60
count 8-185
returned 2-60
from bus 8-174, 8-220
power flow 2-136
data returned 2-57
type 2-135, 8-197
converter ac control 8-220
code 7-197
converter ac current rating 2-158
converter ac MVA rating 2-158
converter angle 7-34, 7-46, 8-199
converter type 8-163
from bus, 2-terminal dc line converter 8-176
limit 8-197
from bus, 2-terminal dc line converter 8-174
steady-state 8-199
rectifier/inverter 8-163
converter bus 2-54, 2-56, 2-135, 2-158, 8-197, 8-204, 8-208, 8-214
dc voltage 8-199
converter current returned 2-57
converter dc control 8-220
code 7-197
converter losses 2-158
VSC dc line converter from bus 8-222
converter participation factor 2-57, 8-199
converter tap ratio limit 8-197
from bus, 2-terminal dc line converter 8-174
converter, capacitor commutated 8-161, 8-163
corrective actions 1-31, 1-35, 1-44, 1-48, 1-149, 1-255, 1-261, 1-274, 1-280
generation dispatch count 9-3
load shed count 9-3
phase shifter change count 9-3
report 1-21
cost coefficient 3-54–3-55
cost consumption 3-92
cost curve type 3-76
cost integration constant 3-91–3-92

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

cost scale coefficient 3-64, 3-66, 3-71, 3-74
cost table 3-76
CRT output channel 4-34, 4-46, 4-52, 4-83
index 4-18
current calculation 1-232
simultaneous unbalance 1-557
current demand 7-34, 8-162
current expressed as MVA 1-374, 1-627
current fault 1-558
current flow
sending bus 8-233–8-234
series element
from bus end, FACTS device 8-247
sending bus 7-61
terminal bus 7-61
shunt element 7-61, 8-234, 8-247
terminal bus 8-234
current loading
3-winding transformer winding 7-204
branch 7-16, 8-93, 8-106
FACTS device 7-61
from bus end, FACTS device 8-247
sending bus 8-234
shunt 7-61, 8-234, 8-247
Slider Diagram 6-30, 6-56–6-57
terminal bus 8-234
current margin 7-34, 7-47
current rating
from bus, VSC dc line converter 8-222
current setpoint 8-199

D

data checking mode, dynamics simulation 4-20
dc base case
dc contingency checking 1-146
interchange limits 1-393
transmission interchange limits 1-600, 1-620
dc bus count 8-185
area 8-255
area assignment 8-255
owner 8-266
returned 2-60
zone 8-277
dc contingency checking 1-146
corrective actions 1-148
dc control code 2-158
converter 8-220
dc converter index 2-158
dc corrective actions 1-148
dc current 7-46, 7-196
2-terminal dc line 8-162, 8-176
converter 8-199
VSC dc line 8-210
dc current margin 2-141, 8-162

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

rectifier 8-199
dc line 1-130, 1-132, 1-229
blocked 1-91, 1-102–1-104, 1-106, 1-109, 1-197, 1-200, 1-203–1-204, 1-557–1-558, 1-585–1-586, 4-368, ... 4-371, 4-373, 9-18
constant admittance load conversion . . 1-91, 1-103, 1-557–1-558, 1-585–1-586, 4-368, 4-371, 4-373
control mode 7-37, 7-49
current 7-34, 7-75–7-77, 7-151
data returned 2-141
ignored 1-538
power added to load 1-538
resistance 2-141, 2-160, 7-34, 7-196, 8-162, 8-210
tap setting 2-57
tap step 2-57
dc line model 4-16, 4-196
array allocation table 4-196
connection table 4-196
table 4-174, 4-176–4-177
unconnected 4-197
user-written 4-198
dc link 1-449
count 8-185
returned 2-60
multi-terminal dc line 2-62
resistance/inductance 2-62
dc network solution/report 1-151
dc power margin 2-57, 2-141, 7-34, 7-47, 8-162
rectifier 8-199
dc setpoint 2-158, 7-196
from bus, VSC dc line converter 8-222
dc tap adjustment 1-145
ac contingency analysis 1-2
corrective actions 1-31, 1-35, 1-44, 1-48, 1-75, 1-78, 1-82
generation dispatch 1-56
post-contingency tripping 1-65
tripping simulation 1-31, 1-35
contingency case 1-468, 1-473, 1-479, 1-485
decoupled Newton-Raphson power flow analysis 1-375
fixed slope decoupled Newton-Raphson power flow analysis 1-171
full Newton-Raphson power flow analysis 1-179
Gauss-Seidel power flow analysis 1-334, 1-596
inertial/governor resposne power flow analysis 1-216, 1-218
multi-level ac contingency analysis 1-238
corrective actions 1-253, 1-259, 1-272, 1-278
tripping simulation 1-253, 1-259, 1-289
PV analysis 1-450, 1-455, 1-465, 1-470, 1-475, 1-481
QV analysis 1-488, 1-490, 1-492, 1-495
dc voltage 2-141, 7-34, 7-46, 7-196
compounded 2-141, 8-162
converter bus 8-199
data returned 2-54, 2-136
from bus, 2-terminal dc line converter 8-176
from bus, VSC dc line converter 8-222
power flow 1-151
rectifier/inverter 8-163
DCCC 1-146, 1-148
DCLF 1-151

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

deceleration factor, tap movement 2-101, 2-104
decoupled Newton-Raphson power flow solution (NSOL) 1-375
default output
character 12-8
integer 12-9
real value 12-10
default values 1-ii
Detailed Branch Output File (*.dat) 1-339–1-340
deterministic reliability results, Slider Diagram 6-59, 6-61
DFAX 1-153, 1-155, 1-452, 1-457, 1-467, 1-472, 1-477, 1-483, 6-60, 6-62, 9-22
DFTI 1-156
DIFF 1-158
difference threshold, case comparison 1-135, 1-160
direct-axis synchronous reactance 3-93, 3-95–3-96
directory pathname 1-390
disconnect/reconnect to bus 1-325–1-326
discrete mode, switched shunt 1-612, 3-41
dispatch 1-162
fraction 3-98–3-99
load shed count 9-3
units 1-163
dispatch mode 1-184
ac contingency analysis
corrective actions 1-31, 1-35, 1-44, 1-48
generation dispatch 1-57
post-contingency tripping 1-65
tripping simulation 1-31, 1-35
multi-level ac contingency analysis 1-238
corrective actions 1-253, 1-259, 1-272, 1-278
tripping simulation 1-253, 1-259, 1-290
distribution factor 1-153, 1-232, 1-601–1-602, 1-621, 9-23
line loading relief 1-233
magnitude 1-232, 1-395
summary table 1-601, 1-622
threshold 1-340
tolerance 1-233
distribution factor analysis 1-380
Distribution Factor Data File (*.dfx) 1-153
ac contingency analysis 1-3
corrective actions 1-32, 1-45, 1-49, 1-77, 1-80, 1-84
generation dispatch 1-57
post-contingency tripping 1-65
tripping simulation 1-32, 1-36
Contingency Solution Output File 1-94
dc contingency checking 1-147
corrective actions 1-149
distribution factor analysis 1-380
filename 9-3, 9-22
generation dispatch algorithm 1-184
interchange limits calculation 1-395
line loading relief calculation 1-233
multi-level ac contingency analysis 1-240
corrective actions 1-256, 1-262, 1-275, 1-281
tripping simulation 1-256, 1-262, 1-292
multiple event contingencies 1-155

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

probabilistic reliability assessment 1-525, 1-528
PV analysis 1-453, 1-458, 1-469, 1-474, 1-480, 1-486, 9-24
QV analysis 1-489, 1-491, 1-494, 1-497, 9-31
substation reliability analysis 1-607
transaction event impact, MW flow 1-209
transfer dispatch 1-211, 1-215
transmission interchange limits calculation 1-602, 1-622
voltage limit record array 6-60, 6-62
DLST 4-18
DMPC 4-85
DOCU 4-19
double-precision fault analysis 1-101, 1-105
Draw Data File (*.drw) 1-536, 6-14
DSCN 1-161
dummy bus 2-51, 7-84
multi-section line 7-26, 7-87, 7-96–7-97, 7-185, 8-4, 8-13, 8-36, 8-57, 8-78, 8-290
data returned 7-86, 7-95, 7-145
dummy filename 6-13
dummy series element reactance 2-24, 7-60, 8-233
DYDA 4-24
dynamic simulation fault cleared 4-366
dynamic simulation run from Event Study file 13-6
dynamic simulation solution parameters changed 4-7, 4-29
dynamics activity selector 7-163
Dynamics Compiling Command File 4-33, 4-74
dynamics data
display, Slider Diagram 6-71
returned 7-52–7-54
storage array 4-18
utilization 4-72
dynamics mode 4-31
Dynamics Model Raw Data File (*.dyr) 1-536, 4-25, 4-32–4-33
dynamics output channel 7-29
3-winding transformer from bus active power flow 4-109
3-winding transformer from bus apparent impedances 4-103
identifier changed 4-82
index 4-4, 4-18
monitored 4-29
monitoring model list 4-87
range 4-4
saved 4-73
specifications 4-85
variable, algebraic (VAR) 4-114
Dynamics Snapshot Data File (*.snp) 4-51, 4-73
filename returned 7-164
read 4-51
save 4-73
summary 1-594
Dynamics Snapshot Raw Data File (*.srs) 4-74
DYRE 4-32–4-33

E

ECDI 1-162, 1-210–1-211, 1-213–1-214, 1-217, 1-219, 1-452, 1-457, 1-467, 1-472, 1-477, 1-483, 3-10

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

ECHO 1-138–1-139, 1-164
economic dispatch analysis 1-162
Economic Dispatch Data File (*.ecd) 1-162, 1-212, 1-215, 1-217, 1-219, 1-453, 1-458, 1-469, 1-474, ... 1-480, 1-486, 1-494, 1-497, 1-536, 3-10, 9-24
import 3-10
EEQV 1-165
electrical control model, wind 4-20, 4-25, 4-78, 4-134, 4-142–4-143, 7-201, 7-203
electrical equivalencing options 1-165–1-166, 1-513, 1-642
element adjustment MW 1-233
element outage severity 1-501
emergency flow limits 3-60, 3-79, 3-81, 3-83
emergency voltage limits 3-61, 3-86, 3-88
end report display in CLI 1-134
endpoint bus, multi-section line 7-85
energy consumption 3-90, 3-92
EQRD 1-166
equipment
out-of-service 1-382
out-of-service deleted 1-207–1-208, 1-424
equipment annotation, Slider Diagram 6-41, 6-55, 6-57
equipment model 4-19
equivalence
2-point type 1 bus 1-166, 1-513
radial bus 1-166, 1-513
subsystem 1-165
working case outside specified subsystem 1-642
error code 1-ii
ERUN 4-34–4-36, 7-74
ESTR 4-35–4-36, 7-74
Event Study file (*.evs) 13-2–13-7
events description 9-5
EXAM 1-168
Excel spreadsheet
ac contingency analysis 9-35
address 9-41
alignment 9-41, 9-44
autofit 9-44
border 9-42, 9-44
cell values 9-46
column width 9-48
font 9-42, 9-45
freeze 9-45
merge cells 9-45
number format 9-42
overwrite 9-36, 9-42
page 9-45
show/hide 9-36
show/hide messages 9-47
transpose data 9-48
wrap text 9-49
Excel workbook
create from Python functions 9-41, 9-50

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

hide 9-45
show 9-46
exception 1-ii
excitation limiter model 4-20, 4-25, 4-44, 4-118, 4-130–4-131, 7-122, 7-126
excitation limiter output signal 4-80, 4-91
excitation system
open circuit response simulation 4-35
open circuit step response test 7-74
response ratio test simulation 4-36
response test 4-34
excitation system model . 4-13, 4-15, 4-20, 4-25, 4-35–4-36, 4-44, 4-118, 4-130–4-131, 7-122, 7-126
exciting current 2-121, 2-124, 2-150, 2-154
exponential cost coefficient 3-91
export plot 11-5
export summary table 1-601, 1-622
extended bus name
format 1-535
returned 7-128, 7-132
setting 1-126
extended term dynamic simulation 4-29, 4-46–4-47, 7-74
extinction angle measuring bus, inverter 8-160
EXTR 1-169

F

FACT 1-170, 4-84
FACTS device 1-430, 2-18, 2-23
auxiliary signal model 4-231
name returned 7-56
blocked 1-91, 1-102–1-104, 1-106, 1-109, 1-197, 1-200, 1-203–1-204, 1-557–1-558, 1-585–1-586, 4-368, ... 4-371, 4-373, 9-18
count, by owner 8-266
data
array requirement 8-230
changed 2-18, 2-23
deleted 1-430
returned 7-60, 7-64, 8-231, 8-233, 8-236, 8-238
ignored 1-538
name changed 1-311
name returned 7-88, 7-133
next 7-133
out-of-service 1-424
owner reassignment 1-385, 1-387
sending end 1-229
terminal end 1-229
FACTS device auxiliary signal model
array index 7-55
status 7-55
FACTS device branch
count 8-232, 8-235, 8-237, 8-239
FACTS device bus 8-245, 8-252
count 8-245, 8-248, 8-250, 8-252
data
array requirement 8-242
returned 8-244, 8-246, 8-249, 8-251

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

FACTS device constant admittance load conversion . 1-91, 1-103, 1-557–1-558, 1-585–1-586, 4-368, ... 4-371, 4-373
FACTS device control mode 7-64, 8-232, 8-245
FACTS device end code 8-245
FACTS device model 4-20, 4-25, 4-37, 4-205, 4-213–4-215
array allocation table 4-213
array index 7-65
character ICON (CHRICN) changed 4-207
connection table 4-213
constant, real (CON) changed 4-208
FACTS device 4-214
status changed 4-215
GMB 4-218
integer ICON changed 4-211
name returned 7-66
removed from FACTS device 4-214
status 7-65
unconnected 4-216
user-written 4-217
variable, algebraic (VAR) changed 4-212
FACTS device power added to load 1-538
FACTS device series element 8-230–8-231, 8-233, 8-236, 8-238, 8-242, 8-244, 8-246, 8-249, 8-251
fault admittance 4-357, 4-361, 4-364
fault analysis 1-175, 1-177, 1-605
coordinate setting 1-590
fault application 1-101, 1-103, 1-105, 1-108
IEC 60909 . . . . . 1-96, 1-194, 1-196, 1-199, 1-202, 1-643
sequence data 1-371
sequence network setup 1-587
unit setting 1-592
warning 1-593
fault cleared, dynamic simulation 4-366
Fault Control Data File (*.fcd) 1-102, 1-104, 1-106, 1-110, 1-197, 1-201, 1-204, 1-536, 9-18
fault current 1-115, 9-15, 9-19
Slider Diagram 6-37, 6-46
summary table 1-102, 1-104
fault current analysis 1-88, 1-91
fault current analysis, ANSI 1-88, 1-91
fault display, Slider Diagram 6-69
fault duty time 1-115
fault location 1-561–1-562, 1-599, 4-371, 4-373, 9-17
fault memory table 4-366
Fault Specification Data File (*.bkf) 1-115, 1-536
faulted bus 6-46, 9-15, 9-19
Slider Diagram 6-37
FDNS 1-3, 1-31, 1-35, 1-44, 1-48, 1-57, 1-65, 1-171, 1-238, 1-253, 1-259, 1-272, 1-278, 1-290, 1-451, ... 1-456, 1-466, 1-471, 1-476, 1-482, 1-489, 1-491, 1-493, 1-496
fetch routine
2-terminal dc line 7-78
branch 7-84–7-85
bus 7-86–7-87, 7-95–7-96
FACTS device 7-88
fixed shunt 7-89

