isCOBOL Evolve: Appendices

Key Topics:

- Performance Tuning
- Library Routines
- Intrinsic Functions
- Internal Objects
- File Status Codes
- Troubleshooting - Compiler Errors - Runtime Errors
- Copybooks

Copyrights

Copyright (c) 2020 Veryant
6390 Greenwich Drive, #225, San Diego, CA 92122, USA

All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution and recompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Veryant and its licensors, if any.

Table of Contents

A. isCOBOL Reserved Words 1

B. Library Routines 6

$WINHELP 11
HELP-CONTENTS 12
HELP-CONTEXT 13
HELP-CONTEXTPOPUP 14
HELP-FINDER 15
HELP-HELPONHELP 15
HELP-QUIT 16
A$CURRENT_USER 16
A$DECRYPT 17
A$ENCRYPT 18
A$GET_DIGEST 19
A$GET_USER 20
A$GETTHREAD 22
A$LIST_LOCKS 22
LISTLOCK-OPEN 25
LISTLOCK-NEXT 27
LISTLOCK-CLOSE 28
A$LIST_USERS 28
LISTUSR-OPEN 29
LISTUSR-NEXT 31
LISTUSR-CLOSE 31
A$USERINFO 32
AUSERINFO-SET 33

AUSERINFO-GET 33
AUSERINFO-CLEAR 33
AS$COPY 34
ASCII2HEX 35
ASCII2OCTAL 35
C$ASYNCPOLL 36
C$ASYNCRUN 37
C$CALLEDBY 38
C$CALLERR 39
C$CHDIR 40
C$CODESET 42
C$COPY 43
C$CREATE_TMP_FILE 45
C$DELETE 47
C$DELTREE 48
C$EASYOPEN 48
C$ENVMAP 49
C$FILEINFO 50
C$FSCOPY 52
C$FSFULLNAME 52
C$FULLNAME 53
C$GETENV 54
C$GETLASTFILENAME 56
C$GETLASTFILEOP 57
C$GETPID 58
C$GUICFG 58
C$JUSTIFY 59
C$KEYMAP 60
C$KEYSTROKE 61
C$LIST_DIRECTORY 64
LISTDIR-OPEN 65
LISTDIR-NEXT 66
LISTDIR-CLOSE 68
C$LIST_ENVIRONMENT 68
LISTENV-OPEN 69

LISTENV-NEXT 70
LISTENV-CLOSE 70
C$MAKEDIR 70
C$MYFILE 71
C$NARG 72
C$OPENSAVEBOX 74
OPENSAVE-SUPPORTED 77
OPENSAVE-OPEN-BOX 77
OPENSAVE-SAVE-BOX 79
OPENSAVE-BROWSE-FOLDER 81
OPENSAVE-OPEN-BOX-MULTI 82
OPENSAVE-BROWSE-FOLDER-MULTI 84
OPENSAVE-NEXT 85
OPENSAVE-SAVE-BOX-CHECKED 87
C$PARAMSIZE 89
C$PRELOAD 90
C$PROGINMEM 91
C$REPLACE_ALL 91
C$RERR 93
C$RERRNAME 94
C$RUN 95
C$SETDEVELOPMENTMODE 96
C$SETENV 96
C$SLEEP 96
C$SOCKET 97
CSOCKET-CREATE-SERVER 102
CSOCKET-ACCEPT 102
CSOCKET-CREATE-CLIENT 103
CSOCKET-CLOSE 103
CSOCKET-WRITE 104
CSOCKET-READ 104
CSOCKET-READ-LINE 105
CSOCKET-FLUSH 106
CSOCKET-EMPTY 107
CSOCKET-GETHOSTNAME 107

CSOCKET-REMOTE-NAME 108
CSOCKET-REMOTE-ADDR 108
CSOCKET-GETREMOTEADDRESS 109
CSOCKET-LAST-ERROR 110
CSOCKET-NEXT-READ 112
C$SYSINFO 113
C$SYSTEM 115
C$TOLOWER 116
C$TOUPPER 116
C$TRIM 117
C$UNSET 118
C$VERSION 118
C$WRITELOG 119
CBL_AND 120
CBL_CHANGE_DIR 120
CBL_CHECK_FILE_EXIST 121
CBL_COPY_FILE 123
CBL_CREATE_DIR 125
CBL_DELETE_DIR 125
CBL_DELETE_FILE 126
CBL_DIR_SCAN_END 127
CBL_DIR_SCAN_READ 132
CBL_DIR_SCAN_START 136
CBL_ERROR_PROC 140
CBL_EXIT_PROC 140
CBL_NOT 141
CBL_OR 142
CBL_READ_SCR_ATTRS 143
CBL_RENAME_FILE 144
CBL_XOR 145
CBL_WRITE_SCR_ATTRS 146
EDBI_DISCONNECT 147
EDBI-DISCONNECT-CONNECTION 148
EDBI-DISCONNECT-ALL 149
ESQL$BLOB 149

GET-BLOB-FROM-FILE 152
PUT-BLOB-INTO-FILE 152
FREE-BLOB-HANDLE 152
HEX2ASCII 153
I$IO 153
OPEN-FUNCTION 157
CLOSE-FUNCTION 160
MAKE-FUNCTION 160
INFO-FUNCTION 162
READ-FUNCTION 163
NEXT-FUNCTION 164
PREVIOUS-FUNCTION 165
START-FUNCTION 166
WRITE-FUNCTION 168
REWRITE-FUNCTION 168
DELETE-FUNCTION 169
UNLOCK-FUNCTION 170
REMOVE-FUNCTION 171
FLUSH-FUNCTION 172
START-TRANSACTION-FUNCTION 172
COMMIT-TRANSACTION-FUNCTION 173
ROLLBACK-FUNCTION 173
RECOVER-FUNCTION 173
IN-TRANSACTION-FUNCTION 174
J$GETFROMLAF 174
JGET-LAF-COLOR 179
JGET-LAF-FONT 186
J$NETADDRESS 188
KEISEN 189
KEISEN1 194
KEISEN2 197
KEISEN_SELECT 199
M$ALLOC 199
M$COPY 200
M$FILL 201

M$FREE 201
M$GET 202
M$PUT 203
M$SIZE 204
OCTAL2ASCII 205
P$ 206
P$CLEARDIALOG 211
P$CLEARFONT 212
P$DISBLEDIALOG 212
P$DISPLAYDIALOG 212
P$DRAWBITMAP 212
P$DRAWBOX 213
P$DRAWLINE 214
P$DRAWROUNDBOX 215
P$ENABLEDIALOG 216
P$GETDEVICECAPABILITIES 216
P$GETDIALOG 218
P$GETFONT 222
P$GETTEXTMETRICS 224
P$NEWPAGE 226
P$SETDEFAULTMODE 226
P$SETDEFAULTUNITS 226
P$SETDIALOG 227
P$SETDOCUMENTNAME 231
P$SETFONT 232
P$SETPEN 234
P$SETPOSITION 235
P$SETTEXTCOLOR 236
P$SETTEXTPOSITION 237
P$SETTOPMARGIN 238
P$TEXTOUT 238
R$IO 239
R-OPEN-FUNCTION 244
R-CLOSE-FUNCTION 245
R-MAKE-FUNCTION 246

R-READ-FUNCTION 247
R-NEXT-FUNCTION 247
R-PREVIOUS-FUNCTION 248
R-START-FUNCTION 248
R-WRITE-FUNCTION 249
R-REWRITE-FUNCTION 250
R-DELETE-FUNCTION 251
R-UNLOCK-FUNCTION 251
REG_CLOSE_KEY, DISPLAY_REG_CLOSE_KEY 252
REG_CREATE_KEY, DISPLAY_REG_CREATE_KEY 253
REG_CREATE_KEY_EX, DISPLAY_REG_CREATE_KEY_EX 255
REG_DELETE_KEY, DISPLAY_REG_DELETE_KEY 258
REG_DELETE_VALUE, DISPLAY_REG_DELETE_VALUE 260
REG_ENUM_KEY, DISPLAY_REG_ENUM_KEY 262
REG_ENUM_VALUE, DISPLAY_REG_ENUM_VALUE 263
REG_OPEN_KEY, DISPLAY_REG_OPEN_KEY 266
REG_OPEN_KEY_EX, DISPLAY_REG_OPEN_KEY_EX 268
REG_QUERY_VALUE, DISPLAY_REG_QUERY_VALUE 271
REG_QUERY_VALUE_EX, DISPLAY_REG_QUERY_VALUE_EX 273
REG_SET_VALUE, DISPLAY_REG_SET_VALUE 276
REG_SET_VALUE_EX, DISPLAY_REG_SET_VALUE_EX 278
RENAME 280
S$IO 282
S-OPEN-FUNCTION 284
S-CLOSE-FUNCTION 286
S-MAKE-FUNCTION 286
S-READ-FUNCTION 287
S-WRITE-FUNCTION 288
S-REWRITE-FUNCTION 288
SYSTEM 289
W$BITMAP 289
WBITMAP-DESTROY 292
WBITMAP-DISPLAY 292
WBITMAP-LOAD 293
W$CAPTURE 294

W$CREATEFONT 297
W$FLUSH 298
WFLUSH-REFRESH 299
W$FONT 299
WFONT-CHOOSE-FONT 301
WFONT-DESCRIBE-FONT 304
WFONT-GET-CLOSEST-FONT 306
WFONT-GET-FONT 309
WFONT-SUPPORTED 313
W$IMAGESIZE 314
W$KEYBUF 314
WKBUF-ADD-TO-END 315
WKBUF-ADD-TO-BEGINNING 316
WKBUF-ADD-CLEAR-BUFFER 317
WKBUF-START-RECORDING 318
WKBUF-STOP-RECORDING 319
WKBUF-LOAD-FROM-FILE 319
W$MENU 320
WMENU-ADD 323
WMENU-ADD-BITMAP 324
WMENU-BLOCK 325
WMENU-CHANGE 326
WMENU-CHANGE-BITMAP 327
WMENU-CHECK 328
WMENU-DELETE 329
WMENU-DELETE-BITMAP 329
WMENU-DESTROY 330
WMENU-DESTROY-DELAYED 331
WMENU-DISABLE 331
WMENU-ENABLE 332
WMENU-ENSURE-VISIBLE 332
WMENU-GET-BLOCK 333
WMENU-GET-MENU 334
WMENU-NEW 334
WMENU-NEW-POPUP 334

WMENU-POPUP 335
WMENU-RELEASE 336
WMENU-SET-BLOCK 336
WMENU-SHOW 337
WMENU-UNBLOCK 337
WMENU-UNCHECK 338
W$MOUSE 339
SET-MOUSE-SHAPE 339
GET-MOUSE-STATUS 341
TEST-MOUSE-PRESENCE 341
W$PALETTE 342
WPALETTE-CHOOSE-COLOR 344
WPALETTE-GET-COLOR 345
WPALETTE-NUM-COLORS 346
WPALETTE-SET-COLOR 347
WPALETTE-SUPPORTED 348
WPALETTE-UPDATE 349
W$ROTATE 349
W$SCALE 350
W$TEXTSIZE 352
WD2$CLIENT_INFO 354
WD2-GET-USER-AGENT 355
WD2-GET-BROWSER-NAME 356
WD2$RUN_JS 356
WD2$SESSION 357
WD2-GET-SESSION-VALUE 358
WD2-PUT-SESSION-VALUE 359
WIN$PLAYSOUND 360
WIN$PRINTER 361
WINPRINT-CANCEL-JOB 372
WINPRINT-CLEAR-DATA-COLUMNS 373
WINPRINT-CLEAR-PAGE-COLUMNS 373
WINPRINT-GET-CURRENT-INFO 374
WINPRINT-GET-CURRENT-INFO-EX 375
WINPRINT-GET-NO-PRINTERS 378

WINPRINT-GET-PAGE-LAYOUT 379
WINPRINT-GET-PRINTER-INFO 379
WINPRINT-GET-PRINTER-INFO-EX 381
WINPRINT-GET-PRINTER-MEDIA 384
WINPRINT-GRAPH-BRUSH 386
WINPRINT-GRAPH-DRAW 387
WINPRINT-GRAPH-PEN 389
WINPRINT-PRINT-BITMAP 390
WINPRINT-SET-ATTRIBUTE 393
WINPRINT-SET-CURSOR 395
WINPRINT-SET-CUSTOM-PAPER 397
WINPRINT-SET-DATA-COLUMNS 397
WINPRINT-SET-FONT 398
WINPRINT-SET-LINES-PER-PAGE 399
WINPRINT-SET-MARGINS 400
WINPRINT-SET-PAGE-COLUMN 401
WINPRINT-SET-PRINTER 404
WINPRINT-SET-PRINTER-AS 406
WINPRINT-SET-PRINTER-EX 406
WINPRINT-SET-STD-FONT 413
WINPRINT-SET-TEXT-COLOR 414
WINPRINT-SETUP 415
WINPRINT-SUPPORTED 416
WINPRINT-UPDATE-PRINTERS 416
WIN$VERSION 417

C. Graphical Control List 420

D. Intrinsic Functions 446

E. File Status Codes 450

F. Copybooks 455

iscoblib.def 457
iscobol.def 457
iscontrols.def 462
iscrt.def 470

isfilesys.def 471
isfonts.def 476
isgui.def 478
iskeisen.def 486
isopensave.def 487
ispalette.def 487
isprint.def 488
isresize.def 499
isreg.def 500
issocket.def 502
iswinhelp.def 503
iswinvers.def 503
efdParser.def 504
SQLCA 507

G. Compiler Errors 509

H. Runtime Errors 517

I. Table of ASCII values 521

J. External Links 526

K. Handling Different Character Sets 528

Introduction 528
National Items 528
The isCOBOL Encoding 529
The Java Encoding 529

L. Performance Tuning 530

Compile-time optimizations 530
Latest Compiler 530
Smaller classes 530
Smaller methods 531
Proper data types 531
Arithmetic operations on integers 531
Strings management 531
Relaxed size checking on computional data-items 532

Better run time performance 532
No logging 532
More memory to Java 532
More memory to isCOBOL 532
Class loading optimizations 533
Additional JVM optimizations 533
Client/Server Optimization 534
Database Bridge and SQL 535
Printing 536

M. Troubleshooting 537

Licensing and Configuration issues 537
Compiler Issues 537
Runtime Issues 538
Data Access Issues 541

N. Internal Objects 543

CurrentDate Class (com.iscobol.rts.CurrentDate) 544
efdParser Class (com.iscobol.lib.efdParser) 545
Constructor 545
efdDescr 545
efdCondInfo 546
efdFieldInfo 548
efdFieldAllInfo 550
efdKeyInfo 552
HTTPClient, HTTPData.params and HTTPHandler Classes 553
CobolGUIJavaBean Class (com.iscobol.gui.server.CobolGUIJavaBean) 554
callMethod 554
callStaticMethod 554
callStaticMethodOnServer 555
getProperty 555
setProperty 556
JSONStream Class (com.iscobol.rts.JSONStream) 556
Constructor 556
readFromStream 557
setPrintWriter 557

write 558
SpoolPrinter class (com.iscobol.rts.print.SpoolPrinter) and the Print Preview 558
StoreProcedure Class (com.iscobol.lib.StoreProcedure) 561
call 561
end 562
input 562
inout 562
output 563
XMLStream Class (com.iscobol.rts.XMLStream) 564
Constructor 564
getPrintWriter 565
read 565
readFromFile 566
readFromString 567
readFromStream 567
setOutputProperty 568
write 569
writeToFile 570
writeToPrintWriter 571
writeToStream 572
writeToSringBuffer 572
Usage example 573

Appendix A

isCOBOL Reserved Words

Following, is a complete list of reserved words. Many of these are ANSI reserved words, some are reserved only in isCOBOL.