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

load 7-91
machine 7-90, 7-92
multi-section line member 7-94
multi-terminal dc line 7-93
tie branch 7-97–7-98
VSC dc line 7-99
file
appended 1-531
conversion
PSAS Command File to Response File 4-49
PSEB Command File to Response File 1-421
input to PSS®E 12-19
overwritten 1-173
FIND 1-174
firing angle 2-57, 2-136
bus 2-135
data returned 2-57, 2-136
dc line 7-43
measuring bus
inverter 8-169
rectifier 8-160
rectifier/inverter 7-37
first in, last out adjustment 1-232
fixed shunt 1-229, 1-443, 1-589, 8-73
admittance 2-95
changed 2-94–2-95
data
array requirement 8-66
changed 2-94–2-95
deleted 1-442–1-443
returned 7-68, 7-70, 8-67, 8-69, 8-71, 8-73
sequence data 1-608
zero sequence data changed 2-74
disconnect/reconnect 1-331–1-332
IBUS connection returned 7-89
identifer changed 1-320
identifier returned 7-134
in-service
total 7-21, 7-23, 8-8, 8-60–8-61, 8-69, 8-71
magnitude returned 7-67
next 7-134
out-of-service 1-424
positive/negative/zero sequence 1-175, 1-177–1-178
status 7-70
case comparison 1-159
zero sequence 8-69, 8-71
in-service, total 7-21, 7-23, 8-60–8-61
zero sequence admittance 2-74
fixed shunt bus
data
array requirement 8-56
returned 8-57, 8-59, 8-61, 8-63
fixed slope decoupled Newton-Raphson power flow solution (FDNS) 1-171
ac contingency analysis 1-3, 1-48
corrective actions 1-31, 1-35, 1-44
generation dispatch 1-57
post-contingency tripping 1-65

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

tripping simulation 1-31, 1-35
multi-level ac contingency analysis 1-238
corrective actions 1-253, 1-259, 1-272, 1-278
tripping simulation 1-253, 1-259, 1-290
PV analysis 1-451, 1-456, 1-466, 1-471, 1-476, 1-482
QV analysis 1-489, 1-491, 1-493, 1-496
fixed voltage excursion penalty 3-33
FLAT 1-175, 1-177
flat conditions 1-101, 1-103, 1-105, 1-108, 1-175, 1-177
flat start 1-172, 1-180, 1-219, 1-334, 1-376, 1-456, 1-471, 1-474, 1-476, 1-479, 1-482, 1-485, 1-491, 1-493, ... 1-496, 1-596, 1-630
flow
case comparison 1-157, 1-159
direction, Slider Diagram 6-55, 6-57
limits 2-150, 2-154, 3-83
overload changed 1-23, 1-26, 1-29
flow band 1-624–1-625
flow constraint 3-78
flow controlling transformer 1-624
flow monitored
PV analysis 9-27
QV analysis 9-29
flow output 4-80
flow overload elimination 1-75
flow table 1-601, 1-621
FLOW1 monitoring model 4-97, 4-99, 4-101
FLOW3 monitoring model 4-105, 4-107, 4-109
flush buffers to disk 12-6
FNSL 1-3, 1-31, 1-35, 1-44, 1-48, 1-57, 1-65, 1-179, 1-238, 1-253, 1-259, 1-272, 1-278, 1-290, 1-451, ... 1-456, 1-466, 1-471, 1-476, 1-482, 1-489, 1-491, 1-493, 1-496
folder
index 4-17, 4-45, 4-70
search order 4-45
Fortran call statement syntax 1-ii
Fortran logical unit 7-144, 12-5
frequency cutoff 1-525, 1-528, 1-606
fuel cost 3-15
scale coefficient 3-76
full Newton-Raphson power flow solution (FNSL) 1-179
ac contingency analysis 1-3, 1-48
corrective actions 1-31, 1-35, 1-44
generation dispatch 1-57
post-contingency tripping 1-65
tripping simulation 1-31, 1-35
multi-level ac contingency analysis 1-238
corrective actions 1-253, 1-259, 1-272, 1-278
tripping simulation 1-253, 1-259, 1-290
PV analysis 1-451, 1-456, 1-466, 1-471, 1-476, 1-482
QV analysis 1-489, 1-491, 1-493, 1-496

G

GAMMA angle 7-46
Gauss-Seidel solution 1-334, 1-596

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

iteration limit 2-100, 2-103
GCAP 1-181
GDIF 1-182
generation
case comparison 1-136
megawatt
scaling 1-549, 1-553
total returned 1-548, 1-551
participating bus 2-65
plant 7-72–7-73
limits 1-211, 1-214, 1-452, 1-457, 1-467, 1-472, 1-477, 1-483
loading 1-186
reactive capability 3-93, 3-95
data purged 3-140–3-141
reactive capability, OPF report 3-13
reserve unit 3-122, 3-124
OPF report 3-13
shift 1-602, 1-622
shift factor
megawatt-mile 1-341
megawatt-ohm 1-342
total 7-146, 7-213
generation dispatch 1-43, 1-47, 1-184, 9-6
corrective actions 9-3
subsystem 1-32, 1-36, 1-45, 1-49, 1-57, 1-65, 1-184, 1-240, 1-256, 1-262, 1-275, 1-281, 1-292
generation limits, reactive 3-93, 3-95
generation scaled 1-549, 1-552
generator 1-130, 1-132, 1-229
angle deviation 4-54–4-55
case comparison 1-159
conversion 1-140
data
conversion data returned 7-71
reserve data purged 3-144–3-145
sequence data 1-608
main field voltage 4-5
fixed 3-93, 3-95
main field voltage output 4-80, 4-91
non-optimized 3-29
out-of-service plant 1-169
power factor 1-176, 1-178
power output 7-81, 7-112, 7-114
power output returned 1-548, 1-551
power unbalance 4-57
PV analysis 10-3, 10-7, 10-9
reserve 3-100–3-101
reserve unit 3-123
scheduled voltage 3-45
terminal 1-187, 1-234
unit capability 3-100–3-101
unit ramp rate 3-100–3-101
var limits 1-217, 1-219
generator angle deviation 4-54–4-55
generator control 1-31, 1-35, 1-44, 1-48, 1-76, 1-78–1-79, 1-82–1-83, 1-148, 1-255, 1-261, 1-274, 1-280
subsystem 1-32, 1-36, 1-45, 1-49, 1-77, 1-80, 1-84, 1-149, 1-256, 1-262, 1-275, 1-281

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

weighting factor 1-32, 1-36, 1-45, 1-49, 1-76, 1-80, 1-84, 1-149, 1-256, 1-262, 1-275, 1-281
generator dispatch 3-98–3-99
ac contingency analysis 9-35
data purged 3-142–3-143
OPF report 3-13
generator model 4-20, 4-25, 4-44, 4-118, 4-130–4-131, 7-122, 7-126
generator model, wind 4-20, 4-25, 4-78, 4-134, 4-142–4-143, 7-201, 7-203
GENS 1-186
GENTMC monitoring model 4-93
GENTMZ monitoring model 4-89
GEOL 1-187, 1-234
GEXM 6-9
GMB FACTS device model 4-218
GMB model 4-126
GMB switched shunt model 4-225
GNE device 1-207–1-208, 1-229, 1-424
data
changed 2-26, 2-28, 2-31, 2-33
deleted 1-431
ignored 1-538
name changed 1-312
owner reassignment 1-385, 1-387
power added to load 1-538
status 2-28
GNECHR 7-75
GNEDAT 7-76
GNEINT 7-77
GNET 1-193
GOUT 6-9
governor model 4-11, 4-40, 4-44
governor response
power flow analysis 1-216, 1-218
simulation 4-39
test 7-74
graphical display setting 1-394
Graphical Report Data File (*.grp) 6-13
graphics file format 11-5
GREF array element 4-40
changed 4-11
grid, Slider Diagram 6-29
grounding resistance/reactance 2-88, 2-90, 2-92
grow bus, Slider Diagram 6-10, 6-12
GRPG 6-13
GRUN 4-38–4-39, 7-74
GSTR 4-39, 7-74
gust model, wind 4-20, 4-25, 4-78, 4-134, 4-142–4-143, 7-201, 7-203

I

IBUS
branch data returned 7-84–7-85
connection code 7-210
load data returned 7-91

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

machine data returned 7-90, 7-92
ICON index 4-18, 4-60, 4-79, 4-85
2-terminal dc line auxiliary signal model 7-30
2-terminal dc line model 7-38
branch relay model 7-153
FACTS device auxiliary signal model 7-55
FACTS device model 7-65
load model 7-102, 7-165
N-terminal dc line auxiliary signal model 7-40
N-terminal dc line model 7-50
plant model 7-122
switched shunt model 7-179
VSC dc line auxiliary signal model 7-194
VSC dc line model 7-199
wind model 7-201
identifier returned 1-222
IEC 60909 standard 9-17
annotation, Slider Diagram 6-46
data export to Excel spreadsheet 9-37
fault analysis 1-175, 1-177, 1-196, 1-199, 1-202
fault display, Slider Diagram 6-73
IEC Fault Calculation Data File (*.iec) 1-197, 1-201, 1-204, 9-18, 9-37
IECS 1-96, 1-194, 1-196, 1-199, 1-202, 1-643, 9-17, 9-37
IEEE Common Format Data File 1-538
image
export, Slider Diagram 6-8
import, Slider Diagram 6-15
type 6-8
IMPC 1-209
impedance 1-140, 4-357, 4-361, 4-364, 8-29
annotation, Slider Diagram 6-39, 6-42
between buses 8-135
branch 8-110
zero sequence 8-110, 8-124, 8-152
branch check 1-116, 1-118
case comparison 1-159
correction table 1-130, 1-132, 1-229, 2-129, 2-149, 2-153
display, Slider Diagram 6-70
I/O code 2-120, 2-123, 2-149, 2-153, 7-189, 7-210, 8-117, 8-132, 8-145
machine
negative sequence 8-29
positive sequence 8-29
zero sequence 8-29
measured 8-135
step-up transformer 8-29
transformer 8-123, 8-152
zero-sequence grounding 8-124, 8-136
import summary table 1-601, 1-622
incremental transfer, PV analysis 10-7
INDDT1 7-79
indicator, activity selector 7-163
INDINT 7-82
Induction 8-298
inertial power flow analysis 1-216, 1-218

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

inertial power flow dispatch error 7-168
infeasible reactive power generation 3-58
inhibiting field increase/decrease 3-93, 3-95
INILOD 7-91
INITIAL CONDITIONS SUSPECT error 7-143
INLF 1-216, 1-218, 4-53, 7-168
in-line fault 1-561, 1-598, 4-370, 4-372
input from interactive device 12-22
input values
2-terminal dc line 7-34, 7-37, 7-75–7-77, 7-151–7-152, 8-160, 8-162, 8-166, 8-168
2-terminal dc line auxiliary signal model 7-30
2-terminal dc line converter 8-174, 8-176, 8-178, 8-181
2-terminal dc line model 7-38
2-winding transformer 7-208, 7-210
3-winding transformer 7-186–7-187, 7-189, 8-147
short circuit 7-156
winding 7-206–7-207
3-winding transformer winding 7-204
area 7-2, 7-4–7-5, 8-255, 8-258, 8-261–8-262
branch 7-11–7-12, 7-14, 7-16, 7-157, 8-91, 8-93, 8-96, 8-98, 8-104, 8-106, 8-110, 8-112
branch relay model 7-153
bus 7-20–7-21, 7-23, 7-26, 7-158, 8-4, 8-6, 8-8, 8-10–8-11, 8-13, 8-15, 8-17, 8-19, 8-21, 8-23, 8-26, 8-29–8-30, 8-32, 8-36, 8-38, 8-41, 8-44–8-45, 8-47, 8-49, 8-51, 8-53, 8-55, 8-57, 8-59, 8-61, 8-63, 8-65, 8-67, 8-69, 8-71, 8-73, 8-75, 8-78, 8-80, 8-83, 8-85, 8-87, 8-99, 8-113, 8-128, 8-140, 8-157, 8-170, 8-182, 8-193, 8-205, 8-216, 8-228, 8-240, 8-253, 8-264, 8-275, 8-286, 8-290, 8-292, 8-294, 8-296– 8-297, 8-301, 8-303, 8-306, 8-308–8-309
converter 7-34, 7-46
dc line 7-43, 7-46, 7-49
dynamics 7-52–7-54
FACTS device 7-60, 7-64, 8-232–8-233, 8-236, 8-238, 8-245, 8-247, 8-250, 8-252
FACTS device auxiliary signal model 7-55
FACTS device model 7-65
fixed shunt 7-67–7-68, 7-70, 8-57, 8-59, 8-61, 8-63, 8-67, 8-69, 8-71, 8-73
load 7-111, 8-36, 8-38, 8-41, 8-44, 8-47, 8-49, 8-51, 8-53, 8-290, 8-292, 8-294, 8-296, 8-301, 8-303, ... 8-306, 8-308
load characteristic 7-106, 7-108
load model 7-102, 7-165
machine 7-79, 7-81–7-82, 7-112, 7-114, 7-117, 7-162, 8-23, 8-26, 8-29–8-30
multi-terminal dc line converter 8-185, 8-187, 8-189, 8-191, 8-197, 8-199, 8-201, 8-204
N-terminal dc line auxiliary signal model 7-40
N-terminal dc line model 7-50
owner 7-146, 8-266, 8-269, 8-272–8-273
plant 8-13, 8-15, 8-17, 8-19
plant model 7-201
switched shunt 7-177–7-178, 8-78, 8-80, 8-83, 8-85
switched shunt model 7-179
system totals 7-183
transformer 8-116, 8-119, 8-123, 8-126, 8-131, 8-133, 8-135, 8-138, 8-145, 8-152, 8-155
VSC dc line 7-196–7-198, 8-208, 8-210, 8-212, 8-214
VSC dc line auxiliary signal model 7-194
VSC dc line converter 8-220, 8-222, 8-225, 8-227
VSC dc line model 7-199
zone 7-213, 8-277, 8-280, 8-283–8-284
INTA 1-220
integer ICON changed 4-9

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

2-terminal dc line model 4-188
auxiliary signal model, 2-terminal dc line 4-260
auxiliary signal model, FACTS device 4-247
auxiliary signal model, multi-terminal dc line 4-254
auxiliary signal model, VSC dc line 4-266
FACTS device model 4-211
line relay model 4-166
load model 4-152
multi-terminal dc line model 4-182
plant model 4-124, 4-280
switched shunt model 4-223
VSC dc line model 4-194
wind model 4-138
integration algorithm (STORE) 4-12
inter-area transfer 1-130, 1-229, 1-428
changed 2-133–2-134
deleted 1-428
identifier changed 1-309
megawatt returned 7-193
interchange
area tie flows 1-220
case comparison 1-136
limits 1-601, 1-621
two opposing systems 1-393
tolerance 2-2, 7-4, 8-258
zone tie flows 1-221
interface 1-525, 1-528
limits summary table 1-602, 1-622
loading violations 1-5, 1-9
interface flow 3-12, 3-16, 9-5
constraint 3-8, 3-105, 3-111, 3-153
3-winding transformer winding 3-103
branch 3-104, 3-107
OPF report 3-13
cost coefficient 3-12
flow constraint data purged 3-146
limit type 3-105
overload violations 1-43
PV analysis 10-3, 10-7, 10-9
type 3-105
interface flow constraint 3-148
interior shift factor multiplier, OPF 3-35
interior subsystem branch 8-101
INTZ 1-221
inverter 2-135, 8-197
angle limit 8-161
bus 8-160, 8-168
commutating capacitor reactance magnitude 8-163
commutating transformer resistance/reactance 8-163
converter angle 8-163
dc voltage 8-163
extinction angle measuring bus 8-160
firing angle measuring bus 8-169
tap ratio limit 8-161
tap settings 8-163
transformer ratio 8-163

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

IPFC slave device 2-18, 2-25
IPLAN File (*.irf) 12-18
IPLAN programming language
command syntax 1-ii
unit number for Fortran unit 7-144
island
analysis 1-629
count 9-5, 9-27, 9-29
disconnected 1-22, 7-100
frequency simulation 4-29
islanding contingencies threshold 1-240, 1-256, 1-262, 1-275, 1-281, 1-292, 1-500, 1-502
isolated buses 3-57
iteration count returned 7-101
iteration limit 7-168
capacitor commutated converter 8-161
Gauss-Seidel 2-100, 2-103
Newton solution 2-100, 2-103, 2-140

J

JBUS connection code 7-210
JOIN 1-222
join pairs of buses 1-222
JPEG image 6-8

K

kV display setting 1-635–1-636
kVA display setting 1-398

L

lagging power factor 3-93, 3-95
rated 3-96
Lagrange multipliers, OPF Optimization Log File 3-2
lambda mismatch step size 3-3
lambda tolerance 3-36
LAMP 1-224–1-225
leading power factor 3-93, 3-95
rated 3-96
library, model programs 4-3, 4-22
line charging 1-203, 9-18
total 2-6
zero sequence total 2-70–2-71
line end fault 1-197, 1-200, 1-203, 9-17
line fault 6-37, 6-46
line flow 1-393
line length 2-6, 7-11, 8-107
case comparison 1-157, 1-159
line loading 1-323
limit 1-504
relief 1-232–1-233
transactions 1-233
line loading relief calculation 1-232
line losses, case comparison 1-157, 1-159

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

line out fault 1-197, 1-200, 1-203, 9-17
line overload contingencies 1-260, 1-279
line R, X, B 1-157
case comparison 1-159
line rating 1-159
case comparison 1-157, 1-159
Slider Diagram 6-48
line relay model 4-20, 4-25, 4-50, 4-160, 4-168–4-170
array allocation table 4-168
branch from bus status 4-170
character ICON (CHRICN) changed 4-162
connection table 4-168
constant, real (CON) changed 4-163
integer ICON changed 4-166
unconnected 4-171
user-written 4-172
variable, algebraic (VAR) changed 4-167
line relay model removed from branch 4-169
line resistance 1-323
changed, in-service nontransformer branches 1-323
scaling factor 1-323
line shunt 1-130, 1-132, 1-229, 2-6
admittance 7-12
admittance at JBUS 7-12
case comparison 1-157, 1-159
deleted branches 1-222
display setting 1-225
from bus end 8-110
positive sequence 1-175, 1-177–1-178
reactive 2-6
reporting 1-225
sequence data 1-608
to bus end 8-110
line style, Slider Diagram 6-29, 6-50, 6-52
linear constraint
dependency 3-112
OPF report 3-13
record, data purged 3-149
slack value 3-112
linear constraint dependency equation
active power dispatch 3-113
variable purged 3-154
adjustable bus load 3-108
variable purged 3-150
adjustable bus shunt 3-109
variable purged 3-151
branch flow 3-110
variable purged 3-152
bus voltage angle 3-118
variable purged 3-158
bus voltage magnitude 3-119
variable purged 3-159
interface flow constraint 3-111
variable purged 3-153
purge variables 3-115
reactive power dispatch 3-114

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

variable purged . . . 3-155
switched shunt . . . 3-116
variable purged . . . 3-156
transformer control . . . 3-117
variable purged . . . 3-157
linear cost coefficient . . . 3-91
linear cost curve . . . 3-76
linear cost table . . . 3-90
data purged . . . 3-137
OPF report . . . 3-13
Linear Dynamic Analysis Program . . . 4-4
linear network model . . . 1-393, 10-2
line-end fault . . . 1-101, 1-103, 1-105, 1-108
line-line-to-ground fault 1-106, 1-109, 1-196, 1-199, 1-202, 1-558, 1-561, 1-598, 4-368, 4-370, 4-372, . . . 9-17
Slider Diagram . . . 6-37, 6-46
line-out fault . . . 1-101, 1-103, 1-105, 1-108
line-to-ground fault . 1-101–1-106, 1-108–1-110, 1-196, 1-199, 1-202, 1-558, 1-561, 1-598, 4-368, 4-370, . . . 4-372, 9-17, 9-20
Slider Diagram . . . 6-37, 6-46
line-to-line fault . . . 1-196, 1-199, 1-202, 1-558, 1-561, 4-368, 9-17
Slider Diagram . . . 6-37, 6-46
LIST . . . 1-229
LLRF . . . 1-232
load . . . 1-130, 1-229, 1-434
area assignment . . . 7-2
area number . . . 7-111
array index returned . . . 7-110
case comparison . . . 1-136
characteristic . . . 1-159
constant current, in-service . . . 8-49–8-51, 8-303–8-304, 8-306
constant MVA, in-service . . . 8-49, 8-51, 8-303
count . . . 8-47, 8-50, 8-52–8-53, 8-301, 8-304, 8-306, 8-308
area assignment . . . 8-255
curtailment . . . 1-44
data
array requirement . . . 8-46, 8-298
changed . . . 2-36, 2-41
deleted . . . 1-434
deleted from bus . . . 1-433, 1-435
returned . . . 8-47, 8-49, 8-51, 8-53, 8-300, 8-302, 8-305, 8-307
dynamic output channel . . . 4-112
identifier changed . . . 1-316
identifier returned . . . 1-222
in-service . . . 8-49, 8-51
monitoring . . . 4-79
out-of-service . . . 1-424
owner number . . . 7-111
owner reassignment . . . 1-385, 1-387
zone number . . . 7-111
load bus data
array requirement . . . 8-33, 8-287
returned . . . 8-35, 8-37, 8-40, 8-43, 8-289, 8-291, 8-293, 8-295
load characteristic model . . . 4-20, 4-25, 4-146, 4-155–4-157, 7-102, 7-104, 7-165, 7-167
load control . . . 1-32, 1-35, 1-48, 1-76, 1-79, 1-83, 1-148, 1-255, 1-261, 1-274–1-275, 1-280

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

subsystem 1-32, 1-36, 1-45, 1-49, 1-77, 1-80, 1-84, 1-149, 1-256, 1-262, 1-275, 1-281
weighting factor 1-32, 1-36, 1-45, 1-49, 1-76, 1-80, 1-84, 1-149, 1-256, 1-262, 1-281
load cost scale coefficient 3-69
load drop
compensating impedance 7-208, 8-124, 8-152
compensating resistance/reactance 2-150, 2-154
compensating transformer 1-137
voltage returned 1-138
load model 4-25, 4-43, 4-146, 4-154–4-157
array allocation table 4-154
array index 7-102
character ICON (CHRICN) changed 4-148
connection report 4-43
connection table 4-154
constant, real (CON) changed 4-149
integer ICON changed 4-152
name returned 7-104, 7-167
removed from load 4-155
removed from subsystem 4-155
status 4-157, 7-102
unconnected 4-158
user-written 4-159
load relay model 4-20, 4-25, 4-146, 4-155–4-157, 7-102, 7-104, 7-165, 7-167
load shed 1-43, 1-47, 1-75, 9-6
ac contingency analysis 9-35
count 9-3
load status 2-39, 2-41, 7-82, 7-111, 8-47, 8-301
case comparison 1-159
inferred 8-36, 8-290
load subsystem selection 7-165, 7-167
Load Throwover Data File (*.thr) 1-3, 1-32, 1-37, 1-45, 1-49, 1-57, 1-65, 1-184, 1-240, 1-256, 1-262, 1-275, 1-281, 1-292, 1-453, 1-458, 1-469, 1-474, 1-480, 1-486, 1-489, 1-491, 1-494, 1-497, 1-536, 9-4, ... 9-24, 9-31
loadflow control quadratic penalty 3-37
loading basis 1-163
bridge active power, percent 8-234, 8-247
from bus series current, percent 8-247
machine MVA as percent of machine base 8-26
MVA percent of plant MVA base 8-15
series element, sending/terminal bus, percent 8-234
shunt, percent 8-234, 8-247
Slider Diagram 6-56–6-57
loading table 1-21, 1-24, 1-27
loading violation report 1-5–1-6, 1-9–1-10
loss coefficient 2-158
from bus, VSC dc line converter 8-222
LOUT 1-225, 1-235
LSTO 3-13
LTAP 1-236
LV bus fault at Power Station Unit 1-197, 1-200, 1-203

M

machine 1-432, 1-436
array index returned 7-116

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

count 8-23, 8-26, 8-29–8-30
count, by owner 8-266
data
array requirement 8-22
changed 2-44, 2-48
deleted 1-432, 1-436
returned 7-79, 7-81–7-82, 7-112, 7-114, 7-117, 8-23, 8-25, 8-28, 8-30
sequence data changed 2-75, 2-77, 2-79–2-80
status returned 7-118
disconnect/reconnect 1-314–1-315, 1-329–1-330
electrical power output 4-80
identifier changed 1-313, 1-317
identifier returned 1-222, 7-135, 7-137
monitoring 4-79
next 7-135, 7-137
out-of-service 1-424, 1-539, 4-367
owner
generation total 7-146
reassignment 1-385, 1-387
power output 2-49
reactive power output 4-80
rotor angle output 4-80
speed deviation output 4-91
status 1-322, 2-48, 7-82, 7-117, 8-23
case comparison 1-159
tripped 4-367
machine angle 4-79
machine capability curve 1-181
Machine Capability Data File (*.gcp) 1-181, 1-536
machine contingencies overload ranking 1-239, 1-254, 1-260, 1-273, 1-279, 1-291, 1-501
machine electrical power output 4-91
machine fault analysis resistance/reactance 2-75, 2-79–2-80
machine impedance 1-140, 7-81, 7-114
output 4-80
Machine Impedance Data File (*.rwm) 1-322, 1-536, 1-539
machine load 4-39, 7-80, 7-112
machine MVA base 2-33, 2-49, 8-26
machine power limits 1-549, 1-552, 4-53
machine reactive power
limits changed 1-181
machine reactive power output 4-91
machine resistance/reactance 2-34, 2-49
machine rotor angle output 4-91
machine short circuit current returned 7-162
machine speed deviation output 4-80
machine terminal
current
dynamic output channel 4-93
output 4-80
voltage output 4-80, 4-91
magnetizing admittance 7-187, 8-124, 8-136
I/O code 2-120, 2-123, 2-149, 2-153, 7-189, 7-210, 8-117, 8-132, 8-145
magnetizing conductance/susceptance 2-121, 2-124, 2-150, 2-154
master legend report 1-5, 1-9

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

matrix growth factor 1-306
Matrix Output File 4-5
maximum fault current 9-16, 9-21
MBID 1-307–1-312, 1-318–1-321
MBIDGNE 1-312
MCRE 1-322
mechanical model, wind 4-20, 4-25, 4-78, 4-134, 4-142–4-143, 7-201, 7-203
megawatt transfer 9-25
changed 9-24
Megawatt-Mile (MWMI) Data Input File (*.dat) 1-340
megawatt-mile shift factor 1-341
megawatt-ohm shift factor 1-342
memory cell, time derivative 4-12
metered bus 2-5, 2-51, 2-53, 2-149, 2-153, 7-14, 8-104, 8-112, 8-116, 8-127, 8-160, 8-168, 8-174, 8-181
case comparison 1-159
tie line case comparison 1-157
metered dc bus 2-62
microfarads display setting 1-628
Midwest megawatt-mile calculation 1-339
mismatch case comparison 1-136
mismatch plot, QV analysis 10-7, 10-9
mismatch tolerance
ac contingency analysis 1-2, 1-30, 1-34, 1-43, 1-56, 1-64
ac corrective actions 1-80, 1-84
corrective actions 1-30, 1-34, 1-43, 1-76, 1-149, 1-252, 1-258, 1-271, 1-277
dc contingency checking 1-147, 1-149
dual variable problem, OPF 3-59
generation dispatch 1-56
interchange limits 1-395
multi-level ac contingency analysis 1-237, 1-252, 1-258, 1-271, 1-277, 1-289
post-contingency tripping 1-64
PV analysis 1-453, 1-458, 1-469, 1-474, 1-479, 1-486
QV analysis 1-489, 1-491, 1-494, 1-497
transmission interchange limits 1-601, 1-622
tripping simulation 1-30, 1-34, 1-252, 1-258, 1-289
missing contingencies report 1-5, 1-9
missing monitored elements report 1-4, 1-9
missing monitored voltage buses report 1-5, 1-9
MLST 4-44
mode switch dc voltage 2-61, 2-141, 7-34, 7-46, 8-162, 8-187
model deleted from dynamic simulation library 4-23
model folder 4-71
count returned 7-120
deleted from search 4-17
index 7-121
name changed 4-70
name returned 7-121
model library
count returned 7-124–7-125
delete model 4-23
removed from search 4-22