All of these are treated as reserved words by isCOBOL. You need to avoid using any of these reserved words as user-defined words.

A	B	C	D	E	F	G - H	I - J - K	L
M	N	O	P - Q	R	S	T	U - V	W - X - Y - Z

A

ACCEPT	ACCESS	ADD	ADDRESS	ADVANCING
AFTER	ALL	ALLOCATE	ALLOWING	ALPHABET
ALPHABETIC	ALPHABETIC_LOWER	ALPHABETIC_UPPER	ALPHANUMERIC	ALPHANUMERIC_EDITED
ALSO	ALTER	ALTERNATE	AND	ANY
APPLY	ARE	AREA	AREAS	AS
ASCENDING	ASSERT	ASSIGN	AT	AUTHOR
AUTO	AUTO_MINIMIZE	AUTO_RESIZE	AUTO_SKIP	AUTOMATIC
AUTOTERMINATE				

B

BACKGROUND_COLOR	BACKGROUND_COLOUR	BACKGROUND_LOW	BACKGROUND_HIGH	BACKGROUND_STANDARD
BACKWARD	BEEP	BEFORE	BELL	BINARY
BIND	BLANK	BLINK	BLINKING	BLOCK
BOLD	BOTTOM	BOX	BOXED	BY

C

CALL	CANCEL	CATCH	CBL	CCOL
CELL	CELLS	CENTERED	CENTURY_DATE	CENTURY_DAY
CHAIN	CHAINING	CHARACTER	CHARACTERS	CLASS
CLASS_CONTROL	CLASS_ID	CLIENT	CLINE	CLINES

CLOCK_UNITS	CLOSE	CODE-SET	COL	COLLATING
COLOR	COLOUR	COLUMN	COMMA	COMMAND_LINE
COMMIT	COMMON	COMP	COMP_0	COMP_1
COMP_2	COMP_3	COMP_4	COMP_5	COMP_6
COMP_9	COMP_X	COMP_N	COMPRESSION	COMPUTATIONAL
COMPUTATIONAL_0	COMPUTATIONAL_1	COMPUTATIONAL_2	COMPUTATIONAL_3	COMPUTATIONAL_4
COMPUTATIONAL_5	COMPUTATIONAL_6	COMPUTATIONAL_9		COMPUTATIONAL_N
COMPUTE	CONFIGURATION	CONSOLE	CONSTANT	CONTAINS
CONTENT	CONTROLS	CONTROLS_UNCROPPED	CONVERSION	CONVERT
CONVERTING	CONTINUE	CONTROL	COPY	CORR
CORRESPONDING	COUNT	CRT	CSIZE	CURRENCY
CURSOR	CYCLE			

D

DATA	DATE	DATE_COMPILED	DATE_WRITTEN	DAY
DAY_OF_WEEK	DECIMAL_POINT	DECLARATIVES	DEFAULT	DELETE
DELIMITED	DELIMITER	DEPENDING	DESCENDING	DESTROY
DESTINATION	DETAIL	DISABLE	DISC	DISK
DISPLAY	DIVIDE	DIVISION	DOUBLE	DOWN
DUPLICATES	DYNAMIC			

E

ECHO	EJECT	ELSE	EMI	EMPTY_CHECK
ENABLED	ENCRYPTION	END	END_ACCEPT	END_ADD
END_CALL	END_DISPLAY	END_CHAIN	END_COMPUTE	END_DELETE
END_DIVIDE	END_EVALUATE	END_EXEC	END_IF	END_INVOKE
END_MODIFY	END_MOVE	END_MULTIPLY	END_OF_PAGE	END_PERFORM
END_READ	END_RECEIVE	END_RETURN	END_REWRITE	END_SEARCH
END_START	END_STRING	END_SUBTRACT	END_SYNCHRONIZED	END_TRY
END_UNSTRING	END_WAIT	END_WRITE	END_XML	ENTER
ENTRY	ENVIRONMENT	EOL	EOP	EOS
EQUAL	EQUALS	ERASE	ERROR	ESCAPE
EVALUATE	EVENT	EVERY	EXAMINE	EXCEEDS
EXCEPTION	EXCLUSIVE	EXEC	EXIT	EXTEND
EXTERNAL				

F

FACTORY	FALSE	FD	FILE	FILE_CONTROL
FILE_ID	FILE_PREFIX	FILLER	FINAL	FINALLY
FIRST	FLOAT	FLOATING	FONT	FOOTING
FOR	FOREGROUND_COLOR	FOREGROUND_COLOUR	FREE	FROM

FULL	FUNCTION_ID

G - H

GENERATE	GLOBAL	GO	GOBACK	GIVING
GRAPHICAL	GREATER	GRID	GROUP_USAGE	HANDLE
HEADING	HEIGHT	HELP_ID	HIGH	HIGHLIGHT
HIGH_VALUE	HIGH_VALUES			

I - J - K

I_O	I_O_CONTROL	ICON	ID	IDENTIFICATION
IDENTIFIED	IF	IMPLEMENTS	IN	INDEPENDENT
INDEX	INDEXED	INHERITS	INITIAL	INITIALIZE
INITIATE	INPUT	INPUT_OUTPUT	INQUIRE	INSPECT
INSTALLATION	INT	INTERFACE	INTERFACE_ID	INTO
INVALID	INVOKE	IS	JUST	JUSTIFIED
JUSTIFY	KEPT	KEY	KEYBOARD	

L

LABEL	LAST	LAYOUT_MANAGER	LEADING	LEFT
LENGTH	LENGTH_CHECK	LESS	LIMIT	LIMITS
LINAGE	LIKE	LINE	LINES	LINK
LINKAGE	LOCK	LOCK_HOLDING	LONG	LOW
LOWER	LOWLIGHT	LOW_VALUE	LOW_VALUES	

M

MANUAL	MASS_UPDATE	MENU	MERGE	MESSAGE
MESSAGES	METHOD	METHOD_ID	MODAL	MODE
MODELESS	MODIFY	MOVE	MULTIPLE	MULTIPLY

N

NATIONAL	NEGATIVE	NEXT	NO	NO_ECHO
NOT	NOTE	NULL	NULLS	NUMBER
NUMERIC	NUMERIC_EDITED	NUMERIC_FILL		

O

OBJECT	OBJECT_COMPUTER	OCCURS	OF	OFF
OLE	OMITTED	ON	ONLY	OPEN
OPTIONAL	OR	ORDER	ORGANIZATION	OTHER
OTHERS	OTHERWISE	OUTPUT	OVERFLOW	OVERLAPPED
OVERRIDE				

P - Q

PACKED_DECIMAL	PADDING	PAGE	PARAGRAPH	PARSE
PERFORM	PIC	PICTURE	PIXEL	PIXELS
PLUS	POINTER	POP_UP	POS	POSITION
POSITIVE	PREVIOUS	PRINT	PRINTER	PRINTER_1
PRINTING	PRIORITY	PRIVATE	PROCEED	PROCEDURE
PROCESS	PROGRAM	PROGRAM_ID	PROGRAM_STATUS	PROMPT
PROPERTY	PROTECTED	PUBLIC	QUOTE	QUOTES

R

RAISE	RAISING	RANDOM	RD	READ
READY	READERS	RECEIVE	RECORD	RECORDING
RECORDS	REDEFINES	REEL	REFERENCE	RELATIVE
RELEASE	REMAINDER	REMARKS	RENAMES	REPLACE
REPLACING	REPORT	REPORTS	REPOSITORY	REQUIRED
RESERVE	RESET	RESIDENT	RESIZABLE	RESUME
RETURN	RETURNING	REVERSE	REVERSE_VIDEO	REVERSED
REWIND	REWRITE	RIGHT	ROLLBACK	ROUNDED
RUN				

S

SAME	SCREEN	SCROLL	SD	SEARCH
SECTION	SECURE	SECURITY	SELECT	SELF
SEND	SENTENCE	SEPARATE	SEQUENCE	SEQUENTIAL
SERVICE	SET	SHADOW	SHARED	SHARING
SHORT	SIGN	SIGNED_INT	SIGNED_LONG	SIGNED_SHORT
SIZE	SKIP1	SKIP2	SKIP3	SORT
SORT_MERGE	SORT_WORK	SOURCE_COMPUTER	SPACE	SPACES
SPECIAL_NAMES	STANDARD	STANDARD_1	START	STATUS
STOP	STRING	STYLE	SUBTRACT	SUBWINDOW
SUPER	SUPPRESS	SWITCH	SWITCH_0	SWITCH_1
SWITCH_2	SWITCH_3	SWITCH_4	SWITCH_5	SWITCH_6
SWITCH_7	SWITCH_8	SWITCH_9	SWITCH_10	SWITCH_11
SWITCH_12	SWITCH_13	SWITCH_14	SWITCH_15	SWITCH_16

SWITCH_17	SWITCH_18	SWITCH_19	SWITCH_20	SWITCH_21
SWITCH_22	SWITCH_23	SWITCH_24	SWITCH_25	SWITCH_26
SYMBOLIC	SYNC	SYNCHRONIZED	SYSTEM	SYSTEM_INFO

T

TAB	TABLE	TALLYING	TERMINAL_INFO	TERMINATE
TEST	THAN	THEN	THREAD	THREADS
THROUGH	THRU	TIME	TIMES	TITLE
TITLE_BAR	TO	TOOL_BAR	TOP	TRAILING
TRANSACTION	TRANSFORM	TRY	TRUE	TYPE

U - V

UNDERLINE	UNDERLINED	UNEQUAL	UNIT	UNLOCK
UNSIGNED_INT	UNSIGNED_LONG	UNSIGNED_SHORT	UNSTRING	UNTIL
UP	UPDATE	UPDATERS	UPON	UPPER
USAGE	USE	USER_COLORS	USER_GRAY	USER_WHITE
USING	VALUE	VALUES	VALIDATE	VARIANT
VARYING	VISIBLE			

W - X - Y - Z

WAIT	WHEN	WHEN_COMPILED	WIDTH	WINDOW
WITH	WORKING_STORAGE	WRAP	WRITE	WRITERS
XML	YIELD	YYYYDDD	YYYYMMDD	ZERO
ZERO_FILL	ZERO	ZEROES	ZEROS	

Appendix B

Library Routines

isCOBOL provides several library routines.

The table below lists the library routines along with their usability in the various scenarios. For each routine it's specified if it can be used in stand-alone, thin client and web, with some advice where applicable. A detailed documentation of each single routine follows.

Some of the library routines are affected by dedicated configuration properties. See Library Routines Configuration for details.

Note - Unless differently specified in the notes, routines that access client resources in a webClient environment work on the server where the webClient service is running and not on the end user PC where the web browser is running. This rule applies to routines called via CALL CLIENT as well as routine functions that access to the client machine (e.g. C$COPY when one of the parameters start with "@[DISPLAY]:").

Library Routine / op-code	Kind of execution						Notes
	Stand-Alone	Thin Client default on		Thin Client	webClient	webDirect	EIS	
		Server	Client	CALL CLIENT				
$WINHELP	YES	X		YES	NO[1]	NO	NO	In thin client, if CALL CLIENT is not used, then the help is opened on the server machine. [1]In webClient, the help is opened on the server where webClient is running, so the user can't see it in the browser.
A$CURRENT_USER	NO	X		NO	YES	NO	NO	
A$DECRYPT	YES	X		NO	YES	YES	YES	
A$ENCRYPT	YES	X		NO	YES	YES	YES	
A$GETTHREAD	NO	X		NO	YES	NO	NO	

Library Routine / op-code	Kind of execution						Notes
	Stand-Alone	Thin Client default on		Thin Client	webClient	webDirect	EIS	
		Server	Client	CALL CLIENT				
A$GET_DIGEST	YES	X		NO	YES	YES	YES	
A$GET_USER	NO	X		NO	YES	NO	NO	
A$LIST_LOCKS	NO	X		NO	YES	NO	NO	
A$LIST_USERS	NO	X		NO	YES	NO	NO	
A$USERINFO	NO	X		NO	YES	NO	NO	
AS$COPY	YES	X		NO	YES	NO	NO	
ASCII2HEX	YES	X		NO	YES	YES	YES	
ASCII2OCTAL	YES	X		NO	YES	YES	YES	
C$ASYNCPOLL	YES	X		NO	YES	YES	YES	
C$ASYNCRUN	YES	X		NO	YES	YES	YES	
C$CALLEDBY	YES	X		NO	YES	YES	YES	
C$CALLERR	YES	X		NO	YES	YES	YES	
C$CARG	YES	X		NO	YES	YES	YES	
C$CENTURY	YES	X		YES	YES	YES	YES	
C$CHDIR	YES	X		YES	YES	YES	YES	
C$CODESET	YES	X		NO	YES	YES	YES	
C$COPY	YES	X		YES[1]	YES	YES	YES	[1]For the copy client to client there's no difference between using the *@[display]:* prefix before file names or calling the routine via CALL CLIENT.
C$CREATE_TMP_FILE	YES	X		YES	YES	YES	YES	
C$DARG	YES	X		NO	YES	YES	YES	
C$DECRYPT	YES	X		NO	YES	YES	YES	
C$DELAY	YES	X		NO	YES	YES	YES	

Library Routine / op-code	Kind of execution						Notes
	Stand-Alone	Thin Client default on		Thin Client	webClient	webDirect	EIS	
		Server	Client	CALL CLIENT				
C$DELETE	YES	X		YES[1]	YES	YES	YES	[1]For the deletion of a file on the client machine there's no difference between using the *@[display]:* prefix before the file name or calling the routine via CALL CLIENT.
C$DELTREE	YES	X		YES	YES	YES	YES	
C$DESKTOP	YES	X		YES[1]	YES[2]	NO	NO	[1]CALL CLIENT should be used to open a file that resides on the client PC. If the file resides on the server, it's better to use the csFlag parameter. [2]The CDESKTOP-BROWSE op-code opens the URL in a new browser tab. If the URL doesn't start with "http:", then it's appended to the webapp URL. The CDESKTOP-OPEN, CDESKTOP-EDIT and CDESKTOP-PRINT op-codes trigger the download of the file to the end user PC.

Library Routine / op-code	Kind of execution						Notes
	Stand-Alone	Thin Client default on		Thin Client	webClient	webDirect	EIS	
		Server	Client	CALL CLIENT				
C$EASYOPEN	YES	X		YES[1]	YES[2]	NO[3]	NO[3]	[1]CALL CLIENT should be used to open a file that resides on the client PC. If the file resides on the server, it's better to use the csFlag parameter. [2]In webClient environment it triggers the download of the file to the end user PC. [3]In webDirect environment you can call the WD2$REDIRECT routine, in EIS environment you can use the displayBinaryFile method of the HttpHandler class (for example comm-Area:>displayBinaryFile(tmp-file-name "application/pdf").
C$ENCRYPT	YES	X		NO	YES	YES	YES	
C$ENVMAP	YES	X		NO	YES	YES	YES	
C$FILEINFO	YES	X		YES	YES	YES	YES	
C$FORNAME	YES	X		YES	YES	YES	YES	
C$FSCOPY	YES	X		YES	YES	YES	YES	
C$FSDELETE	YES	X		YES	YES	YES	YES	
C$FSFULLNAME	YES	X		NO	YES	YES	YES	
C$FSRENAME	YES	X		YES	YES	YES	YES	
C$FULLNAME	YES	X		NO	YES	YES	YES	
C$GETCGI	NO			NO	NO	NO	YES	
C$GETENV	YES	X		YES	YES	YES	YES	
C$GETLASTFILEOP	YES	X		NO	YES	YES	YES	

Library Routine / op-code	Kind of execution							Notes
	Stand-Alone	Thin Client default on		Thin Client	webClient	webDirect	EIS	
		Server	Client	CALL CLIENT				
C$GETLASTFILENAME	YES	X		NO	YES	YES	YES	
C$GETPID	YES	X		YES	YES	YES	YES	
C$GETRUNENV	YES	X		NO	YES	YES	YES	
C$GUICFG	YES		X	NO	YES[1]	YES[2]	YES[2]	By default print jobs are executed client side. [1] See Known limitations and differences between webClient and Thin Client for more information about printing in webClient environment. [2] The webDirect environment and the EIS environment don't support the print preview.
C$JUSTIFY	YES	X		NO	YES	YES	YES	
C$KEYMAP	YES	X		NO	YES	YES	NO	
C$KEYSTROKE	YES	X		NO	YES	YES	NO	
C$LIST_DIRECTORY	YES	X		YES	YES	YES	YES	
C$LIST_ENVIRONMENT	YES	X		YES	YES	YES	YES	
C$LOCKPID	YES	X		YES	YES	YES	YES	
C$MAKEDIR	YES	X		YES	YES	YES	YES	
C$MBAR	YES	X		NO	YES	NO	NO	
C$MONITOR	YES		X	NO	NO	NO	NO	
C$MYFILE	YES	X		NO	YES	YES	YES	
C$NARG	YES	X		NO	YES	YES	YES	

Library Routine / op-code	Kind of execution							Notes
	Stand-Alone	Thin Client default on		Thin Client	webClient	webDirect	EIS	
		Server	Client	CALL CLIENT				
C$OPENSAVEBOX	YES		X	NO	YES[1]	NO	NO	[1] See Known limitations and differences between webClient and Thin Client for more information about opensave dialogs in webClient environment.
C$PARAMSIZE	YES	X		NO	YES	YES	YES	
C$PARSEEFD	YES	X		YES	YES	YES	YES	
C$PRELOAD	YES	X		NO	YES	YES	YES	
C$PROGINMEM	YES	X		NO	YES	YES	YES	
C$RBMENU	YES	X		NO	YES	NO	NO	
C$REPLACE_ALL	YES	X		NO	YES	YES	YES	
C$RERR	YES	X		NO	YES	YES	YES	
C$RERRNAME	YES	X		NO	YES	YES	YES	
C$RUN	YES	X		YES	YES	YES	YES	
C$SBAR	YES	X		NO	YES	NO	NO	
C$SCRD	YES	X		NO	YES	NO	NO	
C$SCWR	YES	X		NO	YES	NO	NO	
C$SETDEVELOPMENTMODE	YES	X		NO	YES[1]	YES	YES	[1] See Known limitations and differences between webClient and Thin Client for more information about printing in webClient environment.
C$SETENV	YES	X		YES	YES	YES	YES	
C$SHOW	YES	X		NO	YES	YES	NO	
C$SLEEP	YES	X		NO	YES	YES	YES	
C$SOCKET	YES	X		NO	YES	YES	YES	

Library Routine / op-code	Kind of execution						Notes
	Stand-Alone	Thin Client default on		Thin Client	webClient	webDirect	EIS	
		Server	Client	CALL CLIENT				
C$SORT	YES	X		YES	YES	YES	YES	
C$SYSINFO	YES	X		YES	YES	YES	YES	
C$SYSTEM	YES	X		YES[1]	YES	YES	YES	[1]To run a command client-side is better to rely on the CSYS-DESKTOP flag parameter instead of using CALL CLIENT.
C$TBAR	YES	X		NO	YES	NO	NO	
C$TOLOWER	YES	X		NO	YES	YES	YES	
C$TOUPPER	YES	X		NO	YES	YES	YES	
C$TRIM	YES	X		NO	YES	YES	YES	
C$UNLOAD	YES	X		YES	YES	YES	YES	
C$UNLOAD_NATIVE	YES	X[1]		YES	YES	NO[1]	NO[1]	[1]The routine will always fail when called in these cases to avoid unloading a library that other clients may need.
C$UNSET	YES	X		YES	YES	YES	YES	
C$VERSION	YES	X		YES	YES	YES	YES	
C$WRITELOG	YES	X		YES	YES	YES	YES	
C$WRU	YES	X		NO	YES	YES	YES	
C$XML	YES	X		YES	YES	YES	YES	
CBL_AND	YES	X		NO	YES	YES	YES	
CBL_CHANGE_DIR	YES	X		YES	YES	YES	YES	
CBL_CHECK_FILE_EXIST	YES	X		YES	YES	YES	YES	
CBL_CLEAR_SCR	YES	X		NO	YES	NO	NO	
CBL_CLOSE_FILE	YES	X		YES	YES	YES	YES	
CBL_COPY_FILE	YES	X		YES	YES	YES	YES	
CBL_CREATE_DIR	YES	X		YES	YES	YES	YES	

Library Routine / op-code	Kind of execution						Notes
	Stand-Alone	Thin Client default on		Thin Client	webClient	webDirect	EIS	
		Server	Client	CALL CLIENT				
CBL_CREATE_FILE	YES	X		YES	YES	YES	YES	
CBL_DELETE_DIR	YES	X		YES	YES	YES	YES	
CBL_DELETE_FILE	YES	X		YES	YES	YES	YES	
CBL_DIR_SCAN_START	YES	X		YES	YES	YES	YES	
CBL_DIR_SCAN_READ	YES	X		YES	YES	YES	YES	
CBL_DIR_SCAN_END	YES	X		YES	YES	YES	YES	
CBL_EQ	YES	X		NO	YES	YES	YES	
CBL_ERROR_PROC	YES	X		NO	YES	YES	YES	
CBL_EXEC_RUN_UNIT	YES	X		YES	YES	YES	YES	
CBL_EXIT_PROC	YES	X		NO	YES	YES	YES	
CBL_FLUSH_FILE	YES	X		YES	YES	YES	YES	
CBL_GET_SCR_SIZE	YES	X		NO	YES	NO	NO	
CBL_IMP	YES	X		NO	YES	YES	YES	
CBL_JOIN_FILENAME	YES	X		NO	YES	YES	YES	
CBL_NOT	YES	X		NO	YES	YES	YES	
CBL_OPEN_FILE	YES	X		YES	YES	YES	YES	
CBL_OR	YES	X		NO	YES	YES	YES	
CBL_READ_FILE	YES	X		YES	YES	YES	YES	
CBL_RENAME_FILE	YES	X		YES	YES	YES	YES	
CBL_READ_SCR_CHARS	YES	X		NO	YES	NO	NO	
CBL_READ_SCR_CHATTRS	YES	X		NO	YES	NO	NO	
CBL_SPLIT_FILENAME	YES	X		NO	YES	YES	YES	
CBL_TOLOWER	YES	X		NO	YES	YES	YES	
CBL_TOUPPER	YES	X		NO	YES	YES	YES	
CBL_WRITE_FILE	YES	X		YES	YES	YES	YES	
CBL_WRITE_SCR_CHARS	YES	X		NO	YES	NO	NO	

Library Routine / op-code	Kind of execution						Notes
	Stand-Alone	Thin Client default on		Thin Client	webClient	webDirect	EIS	
		Server	Client	CALL CLIENT				
CBL_WRITE_SCR_CHATTRS	YES	X		NO	YES	NO	NO	
CBL_WRITE_SCR_N_CHAR	YES	X		NO	YES	NO	NO	
CBL_WRITE_SCR_N_CHATTR	YES	X		NO	YES	NO	NO	
CBL_XOR	YES	X		NO	YES	YES	YES	
DCI	YES	X		NO	YES	YES	YES	
DELETE	YES	X		YES	YES	YES	YES	
EDBI_DISCONNECT	YES	X		NO	YES	YES	YES	
ESQL$BLOB	YES	X		NO	YES	YES	YES	
HEX2ASCII	YES	X		NO	YES	YES	YES	
I$IO	YES	X		YES	YES	YES	YES	
J$GETFROMLAF	YES		X	NO	NO	NO	NO	
J$NETADDRESS	YES	X		YES	YES	YES	YES	
KEISEN	YES	X		NO	YES	NO	NO	
KEISEN1	YES	X		NO	YES	NO	NO	
KEISEN2	YES	X		NO	YES	NO	NO	
KEISEN_SELECT	YES	X		NO	YES	NO	NO	
M$ALLOC	YES	X		NO	YES	YES	YES	
M$COPY	YES	X		NO	YES	YES	YES	
M$FILL	YES	X		NO	YES	YES	YES	
M$FREE	YES	X		NO	YES	YES	YES	
M$GET	YES	X		NO	YES	YES	YES	
M$PUT	YES	X		NO	YES	YES	YES	
M$SIZE	YES	X		NO	YES	YES	YES	
OCTAL2ASCII	YES	X		NO	YES	YES	YES	

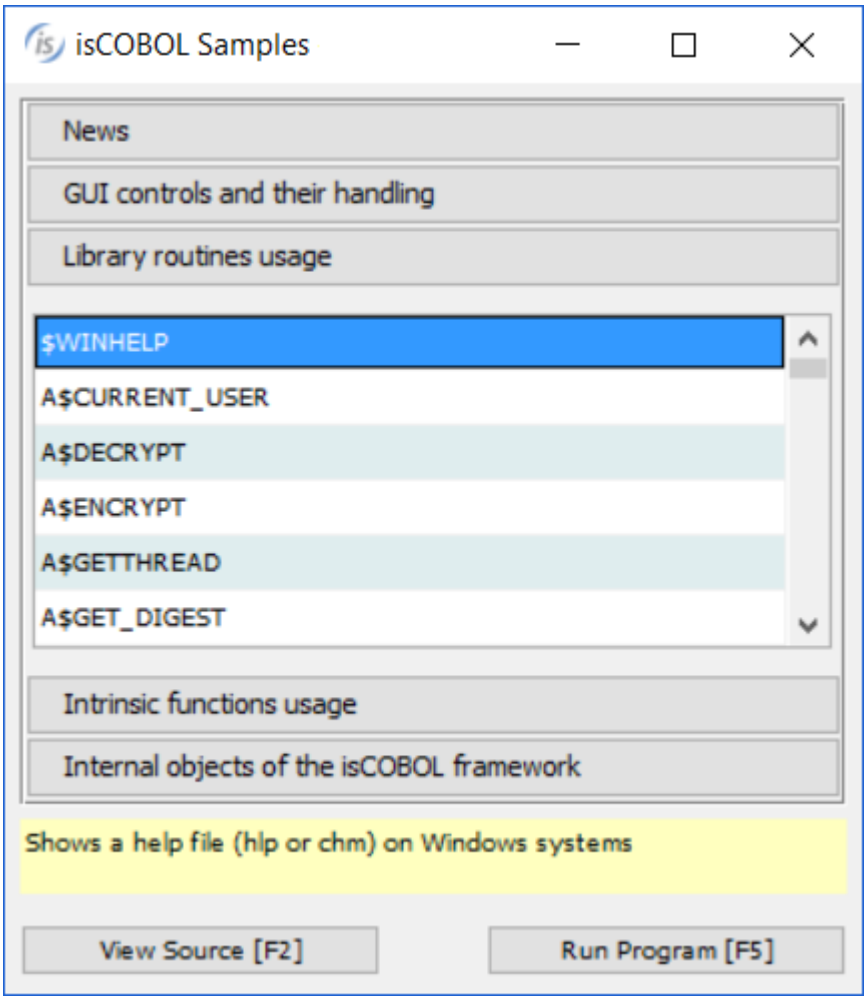
Library Routine / op-code	Kind of execution							Notes
	Stand-Alone	Thin Client default on		Thin Client	webClient	webDirect	EIS	
		Server	Client	CALL CLIENT				
P$	YES		X	NO	YES[1]	YES[2]	YES[2]	[1] See Known limitations and differences between webClient and Thin Client for more information about printing in webClient environment.] [2]The answer is NO for P$DISPLAYDIALOG
R$IO	YES	X		YES	YES	YES	YES	
REG	YES	X		YES[1]	YES	YES	YES	[1]Instead of calling REG routines via CALL CLIENT it's better to call DISPLAY_REG routines.
RENAME	YES	X		YES	YES	YES	YES	
S$IO	YES	X		YES	YES	YES	YES	
SYSTEM	YES	X		YES	YES	YES	YES	
W$BITMAP	YES	X		NO	YES	YES[1]	NO	[1]The WBITMAP-LOAD-FROM-CLIENT function is not supported under webDirect.
W$CAPTURE	YES		X	NO	NO	NO	NO	
W$CENTER_WINDOW	YES		X	NO	YES	NO	NO	
W$CREATEFONT	YES	X		NO	YES	YES	YES	
W$FLUSH	YES		X	NO	YES	YES	NO	
W$FONT	YES	X		NO	YES	YES[1]	YES[1][2]	[2]The WFONT-CHOOSE-FONT function is not supported under EIS and webDirect. [2]Supported only to manage printer fonts.
W$HINT	YES		X	NO	YES	NO	NO	
W$IMAGESIZE	YES	X		NO	YES	YES	NO	

Library Routine / op-code	Kind of execution						Notes
	Stand-Alone	Thin Client default on		Thin Client	webClient	webDirect	EIS	
		Server	Client	CALL CLIENT				
W$KEYBUF	YES	X		NO	YES	NO	NO	
W$MENU	YES		X	NO	YES[1]	YES[1]	NO	[1]In webClient and webDirect it's not possible to display a menu on the try icon.
W$MOUSE	YES		X	NO	YES	YES[1]	NO	[1]In webDirect it's not possible to set a custom mouse shape and get the mouse status
W$PALETTE	YES		X	NO	YES	YES[1]	NO	[1]The WPALETTE-CHOOSE-COLOR function is not supported under webDirect.
W$PROGRESSDIALOG	YES	X		NO	YES	NO	NO	
W$ROTATE	YES	X		NO	YES	YES	YES	
W$SAVE_IMAGE	YES	X		NO	YES	YES	YES	
W$SCALE	YES	X		NO	YES	YES	YES	
W$TEXTSIZE	YES		X	NO	YES	YES	NO	
WD2$CLIENT_INFO	NO			NO	NO	YES	NO	
WD2$EXECJS	NO			NO	NO	YES	NO	
WD2$REDIRECT	NO			NO	NO	YES	NO	
WD2$RUN_JS	NO			NO	NO	YES	NO	
WD2$SESSION	NO			NO	NO	YES	NO	
WIN$PLAYSOUND	YES		X	NO	NO[1]]	NO	NO	[1]In webClient, the sound is played on the server where webClient is running, so the end user can't hear it.

Library Routine / op-code	Kind of execution						Notes
	Stand-Alone	Thin Client default on		Thin Client	webClient	webDirect	EIS	
		Server	Client	CALL CLIENT				
WIN$PRINTER	YES		X	NO	YES[1]	YES	YES	[1] See Known limitations and differences between webClient and Thin Client for more information about printing in webClient environment.]
WIN$VERSION	YES	X		YES	YES	YES	YES	

To override one of these library routines you can compile with the `-sysc` option.

Sample programs for each library routine are available among the isCOBOL Samples.



$WINHELP

The $WINHELP library routine provides a number of functions to manage Windows Help files.

This routine interfaces with the Windows APIs (for hlp files) and hh.exe (for chm files) and therefore, it is not supported on Operating Systems other than Windows.

Microsoft stopped including the 32-bit Help file viewer in Windows releases beginning with Windows Vista and Windows Server 2008. To support customers who still rely on legacy .hlp files, the Microsoft Download Center provides Help file viewer downloads for Windows Vista, Windows 7, Windows Server 2008, and Windows Server 2008 R2.