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

MODEL NOT AVAILABLE error 7-143
model parameters 7-52–7-54
model references 4-32–4-33
deleted from output channel monitoring model table 4-88
model search 4-17, 4-22, 4-70–4-71
folder 4-2
library 4-3
model variables 7-54
MODR 1-323
monitored branch 9-3, 9-22–9-24, 9-27
monitored branch flow 9-5, 9-27, 9-38
ac contingency analysis 9-35
monitored bus voltage 9-5, 9-27, 9-29, 9-38, 9-40
ac contingency analysis 9-35
monitored element 1-233
rating set 9-4
report 1-4, 1-8
sorted 1-153
summary report 1-4, 1-8
Monitored Element Description File (*.mon) 1-153, 1-500, 1-502, 9-4, 9-22, 9-24, 9-31
monitored generator bus 9-24–9-25
count 9-31
monitored interface 9-23
monitored interface flow 9-3, 9-24–9-25, 9-27, 9-38
ac contingency analysis 9-35
monitored load 9-27, 9-38
monitored load bus 9-24–9-25
monitored plant 9-27, 9-29, 9-38, 9-40
monitored tie line 1-394, 1-601, 1-621
monitored voltage bus 9-4, 9-25, 9-31
monitored voltage record 9-4, 9-25, 9-31
motor load
active power output returned 1-548, 1-551
megawatt scaling 1-549, 1-553
megawatt total returned 1-548, 1-551
motor scaled 1-549, 1-552
motors in subsystem, data returned 1-549, 1-552
MOVE 1-314–1-315, 1-325–1-333
MoveIndMacs 1-315
MRUN 4-46–4-47, 7-74, 7-143
MSLV 1-334, 2-96, 2-98, 2-101, 2-103
MSTR 4-47, 7-74, 7-143
MTDC 1-337
multi-level ac contingency analysis 1-237–1-238, 1-253, 1-259, 1-272, 1-278, 1-290
corrective actions 1-252, 1-258, 1-271, 1-277
tripping simulation 1-252, 1-258, 1-289
multiple event contingencies 1-155
Multiple Run Report, ACCC 1-4, 1-8
multi-section line 1-130, 1-229, 1-438, 7-84
branch identifier returned 7-139
branch, next 7-139

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

case comparison 1-159
data
changed 2-51, 2-53
deleted 1-438
returned 7-94, 7-97
dummy bus 7-98, 7-185
identifier changed 1-319
metered end case comparison 1-159
metered end changed 2-53
reporting 1-338
status changed 2-53
multi-terminal dc line 1-437, 2-54, 8-184–8-185, 8-187, 8-189, 8-191
auxiliary signal model 4-233
branch count 8-186–8-187, 8-189, 8-192
bus data
changed 2-54
deleted 1-447
control mode 8-185
changed 2-59–2-60
data
array requirement 8-184
deleted 1-437
returned 7-46, 7-49, 8-185, 8-187, 8-189, 8-191
dc link
changed 2-62
data deleted 1-449
mode switch voltage changed 2-59–2-60
multi-terminal dc line model 4-199
name changed 1-318
name returned 7-93, 7-138
next 7-138
out-of-service 1-424
multi-terminal dc line converter 1-448, 8-195–8-196, 8-198, 8-201, 8-203
bus count 8-197, 8-200, 8-202, 8-204
data
array requirement 8-195
changed 2-56
deleted 1-448
returned 7-43, 8-196, 8-198, 8-201, 8-203
firing angle status 7-43
tap status 7-43
multi-terminal dc line model 4-16, 4-20, 4-25, 4-173, 4-199–4-200
character ICON (CHRICN) changed 4-178, 4-303–4-311, 4-313, 4-315, 4-317, 4-319–4-338
connection table 4-174
constant, real (CON) changed 4-179
integer ICON changed 4-182
multi-terminal line status changed 4-200
removed from multi-terminal dc line 4-199
variable, algebraic (VAR) changed 4-183
Multi-Terminal DC Line Solution Report 1-337
mutual resistance/reactance, branch-to-branch 2-82
mutual sequence data 1-608
mutually coupled branch, suppress equivalencing 1-166, 1-513
MVA display setting 1-374, 1-398, 1-627
MVA loading 7-80, 7-112, 7-117
3-winding transformer winding 7-204
branch 7-16, 8-93, 8-106

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

percent of machine MVA 8-26
percent of plant MVA 8-15
Slider Diagram 6-56–6-57, 6-59
VSC dc line 8-222
MVA mismatch 7-181, 9-5, 9-27, 9-29
MVA rating conversion 1-146, 1-148, 1-394, 1-601, 1-621
MVA rating, from bus, VSC dc line converter 8-222
Mvar controlling transformer 1-625
MW generations dispatch 1-75
MW mismatch 1-146, 1-148, 1-151, 1-339, 1-394, 1-601, 1-621
threshold 1-500, 1-502
MWMI 1-339

N

narrow format power flow solution 1-397
negative impedance, branch check 1-116, 1-118
negative pole
converter 2-56, 8-197
voltage controlling converter
bus 8-185, 8-191
index returned 2-60
negative sequence
bus shunt
admittance 2-73, 2-77
case comparison 1-159
current 7-158
current 9-15–9-16, 9-19
fault analysis 2-75, 2-79–2-80
fault current 7-158
generator impedance, case comparison 1-159
machine impedance 7-114, 8-29
short circuit current 7-156–7-157, 7-162
Thevenin impedance 9-16, 9-21
voltage 7-158
net area interchange 7-2, 7-4
net boundary bus mismatches 1-114
net generation with load 1-193, 1-367
net load 1-165, 1-642
net zone interchange 7-213
NETG 1-367
network admittance matrix 1-381
factorized 1-170, 4-84
Network Admittance Matrix Data File (*.dat) 1-381
network bus fault 1-197, 1-200, 1-203
network case comparison, Slider Diagram 6-3
network connectivity check 1-143
network equivalent 1-554
network frequency dependence 4-58
network items, unbound, Slider Diagram 6-29
network solution
convergence monitor 4-4
iterations 4-29
new power flow case 1-370

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

Newton solution
acceleration factor 2-101, 2-103, 2-141
convergence tolerance 1-237, 1-372, 1-622
ac contingency analysis 1-2, 1-80, 1-84
ac contingency analysis, corrective actions 1-30, 1-34, 1-43, 1-76
ac contingency analysis, generation dispatch 1-56
ac contingency analysis, post-contingency tripping 1-64
ac contingency analysis, tripping simulation 1-30, 1-34
dc contingency analysis 1-147
dc contingency analysis, corrective actions 1-149
interchange limits 1-395
multi-level ac contingency analysis 1-252, 1-258
multi-level ac contingency analysis, corrective actions 1-271, 1-277
multi-level ac contingency analysis, tripping simulation 1-289
PV analysis 1-453, 1-458, 1-469, 1-474, 1-479, 1-486
QV analysis 1-489, 1-491, 1-494, 1-497
single branch outage contingency 1-500, 1-502
transmission interchange limits 1-601
decoupled Newton-Raphson 1-375
fixed slope decoupled Newton-Raphson 1-171, 1-253, 1-259
ac contingency analysis 1-3
ac contingency analysis, corrective actions 1-31, 1-35, 1-44
ac contingency analysis, generation dispatch 1-57
ac contingency analysis, post-contingency tripping 1-65
ac contingency analysis, tripping simulation 1-31, 1-35
multi-level ac contingency analysis 1-48, 1-238
multi-level ac contingency analysis, corrective actions 1-253, 1-259, 1-272, 1-278
multi-level ac contingency analysis, tripping simulation 1-290
PV analysis 1-451, 1-456, 1-466, 1-471, 1-476, 1-482
QV analysis 1-489, 1-491, 1-493, 1-496
full Newton-Raphson 1-179
ac contingency analysis 1-3
ac contingency analysis, corrective actions 1-31, 1-35, 1-44
ac contingency analysis, generation dispatch 1-57
ac contingency analysis, post-contingency tripping 1-65
ac contingency analysis, tripping simulation 1-31, 1-35
multi-level ac contingency analysis 1-48, 1-238
multi-level ac contingency analysis, corrective actions 1-253, 1-259, 1-272, 1-278
multi-level ac contingency analysis, tripping simulation 1-253, 1-259, 1-290
PV analysis 1-451, 1-456, 1-466, 1-471, 1-476, 1-482
QV analysis 1-489, 1-491, 1-493, 1-496
iteration limit 2-100, 2-103, 2-140
mismatch convergence tolerance 2-101, 2-103
non-divergent improvement factor 2-101, 2-104
no-load losses 2-121, 2-124, 2-150, 2-154
non-converged contingencies report 1-5, 1-9
non-converged networks report 1-21, 1-24, 1-27
non-dispatchable machines 4-53
non-divergent improvement factor 2-101, 2-104
non-divergent solution 7-168
ac contingency analysis 1-3, 1-44, 1-48, 1-57
corrective actions 1-31, 1-35
post-contingency tripping 1-65
tripping simulation 1-31, 1-35
multi-level ac contingency analysis 1-238, 1-253, 1-259, 1-272, 1-278
tripping simulation 1-290
Newton power flow setting 1-373
Newton solution 1-172, 1-180
PV analysis 1-451, 1-456, 1-466, 1-471, 1-476, 1-482

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

QV analysis 1-489, 1-491, 1-493, 1-496
non-metered bus 2-120, 2-123, 7-14, 8-91, 8-98, 8-104, 8-112, 8-116, 8-127, 8-139, 8-155, 8-160, 8-168, ... 8-174, 8-181
number 7-189
non-metered end bus 2-28
non-recording routines 7-1, 8-1, 9-1–9-2, 9-34, 9-41, 12-1
non-transformer branch 1-429
data
changed 2-3, 2-5
deleted 1-429
zero sequence data changed 2-70–2-71
disconnect/reconnect to bus 1-326
IBUS fault 4-361
identifier changed 1-310
identifier returned 1-222
percent units 1-374
rating units 1-370
set to in-service 4-360
set to out-of-service 4-363
NOPF 3-24, 7-168
normal flow limits 3-79, 3-81, 3-83
normal voltage limits 3-86, 3-88
NSOL 1-375
N-terminal dc line auxiliary signal model
array index 7-40
name returned 7-41
status 7-40
N-terminal dc line model
array index 7-50
name returned 7-51
status 7-50
numerical order return
branch 7-130–7-131
bus 7-86–7-87, 7-132, 7-145
specified area 7-170
specified base voltage 7-171
specified zone 7-172
fixed shunt identifier 7-134
load identifier 7-136
machine identifier 7-135, 7-137
multi-section line branch 7-139
tie branch identifier 7-140–7-141
NXTLOD 7-136

O

off-diagonal terms, ties 1-381
off-line generator control 1-36, 1-47–1-48, 1-79, 1-83, 1-261, 1-280
subsystem 1-36, 1-49, 1-80, 1-84, 1-262, 1-281
weighting factor 1-36, 1-49, 1-80, 1-84, 1-262, 1-281
ohms display setting 1-628
OLTL 1-504
OLTR 1-504
on-line plants 1-186
OPEN 7-144

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

open bus voltage limits 3-25
opened end bus 1-561
operating voltages 1-89, 1-92
OPF data
default 3-63
purged 3-125
OPF Optimization Log File (*.log) 3-53
Lagrange multipliers appended 3-2
sensitivity values appended 3-2
OPF solution 3-24
Optimal Power Flow Data File (*.rop) 3-56–3-57
Optimal Power Flow Options File (*.opt) 3-62
OPTN . 1-113, 1-125–1-127, 1-144–1-145, 1-173, 1-192, 1-225–1-226, 1-228, 1-306, 1-338, 1-372–1-374, . . 1-392, 1-398, 1-404–1-405, 1-506, 1-587, 1-590–1-593, 1-612–1-613, 1-627–1-628, 1-635–1-636, ... 1-641
ORDR 1-379
OTDF 1-380, 9-22–9-23
Outage Statistics Data File (*.prb) 1-525, 1-528, 1-607
outage statistics, contingency summary report 1-524, 1-527
out-of-service display, Slider Diagram 6-40, 6-44, 6-48
out-of-step scanning 4-62
Output Outage Statistics Data File (*.sts) 1-525, 1-528
OUTS 1-382
overload display setting 1-146
overload ranking 1-239, 1-254, 1-260, 1-273, 1-279, 1-291, 1-501
branch contingencies 1-239, 1-254, 1-273, 1-291, 1-499, 1-501
line contingencies 1-260, 1-279
overload report 1-21–1-22, 1-24–1-25, 1-27–1-28, 1-146–1-147
overload voltage, Slider Diagram 6-51, 6-53
overloaded machines 1-187, 1-234
overwrite file 1-173
owner 1-130, 1-229
assignment 1-383, 1-385, 1-387
count
2-winding transformer 8-116
3-winding transformer 7-189, 8-132
3-winding transformer winding 8-145
branch 7-14, 8-91, 8-104
machine 7-82, 7-117, 8-23
subsystem 8-266, 8-270, 8-272–8-273
VSC dc line 8-208
data
array requirement 8-265
changed 2-64
returned 7-146, 8-266, 8-268, 8-271, 8-273
fraction
2-winding transformer 2-150, 2-154, 8-120
3-winding transformer 2-121, 2-124, 7-186, 8-133
3-winding transformer winding 8-149
branch 7-11, 8-93, 8-107
machine 2-34, 2-49, 7-113, 8-26
non-transformer branch 2-6
VSC dc line 2-160, 8-210
name returned 7-148

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

number 7-189
number returned 7-149
status returned 7-150
totals 1-135, 1-389
owner bus load 8-272
owner generation 8-272
owner load 8-272
data returned 7-146
owner model 4-146, 4-155–4-156
owner power losses 8-272
owner power total 4-79
owner subsystem 5-13–5-14
initialization 5-15
selection 7-165, 7-167
2-winding transformer 8-115–8-116, 8-119, 8-123, 8-126
3-winding transformer 8-130–8-131, 8-133, 8-135, 8-138
3-winding transformer winding 8-142, 8-144, 8-147, 8-152, 8-154
branch 8-101, 8-103, 8-105, 8-109, 8-111
branch flow 8-89–8-90, 8-92, 8-95, 8-97
OWNM 1-383, 1-385, 1-387
OWNM_2 1-385
OWNM_3 1-387
OWNR 1-389

P

page length, reporting device 1-226, 1-228
parallel transformer branch check 1-116, 1-118
parameter set display 10-2, 10-4, 10-6
parameters changed, dynamic simulation 4-29
partial bus name 1-174
participating bus 2-65
participation factor, converter 2-57, 8-199
PATH 1-390
path to ground 1-598, 4-370, 4-372
pause, dynamic simulation 4-34, 4-38, 4-46, 4-52
peak current 9-20
per unit display setting 1-592, 1-628, 1-635–1-636
percent loading 1-6, 1-10, 9-11
increase in overload 1-23, 1-26, 1-29
line rating, Slider Diagram 6-50, 6-52
reduced capacity 1-340
Slider Diagram 6-59
percent of contributed reactive power 7-113, 7-177
converter 8-222
FACTS device 2-24, 8-233
plant 2-67, 8-15
switched shunt 2-111, 8-80
VSC dc line 2-158
percent of flow rating 1-6, 1-10, 1-23, 1-26, 1-29, 1-403, 1-606, 9-12
percent of peak load 1-606
percent of power load to convert 1-142
percent of rating set

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

ac corrective actions 1-76, 1-80, 1-84, 1-256, 1-262, 1-275, 1-281
current loading
3-winding transformer winding 7-204
branch 7-16, 8-93, 8-106
dc corrective actions 1-149
excessive branch loading check 1-453, 1-458, 1-469, 1-474, 1-479, 1-486
MVA loading
3-winding transformer 7-204
branch 7-16, 8-93, 8-106
overloaded element report 1-147, 1-395, 1-601, 1-622
ranking 1-240, 1-256, 1-262, 1-275, 1-281, 1-292, 1-500, 1-502
reliability assessment 1-525, 1-528
period reserve
constraint 3-122–3-124
data purged 3-162
OPF report 3-13
unit, data purged 3-163
perturbation 4-4–4-5
phase closed 1-558
unbalance 1-560
phase open 1-598, 4-370, 4-372
phase shift adjustment 1-192, 1-392
ac contingency analysis 1-2, 1-43, 1-47, 1-56, 1-75, 1-78, 1-82
corrective actions 1-30, 1-34
post-contingency tripping 1-64
tripping simulation 1-30, 1-34
decoupled Newton-Raphson power flow solution 1-375
fixed slope decoupled Newton-Raphson power flow solution 1-171
full Newton-Raphson power flow solution 1-179
inertial/governor response power flow analysis 1-216, 1-218
multi-level ac contingency analysis 1-237, 1-252, 1-259, 1-271, 1-277
tripping simulation 1-289
PV analysis 1-450, 1-455
base case 1-465, 1-468, 1-470, 1-473, 1-475, 1-479, 1-481, 1-485
contingency case 1-468, 1-473, 1-479, 1-485
QV analysis 1-488, 1-490, 1-492, 1-495
phase shift angle 9-6
ac contingency analysis 9-35
OPF output 3-58
winding 7-204, 8-121, 8-149
phase shifter control 1-32, 1-36, 1-44, 1-48, 1-76, 1-79, 1-83, 1-148, 1-255, 1-261, 1-274–1-275, 1-280
angle adjustment 1-47, 1-75
corrective actions 9-3
locked/regulating 1-393, 1-600, 1-620
subsystem 1-32, 1-36, 1-45, 1-49, 1-77, 1-80, 1-84, 1-149, 1-256, 1-262, 1-275, 1-281
suppress equivalencing 1-165, 1-642
weighting factor 1-32, 1-36, 1-45, 1-49, 1-76, 1-80, 1-84, 1-149, 1-256, 1-262, 1-281
physical units display setting 1-592
pitch angle output 4-80, 4-91
pitch control model, wind 4-20, 4-25, 4-78, 4-134, 4-142–4-143, 7-201, 7-203
plant 1-130, 1-229, 1-441
bus count 8-14, 8-16, 8-18–8-19
data
changed 2-66–2-67
deleted 1-441
out-of-service 1-424

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

status, inferred 8-13
total power output 7-72–7-73
plant bus data
array requirement 8-12
returned 8-13, 8-15, 8-17, 8-19
plant model 4-20, 4-25, 4-44, 4-91, 4-118, 4-128–4-131, 4-276, 4-282–4-284
array index 7-122
character ICON (CHRICN) changed 4-120, 4-278
connection table 4-129, 4-282
constant, real (CON) changed 4-121, 4-279
integer ICON changed 4-124, 4-280
machine status changed 4-131, 4-284
machine type 4-118, 4-130, 4-276, 4-283
name returned 7-126
status 7-122
type removed from machine 4-130, 4-283
unconnected 4-44, 4-132, 4-285
user-written 4-133, 4-286
variable, algebraic (VAR) changed 4-125, 4-281
variable, dynamic output channel 4-91
plot constraints 1-394
plot, PV/QV analysis
area parameters 10-9
boundary value 10-2
line style 10-4–10-5
scale 10-4–10-5, 10-7
plotting book 11-8
active page 11-11–11-12
delete plot 11-4
display editor 11-7
export plot 11-5
insert page 11-2
insert plot 11-3
load Channel Output File 11-9–11-10
print plot 11-6
polar coordinates 1-590, 9-15, 9-19
pole converter 2-56, 8-197
POLY 1-393, 10-2
POLY Results File (*.pol) 1-395, 10-2
polynomial and exponential cost curve 3-76
polynomial and exponential cost table 3-91
data purged 3-138
OPF report 3-13
positive load, transfer methods 1-211, 1-214, 1-453, 1-458, 1-468, 1-473, 1-478, 1-484
positive pole converter 2-56, 8-197
positive pole voltage controlling converter
bus 8-185, 8-191
data returned 2-60
positive sequence
current 9-15–9-16
fault admittance 4-368
fault analysis 2-75, 2-79–2-80
fault current 7-158
impedance

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

generator 1-140, 1-159
machine 7-113–7-114, 8-26, 8-29
pi-equivalent
branch unbalance 1-598
branch, dynamic simulation 4-370, 4-372
short circuit current 7-156–7-157, 7-162
shunt current 7-158
Thevenin impedance 9-16, 9-20
three phase fault current 9-15, 9-19
voltage 7-158
positive sequence network 1-554
branch 1-89, 1-92
line charging 9-18
load option 1-175, 1-177–1-178, 9-18
machine 1-89, 1-92
shunt option 1-175, 1-177–1-178, 9-18
post corrective actions
load shed 9-6
reliability assessment 1-524, 1-527
Slider Diagram 6-59, 6-61
post tripping
load shed 9-6
reliability assessment 1-524, 1-527
report 1-22, 1-25, 1-28
Slider Diagram 6-59, 6-61
post-contingency
load shed 9-6
reliability assessment 1-524, 1-527
Slider Diagram 6-59, 6-61
POUT 1-225, 1-397
power demand 7-34, 8-162
power dispatch, reactive 3-114, 3-154–3-155
power display setting 1-398
power exchange, active bridge 8-234, 8-247
power factor 1-550, 1-553, 3-93, 3-95, 4-36
power flow 1-168
annotation, Slider Diagram 6-55, 6-57
branch from bus or to bus 8-107
case removed from working memory 1-131
count per contingency 1-31, 1-35, 1-65, 1-255, 1-261, 1-292
data
added to working case 1-517, 1-519
returned 2-136, 7-16
display, Slider Diagram 6-74
from bus 8-96, 8-110
from bus end, 2-winding transformer 8-121, 8-124
from bus or to bus 8-107, 8-121
run from active Event Study 13-6
sending bus 8-234
series element
from bus end, FACTS device 8-247
sending end 7-61
terminal end 7-61
shunt element 8-234, 8-247
solution parameters changed 2-100, 2-103
subsystem branches 8-93

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

terminal bus 8-234
voltage, Slider Diagram 6-48, 6-52
winding bus, 3-winding transformer 8-150, 8-153
power flow activity selector 7-163
Power Flow Data File, Options 1-509, 1-511
Power Flow Raw Data File (*.raw) . . 1-509–1-510, 1-512, 1-515–1-517, 1-519, 1-554, 1-639–1-640
power losses 8-96, 8-110, 8-124, 8-137, 8-153, 8-236, 8-250
case comparison 1-136
power output 8-29
in-service machines 8-17
power output option setting
2-terminal dc line 7-34, 8-164, 8-166
2-terminal dc line converter 8-176, 8-179
2-winding transformer 8-122, 8-124
3-winding transformer 7-206, 8-134, 8-137
3-winding transformer winding 7-204, 8-150, 8-153
area 8-259, 8-261
area data 7-4
branch 8-107, 8-110
branch flow 8-94, 8-96
branch flow data 7-16
bus 7-22, 7-24, 8-6, 8-8
FACTS device 7-60–7-61, 8-234, 8-237
FACTS device bus 8-247, 8-250
fixed shunt 7-67–7-68, 8-69, 8-71
fixed shunt bus 8-60, 8-62
load 7-106, 7-108, 8-50, 8-52
load bus 8-38, 8-41
machine 7-81, 7-112, 7-114, 8-26, 8-29
multi-terminal dc line 7-46, 8-187, 8-189
multi-terminal dc line converter 8-199, 8-202
owner 8-270, 8-272
plant bus 8-16–8-17
switched shunt 8-81, 8-83
VSC dc line 7-196, 8-211–8-212
VSC dc line converter 8-222, 8-225
zone 8-280, 8-283
power setpoint 8-199
power unbalance threshold 4-57
power weighting factor fraction 2-158
from bus, VSC dc line converter 8-222
precision, Slider Diagram 6-27
pre-shift flow changed 1-602, 1-622
primary base ac voltage 8-199
primary voltage recovery 4-68, 4-77
print plot 11-6
print Slider Diagram 6-20, 6-29
priority level, line loading relief calculation 1-232
probabilistic reliability
assessment, transmission systems 1-524, 1-527
results, Slider Diagram 6-59, 6-61
probability cutoff 1-525, 1-528
program settings, psse.opt file 1-641

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

progress device setting 1-406, 12-12
PROGRESS_OUTPUT 1-406
prompt device setting 1-407, 12-13
PROMPT_OUTPUT 1-407
PRTI 1-408
PSAS 4-49
PSAS Command File (*.psa) 4-49
PSEB 1-421
PSEB Command File (*.pse) 1-421
PSS®E Options File (*.opt) 1-641
PSS®E version 1-533, 1-639–1-640, 12-15
PSSPLT Relay Characteristic Data File 4-53
psspy module 1-iii, 9-1, 11-1, 12-2, 14-3
PURG 1-207–1-208, 1-424–1-426, 1-428–1-439, 1-441–1-445, 1-447–1-449
PurgIndMacs 1-433
PV analysis 9-27
calculation engine 1-450, 1-455, 1-460, 1-465, 1-470, 1-475, 1-481
export to Excel spreadsheet 9-38
report 9-28
summary 9-24, 9-26, 9-38
PV Results File (*.pv) . 1-453, 1-458, 1-464, 1-469, 1-474, 1-480, 1-486, 9-24, 9-26–9-28, 9-38, 10-3
Python format recording 12-20
Python functions 9-41, 9-44, 9-50
Python modules 1-iii
Python scripts 12-2
Python syntax 1-i

Q

quadratic cost coefficient 3-91
quadratic cost curve 3-76
quadratic cost table 3-92
data purged 3-139
OPF report 3-13
quit Excel application 9-44
quit PSS®E application 1-423
QV analysis 9-29
calculation engine 1-488, 1-490, 1-492, 1-495
export to Excel spreadsheet 9-40
report 9-30
summary 9-31, 9-33, 9-40
QV Results File (*.qv) 1-489, 1-491, 1-494, 1-497, 9-29–9-31, 9-33, 9-40, 10-5

R

radial bus equivalence 1-166
range checking annotation, Slider Diagram 6-39, 6-42–6-43, 6-48, 6-52
RANK 1-499, 1-501
RAT3 1-504
RATE 1-504
rating conversion to MW 1-239, 1-254, 1-260, 1-273, 1-279, 1-291, 1-499, 1-501
rating set
ac contingency analysis with corrective actions 1-43

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

ac corrective actions 1-76, 1-79, 1-83
base case
AC Contingency Multiple Run Report 1-4, 1-8
AC Contingency Report 1-21, 1-24, 1-27
interchange limits 1-393
transmission interchange limits 1-600, 1-620
branch 8-93, 8-107
branch flow 3-83
branch loading 1-504
branch resistance 1-323
contingency
multi-level ac contingency analysis 1-239, 1-254, 1-260, 1-273, 1-279, 1-291
contingency case
AC Contingency Multiple Run Report 1-4, 1-8
AC Contingency Report 1-21, 1-24, 1-27–1-28
dc contingency checking 1-146
interchange limits 1-393
transmission interchange limits 1-600, 1-620
dc corrective actions 1-148
dc network solution 1-151
default 1-506
line 2-6, 2-150, 2-154
Midwest megawatt-mile calculation 1-339
monitored elements 9-4
percentage loading 9-11–9-12
probabilistic reliability assessment 1-525, 1-527
PV analysis 1-451, 1-456, 1-466, 1-471, 1-476, 1-482, 9-25
single branch outage contingency 1-499, 1-501
substation reliability analysis 1-606
three-winding transformer 8-149
two-winding transformer 8-121
winding 2-130
ratio limits, transformer winding 8-149
ratio table, transformer impedance correction 2-30
RAWD 1-509
RDCH 1-509, 1-511–1-512
RDEQ 1-513
reactance 2-121, 2-124
commutating transformer 8-199
dummy series element 2-24, 7-60, 8-233
limits 1-117, 1-119
line-to-ground fault impedance 1-559, 1-561–1-562, 4-368
line-to-line fault impedance 1-560–1-562, 4-368
load drop compensating 2-150, 2-154
magnitude 8-163
phase impedance 1-560
synchronous 3-93, 3-95
reactance multiplier 3-64, 3-66
reactance threshold 1-240, 1-256, 1-262, 1-275, 1-281, 1-292, 1-500, 1-502
reactive absorption 3-93, 3-95–3-96
reactive capability 3-93, 3-95, 3-140–3-141
reactive generation limits 3-95
scaled 3-43
reactive power
animation arrow 6-29
into converter 2-136