Syntax:

```
CALL "$WINHELP" USING helpFile
                      opCode
                      parameters
               GIVING returnCode
```

Parameters:

helpFile	Name of the help file	
opCode	Function to be executed. Valid values, defined in iswinhelp.def, are:	
	HELP-CONTENTS	Display the topic specified by the Contents option in the [OPTIONS] section of the .hpj file.
	HELP-CONTEXT	Display the topic identified by the specified context identifier defined in the [MAP] section of the .hpj file.
	HELP-CONTEXTPOPUP	Display the topic identified by the specified context identifier defined in the [MAP] section of the .hpj file in a pop-up window.
	HELP-FINDER	Display the Help Topics dialog box.
	HELP-HELPONHELP	Display help on how to use Windows Help, if the Winhlp32.hlp file is available.
	HELP-QUIT	Inform Windows Help that it is no longer needed. If no other applications have asked for help, Windows closes Windows Help.
parameters	Parameters depend on the opcode.	

Return code:

returnCode can be any signed numeric data item. The meaning depends on the opcode.

Examples:

Example - Display the contents of a Windows Help file contained in the help.hlp file

```
working-storage section.
copy "iscoblib.def".
77  path-help      pic x(255).
...

procedure division.
...
display-help.
   move "c:\myapp\mydocs\help.hlp" to path-help
   call "$winhelp" using path-help , help-contents.
```

HELP-CONTENTS

The HELP-CONTENTS function displays the topic specified by the Contents option in the [OPTIONS] section of the .hpj file. This command is for backward compatibility. New applications should provide a .cnt file and use the HELP_FINDER command.

Syntax:

```
 CALL "$WINHELP" USING helpFile
                       HELP-CONTENTS
                 GIVING returnCode
```

Parameters:

HELP-CONTENTS	Constant	
helpFile	PIC X(n)	Specifies the name of the help file to be opened. It can be either a .HLP or a .CHM file. The file name can be followed by an angle bracket (>) and the name of a secondary window if the topic is to be displayed in a secondary window rather than in the primary window. You must define the name of the secondary window in the [WINDOWS] section of the Help project (.hpj) file.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The $WINHELP library routine is not supported.
WPRTERR-BAD-ARG	The $WINHELP library routine has been called with bad parameters.

HELP-CONTEXT

The HELP-CONTEXT function displays the topic identified by the specified context identifier defined in the [MAP] section of the .hpj file.

Syntax:

```
CALL "$WINHELP" USING helpFile
                      HELP-CONTEXT
                      contextID
                GIVING returnCode
```

Parameters:

HELP-CONTEXT	Constant	
helpFile	PIC X(n)	Specifies the name of the help file to be opened. It can be either a .HLP or a .CHM file. The file name can be followed by an angle bracket (>) and the name of a secondary window if the topic is to be displayed in a secondary window rather than in the primary window. You must define the name of the secondary window in the [WINDOWS] section of the Help project (.hpj) file.
contextID	any numeric data item	Specifies the context identifier for the topic.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The $WINHELP library routine is not supported.
WPRTERR-BAD-ARG	The $WINHELP library routine has been called with bad parameters.

HELP-CONTEXTPOPUP

The HELP-CONTEXTPOPUP function displays the topic identified by the specified context identifier defined in the [MAP] section of the .hpj file in a popup window.

Syntax:

```
CALL "$WINHELP" USING helpFile
                      HELP-CONTEXTPOPUP
                      contextID
                GIVING returnCode
```

Parameters:

HELP-CONTEXTPOPUP	Constant

helpFile	PIC X(n)	Specifies the name of the help file to be opened. It can be either a .HLP or a .CHM file. The file name can be followed by an angle bracket (>) and the name of a secondary window if the topic is to be displayed in a secondary window rather than in the primary window. You must define the name of the secondary window in the [WINDOWS] section of the Help project (.hpj) file.
contextID	any numeric data item	Specifies the context identifier for the topic.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The $WINHELP library routine is not supported.
WPRTERR-BAD-ARG	The $WINHELP library routine has been called with bad parameters.

HELP-FINDER

The HELP-FINDER function displays the Help Topics dialog box.

Syntax:

```
CALL "$WINHELP" USING helpFile
                      HELP-FINDER
               GIVING returnCode
```

Parameters:

HELP-FINDER	Constant	
helpFile	PIC X(n)	Specifies the name of the help file to be opened. It can be either a .HLP or a .CHM file. The file name can be followed by an angle bracket (>) and the name of a secondary window if the topic is to be displayed in a secondary window rather than in the primary window. You must define the name of the secondary window in the [WINDOWS] section of the Help project (.hpj) file.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The $WINHELP library routine is not supported.
WPRTERR-BAD-ARG	The $WINHELP library routine has been called with bad parameters.

HELP-HELPONHELP

The HELP-HELPONHELP function displays help on how to use Windows Help, if the Winhlp32.hlp file is available.

Syntax:

```
CALL "$WINHELP" USING helpFile
                      HELP-HELPONHELP
               GIVING returnCode
```

Parameters:

HELP-HELPONHELP	Constant	
helpFile	PIC X(n)	Specifies the name of the help file to be opened. It can be either a .HLP or a .CHM file.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The $WINHELP library routine is not supported.
WPRTERR-BAD-ARG	The $WINHELP library routine has been called with bad parameters.

HELP-QUIT

The HELP-QUIT function informs Windows Help that it is no longer needed. If no other applications have asked for help, Windows closes Windows Help.

Syntax:

```
CALL "$WINHELP" USING helpFile
                      HELP-QUIT
               GIVING returnCode
```

Parameters:

HELP-QUIT	Constant	
helpFile	PIC X(n)	Specifies the name of the help file to be opened. It can be either a .HLP or a .CHM file

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The $WINHELP library routine is not supported.
WPRTERR-BAD-ARG	The $WINHELP library routine has been called with bad parameters.

A$CURRENT_USER

In an Application Server environment, the A$CURRENT_USER returns information about the logged user.

Syntax:

```
CALL "A$CURRENT_USER" USING ID
                            userName
                            userAddr
                            userComp
                            threadID
                            prog
                           [type]
```

Parameters:

ID	PIC X(n)	Returns the user ID. A value of zero means that the user is administrator. A value of -1 means that no login has been made.
userName	PIC X(n)	Returns the login user name. If no login has been made, it's set to operating system user name
userAddr	PIC X(n)	Returns the login IP address[*]
userComp	PIC X(n)	Returns the login computer name[*]. If the computer name can't be retrieved, the IP address is returned in this field.
threadID	PIC 9(n)	Returns the thread ID
prog	PIC X(n)	Returns the name of the program launched by the client. The special value "Server Call Session" identifies a remote call. The text between square brackets tells the name of the program that was remotely called. See Remote objects for details.
type	PIC 9(1)	Optional parameter. Returns the client type. The value is the sum between one or more of these values: 0 - standard isCOBOL Client 1 - webClient 2 - client running in a separate process due to the iscobol.as.multitasking setting

[*] A computer may have multiple IPs and multiple alias name for the same IP too, so you might not receive the expected IP and name. Usually a safe method to get the desired information is to change the hosts configuration file (/etc/hosts on Linux/Unix, %SystemRoot%\System32\drivers\etc\hosts on Windows) appropriately.

Examples:

Example - Get current user information when running in thin-client mode

```
*> All parameters are to return information from the routine
*> In working-storage define thread-id as pic 9(n), all others as pic x(n)

call "a$current-user" using usr-id
                            usr-name
                            usr-ip-addr
                            usr-pc-name
                            thread-id
                            usr-program
```

A$DECRYPT

This routine decrypts data using a given key.

It uses the Blowfish algorithm. If you wish to decrypt data using a different algorithm, consider using C$DECRYPT instead.

Syntax:

```
CALL "A$DECRYPT" USING dataToDecrypt
                       encryptionKey
                       decryptedData
```

Parameters:

dataToDecrypt	PIC X(n) or string literal.	Specifies the data to decrypt. Trailing spaces are processed too. The size of this field should be multiple of 8.
encryptionKey	PIC X(n) or string literal.	Specifies the key to use during decryption. Since Blowfish is used, the length of the key should be a multiple of 8 and shouldn't be greater than 56.
decryptedData	PIC X(n)	Returns the decrypted data. The size of this field can be one byte smaller than the size of *dataToDecrypt*. For example, if *dataToDecrypt* is 16 bytes in size, *decryptedData* will be 8 to 15 bytes in size, while if *dataToDecrypt* is 8 bytes in size, *decryptedData* will be 1 to 7 bytes in size. In order to be more flexible and avoid specific size calculations, a PIC X ANY LENGTH item can be used.

Examples:

Example - Decrypt a previously encrypted text

```
*> All parameters used by a$decrypt may be defined as pic x(n)
*> encrypted-data should come from a saved encrypted source
*> previously encrypted with a$encrypt

move "Veryant0"     to source-pwd
call "a$decrypt" using encrypted-data, source-pwd, decrypted-data
```

A$ENCRYPT

This routine encrypts data using a given key.

It uses the Blowfish algorithm. If you wish to encrypt data using a different algorithm, consider using C$ENCRYPT instead.

Syntax:

```
CALL "A$ENCRYPT" USING dataToEncrypt
                       encryptionKey
                       encryptedData
```

Parameters:

dataToEncrypt	PIC X(n) or string literal.	Specifies the data to encrypt. Trailing spaces are processed too.
encryptionKey	PIC X(n) or string literal.	Specifies the key to use during ecryption. Since Blowfish is used, the length of the key should be a multiple of 8 and shouldn't be greater than 56.
encryptedData	PIC X(n)	Returns the encrypted data. The size of this field must be the first multiple of 8 bytes greater than the size of *dataToEncrypt*. For example, if *dataToEncrypt* is 1 to 7 bytes in size, *encryptedData* should be at least 8 bytes in size; if *dataToEncrypt* is 8 to 15 bytes in size, *encryptedData* should be at least 16 bytes in size; if *dataToEncrypt* is 16 to 23 bytes in size, *encryptedData* should be at least 24 bytes in size; and so on... In order to be more flexible and avoid specific size calculations, a PIC X ANY LENGTH item can be used.

Examples:

Example - Encrypt a text with specific password

```
*> All parameters used by a$encrypt may be defined as pic x(n)

move "this is test data" to source-str
move "Veryant0"          to source-pwd
call "a$encrypt" using source-str, source-pwd, encrypted-data

*> Some characters in the encrypted-data could be not displayable
*> You may use function dec2hex to convert it to displayable hex codes
```

A$GET_DIGEST

The A$GET_DIGEST routine returns the digest of one or more items.

Message digests are secure one-way hash functions that take arbitrary-sized data and output a fixed-length hash value.

The resulting digest depends on the setting of iscobol.as.digest.

Syntax:

```
CALL "A$GET-DIGEST" using digest
                          param1 [,... , param(n)]
```

Parameters:

digest	PIC X(n).	Returns the digest of the other parameters.
params	PIC X(n) or string literal.	Specifies the items for which the digest will be retrieved.

Examples:

Example - Calculate the digest of the “Veryant” word

```
       working-storage section.
       77  input-string pic x(256).
       77  digest       pic x any length.
       ...
       procedure division.
       ...
           move "Veryant" to input-string.
           call "A$GET_DIGEST" using digest, input-string.
       ...
```

A$GET_USER

The A$GET_USER routine returns information about a given thread ID.

Syntax:

```
CALL "A$GET_USER" USING threadID
                        ID
                        userName
                        userAddr
                        userComp
                        prog
                  GIVING returnCode
```

Parameters:

threadID	PIC 9(n)	Specifies the threadID to query
ID	PIC X(n)	Receives the user ID. If it is zero, it means that the user is an administrator, if it is –1, it means that no login has been made
userName	PIC X(n)	Receives the user name. If no login has been made, it's set to the operating system user name
userAddr	PIC X(n)	Receives the IP address of the client machine[*]
userComp	PIC X(n)	Receives the name of the client machine[*]. If the computer name can't be retrieved, the IP address is returned in this field.
prog	PIC X(n)	Receives the name of the program launched by the client or the name of the last program called through CHAIN statement. The special value "File server" identifies a connection to the isCOBOL File Server. The special value "Server Call Session" identifies a remote call. The text between square brackets tells the name of the program that was remotely called. See Remote objects for details.

[*] A computer may have multiple IPs and multiple alias name for the same IP too, so you might not receive the expected IP and name. Usually a safe method to get the desired information is to change the hosts configuration file (/etc/hosts on Linux/Unix, %SystemRoot%\System32\drivers\etc\hosts on Windows) appropriately.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful
-1	An error occurred

Examples:

Example - Return user information given a thread Id

```
*> Define in working-storage threadID and returnCode as pic 9(n)
*> and all other parameters as pic x(n)
*> threadID should be sent as input parameter, all others will be
*> output parameters (use a$getthread or a$list-users to obtain
*>                    a threadID before the following call)

call "a$get_user" using threadID
                        user-id
                        user-name
                        user-ip-address
                        user-computer-name
                        user-program-name
                  giving returnCode
```

A$GETTHREAD

The A$GETTHREAD routine returns the client's thread ID.

Syntax:

```
CALL "A$GETTHREAD" GIVING threadId
```

Parameters:

threadId	PIC 9(n)	Returns the thread id. The returned value is between 1 and 2147483647.

Examples:

Example - Return the current client thead Id when running in thin-client mode

```
*> threadID is an output parameter, defined as pic 9(n)

call "a$getthread" giving threadID
```

A$LIST_LOCKS

The A$LIST_LOCKS routine returns the list of active locks in an Application Server environment.

The list of locks is available only if iscobol.file.lock_manager * is set in the server configuration.

isCOBOL supports two types of lock:

- LOCK READ: acquired by locking records in a file open in INPUT mode. There can be more than one lock of this type on the same record.
- LOCK WRITE: acquired by locking records in a file open in I-O or OUTPUT mode. There can be only one lock of this type on the same record.

The routine lists both of them.

It's possible to filter the information and retrieve all the locks associated to a specific client (see *threadID* in LISTLOCK-OPEN) or all the locks associated to a specific file (see *fileName* in LISTLOCK-OPEN).

Syntax:

```
CALL "A$LIST_LOCKS" USING opCode
                          parameters
                    GIVING returnCode
```

Parameters:

opCode	The function to be executed. Valid values, defined in iscobol.def, are:	
	LISTLOCK-OPEN	Open the list.
	LISTLOCK-NEXT	Retrieve the next item in the list.
	LISTLOCK-CLOSE	Close a previously open list.
parameters	Parameters depend on the opcode.	

Return-Code:

The definition and meaning of the *returnCode* depend on the opcode.

Examples:

Example - Open the list of locks for an Application Server, loop through all the locks to show them and close the list after that.

```
working-storage section.
copy "iscobol.def".
77  locklist                handle.
77  th-id                   pic 9(5).
77  usr-id                  pic x(3).
77  usr-name                pic x(32).
77  usr-addr                pic x(32).
77  usr-pcname              pic x(32).
77  usr-tid                 pic x(32).
77  usr-prog                pic x(32).
77  usr-count               pic 9(5).
77  lock-count              pic 9(5).
77  lock-filename           pic x(50).
77  lock-tid                pic 9(5).
77  lock-open-mode          pic 9(5).
77  lock-mode               pic 9(5).
77  lock-key-val            pic x(256).
77  lock-key-len            pic 9(3).

get-locks-list.
   initialize lock-filename lock-tid
   call "a$list-locks" using listlock-open
                             null
                             lock-filename
                             lock-tid
                     giving locklist

   if locklist < 1
      display message
              "No Locks Found (verify iscobol.file.lock_manager)"
      exit paragraph
   end-if
   move 0 to lock-count
   perform until 1 = 2
      call "a$list-locks" using listlock-next
                                locklist
                                lock-filename
                                lock-tid
                                lock-open-mode
                                lock-mode
                                lock-key-val
                                lock-key-len
```

```
      if return-code = 0
         exit perform
      end-if
      add 1 to lock-count
      | to get user information from lock-tid
      call "a$get-user" using lock-tid
                              usr-id
                              usr-name
                              usr-addr
                              usr-pcname
                              usr-prog
   end-perform
   call "a$list-locks" using listlock-close
                             locklist
   display message "Number of locks : " lock-count.
```

LISTLOCK-OPEN

The LISTLOCK-OPEN function opens the list of active locks.

Syntax:

```
CALL "A$LIST_LOCKS" USING LISTLOCK-OPEN
                          NULL
                          fileName
                          threadID
                    GIVING listHandle
```

Parameters:

LISTLOCK-OPEN	Constant	
NULL	Constant value	This parameter must be Null
fileName	PIC X(n)	Name of the file you want to inquire. If omitted or set to spaces, then all the files are inquired.
threadID	PIC 9(n)	Thread ID of the client you want to inquire. If omitted or set to 0, then all clients are inquired.

Return code:

listHandle must be a USAGE HANDLE data item. It receives the handle of the list of locks and will be used with the LISTLOCK-NEXT and LISTLOCK-CLOSE functions.

A value of 0 means that the function failed. Possible failure causes are:

- the configuration property `iscobol.file.lock_manager` was not set
- there are no active locks at the moment
- the program is not running in an Application Server environment

LISTLOCK-NEXT

The LISTLOCK-NEXT function retrieves the next item in the list.

Syntax:

```
CALL "A$LIST_LOCKS" USING LISTLOCK-NEXT
                          listHandle
                          fileName
                          threadID
                          openMode
                          lockMode
                          keyVal
                          keyLen
```

Parameters:

LISTLOCK-NEXT	Constant	
listHandle	USAGE HANDLE	Specifies the handle of a list, returned by the LISTLOCK-OPEN function
fileName	PIC X(n)	Receives the name of the file where the lock is.
threadID	PIC 9(n)	Receives the thread ID of the Client that is locking the record. To have additional information you can call the A$GET_USER routine passing this ID to it
openMode	PIC 9	Receives the way by which the file was open. Possible values are: **1...** Input **2...** Output **3...** I-O **6...** Extend **8...** Transaction
lockMode	PIC 9(4)	Receives the lock mode applied to the file. The value is the sum between one or more of the following values: **0...** None **1...** Exclusive **2...** Allowing No Others **3...** Allowing Readers **4...** Allowing Writers **5...** Allowing Updaters **6...** Allowing All **128...** Automatic **256...** Multiple Records **512...** Mass Update **1024...** Bulk Addition **2048...** Transaction
keyVal	PIC X(n)	Receives the value of the primary key of the locked record. This field is intialized to "" when the lock is on the whole file.

keyLen	PIC 9(n)	Receives the length in bytes of the primary key of the locked record. A length of zero means that the lock is on the whole file.

LISTLOCK-CLOSE

The LISTLOCK-CLOSE function closes the list of active locks.

Syntax:

```
CALL "A$LIST_LOCKS" USING  LISTLOCK-CLOSE
                           listHandle
```

Parameters:

LISTLOCK-CLOSE	Constant	
listHandle	USAGE HANDLE	Specifies the handle of a list returned by the LISTLOCK-OPEN function

A$LIST_USERS

The A$LIST_USERS library routine provides a number of functions to retrieve the list of users currently connected to the Application Server.

Syntax:

```
CALL "A$LIST_USERS" USING opCode
                          parameters
                    GIVING returnCode
```

Parameters:

opCode	It is the function to be executed. Valid values, defined in iscobol.def, are:	
	LISTUSR-OPEN	Open the list.
	LISTUSR-NEXT	Retrieve the next item in the list.
	LISTUSR-CLOSE	Close a previously open list.
parameters	Parameters depend on the opcode.	

Return code:

The definition and meaning of the *returnCode* depend on the opcode.

Examples:

Example - Get the list of users connected to an Application Server. Call a$list-users to open the list, loop through all the next users and then close the list

```
working-storage section.
copy "iscobol.def".
77  usrlist                 handle.
77  usr-id                  pic x(3).
77  usr-name                pic x(32).
77  usr-addr                pic x(32).
77  usr-pcname              pic x(32).
77  usr-tid                 pic x(32).
77  usr-prog                pic x(32).
77  usr-count               pic 9(5).
...
procedure division.
...
get-users-list.
    call "a$list-users" using listusr-open
                        giving usrlist
    if usrlist < 1
       display message "Error on User Count!"
       exit paragraph
    end-if
    move 0 to usr-count
    perform until 1 = 2
       call "a$list-users" using listusr-next
                                 usrlist
                                 usr-id
                                 usr-name
                                 usr-addr
                                 usr-pcname
                                 usr-tid
                                 usr-prog
       if return-code = 0
          exit perform
       end-if
       display "User ID: " usr-id "  User Name: " usr-name
       add 1 to usr-count
    end-perform
    call "a$list-users" using listusr-close
                              usrlist
    display message "Number of users connected : " usr-count
    .
```

LISTUSR-OPEN

The LISTUSR-OPEN function opens the list of connected users.

Syntax:

```
CALL "A$LIST_USERS" USING LISTUSR-OPEN
                    GIVING listHandle
```

Parameters:

LISTUSR-OPEN	Constant

Return code:

listHandle must be a USAGE HANDLE data item. It receives the handle of the list of users and will be used with the LISTUSR-NEXT and LISTUSR-CLOSE functions.

LISTUSR-NEXT

The LISTUSR-NEXT function retrieves the next item in the list.

Syntax:

```
CALL "A$LIST_USERS" USING LISTUSR-NEXT
                          listHandle
                          ID
                          userName
                          userAddr
                          userComp
                          threadID
                          prog
                         [type]
```

Parameters:

LISTUSR-NEXT	Constant	
listHandle	USAGE HANDLE	Specifies the handle of a list returned by the LISTUSR-OPEN function
ID	PIC X(n)	Receives the user ID. If it is zero, it means that the user is an administrator, if it is –1, it means that no login has been made
userName	PIC X(n)	Receives the user name. If no login has been made, it's set to the operating system user name
userAddr	PIC X(n)	Receives the IP address of the client machine[*]
userComp	PIC X(n)	Receives the name of the client machine[*]. If the computer name can't be retrieved, the IP address is returned in this field.
threadID	PIC 9(n)	Receives the unique thread ID of the client. The returned value is between 1 and 2147483647.
prog	PIC X(n)	Receives the name of the program launched by the client or the name of the last program called through the CHAIN statement. The special value "File server" identifies a connection to the isCOBOL File Server. The special value "Server Call Session" identifies a remote call. The text between square brackets tells the name of the program that was remotely called. See Remote objects for details.

type	PIC 9(1)	Optional parameter. Returns the client type. The value is the sum between one or more of these values: 0 - standard isCOBOL Client 1 - webClient 2 - client running in a separate process due to the iscobol.as.multitasking setting

[*] A computer may have multiple IPs and multiple alias name for the same IP too, so you might not receive the expected IP and name. Usually a safe method to get the desired information is to change the hosts configuration file (/etc/hosts on Linux/Unix, %SystemRoot%\System32\drivers\etc\hosts on Windows) appropriately.

LISTUSR-CLOSE

The LISTUSR-CLOSE function closes the list of connected users.

Syntax:

```
CALL "A$LIST_USERS" USING  LISTUSR-CLOSE, listHandle
```

Parameters:

LISTUSR-CLOSE	Constant	
listHandle	USAGE HANDLE	Specifies the handle of a list returned by the LISTUSR-OPEN function

A$USERINFO

In an Application Server environment, A$USERINFO sets and retrieves custom information shared between all connected clients.

For this routine to work correctly, the configuration property iscobol.as.multitasking must be either omitted or set to 0.

The custom information set by this routine is also shown in the Application Server administration panel. See Format 5 in Usage of isCOBOL Client for details.

Syntax:

```
CALL "A$USERINFO" USING opCode
                        parameters
```

Parameters:

opCode	Function to be executed. Valid values, defined in iscobol.def, are:	
	AUSERINFO-SET	Stores custom information for the current client.
	AUSERINFO-GET	Retrieves custom information for the current client or for a specific client.
	AUSERINFO-CLEAR	Clears the custom information for the current client.
parameters	Parameters depend on the opcode.	

Examples:

Example - Save, restore and clear custom user data

```
*> copy iscobol.def in the working-storage section
*> info-to-set is pic x(n) and contains any custom data

move "UserType=Operator1,UserLevel=A" to info-to-set
call "A$USERINFO" using auserinfo-set, info-to-set
...
*> Retrieve the custom info, user-info is pic x(n)
call "A$USERINFO" using auserinfo-get, user-info
...
*> Clear the current use custom info
call "A$USERINFO" using auserinfo-clear, user-info
```

AUSERINFO-SET

The AUSERINFO-SET function sets custom information for the current client or for another client running in the same Application Server depending on the *threadID* parameter. Each time this function is called on the same client, it stores new information overwriting the previous one.

Syntax:

```
CALL "A$USERINFO" USING AUSERINFO-SET
                        clientData
                       [threadID]
```

Parameters:

AUSERINFO-SET	Constant	
clientData	PIC X(n)	specifies the information to be stored.
threadID	PIC 9(n)	specifies the thread ID of the client. If omitted, then the current client is used.

AUSERINFO-GET

The AUSERINFO-GET function inquires custom information for the current client or for another client running in the same Application Server depending on the *threadID* parameter.

Syntax:

```
CALL "A$USERINFO" USING AUSERINFO-GET
                        clientData
                       [threadID]
```

Parameters:

AUSERINFO-GET	Constant	
clientData	PIC X(n)	receives the custom information.
threadID	PIC 9(n)	specifies the thread ID of the client you want to inquire. If omitted, then the current client is inquired.

AUSERINFO-CLEAR

The AUSERINFO-CLEAR function deletes the custom information for the current current client.

Syntax:

```
CALL "A$USERINFO" USING AUSERINFO-CLEAR
```

Parameters:

AUSERINFO-CLEAR	Constant

AS$COPY

In an Application Server environment, the AS$COPY library routine copies a file from the server to the client or vice versa.

The AS$COPY library routine is deprecated and supported only for compatibility to previous versions. The C$COPY routine should be used instead.

Syntax:

```
CALL "AS$COPY" USING sourceFile
                     destinationFile
```

Parameters:

sourceFile	PIC X(n)	Specifies the name of the file to be copied. In an Application Server environment, when the file name starts with "CLIENT:", the file will be copied from the client.
destinationFile	PIC X(n)	Specifies the name of the destination file. In an Application Server environment, when the file name starts with "CLIENT:", the file will be copied to the client.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful, the file has been copied.
1	An error occurred, the file has not been copied.

Examples:

Example - Copy a report text file from the server to the client

```
*> define serverReport and clientReport variables as pic x(n)
move "/myapp/myreports/custlist.txt" to serverReport
move "CLIENT:c:/tmp/reports/custlist.txt" to clientReport
call "as$copy" using serverReport
                     clientReport
```

ASCII2HEX

The ASCII2HEX library routine converts a string to a hexadecimal number according to the ASCII table.

Syntax:

```
CALL "ASCII2HEX" USING asciiValue
                       hexValue
```

Parameters:

asciiValue	PIC X(n)	Specifies the ASCII characters to be converted.
hexValue	PIC X(n) It should be twice the size of *asciiValue*	Receives the hexadecimal value of *asciiValue*.

Examples:

Example - Convert "AZ" state code to its representation in HEX

```
*> both variables should be pic x(n), however, the hexValue variable
*> should have at least double of the lenght of the asciiValue

move "AZ" to asciiValue
call "ascii2hex" using asciiValue
                       hexValue

*> hexValue will contain: 415A
```

ASCII2OCTAL

The ASCII2OCTAL library routine converts a string to an octal number according to the ASCII table.

Syntax:

```
 CALL "ASCII2OCTAL" USING asciiValue
                          octalValue
```

Parameters:

asciiValue	PIC X(2)	Contains the ASCII characters to be converted. If characters outside the ASCII table are passed, results are unpredictable.
octalValue	PIC 9(8)	Receives the octal value of *asciiValue*.

Notes:

The routine always converts both characters in *asciiValue*. To convert a single character make sure that the first byte contains 0x00. If *asciiValue* contains spaces, their value 0x20 will be converted to octal.

```
MOVE X"00" TO asciiValue(1:1)
MOVE "A"   TO asciiValue(2:1)
CALL "ASCII2OCTAL" USING asciiValue, octalValue | Returns 101
```

Examples:

Example - Convert "AZ" state code to its representation in OCTAL

```
*> asciiValue should be pic x(n)
*> octalValue should be pic 9(n) and have double the length of the
*> asciiValue

move "AZ" to asciiValue
call "ascii2octal" using asciiValue
                         octalValue

*> octalValue will contain 00040532
```

C$ASYNCPOLL

The C$ASYNCPOLL library routine tests whether or not a thread still exists.

Syntax:

```
CALL "C$ASYNCPOLL" USING threadHandle
                         threadStatus
```

Parameters:

threadHandle	Usage handle	Must contain a valid thread handle.
threadStatus	PIC S9	Represents the status of the thread. 0 The thread still exists. 1 The thread is terminated.

Examples:

Example - Call a program on a separate thread then check if thread is still running

```
working-storage section.
77 threadHandle usage handle.
77 threadStatus pic s9.
...
procedure division.
...
   call thread "programB" handle in threadHandle
                          using "some data"
...
exit-program.
   call "c$asyncpoll" using threadHandle
                            threadStatus
   if threadStatus = 0 | The thread is still running
      display message "Cannot exit, threads still running"
   else
      goback
   end-if.
```

C$ASYNCRUN

The C$ASYNCRUN library routine creates a thread, passing up to 14 parameters to it.

The thread is a COBOL program that can reside on the local host or on a remote machine (see iscobol.remote.code_prefix).

Syntax:

```
CALL "C$ASYNCRUN" USING threadHandle
                        cobolProgram
                       [parameter(s)]
                  GIVING returnCode
```

Parameters:

threadHandle	Usage handle	Contains the handle of the thread that has been created. You can use this handle with the C$ASYNCPOLL routine to test whether the thread still exists or if it's terminated.
cobolProgram	PIC X(n)	The name of the program to be called asynchronously.
parameter(s)	Any cobol type	Optional. You can pass up to 14 parameters to the thread.

Examples:

Example - Call ProgramB on a separate thread

```
working-storage section.
77 threadHandle usage handle.
...
procedure division.
...
   call "c$asyncrun" using threadHandle
                           "programB"
                           "some data".
```

C$CALLEDBY

The C$CALLEDBY library routine returns the name of the program that has called the currently running program.

If the calling program does not exist or is not known, then spaces are returned. Spaces are returned also if the currently running program was called through the iscobol.remote.code_prefix.

Syntax:

```
CALL "C$CALLEDBY" USING  callingProgram
                  GIVING returnCode
```

Parameters:

callingProgram	PIC X(n)	Receives the name of the caller.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	The currently running program has been called by another isCOBOL program.
0	The currently running program is the main program, no other isCOBOL program has called it.
-1	The currently running program has not been called by another isCOBOL program, the caller is unknown

Examples:

Example - Display what program called the current one in the beginning of the program

```
working-storage section.
77 calling-prg  pic x(256).
...
procedure division.
main.
  call "c$calledby" using calling-prg
  if calling-prg = spaces
     display message "No program called me"
  else
     display message "I was called by program : " calling-prg
  end-if.
```

C$CALLERR

The C$CALLERR routine may be called to retrieve the reason why the last CALL statement failed. For accurate information, it must be called before any other CALL statement is executed.