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

limits
machine . . . 1-181
VSC dc converter . . . 2-158
VSC dc line converter . . . 8-222
wind machine . . . 2-48, 8-23, 8-26
mismatch convergence tolerance . . . 2-101, 2-104
slack generation . . . 3-21
reactive power dispatch . . . 3-114, 3-155
reactive power flow
data . . . 7-196
shunt element . . . 8-234, 8-247
FACTS device . . . 7-61
reactive power generation . . . 3-58
reactive power load . . . 4-112
output . . . 4-80
scaling . . . 1-549, 1-552
reactive power losses . . . 3-20, 8-164, 8-234, 8-247
cost coefficient . . . 3-54
reactive power output, machine . . . 4-80, 4-91
reactive reserve . . . 3-22
cost coefficient . . . 3-55
reactor
Mvar scaling . . . 1-549, 1-553
Mvar total returned . . . 1-548, 1-551
negative value . . . 7-23
power flow data . . . 7-23
READ . . . 1-509, 1-515–1-517, 1-519, 7-185
read into working case
long case title . . . 1-534
Machine Impedance Data File . . . 1-322
Power Flow Change Raw Data File . . . 1-512
power flow changes . . . 1-511
Power Flow Raw Data File . . . 1-515–1-516
Sequence Data File . . . 1-532–1-533
Transactions Raw Data File . . . 1-530
RECN . . . 1-521
reconnect bus . . . 1-521
reconnect to bus . . . 1-325–1-326
reconstruct load . . . 1-141
recording
enable/disable . . . 1-164
options . . . 1-522
start . . . 12-20
stop . . . 12-21
recording format . . . 1-ii
rectangular coordinates . . . 1-590, 9-15, 9-19
rectifier . . . 2-135, 8-197
angle limit . . . 8-160
bus . . . 8-160, 8-168
commutating capacitor reactance magnitude . . . 8-163
commutating transformer resistance/reactance . . . 8-163
converter angle . . . 8-163
dc voltage . . . 8-163
firing angle measuring bus . . . 8-160, 8-169

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

tap ratio limit . . . 8-161
tap setting . . . 8-163
transformer ratio . . . 8-163
refresh GUI . . . 12-16
REGB . . . 1-523
regulated bus . . . 3-58, 8-13, 8-19, 8-85
base voltage . . . 8-15
number . . . 7-178
voltage magnitude . . . 8-15
regulated voltage setpoint . . . 7-112
relative angle calculation . . . 4-63
Relay Output Data File (*.rel) . . . 1-102, 1-104, 1-106, 1-110
RELAY2 monitoring model . . . 4-95
RELAY3 monitoring model . . . 4-103
reliability analysis display, Slider Diagram . . . 6-75
reliability annotation, Slider Diagram . . . 6-59, 6-61
reliability assessment . . . 1-524, 1-527
parameters . . . 1-525, 1-528
REMM . . . 1-530
remote bus . . . 2-23, 2-67, 2-110, 2-158
number . . . 7-64
remote regulated bus number . . . 7-82, 7-117
remote voltage-controlled bus . . . 8-227
remove dc bus . . . 1-447
renumber buses . . . 1-120–1-121
auxiliary files . . . 1-536
Slider Diagram . . . 6-23
report
ac contingency analysis lists . . . 9-11
summary . . . 9-7–9-8
branch . . . 1-339–1-340
branch flow overload . . . 1-524, 1-527
bus load curtailment . . . 1-606
bus loss of load . . . 1-524, 1-527
bus voltage violation . . . 1-524, 1-527
circuit breaker interrupting duty . . . 1-115
contingency . . . 1-606
contingency case . . . 1-5, 1-10
contingency loading violation . . . 1-6, 1-10
contingency summary . . . 1-524, 1-527
corrective actions . . . 1-21
load model connection . . . 4-43
loading violation . . . 1-5–1-6, 1-9–1-10
machine loading/limit . . . 1-181
master legend . . . 1-5, 1-9
missing contingencies . . . 1-5, 1-9
missing monitored elements . . . 1-4, 1-9
missing monitored voltage buses . . . 1-5, 1-9
monitored elements . . . 1-4, 1-8
network with unbalances . . . 1-563, 1-583
non-converged contingencies . . . 1-5, 1-9
non-converged networks . . . 1-21, 1-24, 1-27
OPF working case . . . 3-13
out-of-limit bus voltage . . . 1-634

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

overload 1-21–1-22, 1-24–1-25, 1-27–1-28, 1-146–1-147
overloaded element 1-80, 1-84, 1-147, 1-395, 1-601, 1-622
PV analysis 9-28
summary 9-26
QV analysis 9-30
summary 9-33
substation load curtailment 1-606
system loss of load 1-524, 1-527
system problem summary 1-524, 1-527
vector absolute MW-mile 1-632
vector absolute MW-ohm 1-633
vector MW-ohm 1-637
voltage range violation 1-6, 1-10, 1-22–1-23, 1-25, 1-28
voltage violation 1-5, 1-9
report device
location list 1-138
name returned 1-226, 1-228
setting 1-531, 12-17
report separator, GUI 12-11
report tab, GUI 12-4
reporting mode, dynamics simulation 4-20
represent dc line as load 1-102, 1-104, 1-106, 1-197, 1-200, 9-18
represent FACTS device as load 1-102, 1-104, 1-106, 1-197, 1-200, 9-18
reserve limit 3-123
reserved ICON index, load model 7-102
resistance 2-121, 2-124
commutating transformer 8-163, 8-199
compounding 2-141, 8-162
dc line 2-141, 2-160, 8-162, 8-210
fault impedance 1-559–1-560, 1-562, 4-368
load drop compensating 2-150, 2-154
to ground 2-54
resistance/reactance threshold, branch check 1-117, 1-119
Response File (*.idv) 1-164, 1-421, 1-563, 1-583, 1-588, 4-49, 4-85
RESQ 1-371, 1-532, 1-593
RETI 1-534
return code 1-ii
RNFI 1-536
ROPF 3-56
rotor current/voltage output 4-80, 4-91
RSTR 4-51
RUN 4-52, 4-75, 7-74
run IPLAN program 12-18
run PSS®E from file 12-19
RWCM 1-538
RWDY 4-53
RWMA 1-539
RWMM 1-541
RWOP 3-57
RWSQ 1-542, 1-544

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

S

SAVE 1-546
Saved Case File (*.sav) 1-546
case comparison 1-136, 1-158, 1-160, 6-72
tie lines 1-156–1-157
export IEC data to Excel 9-37
filename 9-3, 9-22
returned 7-164
filenames in Contingency Solution Output File 1-4, 1-8
pre-contingency working case 1-94
PV analysis summary 9-24
QV analysis summary 9-31
read into working case 1-129
sequence data present 1-157, 1-160
summary 1-594
SCAL 1-548, 1-551
scaling
automatic, OPF solution 3-4
bus data 1-548, 1-551
CRT output channel 4-83
percent change 1-550, 1-553
reactive generation limits 3-43
scanning generators
angle deviation 4-54–4-55
power unbalance 4-57
Scenario file (*.pssxml) 13-8–13-11
SCEQ 1-554
scheduled compounded dc voltage 8-162
scheduled current 2-57, 2-141
scheduled dc voltage 2-141, 7-34
scheduled power/voltage 2-57
scheduled voltage magnitude 2-67
SCMU 1-557, 1-563, 1-583, 1-587, 7-156–7-158, 7-161–7-162
SCOP 1-563, 1-583
search pathname 4-2, 4-70
secondary voltage recovery 4-68, 4-77
sending bus 8-232
number 7-64
voltage setpoint 2-24, 8-233
sending end bus 2-23, 8-238
FACTS device end code 8-245
sending end shunt current 2-24
sensitivity values, OPF Optimization Log File 3-2
SEQD 1-557, 1-585–1-587
sequence admittance matrix 1-557, 4-368
sequence data 1-608
fault analysis 1-371
list 1-608
Saved Case File 1-157, 1-160
working case 1-157, 1-160
Sequence Data File (*.seq) 1-532–1-533, 1-536, 1-542, 1-544, 1-554
sequence matrix, factorized 1-585–1-586

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

series compensation 3-23
series current 2-24, 7-60, 8-234
series element, FACTS device 8-230–8-231, 8-233, 8-236, 8-238, 8-242, 8-244, 8-246, 8-249, 8-251
series voltage 7-60
angle data returned 2-24
limit code 7-64, 8-232, 8-245
magnitude 2-24, 7-61, 8-233, 8-247
data returned 2-24
real/imaginary component returned 2-24
reference code 2-23, 8-232, 8-245
series Z real/imaginary component returned 2-24
serving load 1-340
set terminal input device to file 1-588
severity
single branch outage contingency 1-499
single element outage contingencY 1-501
SHNT 1-589
short circuit current 9-17
3-winding transformer 7-156
branch 7-157
machine 7-162
short circuit currents 9-13
short circuit data returned 7-161
short circuit display 1-563, 1-583
annotation, Slider Diagram 6-35–6-37
results, Slider Diagram 6-76
Short Circuit Output file (*.sc)
ASCC results 1-106, 1-110
IEC fault analysis 1-198, 1-201, 1-204
IEC fault calculation data 9-18
short circuit phase modeling 1-591
SHOW 1-594
shunt 7-22
active 8-83
bus count 8-58, 8-60, 8-62–8-63
count 8-67, 8-69, 8-71, 8-73
megawatt scaling 1-549, 1-553
power flow 8-236, 8-250
status 2-95, 8-67
inferred 8-57
treatment options 1-175, 1-177–1-178
shunt current 7-60
sending bus 8-234
shunt MVA 7-22
shunt MW total returned 1-548, 1-551
simulation mode
shift to dynamics mode 4-31
shift to power flow mode 4-48
simulation time 7-54
simulation type 7-74
simulation variable returned 7-29
simultaneous unbalances 1-557
single branch outage 1-238, 1-253, 1-260, 1-272, 1-278, 1-290

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

contingencies 1-499
single element outage contingencies 1-501
single machine outage 1-238, 1-253, 1-260, 1-272, 1-278, 1-290
singular Jacobean matrix 7-168
sink subsystem 1-606
sink system 1-211, 1-214, 1-394, 1-452, 1-457, 1-467, 1-472, 1-477, 1-483, 1-600, 1-621
SIZE 1-595
Slider Binary File (*.sld) 6-19, 6-24
Slider Diagram
bus display 6-9
Bus Location Data File 6-77
export 6-7
open 6-18
case comparison voltage 6-43
close without save prompt 6-4
comparison case display 6-72
contour
parameter change 6-2
parameter definition 6-6
refresh 6-21
removal 6-5
default text font 6-26
Draw Data File 6-14
dynamic simulation annotation options 6-45
dynamics data display 6-71
fault calculation 6-69
Graphical Report Data File 6-13
grow bus 6-10
connected buses 6-12
IEC annotation 6-46
IEC fault calculation display 6-73
image export 6-8
image import 6-15
impedance 6-39, 6-42, 6-70
network case comparison 6-3
new 6-16
open 6-19
power flow
annotation 6-55, 6-57
results display 6-74
voltage 6-48, 6-52
precision 6-27
printing 6-20
properties 6-28, 6-31, 6-33
range checking annotation 6-39, 6-42–6-43, 6-48, 6-52
reliability analysis display 6-75
reliability annotation 6-59, 6-61
renumber buses 6-23
results refresh 6-22
save 6-24
short circuit display 6-37, 6-76
slow voltage stability analysis
PV 10-3
QV 10-5
SNAP 4-73
soft limit penalty weight 3-79, 3-81, 3-83, 3-86, 3-88, 3-105

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

solution convergence indicator 7-168
solution tolerance test 7-168
solution type, ac contingency analysis 9-5, 9-11–9-12, 9-35
SOLV 1-596
source subsystem 1-606
source subsystem ties, probabilistic reliability assessment 1-525, 1-528
source system 1-210, 1-213, 1-393, 1-452, 1-457, 1-467, 1-472, 1-477, 1-483, 1-600, 1-620
SPCB 1-598
SPIL 1-600
split buses, additional bus for each split bus 1-604
SPLT 1-604
SQEX 1-605
SQLI 1-608
SRRS 4-74
stabilizer model 4-20, 4-25, 4-44, 4-118, 4-130–4-131, 7-122, 7-126
stabilizer signal 4-5
stabilizer signal output 4-80, 4-91
stand-alone simulator 7-71
star bus, 3-winding transformer 7-185
voltage angle 2-121, 2-124, 7-186
voltage magnitude 2-121, 2-124, 7-186
actual 8-133
STATE index 4-18
2-terminal dc line auxiliary signal model 7-30
2-terminal dc line model 7-38
branch relay model 7-153
FACTS device auxiliary signal model 7-55
FACTS device model 7-65
load model 7-102
N-terminal dc line auxiliary signal model 7-40
N-terminal dc line model 7-50
plant model 7-122
switched shunt model 7-179
VSC dc line auxiliary signal model 7-194
VSC dc line model 7-199
wind model 7-201
state variable (STATE) 4-29, 4-113, 7-54
changed 4-12
system matrix 4-4
state-space simulation 4-34, 4-38, 4-52, 4-75, 7-74
stator current limit 3-93, 3-95–3-96
status
branch relay model 7-153
FACTS device auxiliary signal model 7-55
FACTS device model 7-65
load model 7-102
machine 8-23
N-terminal dc line auxiliary signal model 7-40
N-terminal dc line model 7-50
plant model 7-122
shunt 8-67
subsystem load model 7-165
switched shunt model 7-179

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

transformer 8-131
two-terminal dc line auxiliary signal model 7-30
two-terminal dc line model 7-38
VSC dc line auxiliary signal model 7-194
VSC dc line model 7-199
wind models 7-201
winding 8-145
status changed
2-terminal dc line model 4-202
auxiliary signal model, 2-terminal dc line 4-273
auxiliary signal model, FACTS device 4-269
auxiliary signal model, multi-terminal dc line 4-271
auxiliary signal model, VSC dc line 4-275
FACTS device model 4-215
GNE device 2-28
line relay model 4-170
load model 4-156–4-157
multi-terminal dc line model 4-200
plant model 4-131, 4-284
switched shunt model 4-228
VSC dc line model 4-204
wind model 4-143
steady-state condition 4-4
steady-state converter angle 8-199
rectifier/inverter 8-163
step count 8-78
step length, barrier function 3-26
step size 8-81
limit 3-3
step-up transformer
impedance 7-81, 7-114, 8-29
resistance/reactance 2-49
tap ratio 2-34, 2-49, 8-26
STOP 1-610
STRT 4-75, 7-74, 7-143
SUBS 1-611
substation load curtailment report 1-606
substation reliability analysis 1-606
substation subsystem 1-606
subsystem
branch count 8-91, 8-94, 8-96, 8-98
bus mismatch 8-60
data added to working case 1-517, 1-519
equipment 1-509, 1-542, 1-544
fixed shunt 8-66–8-67, 8-69, 8-71
monitoring 4-79
tie branches 8-101
ties 1-509, 1-542, 1-544
subsystem definition, OPF 3-58
Subsystem Description File (*.sub) 1-153, 1-500, 1-502, 9-3, 9-22, 9-24, 9-31
subsystem load
bus count 8-36, 8-38, 8-41, 8-44, 8-290, 8-292, 8-294, 8-296
constant current, in-service 8-38, 8-41
constant MVA, in-service 8-38, 8-41