Syntax:

```
CALL "C$CALLERR" USING  errorCode
                       [errorMessage]
```

Parameters:

errorCode	PIC X(2)	It's the error code, it receives the value "01".

errorMessage	PIC X(n)	Optional, It receives a descriptive message about the error encountered.

Examples:

Example - Show the call error after a failed call

```
working-storage section.
77 err-code pic x(2).
77 err-text pic x(256).
...
procedure division.
...
 call "program-not-exist"
      on exception call "c$callerr" using err-code, err-text.
```

C$CARG

The C$CARG library routine returns information about the actual parameter that corresponds to a formal parameter in the USING phrase in the Procedure Division header of a subprogram. This information identifies the type and length of the argument and, when the argument is numeric or numeric edited, the number of digits and scale factor for the argument.

Note: This routine cannot be used in the Procedure Division of a method. It returns information only on parameters passed by CALL, not by INVOKE.

Syntax:

```
CALL "C$CARG" USING okFlag
                    argumentName
                    argumentDesc
```

Parameters:

okFlag	PIC X(1)	Receives "Y" if the argument named by *argumentName* is successfully identified. Receives "N" otherwise.
argumentName	PIC X(n)	Specifies the name of the Linkage Section data item named in the Procedure Division header USING list. If a calling program passes a called program two or more arguments that begin at the same location (either through redefinition, with reference modification, or because one is a group that contains the other), when the called program asks C$CARG for the parameter descriptions, it always receives that of the first actual argument passed that has the same location, regardless of the name specified in argument-name. In these cases, the C$DARG library routine may be used to obtain the distinct descriptions.

argumentDesc	Group item	Receives the information about the data item.
		
		`01 argumentDescription.`
		`  03 argumentType  pic 9(2)  binary(2).`
		`  03 argumentLength  pic 9(8)  binary(4).`
		`  03 argumentDigitCount pic 9(2)  binary(2).`
		`  03 argumentScale  pic s9(2) binary(2).`
		
		argumentType
		Returns a number indicating the type of the argument data item.
		`0: NUMERIC_EDITED`
		`1: UNSIGNED_DISPLAY`
		`2: DISPLAY_EXTERNAL_TRAILING`
		`3: DISPLAY_INTERNAL_TRAILING`
		`4: DISPLAY_EXTERNAL_LEADING`
		`5: DISPLAY_INTERNAL_LEADING`
		`6: SIGNED_COMP_2`
		`7: UNSIGNED_COMP_2`
		`8: SIGNED_COMP_3`
		`9: UNSIGNED_COMP_3`
		`10: COMP_6`
		`11: SIGNED_BINARY`
		`12: UNSIGNED_BINARY`
		`13: SIGNED_NATIVE`
		`14: UNSIGNED_NATIVE`
		`15: FLOATING_POINT`
		`16: ALPHANUMERIC`
		`17: ALPHANUMERIC_JUSTIFIED`
		`18: ALPHANUMERIC_EDITED`
		
		argumentLength
		Returns the BYTE-LENGTH of the argument data item.
		
		argumentDigitCount
		Returns the number of digits defined in the PICTURE character-string for an argument that is a numeric or numeric edited data item as indicated by the *argumentType* field value; otherwise, the value zero is returned for nonnumeric data items. The digit count for a numeric or numeric edited data item does not include any positions defined by the PICTURE symbol P, which represents a scaling position.
		
		argumentScale
		Returns the position of the implied or actual decimal point for an argument that is a numeric or numeric edited data item as indicated by the *argumentType* field value; otherwise, the value zero is returned for nonnumeric data items. If the PICTURE symbol P was used in the description of the data item, the absolute value of the *argumentScale* value will exceed the *argumentDigitCount* value; in this case, a positive scale value indicates an integer with P scaling positions on the right of the PICTURE character-string and a negative scale value indicates a fraction with P scaling positions on the left of the PICTURE character-string.

Examples:

Example - Display information about the ARG2 Linkage data item

```
       working-storage section.
       01  arg-description.
           02 arg-type         pic 99   binary(2).
           02 arg-length       pic 9(8) binary (4).
           02 arg-digit-count  pic 99   binary(2).
           02 arg-scale        pic s99  binary(2).

       77  ok             pic x.

       linkage section.
       77  arg1  pic x(10).
       77  arg2  pic 9(5)v9(5).

       procedure division using arg1 arg2.
       main.
           call "c$carg" using ok arg2 arg-description
           if ok = "y"
              display "type=" arg-type
              display "length=" arg-length
              display "digit-count=" arg-digit-count
              display "scale=" arg-scale
           end-if.
           goback.
```

C$CENTURY

The C$CENTURY library routine retrieves the first two digits of the current year.

Syntax:

```
CALL "C$CENTURY" USING century
```

Parameters:

century	PIC XX or PIC 99	Receives the first two digits of the current year

Examples:

Example - Getting the current century

```
working-storage section.
77 century pic xx.
...
procedure division.
...
  call "c$century" using century.
```

C$CHDIR

The C$CHDIR library routine sets or retrieves the current working directory for relative file paths.

When the current working directory is different from the initial one, all the file names are treated as absolute paths, even if no file-prefix or paths have been specified. This rule could cause strange behaviors with some interfaces.

This routine changes the working directory only for files opened by the COBOL program.

Syntax:

```
CALL "C$CHDIR" USING directoryName
                     [errorNumber]
```

Parameters:

directoryName	PIC X(n)	When set to spaces, it receives the name of the current working directory. When set to a valid path, it represents the working directory to be set.
errorNumber	PIC 9(9) COMP-4	Receives the status of the operation: zero if successful or the operating system's error number if an error has occurred.

Examples:

Example - Getting the current directory and setting a different current directory

```
working-storage section.
77 dirName pic x(256).
77 errNum  pic 9(9) comp-4.
...
procedure division.
...
get-curr-dir.
  move spaces to dirName
  call "c$chdir" using dirName errNum
  if errNum = 0
     display message "Current directory is : " dirName
  else
     display message "Error " errNum " when getting current dir"
  end-if.

set-curr-dir.
  move "c:\myapp\mydir1" to dirName
  call "c$chdir" using dirName errNum
  if errNum = 0
     display message "Current directory was set to : " dirName.
  else
     display message "Error " errNum " when setting current dir"
  end-if.
```

C$CODESET

The C$CODESET library routine translates a string from EBCDIC to ASCII and vice versa.

Syntax:

```
CALL "C$CODESET" USING transFlag
                       length,
                       transString
                      [encoding]
               GIVING ReturnCode
```

Parameters:

transFlag	PIC 9(2) COMP-X	Indicates the type of text in *TransString*, and whether to apply *Length* when performing the translation. *TransFlag* takes one of the following values: 0 Indicates that *TransString* contains EBCDIC and that *Length* specifies the length of the string to translate to ASCII. 1 Indicates that *TransString* contains ASCII and that *Length* specifies the length of the string to translate to EBCDIC. 2 Indicates that *TransString* contains EBCDIC and that 256 bytes of data should be translated to ASCII. The *Length* parameter is ignored. 3 Indicates that *TransString* contains ASCII and that 256 bytes of data should be translated to EBCDIC. The *Length* parameter is ignored.
length	9(9) COMP-X	Specifies the length of the string to translate.
transString	PIC X(n)	Contains the string to translate and the result of the translation.
encoding	PIC X(n)	Contains an alternate java encoding to be used in conversion. By default cp037 is used.

Return code:

returnCode can be any signed numeric data item and provides additional information:

0	Operation successful.
-1	Error converting to EBCDIC
-2	Error converting to ASCII
-3	Bad arguments

Examples:

Example - Translate a string from ASCII to EBCDIC

```
working-storage section.
77 transFlag   pic 9(2) comp-x.
77 strLen      pic 9(9) comp-x.
77 transString pic x(256).
77 retCode     pic s9(1).

procedure division.
...
   move "This is the ASCII text" to transString
   move 1  to transFlag
   move 22 to strLen
   call "c$codeset"
              using transFlag, strLen transString
             giving retCode.
```

C$COPY

The C$COPY library routine copies a file to a destination.

A full path is built according to the working directory before processing the file. This full path may not be valid in c-tree environment where the c-tree server working directory doesn't match with the runtime working directory; in this case, the C$FSCOPY should be used.

Syntax:

```
 CALL "C$COPY" USING sourceFile
                     destinationFile
                    [fileType]
               GIVING returnCode
```

Parameters:

sourceFile	PIC X(n)	Specifies the source file name. If the file name starts with "@[DISPLAY]:", the file will be read from the client in an Application Server. If the file name starts with "isf://", the file will be read from the File Server specified in the name. See The ISF protocol for more information.
destinationFile	PIC X(n)	Specifies the destination file name. If the file name starts with "@[DISPLAY]:", the file will be copied to the client in an Application Server. If the file name starts with "isf://", the file will be copied to the File Server specified in the name. See The ISF protocol for more information.

fileType	PIC X(1)	Specifies the file type. Valid values are: "I" File is Indexed. "R" File is Relative. "RX" File is Relative. Copy file attributes as well. "S" File is binary Sequential, the default. "SX" File is binary Sequential. Copy file attributes as well. “T” File is Line Sequential, handle CR/LF. The default type "S" is suitable to copy generic disk files like PDFs. File type "I" is useful in cases where the original file is held in more than one physical disk file (for example, JIsam and c-tree files are physically held in two separate files). With File type "I" the file name is passed to the indexed file handler APIs so it must be suitable for them. For example, if the file handler is JIsam, avoid the "dat" extension. The routine retrieves input file and ouput file full paths according to the current working directory, so it’s not suitable to copy c-tree files as the working directory of the file server may not match the working directory of the runtime. If you have this need, rely on C$FSCOPY. If the copy of the indexed file is performed in a thin client environment, then it’s important to have the same iscobol.file.index setting on both client and server, otherwise a file conversion will occur. File type "T" is useful while copying a line sequential file between client and server and the line separator of the server operating system is different from the line separator of the client operating system (for example between Linux and Windows). C$COPY will take care of this creating a new line sequential file with the proper line separator on the destination system. The source file attributes such as the last modification date and time are not applied to the destination file unless the type parameter is set to "SX" or "RX".

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful, the file has been copied.
1	An error occurred. Neither the input file nor the output file had "@[DISPLAY]:" in their name. The file has not been copied correctly.
2	An error occurred. Either the input file or the output file had "@[DISPLAY]:" in their name. The file has not been copied correctly.
3	Invalid or missing parameter.

Examples:

Example - Copy file from server to client when running in thin-client mode and backup report on server only

```
working-storage section.
77 sourceFile pic x(256).
77 destFile   pic x(256).
77 retCode    pic s9(9).

procedure division.
...
copy-report-to-client.
   move "/myapp/reports/custlist.txt" to sourceFile
   move "@[DISPLAY]:c:\tmp\reports\custlist.txt" to destFile
   call "c$copy" using sourceFile destFile
          giving retCode.
   if retCode not = 0
      display message "Copy failed with error : " retCode
   else
      display message "File was copied"
   end-if.

backup-report.
   move "/myapp/reports/custlist.txt" to sourceFile
   move "/myapp/backups/reports/custlist.txt" to destFile
   call "c$copy" using sourceFile destFile
          giving retCode.
   if retCode not = 0
      display message "Backup failed with error : " retCode
   else
      display message "File was backed up"
   end-if.
```

C$CREATE_TMP_FILE

The C$CREATE_TMP_FILE library routine creates a temporary file and returns its name.

The file will be handled as a normal sequential file whose physical name is the one returned by the routine. The program logic must take care of deleting the file when it's no more necessary.

Syntax

```
CALL "C$CREATE_TMP_FILE" USING fileName
                               [filePrefix]
                               [fileSuffix]
                               [directory]
                         GIVING returnCode
```

Parameters

fileName	PIC X(n)	Receives the name of the temporary file that has been created. It must be large enough to store the name, otherwise an error occurs.
filePrefix	PIC X(n)	Optional. Prefix to be placed at the beginning of the file name. If omitted, or less than 3 digits, then underscores are placed at the beginning of the file name.
fileSuffix	PIC X(n)	Optional. Suffix to be placed at the end of the file name.
directory	PIC X(n)	Optional. Directory in which to create the file. If omitted, the file is created in the user Temp folder.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
1	The file couldn't be created
2	*fileName* is missing or it is not big enough to store the full path-name, the file is removed

Examples:

Example - Create temporary file to store some customers data

```
working-storage section.
77 dest-file pic x(512).
77 retCode   pic s9(5).
...
procedure division.
...
   call "c$create_tmp_file" using
        dest-file "cust_" ".tmp" "/myapp/tmp"
        giving retCode
   if retCode = 0
      display message "Temp file was created"
   else
      display message "Error creating temp file"
   end-if.
```

C$DARG

The C$DARG library routine returns information about an actual parameter passed in the USING phrase in the CALL statement that called a subprogram. This information identifies the type and length of the argument and, when the argument is numeric or numeric edited, the number of digits and scale factor for the argument.

Note: This routine cannot be used in the Procedure Division of a method. It returns information only on parameters passed by CALL, not by INVOKE.

Syntax:

```
CALL "C$DARG" USING argumentNumber
                    argumentDesc
```

Parameters:

argumentNumber	PIC 9(n)	Specifies the one-relative ordinal position of the actual argument in the USING phrase of the CALL statement used to call the subprogram that calls C$DARG. The value zero obtains the description of the actual argument in the GIVING phrase of that CALL statement. If the value specified is less than zero or greater than the number of actual arguments passed, an argument-description for an omitted argument will be returned (*argumentType* = 32). The actual number of arguments passed can be obtained with the C$NARG library routine. The actual number of arguments may exceed the number of formal arguments declared in the Procedure Division header of the program that calls C$DARG. All of the actual arguments can be accessed using C$DARG even though there is no formal argument name available for accessing the actual arguments beyond the number of formal arguments.

argumentDesc	Group item	Receives the information about the data item.
		
		`01 argumentDescription.`
		`  03 argumentType  pic 9(2)  binary(2).`
		`  03 argumentLength  pic 9(8)  binary(4).`
		`  03 argumentDigitCount pic 9(2)  binary(2).`
		`  03 argumentScale  pic s9(2) binary(2).`
		
		argumentType
		Returns a number indicating the type of the argument data item.
		`0: NUMERIC_EDITED`
		`1: UNSIGNED_DISPLAY`
		`2: DISPLAY_EXTERNAL_TRAILING`
		`3: DISPLAY_INTERNAL_TRAILING`
		`4: DISPLAY_EXTERNAL_LEADING`
		`5: DISPLAY_INTERNAL_LEADING`
		`6: SIGNED_COMP_2`
		`7: UNSIGNED_COMP_2`
		`8: SIGNED_COMP_3`
		`9: UNSIGNED_COMP_3`
		`10: COMP_6`
		`11: SIGNED_BINARY`
		`12: UNSIGNED_BINARY`
		`13: SIGNED_NATIVE`
		`14: UNSIGNED_NATIVE`
		`15: FLOATING_POINT`
		`16: ALPHANUMERIC`
		`17: ALPHANUMERIC_JUSTIFIED`
		`18: ALPHANUMERIC_EDITED`
		`32: OMITTED`
		
		argumentLength
		Returns the BYTE-LENGTH of the argument data item.
		
		argumentDigitCount
		Returns the number of digits defined in the PICTURE character-string for an argument that is a numeric or numeric edited data item as indicated by the *argumentType* field value; otherwise, the value zero is returned for nonnumeric data items. The digit count for a numeric or numeric edited data item does not include any positions defined by the PICTURE symbol P, which represents a scaling position.
		
		argumentScale
		Returns the position of the implied or actual decimal point for an argument that is a numeric or numeric edited data item as indicated by the *argumentType* field value; otherwise, the value zero is returned for nonnumeric data items. If the PICTURE symbol P was used in the description of the data item, the absolute value of the *argumentScale* value will exceed the *argumentDigitCount* value; in this case, a positive scale value indicates an integer with P scaling positions on the right of the PICTURE character-string and a negative scale value indicates a fraction with P scaling positions on the left of the PICTURE character-string.

Examples:

Example - Display information about the ARG2 Linkage data item

```
working-storage section.
01  arg-description.
    02 arg-type         pic 99   binary(2).
    02 arg-length       pic 9(8) binary (4).
    02 arg-digit-count  pic 99   binary(2).
    02 arg-scale        pic s99  binary(2).

77  ok             pic x.

linkage section.
77  arg1  pic x(10).
77  arg2  pic 9(5)v9(5).

procedure division using arg1 arg2.
main.
    call "c$darg" using 2 arg-description.
    display "type=" arg-type.
    display "length=" arg-length.
    display "digit-count=" arg-digit-count.
    display "scale=" arg-scale.
    goback.
```

C$DECRYPT

The C$DECRYPT library routine decrypts data using a specific symmetric-key algorithm.

The algorithm is specified by the iscobol.crypt.algorithm configuration property.

Syntax:

```
CALL "C$DECRYPT" USING encryptedText
                       password
                       decryptedText
                      [errorDescription]
                GIVING returnCode
```

Parameters:

encryptedText	PIC X(n)	Specifies the encrypted text to be decrypted.
password	PIC X(n)	Specifies the encryption key to be used.
decryptedText	PIC X(n)	Returns the decrypted text.
errorDescription	PIC X(n)	Optional parameter. It returns the Java exception behind a failed decryption.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
-1	Error occurred.

Examples:

Example - Encrypt then decrypt a text using the AES algorithm:

```
       WORKING-STORAGE SECTION.
       77  secret     pic x(16) value "0123456789ABCDEF".
       77  clear-text pic x(15) value "some text".
       77  crypt-text pic x(16).
       77  dcrpt-text pic x(15).

       PROCEDURE DIVISION.

       MAIN.
           set environment "crypt.algorithm" to "AES".
           call "C$ENCRYPT" using clear-text secret crypt-text.
           display crypt-text.
           call "C$DECRYPT" using crypt-text secret dcrpt-text.
           display dcrpt-text.
```

C$DELAY

The C$DELAY library routine suspends the running program without using CPU resources.

Syntax:

```
CALL "C$DELAY" USING seconds
```

Parameters:

seconds	any numeric data item or numeric literal	Specifies the time to wait before the program execution is resumed. Decimal values are allowed.

Examples:

Example - Put the program to sleep for half second

```
call "c$delay" using 0.5
```

C$DELETE

The C$DELETE library routine deletes a file.

A full path is built according to the working directory before processing the file. This full path may not be valid in c-tree environment where the c-tree server working directory doesn't match with the runtime working directory; in this case, the C$FSDELETE should be used.

Syntax:

```
CALL "C$DELETE" USING fileName
                      [fileType]
                GIVING returnCode
```

Parameters:

fileName	PIC X(n)	Specifies the name of the file to be deleted. If the file name starts with "@[DISPLAY]:", the file will be searched on the client in an Application Server. If the file name starts with "isf://", the file will be searched via the File Server specified in the name. See The ISF protocol for more information.
fileType	PIC X(1)	Specifies the file type. Valid values are: "I" File is Indexed. "R" File is Relative. "S" File is binary Sequential, the default. The default type "S" is suitable to delete generic disk files like PDFs. File type "I" is useful in cases where the original file is held in more than one physical disk file (for example, JIsam and c-tree files are physically held in two separate files). With File type "I" the file name is passed to the indexed file handler APIs so it must be suitable for them. For example, if the file handler is JIsam, avoid the "dat" extension and use a server side path.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful, the file has been deleted.
1	An error occurred, the file has not been deleted.

Examples:

Example - Delete a temporary file

```
*> define retCode as pic 9(n)

call "c$delete" using "/myapp/tmp/cust_828383838322323.tmp"
     giving retCode
if retCode = 0
   display message "Temp file was deleted"
else
   display message "Error deleting temp file"
end-if.
```

C$DELTREE

The C$DELTREE library routine allows the user to delete a directory and its subdirectories even if there are files in them.

Syntax:

```
CALL "C$DELTREE" USING directoryName
                 GIVING returnCode
```

Parameters:

directoryName	PIC X(n)	Specifies the name of the directory to be deleted. Both full and relative paths are allowed.

Return code:

returnCode can be any signed numeric data item and provides additional information:

0	Operation successful. Directory has been deleted.
1	Operation failed. Directory has not been fully deleted.

Examples:

Example - Delete a logs directory, including its subfolders and files

```
*> define retCode as pic 9(n)

call "c$deltree" using "/myapp/logs"
                 giving retCode
if retCode = 0
   display message "Directory fully deleted"
else
   display message "Directory was not fully deleted"
end-if.
```

C$DESKTOP

The C$DESKTOP library routine allows to perform desktop operations like open/edit, print and mailto. It is useful to open or print a file with the associated application as well as to open the default email client with a pre-compiled email ready to be sent.

Syntax:

```
CALL "C$DESKTOP" USING opCode
                       parameters
                 GIVING returnCode
```

Parameters:

opCode	It is the function to be executed. Valid values, defined in isgui.def, are:	
	CDESKTOP-BROWSE	Browse for a resource
	CDESKTOP-EDIT	Edit a file with the associated application
	CDESKTOP-MAIL	Open the email client
	CDESKTOP-OPEN	Open a file with the associated application
	CDESKTOP-PRINT	Print a file with the associated application
parameters	Parameters depend on the opcode.	

Return code:

returnCode definition and meaning depend on the opcode.

Example - Print a PDF file using the associated application:

```
call "c$desktop" using cdesktop-print "C:\Temp\print_001.pdf".
```

CDESKTOP-BROWSE

The CDESKTOP-BROWSE function opens a web browser or the system file explorer in order to browse for a URI.

Syntax:

```
CALL "C$DESKTOP" USING CDESKTOP-BROWSE
                       URI
                      [csFlag]
                 GIVING returnCode
```

Parameters:

CDESKTOP-BROWSE	Constant	
URI	PIC X(n)	Specifies the URI to browse for. The parameter value must include the protocol. Some examples: • "http://www.veryant.com" (navigate to the Veryant website) • "file://C:/Temp" (show content of C:\Temp)

csFlag	PIC 9	Optional parameter evaluated in thin client environment. If it is set to 1, the operation is executed on the client, otherwise it is executed on the server.

Return code:

returnCode can be any signed numeric data item and provides additional information:

0	Operation successful.
-1	Unsupported operation or routine not supported in this context. This is returned for example if the routine is called in webDirect environment.
-2	Invalid or missing parameters.
-3	I/O error.
-4	A Security exception occured.

CDESKTOP-EDIT

The CDESKTOP-EDIT function opens a file with the associated editor, so the user can edit the file.

Syntax:

```
CALL "C$DESKTOP" USING CDESKTOP-EDIT
                       fileName
                      [csFlag]
               GIVING returnCode
```

Parameters:

CDESKTOP-EDIT	Constant	
fileName	PIC X(n)	Specifies the name of the file to open. It can be either a relative or a full pathname.
csFlag	PIC 9	Optional parameter evaluated in thin client environment. If it is set to 1, the file is edited on the client, otherwise it is edited on the server. In order to edit the file on the client, the routine creates a temporary copy of the file on the client machine, then it opens it with the associated editor. Set this flag to 1 only if the file resides on the server and you want to edit it on the client. If the file reside on the client, call C$DESKTOP with CALL CLIENT statement and omit this flag.

Return code:

returnCode can be any signed numeric data item and provides additional information:

0	Operation successful.
-1	Unsupported operation or routine not supported in this context. This is returned for example if the routine is called in webDirect environment.
-2	Invalid or missing parameters.
-3	I/O error. This is returned for example when the file is not found.
-4	A Security exception occured.

CDESKTOP-MAIL

The CDESKTOP-MAIL function opens a the email client software in "new message" mode, with some fields already filled.

Syntax:

```
CALL "C$DESKTOP" USING CDESKTOP-MAIL
                       URI
                      [csFlag]
                GIVING returnCode
```

Parameters:

CDESKTOP-MAIL	Constant	
URI	PIC X(n)	Mailto URI including the values for the mail fields. The parameter value must include the protocol. Example: • "mailto://support@veryant.com" (send an email to support@veryant.com)
csFlag	PIC 9	Optional parameter evaluated in thin client environment. If it is set to 1, the operation is executed on the client, otherwise it is executed on the server.

Return code:

returnCode can be any signed numeric data item and provides additional information:

0	Operation successful.
-1	Unsupported operation or routine not supported in this context. This is returned for example if the routine is called in webDirect environment.
-2	Invalid or missing parameters.
-3	I/O error.

-4	A Security exception occured.

CDESKTOP-OPEN

The CDESKTOP-OPEN function opens a file with the associated application.

Syntax:

```
CALL "C$DESKTOP" USING CDESKTOP-OPEN
                       fileName
                      [csFlag]
               GIVING returnCode
```

Parameters:

CDESKTOP-OPEN	Constant	
fileName	PIC X(n)	Specifies the name of the file to open. It can be either a relative or a full pathname.
csFlag	PIC 9	Optional parameter evaluated in thin client environment. If it is set to 1, the file is opened on the client, otherwise it is opened on the server. In order to open the file on the client, the routine creates a temporary copy of the file on the client machine, then it opens it. Set this flag to 1 only if the file resides on the server and you want to open it on the client. If the file reside on the client, call C$DESKTOP with CALL CLIENT statement and omit this flag.

Return code:

returnCode can be any signed numeric data item and provides additional information:

0	Operation successful.
-1	Unsupported operation or routine not supported in this context. This is returned for example if the routine is called in webDirect environment.
-2	Invalid or missing parameters.
-3	I/O error. This is returned for example when the file is not found.
-4	A Security exception occured.

CDESKTOP-PRINT

The CDESKTOP-PRINT function prints a file with the associated application.

Syntax:

```
CALL "C$DESKTOP" USING CDESKTOP-PRINT
                       fileName
                      [csFlag]
               GIVING returnCode
```

Parameters:

CDESKTOP-PRINT	Constant	
fileName	PIC X(n)	Specifies the name of the file to print. It can be either a relative or a full pathname.
csFlag	PIC 9	Optional parameter evaluated in thin client environment. If it is set to 1, the file is printed on the client, otherwise it is printed on the server. In order to print the file on the client, the routine creates a temporary copy of the file on the client machine, then it prints it. Set this flag to 1 only if the file resides on the server and you want to print it on the client. If the file reside on the client, call C$DESKTOP with CALL CLIENT statement and omit this flag.

Return code:

returnCode can be any signed numeric data item and provides additional information:

0	Operation successful.
-1	Unsupported operation or routine not supported in this context. This is returned for example if the routine is called in webDirect environment.
-2	Invalid or missing parameters.
-3	I/O error. This is returned for example when the file is not found.
-4	A Security exception occured.

C$EASYOPEN

The C$EASYOPEN library routine opens a file with the associated application, exactly as when the user double click on it in the operating system.

The routine is always asynchronous, it doesn't wait for the user to close the associated application.

By default C$EASYOPEN takes advantage of the *java.awt.Desktop* class in order to open a file with the associated software, but it can be configured to use different methods by setting iscobol.easyopen.method in the configuration.

In thin client environment, if the file is stored on the client machine, then you can open it there by calling the routine via CALL CLIENT, instead, if the file is stored on the server machine, then you can get it downloaded to the client's temp folder and opened on the client by setting the csFlag parameter to 1.

Syntax:

```
CALL "C$EASYOPEN" USING fileName
                        [csFlag]
                  GIVING returnCode
```

Parameters:

fileName	PIC X(n)	Specifies the name of the file to be open.
csFlag	PIC 9	This optional parameter is evaluated in thin client environment. Set it to 0 or omit it in order to open the file on the server. Set it to 1 in order to open the file on the client. In order to open the file on the client, the routine downloads it in the user temp directory on the client machine, then it opens it from there.

Return code:

returnCode can be any signed numeric data item and provides additional information:

0	Operation successful.
1	File not found.
2	An error occurred during the copy from server to client in thin client environment.
3	Generic error. The most common causes are: • invalid or missing parameter • there is no program associated to the *fileName* extension • a Java exception occurred due to missing or incompatible runtime items (only with JDIC method)

Examples:

Example - Open a PDF with the computer's default program for PDF files

```
*> define retCode as pic 9(n)

call "c$easyopen" using "c:\myapp\pdfs\custreport.pdf"
                  giving retCode
if retCode not = 0
   display message "custreport.pdf could not be open"
end-if.
```

C$ENCRYPT

The C$ENCRYPT library routine encrypts data using a specific symmetric-key algorithm.

The algorithm is specified by the iscobol.crypt.algorithm configuration property.

Syntax:

```
CALL "C$ENCRYPT" USING clearText
                       password
                       encryptedText
                      [errorDescription]
                GIVING returnCode
```

Parameters:

clearText	PIC X(n)	Specifies the text to be encrypted.
password	PIC X(n)	Specifies the encryption key to be used.
encryptedText	PIC X(n)	Returns the encrypted text.
errorDescription	PIC X(n)	Optional parameter. It returns the Java exception behind a failed encryption.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
-1	Error occurred.

Examples:

Example - Encrypt then decrypt a text using the AES algorithm:

```
       WORKING-STORAGE SECTION.
       77  secret     pic x(16) value "0123456789ABCDEF".
       77  clear-text pic x(15) value "some text".
       77  crypt-text pic x(16).
       77  dcrpt-text pic x(15).

       PROCEDURE DIVISION.

       MAIN.
           set environment "crypt.algorithm" to "AES".
           call "C$ENCRYPT" using clear-text secret crypt-text.
           display crypt-text.
           call "C$DECRYPT" using crypt-text secret dcrpt-text.
           display dcrpt-text.
```

C$ENVMAP

The C$ENVMAP library routine saves or restores a subset of environment variables. Information is saved to a stack, the routine can be called recursively.

Syntax:

```
CALL "C$ENVMAP" USING opCode
                      [status]
```

Parameters:

opCode	PIC 9(1)	Specifies the operation to be performed. Valid values are: 1 Save variables 0 Restore variables
status	PIC S99 COMP-1.	Receives the function status: 1 Operation completed successfully. 0 Operation failed.

Examples:

Example - Backup the environment, then change one variable temporarily, then restore the value from backup

```
set environment "tempdatapath" to "/myapp/data/temp1"
call "c$envmap" using 1 *> save environment
set environment "tempdatapath" to "/myapp/data/temp2"
accept path-to-temp-data from environment "tempdatapath"
*> path-to-temp-data will be /myapp/data/temp2
call "c$envmap" using 0 *> restore environment
accept path-to-temp-data from environment "tempdatapath"
*> path-to-temp-data will be /myapp/data/temp1
```

C$FILEINFO

The C$FILEINFO library routine retrieves information about a file.

Syntax:

```
CALL "C$FILEINFO" USING fileName
                        fileInfo
                 GIVING returnCode
```

Parameters:

fileName	PIC X(n)	Specifies the name of the file. This should either be a full path name or a name relative to the current directory.

fileInfo	Group Item	Receives the file information. It must have the following structure:
		
		`01 fileInfo.`
		`  03 fileSize  pic x(8) comp-x.`
		`  03 fileDate  pic 9(8) comp-x.`
		`  03 fileTime  pic 9(8) comp-x.`
		
		fileSize
		Size of the file in bytes.
		
		fileDate
		Last modified date of the file, in YYYYMMDD format.
		
		fileTime
		Last modified time of the file.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful, file information has been retrieved.
1	An error occurred, no information retrieved.