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

count 8-46, 8-299
by owner 8-266
in-service 8-38, 8-41, 8-292, 8-294
subsystem load model
array index 7-165
status 7-165
Subsystem Participation Data File (*.prt) 1-602
summary
ac contingency analysis 9-3, 9-7–9-8, 9-35
ANSI reactance and impedance 1-89, 1-92
branch mileage by owner 1-336
controlling transformers 1-619
in-service machines 1-187, 1-234
loadings on all area tie lines 1-616
loadings on all zone tie lines 1-617
power flow conditions 1-611
working case components 1-595
susceptance 2-74, 2-95, 2-121, 2-124, 2-150, 2-154, 7-208, 8-81
zero sequence 8-80–8-81
swing 1-257
swing bus 7-28
swingless island 1-240, 1-255, 1-261, 1-274, 1-280, 1-292, 1-499, 1-501
bus count returned 1-629
switched shunt 1-130, 1-229, 1-444, 1-589, 3-58, 3-116, 3-156, 7-21, 7-23, 8-76–8-77, 8-79, 8-83, 8-85
block, step size returned 7-173
block, zero sequence step size 2-84, 8-81
returned 7-174–7-175
case comparison 1-159
control limits 7-177, 8-80
control mode 7-178, 8-78
count 8-78, 8-82–8-83, 8-86
data
array requirement 8-76
changed 2-105, 2-110
deleted 1-444
returned 7-177–7-178, 8-77, 8-79, 8-83, 8-85
sequence data 1-608
zero sequence data changed 2-84
disconnect/reconnect 1-333
number of blocks 7-178
out-of-service 1-424
positive/negative/zero sequence 1-175, 1-177–1-178
settings 3-31
steps per block returned 7-173–7-175
voltage limits returned 7-176
switched shunt adjustment 1-612, 2-101, 2-104
ac contingency analysis 1-3
corrective actions 1-31, 1-35, 1-44, 1-48, 1-76, 1-79, 1-82
generation dispatch 1-57
post-contingency tripping 1-65
tripping simulation 1-31, 1-35
decoupled Newton-Raphson power flow analysis 1-375
fixed slope decoupled Newton-Raphson power flow analysis 1-171
full Newton-Raphson power flow solution 1-179
Gauss-Seidel power flow analysis 1-334, 1-596
inertial/governor response power flow analysis 1-217, 1-219

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

multi-level ac contingency analysis 1-238
corrective actions 1-253, 1-259, 1-272, 1-278
tripping simulation 1-253, 1-259, 1-290
PV analysis 1-451, 1-456, 1-466, 1-468, 1-471, 1-473, 1-476, 1-479, 1-482, 1-485
QV analysis 1-488, 1-490, 1-493, 1-495
switched shunt admittance 2-111, 7-177
switched shunt control 1-36, 1-47, 1-49, 1-79, 1-83, 1-280
subsystem 1-36, 1-49, 1-80, 1-84, 1-262, 1-281
weighting factor 1-36, 1-49, 1-80, 1-84, 1-262, 1-281
switched shunt model 4-20, 4-25, 4-76, 4-220, 4-226–4-228
array allocation table 4-226
array index 7-179
character ICON (CHRICN) changed 4-221
connection table 4-226
constant, real (CON) changed 4-222
GMB 4-225
integer ICON changed 4-223
list 4-76
name returned 7-180
removed from switched shunt 4-227
status 7-179
switched shunt 4-227
switched shunt device status changed 4-228
unconnected 4-229
user-written 4-230
variable, algebraic (VAR) changed 4-224
switching study
acceleration factor 2-101, 2-103
network solution 1-630
switching time 1-606
symmetrical breaking current 9-19–9-20
system base frequency 1-370
system base MVA 1-370
data returned 7-182
system delta interchange 1-395
system generation shift 1-395
system loss of load indices report 1-524, 1-527
system loss of load report 1-524, 1-527
system mismatch 9-38, 9-40
converged tolerance 1-6, 1-10, 1-23, 1-26, 1-29, 1-403
tolerance 9-5, 9-11–9-12, 9-35
system MVA mismatch 9-27, 9-29
data returned 7-181
system output vector 4-4
system problem indices report 1-524, 1-527
system problem summary report 1-524, 1-527
system totals 1-135
data returned 7-183
system transfer dispatch
sink system 1-452, 1-457, 1-467, 1-472, 1-477, 1-483
source system 1-452, 1-457, 1-467, 1-472, 1-477, 1-483

T

tap adjustment 2-101, 2-104

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

ac contingency analysis 1-2
corrective actions 1-30, 1-34, 1-43, 1-47, 1-75, 1-78, 1-82
generation dispatch 1-56
post-contingency tripping 1-64
tripping simulation 1-30, 1-34
adjustable transformers 1-615
contingency case 1-468, 1-473, 1-478, 1-484, 1-493, 1-496
decoupled Newton-Raphson power flow analysis 1-375
default 1-613
fixed slope decoupled Newton-Raphson power flow analysis 1-171
full Newton-Raphson power flow solution 1-179
Gauss-Seidel power flow analysis 1-334, 1-596
inertial/governor response power flow analysis 1-216, 1-218
multi-level ac contingency analysis 1-237
corrective actions 1-252, 1-258, 1-271, 1-277
tripping simulation 1-252, 1-258, 1-289
PV analysis 1-450, 1-455, 1-465, 1-470, 1-475, 1-481
QV analysis 1-488, 1-490, 1-492, 1-495
subsystem 1-49, 1-80, 1-84
weighting factor 1-49, 1-80, 1-84
tap movement deceleration factor 2-101, 2-104
tap non-transformer ac branch 1-236
tap position count 2-129, 2-149, 2-153, 7-207, 7-210, 8-117, 8-145
tap ratio 1-175, 1-177, 1-203, 3-58, 9-18
branch check 1-116, 1-118
limits 1-117, 1-119
step 3-38–3-39
step-up transformer 8-26
threshold 1-160
winding 7-204, 8-121, 8-149
tap setting 2-136, 7-35, 7-47, 8-199
converter bus 8-199
dc line converter 7-46
from bus, 2-terminal dc line converter 8-176
rectifier/inverter 8-163
tap setting adjustment 1-36, 1-49, 1-79, 1-83, 1-261, 1-280
subsystem 1-36, 1-262, 1-281
weighting factor 1-36, 1-262, 1-281
tap status
dc line 7-43
inverter 7-37
tap step 1-613, 1-624–1-625, 2-136, 7-35, 7-47, 8-199
tapped side bus number 7-210
terminal bus 8-232, 8-238
number 7-64
power 2-24
reactive power 2-24
voltage magnitude 2-24, 8-233
terminal end bus 2-23
FACTS device end code 8-245
terminal end voltage condition 7-64
code 8-232, 8-245
TEXT 1-614
text description, contingency case solution 1-94
text font, Slider Diagram 6-26

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

text passed into progress stream 1-614
text report of lists, ac contingency analysis 9-7–9-8
summary 9-11
TFLG 1-615
Thevenin impedance 1-598, 9-13, 9-16–9-17, 9-20
three phase fault 1-101, 1-103, 1-106, 1-109, 1-196, 1-199, 1-202, 1-558, 1-561, 1-598, 4-370, 4-372, ... 9-15, 9-17, 9-19
analysis 1-591
Slider Diagram 6-37, 6-46
three-winding transformer 1-229, 8-142, 8-144, 8-147, 8-152, 8-154
branch 7-131
count 8-132, 8-134, 8-137, 8-139
bus fault 1-561
bus status 1-152
control 1-229
data
array requirement 8-130
deleted 1-426
flow constraint data purged 3-132, 3-148
impedance data changed 2-118, 2-120, 2-123
returned 7-141, 7-186–7-187, 7-189, 8-131, 8-133, 8-135, 8-138
sequence data 1-608
zero sequence grounding data changed 2-68
zero sequence impedance data changed 2-69
disconnect/reconnect to bus 1-325
flow constraint 3-78
from bus
active power flow 4-109
active/reactive power flow 4-107
apparent impedances 4-103
MVA flow 4-105
IBUS fault 4-357
identifier
changed 1-308
returned 1-222
impedance 1-229
in-service 4-356
name returned 7-191
out-of-service 1-424, 4-359
short circuit currents returned 7-156
star bus 7-87, 7-96
status 7-189
winding connection angle 2-130
three-winding transformer winding 1-229, 3-103
active/reactive power flow 8-150
angle step width 8-121, 8-150
base MVA 2-121, 2-124
branch count 8-145, 8-150, 8-153, 8-155
bus impedance 7-187
bus name 8-138, 8-155
bus number 2-120, 2-123, 8-131, 8-145
connection code 7-189, 7-207
current 7-204
data
array requirement 8-142
changed 2-129
deleted 3-148

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

resistance/reactance data returned 2-130
returned 7-204, 7-206–7-207, 8-144, 8-147, 8-152, 8-154
derived tap step 7-204, 7-208
flow limits 2-130
I/O code 2-120, 2-123, 7-189, 8-132, 8-145
impedance 7-206
load drop compensating resistance/reactance 2-130
MVA base 7-204
MVA flow 7-204
number 7-207, 8-145
phase shift angle 2-130, 7-204, 8-149
power flow 8-153
rating set 7-204
ratio 2-130
ratio step width 8-150
ratio/angle limits 2-130
reporting to buses 1-405
status 7-207, 8-145
tap ratio 7-204, 8-149
to bus display setting 1-405
voltage 2-130, 7-204
zero sequence impedance 7-206
tie branch
data returned 7-97–7-98
identifier returned 7-140–7-141
next 7-140
tie flows
interchange area 1-220
interchange zone 1-221
tie lines, out-of-service 1-207–1-208, 1-424
TIES 1-616
ties added to working case 1-517, 1-519
TIEZ 1-617
TIME 1-618
time derivative (DSTATE) changed 4-12
time elapsed, period reserve constraint 3-123
time step 7-54
count 4-46, 4-52
dynamic simulation 4-29
threshold, extended term dynamic simulation 4-29
timer initialization 1-618
timing statistics 1-618
TLST 1-619
TLTG 1-620
tooltips, Slider Diagram 6-29
total cost 3-90
total fault current 1-196, 1-199, 1-202
total generation
active/reactive power 1-539, 8-258
area 7-2
bus owner 7-146
machine owner 7-146
system 7-183
zone bus assignment 7-213

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

total load 1-159, 7-106
bus area assignment 7-2
bus owner 7-146
load area assignment 7-2
load owner 7-146
Mvar 1-550, 1-553
system 7-183
zone bus assignment 7-213
zone load assignment 7-213
total losses
area 7-2
branch owner 7-146
dc line ac power 8-201
system 7-183
zone 7-213
total MVA base 1-539, 7-79, 7-112, 8-15, 8-26
case comparison 1-159
total power output data returned 7-72–7-73
total system MVA mismatch data returned 7-181
TPCH 1-624
transaction
adjustment 1-233
line loading relief 1-233
schedule 1-232
size, megawatts 1-340, 9-27
transaction event 1-341–1-342, 1-391, 1-626, 2-65, 2-132
changed 2-132
curtailment 2-132
data
changed 1-530
deleted 1-626
impact, MW flow 1-209
magnitude 2-132
participating bus
changed 2-65
priority 2-132
status 2-132
Transactions Raw Data File (*.mwm) 1-340, 1-530, 1-541
transfer dispatch 1-210, 1-213
transfer increment 1-211, 1-214, 1-451, 1-456
tolerance 1-453, 1-458, 1-469, 1-474, 1-479, 1-486
transformer
adjustment
check 1-624
code 7-207, 7-210
control mode 1-615
requirement returned 1-625
bar chart, Slider Diagram 6-30
case comparison 1-157, 1-159
circuit identifier returned 1-138
connection code 2-68, 2-88, 2-90, 2-92, 7-210, 8-117, 8-132, 8-145
case comparison 1-157, 1-159
equivalencing suppressed 1-166, 1-513
from bus returned 1-138
limits 7-208
percent units 1-627

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

status 8-131
to bus returned 1-138
transformer check 1-624
transformer control 3-117, 3-157
automatic adjustments 8-117, 8-145
limits 8-121, 8-150
transformer impedance 8-123, 8-152
adjustment table 7-207, 8-117, 8-145
correction 1-88, 1-91, 1-102, 1-104, 1-106, 1-109, 1-197, 1-200–1-201, 1-204, 1-554, 1-557, 1-585–1-586, ... 4-368, 4-371, 4-373, 9-18
zero sequence network 1-166, 1-513
correction table 2-30, 7-210
changed 2-30
number of entries returned 2-30
transformer magnetizing admittance 7-208
transformer phase shift angle 3-30
transformer rating 1-370
transformer ratio 2-57, 2-136, 7-35, 7-47, 8-199
transformer resistance/reactance 2-121, 2-124, 2-150, 2-154
data returned 2-150, 2-154
transformer tap ratio settings 3-32, 3-42
transformer winding
angle limits 7-204, 7-207, 7-210, 8-117, 8-121, 8-145
MVA base 8-121, 8-150
ratio limits 7-204, 7-207, 7-210, 8-117, 8-121, 8-145
transformer, load drop compensating 1-138
transmission interchange limits 1-600, 10-2
calculation 1-620
transmission line
branch check 1-504
units 1-628
transmission system probabilistic reliability assessment 1-524, 1-527
TREE 1-629, 9-22
trip in-service branch 7-100
trip out-of-step circuit 4-62
Tripping Element Data File (*.trp) 1-32, 1-37, 1-66, 1-256, 1-263, 1-292, 9-4
tripping event count, power flow 1-31, 1-35, 1-65, 1-255, 1-261, 1-292
tripping load shed count 9-3
tripping simulation 1-31, 1-35, 1-65, 1-292
options 1-255, 1-261
turbine governor model 4-20, 4-25, 4-38–4-39, 4-118, 4-130–4-131, 7-122, 7-126
reference output 4-80, 4-91
response test 4-38
turbine load controller model 4-20, 4-25, 4-44, 4-118, 4-130–4-131, 7-122, 7-126
reference output 4-80, 4-91
turbine mechanical power 4-5
turbine mechanical power output 4-80, 4-91
two-point type 1 bus equivalence 1-166
two-terminal dc line 1-425, 7-78, 8-159–8-160, 8-162, 8-166, 8-168
auxiliary signal model 4-235, 7-31
branch count 8-161, 8-164, 8-167, 8-169
control mode 8-160

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

data
array requirement 8-159
deleted 1-425
returned 7-34, 7-37, 7-75–7-77, 7-151–7-152, 8-160, 8-162, 8-166, 8-168
firing angle status 7-37
inverter end 7-37
metered end 7-37
name changed 1-307
next 7-129
OPF report 3-13
out-of-service 1-424
rectifier end 7-37
tap status 7-37
two-terminal dc line auxiliary signal model
array index 7-30
status 7-30
two-terminal dc line converter 8-172–8-173, 8-175, 8-178, 8-180
bus count 8-174, 8-176, 8-179, 8-181
control mode 8-174
data
array requirement 8-172
returned 8-173, 8-175, 8-178, 8-180
two-terminal dc line dc converter, data changed 2-135
two-terminal dc line dc link, data changed 2-138, 2-140
two-terminal dc line model 4-16, 4-20, 4-25, 4-175, 4-201–4-202, 4-287, 4-289, 4-291, 4-293, 4-295, ... 4-297, 4-299, 4-301, 4-339–4-355
array index 7-38
character ICON (CHRICN) changed 4-184
connection table 4-176
constant, real (CON) changed 4-185
integer ICON changed 4-188
name returned 7-39
removed from 2-terminal dc line 4-201
status 7-38
two-terminal dc line 4-201
two-terminal end line status changed 4-202
variable, algebraic (VAR) changed 4-189
two-winding ac transformer 2-135
two-winding transformer 1-429, 8-115–8-116, 8-119, 8-123, 8-126
branch count 8-117, 8-122, 8-124, 8-127
changed 2-142, 2-149, 2-153
control 1-229
data
array requirement 8-115
changed 2-142, 2-149, 2-153
deleted 1-429
flow constraint data purged 3-133
returned 7-208, 7-210, 8-116, 8-119, 8-123, 8-126
sequence data 1-608
zero sequence data changed 2-90, 2-92
disconnect/reconnect to bus 1-326
flow constraint 3-80
IBUS fault 4-361
identifier changed 1-310
identifier returned 1-222
impedance 1-229
name returned 7-212

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

owner fraction 8-120
set to in-service 4-360
set to out-of-service 4-363
status 8-116
winding connection angle 2-150, 2-154
two-winding transformer winding 1-229
base MVA 2-150, 2-154
bus name 8-127
bus number 8-117
I/O code 2-149, 2-153, 7-210, 8-117
MVA base 7-208
phase shift angle 2-150, 2-154, 7-208, 8-121
ratio step width 7-208, 8-121
ratio/angle limits 2-150, 2-154
ratio/voltage 2-150, 2-154
resistance/reactance to ground 2-88–2-90, 2-92
tap ratio 8-121
voltage 2-150, 2-154, 7-208
TYSL 1-630, 2-96, 2-98, 2-100, 2-103
TYSL acceleration factor 2-101, 2-103

U

UCTE Data file (*.uct) 1-378, 1-547
unbalance code 4-370, 4-372
unbalance parameters 1-599, 4-371, 4-373
unbalance type 1-558, 1-598, 4-368
unbalanced fault calculation 1-585–1-586, 4-368
unconnected line relay model removed from branch 4-171
unconnected models added/removed
auxiliary signal model 4-240
dc line model 4-197
FACTS device model 4-216
load model 4-158
switched shunt model 4-229
wind model 4-144
unconnected plant model removed from working case 4-132, 4-285
unconverted load returned 1-142
unit commitment 1-162
Unit Inertia and Governor Data File (*.inl) 1-32, 1-37, 1-45, 1-49, 1-57, 1-65, 1-184, 1-217, 1-219, 1-240, ... 1-256, 1-263, 1-275, 1-281, 1-292, 1-536, 4-53, 9-4
unity voltage load reconstruction 1-141
unused entry deleted from connection/array allocation tables
auxiliary signal model 4-239
dc line model 4-196
FACTS device model 4-213
line relay model 4-168
load model 4-154
plant model 4-129, 4-282
switched shunt model 4-226
wind model 4-141
unused user-written model added/removed from definition table
auxiliary signal model 4-241
dc line model 4-198
FACTS device model 4-217

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

line relay model 4-172
load model 4-159
plant model 4-133, 4-286
switched shunt model 4-230
wind model 4-145
USER 1-631
user specified case 1-238, 1-253, 1-260, 1-272, 1-278, 1-290
user-written activity execution 1-631

V

VAR index 4-18, 4-61, 4-79, 4-85
2-terminal dc line auxiliary signal model 7-30
2-terminal dc line model 7-38
branch relay model 7-153
FACTS device auxiliary signal model 7-55
FACTS device model 7-65
load model 7-102
N-terminal dc line auxiliary signal model 7-40
N-terminal dc line model 7-50
plant model 7-122
switched shunt model 7-179
VSC dc line auxiliary signal model 7-194
VSC dc line model 7-199
wind model 7-201
var limited plants 1-186
var limits 1-172, 1-180–1-181, 1-334, 1-376, 1-451, 1-456, 1-466, 1-471, 1-476, 1-482, 1-489, 1-491, 1-493, ... 1-496, 1-596
var record 3-73
variable type code 4-5
variable, algebraic (VAR) changed 4-14
2-terminal dc line 4-189
auxiliary signal model, 2-terminal dc line 4-261
auxiliary signal model, FACTS device 4-249
auxiliary signal model, multi-terminal dc line 4-255
auxiliary signal model, VSC dc line 4-267
FACTS device model 4-212
line relay model 4-167
multi-terminal dc line model 4-183
plant model 4-125, 4-281
switched shunt model 4-224
VSC dc line model 4-195
wind model 4-139
VCHK 1-634
VHI power flow 1-489, 1-491, 1-493, 1-496
VOLMAG monitoring model 4-115–4-116
voltage
band 1-137, 1-139, 1-624–1-625, 1-634
bus base voltage 8-6, 8-15, 8-38, 8-59, 8-80, 8-292
case comparison 1-159
deviation violation filter 1-22, 1-26, 1-29
display setting 1-636
error, convergence tolerance 2-101, 2-104
input setting 1-635
level
annotation, Slider Diagram 6-39, 6-43

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

threshold, Slider Diagram 6-48
mode switch dc voltage 8-187
range, Slider Diagram 6-39, 6-43, 6-49
reference step 4-35
regulated bus base voltage 8-15
scanning 4-65
schedule, changed 1-137
units display setting 1-619
vector 1-630
voltage collapse
branch ranking 1-261, 1-279
voltage collapse line contingencies 1-260, 1-279
voltage controlled bus 1-523, 2-101, 2-104
voltage controlling band mode switched shunt 2-101, 2-104
voltage controlling converter 2-56, 8-197
voltage controlling device setpoint 1-139
voltage controlling transformer 1-624–1-625
voltage depression
branch ranking 1-240, 1-255, 1-274
contingencies 1-239, 1-254, 1-273, 1-291
number of contingencies 1-499, 1-501
performance index 1-499, 1-501
voltage dip 4-67, 4-77
voltage limits 2-111, 2-150, 2-154, 3-85, 3-87, 4-65
data returned 7-176
limit check 1-22, 1-25, 1-28
voltage magnitude
bus 8-59
scanning 4-65
voltage monitored bus 9-3, 9-24
count 9-31
voltage phase angle
bus 8-6, 8-15, 8-38, 8-59, 8-80, 8-292
star point bus 8-133
voltage primary recovery 4-68, 4-77
voltage range violation 1-6, 1-10, 1-22, 1-26, 1-29
report 1-6, 1-10, 1-22–1-23, 1-25, 1-28
voltage regulation
compensated voltage output 4-80, 4-91
controlled bus 7-177, 8-233
remote bus 7-113
setpoint 4-5
setpoint output 4-80, 4-91
voltage secondary recovery 4-68, 4-77
voltage setpoint 1-137, 1-139, 1-489, 1-491, 1-494, 1-497, 8-199, 9-29, 9-31
changed 1-137
voltage threshold 1-160, 1-453, 1-458, 1-469, 1-474, 1-479, 1-486, 4-65
voltage violation
dynamic simulation, on Slider Diagram 6-45
elimination 1-75
report 1-5, 1-9
voltage/angle output 4-80
VREF array element 4-41–4-42

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

changed 4-13, 4-15
VSC dc line 1-445, 8-207–8-208, 8-210, 8-212, 8-214
auxiliary signal model 4-237
name returned 7-195
branch count 8-209, 8-211–8-212, 8-214
control mode 7-198, 8-208
count, by owner 8-266
data
array requirement 8-207
deleted 1-445
returned 7-99, 7-196, 7-198, 8-208, 8-210, 8-212, 8-214
name changed 1-321
name returned 7-142
next 7-142
out-of-service 1-424
owner reassignment 1-385, 1-387
power at converter bus 7-196
VSC dc line auxiliary signal model
array index 7-194
status 7-194
VSC dc line converter 8-218–8-219, 8-221, 8-224, 8-226
data
array requirement 8-218
returned 7-197, 8-219, 8-221, 8-224, 8-226
VSC dc line converter bus 7-196
count 8-220, 8-223, 8-225, 8-227
VSC dc line dc converter, data changed 2-158
VSC dc line dc link, data changed 2-157, 2-160
VSC dc line model 4-16, 4-20, 4-25, 4-177, 4-203–4-204
array index 7-199
character ICON (CHRICN) changed 4-190
connection table 4-177
constant, real (CON) changed 4-191
integer ICON changed 4-194
name returned 7-200
removed from VSC dc line 4-203
status 7-199
variable, algebraic (VAR) changed 4-195
VSC dc line 4-203
VSC line status changed 4-204

W

warning display setting 1-593
wide format power flow display setting 1-224, 1-235
wind auxiliary control output 4-80, 4-92
wind machine
power factor 2-33–2-34, 2-49, 7-113
reactive power limits 2-48, 7-117, 8-23, 8-26
wind model type 4-134, 4-142
wind model
character ICON (CHRICN) changed 4-136
constant, real (CON) changed 4-137
integer ICON changed 4-138
variable, algebraic (VAR) changed 4-139
wind models 4-20, 4-25, 4-78, 4-134, 4-140–4-143

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

add/remove unconnected 4-144
array allocation table 4-141
array index 7-201
connection table 4-141
list 4-78
name returned 7-203
status 7-201
type removed from wind machine 4-142
user-written 4-145
wind machine status changed 4-143
wind turbine rotor speed deviation output 4-80, 4-91
wind velocity output 4-80, 4-91
working case 1-156, 1-158, 1-229
IEEE Common Format 1-538
OPF data purged 3-125
power flow change data read 1-511
pre-fault conditions 1-101, 1-103, 1-105, 1-108
restoration 7-160
sequence data present 1-157, 1-160
transferred from Saved Case File 1-129
working case deletion
2-terminal dc line 1-425
2-winding transformer 1-429
3-winding transformer 1-426
all fixed shunts 1-443
all loads from specified bus 1-433, 1-435
bus, with all branches 1-169
FACTS device 1-430
GNE device 1-431
inter-area transfer 1-428
load 1-434
machine 1-432, 1-436
multi-section line 1-438
multi-terminal dc line 1-437
multi-terminal dc line bus 1-447
multi-terminal dc line converter 1-448
multi-terminal dc line dc link 1-449
non-transformer branch 1-429
out-of-service equipment 1-207–1-208, 1-424
plant 1-441
switched shunt 1-444
VSC dc line 1-445
zero sequence mutual coupling 1-439
worst case contingency loading violation report 1-5–1-6, 1-9–1-10

Z

zero impedance line
suppress equivalencing 1-166, 1-513
threshold 2-101, 2-104
zero sequence
branch
impedance 8-110, 8-124, 8-152
resistance/reactance 2-70–2-71, 7-12
bus shunt
admittance 2-73, 2-77
case comparison 1-159

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

current . . . 7-158
charging capacitance . . . 8-107
current . . . 9-15–9-16, 9-19
fault analysis . . . 2-75, 2-79–2-80
fixed shunt in-service total . . . 7-21, 7-23
generator impedance, case comparison . . . 1-159
grounding
impedance . . . 7-187, 7-208, 8-124, 8-136
resistance/reactance . . . 2-68
line shunt
active/reactive . . . 2-70–2-71
admittance . . . 7-12
case comparison . . . 1-157, 1-159
from/to bus . . . 8-110
machine impedance . . . 7-114, 8-29
mutual coupling, deleted . . . 1-439
mutual impedance data, changed . . . 2-82
mutuals, case comparison . . . 1-159
R, X, B . . . 1-157
case comparison . . . 1-159
short circuit current . . . 7-156–7-157, 7-162
shunt . . . 7-67–7-68
step size . . . 8-80
switched shunt block . . . 2-84, 8-81
susceptance . . . 8-81
switched on . . . 8-80
switched shunt . . . 7-21, 7-23
admittance data returned . . . 2-84
Thevenin impedance . . . 9-16, 9-21
total line charging . . . 2-70–2-71
transformer resistance/reactance . . . 2-88–2-90, 2-92
voltage . . . 7-158
winding resistance/reactance . . . 2-69
zero sequence data, branch check . . . 1-116, 1-118
zero sequence network . . . 1-554
branch . . . 1-89, 1-92
machine . . . 1-89, 1-92
ZONE . . . 1-644–1-645
zone . . . 1-130, 1-229
bus/load reassignment . . . 1-646
count . . . 8-277, 8-281, 8-283–8-284
data
array requirement . . . 8-276
changed . . . 2-161
returned . . . 8-277, 8-279, 8-282, 8-284
status data returned . . . 7-218
name returned . . . 7-216
number returned . . . 7-217
power losses . . . 8-283
subsystem
definition . . . 5-16–5-17
initialization . . . 5-18
selection . . . 7-165, 7-167
subtotals . . . 1-98
totals . . . 1-135, 1-644
area subtotals . . . 1-645
data returned . . . 7-213
zone boundary bus

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

equivalencing suppressed 1-166, 1-513
retained, electrical equivalent 1-165, 1-642
zone bus load 8-283
zone generation 8-283
zone interchange 7-213, 8-283
data returned 7-215
zone load 7-213, 8-283
count 8-277
zone model 4-146, 4-155–4-156
zone power total 4-79
zone tie monitoring 4-79
ZONM 1-646
zoom factor, Slider Diagram 6-29

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.

All material contained in this documentation is proprietary to Siemens Industry, Inc., Siemens Power Technologies International.