Example:

Example - Retrieve file information for a specific file

```
working-storage section.
01 file-info.
   02 file-size pic x(8) comp-x.
   02 file-date pic 9(8) comp-x.
   02 file-time pic 9(8) comp-x.
77 file-name    pic x(256).
77 retCode      pic s9(2).
...
procedure division.
...
   move "c:\myapp\resources\properties1.txt" to file-name
   call "c$fileinfo" using file-name, file-info
        giving retCode
   if retCode = 0
      display message "File size   : " file-size x"0d0a"
                      "Last update : " file-date " - " file-time
   else
      display message "File information could not be retrieved"
   end-if.
```

C$FORNAME

The C$FORNAME library routine tells if a given class is available in the Classpath.

Syntax:

```
CALL "C$FORNAME" USING className
                       [exceptionMessage]
                 GIVING returnCode
```

Parameters:

className	PIC X(n)	Specifies the name of the class to be searched. If the class is included in a package, the package name should be specified as well. The value of this parameter is case sensitive.
exceptionMessage	PIC X(n)	This optional parameter receives the Java exception behind a "class not found" error or spaces if the class was correctly found.

Return code:

returnCode can be any numeric data item and provides additional information:

0	The class is available.
1	Class not found.

Examples:

Example - Verify if the Oracle JDBC driver class can be loaded:

```
call "c$forname" using "oracle.jdbc.OracleDriver"
if return-code = 0
   display message "Oracle JDBC driver is available"
else
   display message "Oracle JDBC driver not available"
end-if
```

C$FSCOPY

The C$FSCOPY library routine copies an indexed file by invoking the proper file handler functions.

The configuration properties iscobol.file.index and iscobol.file.index.FileName specify which file handler is used.

This routine is particularly suitable for copying files that reside on a remote file server like c-tree. In order to copy files on the local machine, C$COPY routine should be used instead.

Syntax:

```
CALL "C$FSCOPY" USING inputFile
                      outputFile
                GIVING returnCode
```

Parameters:

inputFile	PIC X(n)	Specifies the name of the indexed file to be copied. The name is passed to the indexed file handler APIs so it must be suitable for them. For example, if the file handler is c-tree, avoid the "dat" extension.
outputFile	PIC X(n)	Specifies the name of the indexed file to be created. The name is passed as is to the file handler.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
1	An error occurred.

Examples:

Example - Make a backup copy of an indexed file

```
*> define retCode as pic s9(2)

call "c$fscopy" using "cust"
                      "cust_bk"
                giving retCode
if retCode not = 0
   display message "File backup has failed"
end-if
```

C$FSDELETE

The C$FSDELETE library routine deletes an indexed file by invoking the proper file handler functions.

The configuration properties iscobol.file.index and iscobol.file.index.FileName specify which file handler is used.

This routine is particularly suitable for copying files that reside on a remote file server like c-tree. In order to delete files on the local machine, C$DELETE routine should be used instead.

Syntax:

```
CALL "C$FSDELETE" USING inputFile
                  GIVING returnCode
```

Parameters:

inputFile	PIC X(n)	Specifies the name of the indexed file to be deleted. The name is passed to the indexed file handler APIs so it must be suitable for them. For example, if the file handler is c-tree, avoid the "dat" extension.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
1	An error occurred.

Examples:

Example - Delete an indexed file

```
*> define retCode as pic s9(2)

call "c$fsdelete" using "cust"
                  giving retCode
if retCode not = 0
   display message "Operation failed"
end-if
```

C$FSFULLNAME

The C$FSFULLNAME library routine retrieves an indexed file's full path. The runtime Framework takes advantage of the current file system native functions to check for file existence and obtain its full path.

The routine builds full names according to the iscobol.file.prefix, the working directory and the mappings set in the environment (see iscobol.file.env_naming (boolean)), then it tries to open the file using the file handler specified by iscobol.file.index and iscobol.file.index.FileName. If the opening is successful, the current full name is returned.

The ISF protocol is not supported, so each FILE-PREFIX path that starts with "isf://" is invalid for this routine.

Syntax:

```
CALL "C$FSFULLNAME" USING fileName
                          fullName
                    GIVING returnCode
```

Parameters:

fileName	PIC X(n)	Specifies the name of the file.
fullName	PIC X(n)	Receives the full path of fileName.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful, file found.
1	An error occurred, file not found.

Examples:

Example - Get the full absolute path to an indexed file

```
call "c$fsfullname" using "customers" fsfullname
display fsfullname

*> The value displayed could be : c:\myapp\data\customers
*> This value is possible if for instance, the c:\myapp\data
*> path is included in the iscobol.file.prefix runtime property
```

C$FSRENAME

The C$FSRENAME library routine renames an indexed file by invoking the proper file handler functions.

The configuration properties iscobol.file.index and iscobol.file.index.FileName specify which file handler is used.

This routine is particularly suitable for copying files that reside on a remote file server like c-tree. In order to rename files on the local machine, RENAME routine should be used instead.

Syntax:

```
CALL "C$FSRENAME" USING fileName
                        newName
                        exitStatus
```

Parameters:

fileName	PIC X(n)	Specifies the name of the indexed file to be renamed. The name is passed to the indexed file handler APIs so it must be suitable for them. For example, if the file handler is c-tree, avoid the "dat" extension.
newName	PIC X(n)	Specifies the new name of the indexed file. The name is passed to the indexed file handler APIs so it must be suitable for them. For example, if the file handler is c-tree, avoid the "dat" extension.
exitStatus	any numeric data item	Receives the status of the operation: 0 the operation has been executed successfully. 1 the operation failed.

Examples:

Example - Rename an indexed file

```
*> define retCode as pic s9(2)

call "c$fsrename" using "cust" "cust2" retCode.
if retCode not = 0
   display message "Operation failed"
end-if
```

C$FULLNAME

The C$FULLNAME library routine retrieves a file's full path. The runtime Framework follows the rules described in File names interpretation to resolve the name of the file.

Syntax:

```
CALL "C$FULLNAME" USING fileName
                        fullName
                       [fileInfo]
                  GIVING returnCode
```

Parameters:

fileName	PIC X(n)	Specifies the name of the file. This should either be a full path name or a name relative to the current directory.
fullName	PIC X(n)	Receives the full path of fileName. If FILE-PREFIX includes paths starting with "isf://" and the file is found on the specified File Server, a name in the form "isf://servername[:port]:/path/to/file" is returned. See The ISF protocol for more information.

fileInfo	Group Item	Receives the file information. It must have the following structure:
		
		`01 fileInfo.`
		`  03 fileSize  pic x(8) comp-x.`
		`  03 fileDate  pic 9(8) comp-x.`
		`  03 fileTime  pic 9(8) comp-x.`
		
		fileSize
		Size of the file.
		
		fileDate
		Date of the file, in YYYYMMDD format.
		
		fileTime
		Time of the file.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful, file found.
1	An error occurred, file not found.

Examples:

Example - Get the full absolute path to any file

```
call "c$fullname" using "customers.dat" fullname
display fullname
*> The value displayed could be : c:\myapp\data\customers.dat
```

C$GETCGI

The C$GETCGI routine retrieves CGI variables. This routine is implemented for compatibility with other COBOLs.

Syntax:

```
CALL "C$GETCGI"  USING variableName
                       destItem
                      [valueIndex]
                 GIVING returnCode
```

Parameters:

variableName	PIC X(n) or string literal	Name of the variable.
destItem	PIC X(n)	Value of the variable.
valueIndex	PIC 9(n)	Optional parameter. It contains an index that is used when a CGI variable has multiple values in the CGI input data.

Return Code:

returnCode is a signed numeric data item. It receives the size of the resulting value. This may be "0" to indicate that the variable exists but has no value, or "-1" to indicate that the variable does not exist

C$GETENV

The C$GETENV library routine returns the current value of a configuration property.

The routine can be used to inquire both isCOBOL and Java configuration properties.

Syntax:

```
CALL "C$GETENV" USING propertyName
                      propertyValue
                GIVING returnCode
```

Parameters:

propertyName	PIC X(n) or string literal	Name of the property.
propertyValue	PIC X(n)	Value of the property.

Return code:

returnCode is a signed numeric data item:

0	The property is found.
-1	The property is not found.

For your convenience, below is a list of common Java properties that you can inquire with this routine:

Property Name	Value Description
java.version	The Java version number
java.vm.name	The JVM implementation name
java.vm.vendor	The JVM implementation vendor
java.class.version	The Java class format version number
java.home	The Java installation directory
java.io.tmpdir	The default directory in which Java should create temporary files
line.separator	The line separator (e.g. "\n" on Linux)
path.separator	The path separator (e.g. ";" on Windows)
os.name	The operating system/kernel name on which the program runs
os.version	The operating system version
os.arch	The operating system architecture
user.name	The user's account name
user.country	The two-letter country code of the default locale
user.language	The default language
user.timezone	The default time zone
user.home	The user home directory
user.dir	The user's current working directory
java.class.path	The paths of jar files, zip files, and directories used for finding Java classes.
java.library.path	The directory path used for finding native libraries
file.encoding	The character encoding for the default locale
sun.arc.data.model	The type of address/data manipulation supported for the processor/ architecture
sun.cpu.endian	The byte order of the CPU

Examples:

Example - Get the iscobol.file.prefix and iscobol.code_prefix value

```
*> define varname and varvalue as pic x(n)

move "file.prefix" to varname
call "C$GETENV" using varname varvalue
display "iscobol.file.prefix = " varvalue
move "code_prefix" to varname
call "C$GETENV" using varname varvalue
display "iscobol.code_prefix = " varvalue
```

C$GETLASTFILENAME

The C$GELASTFILENAME library routine is used to retrieve the last file name and path used in a I/O statement.

Syntax:

```
CALL "C$GETLASTFILENAME" USING fileName
                               [filePath]
```

Parameters:

fileName	PIC X(n)	Receives the file name.
filePath	PIC X(n)	Optional. Receives the file path.

Examples:

Example - Get the filename and path of the last file used on a I/O stmt along with the last I/O file operation

```
*> define fileName, filePath and fileOp as pic x(n)
display-io-error.
   ...
   call "c$getlastfilename" using fileName filePath
   call "c$getlastfileop"   using fileOp
   *> assuming the file error code was already moved to a general variable
   display message "Error code    : " fs-code   x"0d0a"
                   "I/O Operation : " fileOp    x"0d0a"
                   "On file       : " fileName  x"0d0a"
                   "File path     : " filePath.
```

C$GETLASTFILEOP

The C$GELASTFILEOP library routine returns the name of the last i-o operation.

Syntax:

```
CALL "C$GETLASTFILEOP" USING fileOp
```

Parameters:

fileOp	PIC X(n)	Receives the operation name. Possible values are: "Close" "Delete" "DeleteFile" "DeleteRandom" "Open" "ReadNext" "ReadPrevious" "ReadRandom" "Rewrite" "RewriteRandom" "Start" "Unlock" "Write" "WriteRandom"

Examples:

Example - Get the filename and path of the last file used on a I/O stmt along with the last I/O file operation

```
*> define fileName, filePath and fileOp as pic x(n)
display-io-error.
   ...
   call "c$getlastfilename" using fileName filePath
   call "c$getlastfileop"   using fileOp
   *> assuming the file error code was already moved to a general variable
   display message "Error code    : " fs-code   x"0d0a"
                   "I/O Operation : " fileOp    x"0d0a"
                   "On file       : " fileName  x"0d0a"
                   "File path     : " filePath.
```

C$GETPID

The C$GETPID library routine returns the process ID (PID) of the Java virtual machine (JVM) where the program is running.

Syntax:

```
CALL "C$GETPID" GIVING  processID
```

Return code:

processID can be any numeric data item. It receives the process ID. Ensure to provide a data item large enough to store the process ID; note that the process ID may have more than five digits in some system architectures.

Examples:

Example - Get the process id of the JVM where the program is running

```
*> define process-id as pic 9(n)

call "c$getpid" giving  process-id
```

C$GETRUNENV

The C$GETRUNENV library routine returns the architecture where the COBOL program is running.

Syntax:

```
 CALL "C$GETRUNENV" GIVING  env
```

Parameters:

env	PIC 9(n)	Receives the environment. Possible values, defined in iscobol.def, are: RUNENV-STANDALONE RUNENV-CHARVA RUNENV-REMOTE-CALL RUNENV-THIN-CLIENT RUNENV-WEB-CLIENT RUNENV-WD2 RUNENV-J2EE RUNENV-MOBILE

Examples:

Example - Check if running in webDirect environment

```
working-storage section.
copy "iscobol.def".
77 env pic 9.
...
procedure division.
...
call "c$getrunenv" giving env.
if env = runenv-wd2
   display "Running in webDirect"
end-if.
```

C$GUICFG

The C$GUICFG library routine is used to dynamically manipulate configuration settings in a RM/COBOL compatible way.

Configuration settings in RM/COBOL syntax are translated to the corresponding iscobol property if applicable.

For example:

```
CALL "C$GUICFG" USING "Printer Dialog Always=true"
```

is equivalent to

```
SET ENVIRONMENT "printer.dialog.always" to "true"
```

Syntax:

```
 CALL "C$GUICFG" USING  setting1
                       [setting2 ... settingN]
                        exit-code
```

Parameters:

setting1 *setting2* ... *settingN*	PIC X(n)	Specifies the configuration setting.
exit-code	PIC 9(n)	Receives the exit status: zero for success and non-zero for failure.

Examples:

Example - Set one RM compatible printing property

```
call "c$guicfg" using "Printer Dialog Always=true"
```

C$JUSTIFY

The C$JUSTIFY library routine changes the alignment of a data item's content.

Syntax:

```
CALL "C$JUSTIFY" USING text
                       alignment
```

Parameters:

text	PIC X(n)	Specifies the text to be justified.
alignment	PIC X(1)	It specifies a value representing the justification alignment. Valid values are: "L" Left justified. "C" Centered. "R" Right justified.

Examples:

Example - Justify a string Center, Right and then Left

```
working-storage section.
77 str1   pic x(10) value "hello".
77 ruler  pic x(10) value "1234567890".
...
 procedure division.
...
   display ruler
   display str1
   call "c$justify" using str1 "C"
   display str1
   call "c$justify" using str1 "R"
   display str1
   call "c$justify" using str1 "L"
   display str1.
```

C$KEYMAP

The C$KEYMAP library routine saves or restores the keyboard configuration. Information is saved to a stack, the routine can be called recursively.

Syntax:

```
CALL "C$KEYMAP" USING opCode
                      [status]
```

Parameters:

opCode	PIC 9(1)	Specifies the operation to be performed. Valid values are: 1 Save configuration 0 Restore configuration

status	PIC S99 COMP-1.	Optional. It returns the function status: 1 Operation completed successfully. 0 Operation failed.

Examples:

Example - Backup the keyboard configuration, then change the F1 exception configuration, then restore the value from backup

```
set environment "key.f1" to "exception=1"
*> from here and on when you press F1 it will return exception 1
...
call "c$keymap" using 1 *> save keyboard configuration
set environment "key.f1" to "exception=101"
*> from here and on when you press F1 it will return exception 101
...
call "c$keymap" using 0 *> restore keyboard configuration
*> from here and on when you press F1 it will return exception 1
...
```

C$KEYSTROKE

The C$KEYSTROKE routine allows keystroke configuration using the Acucobol-GT syntax.

Syntax:

```
CALL "C$KEYSTROKE" USING keySetting
```

Parameters:

keySetting	PIC X(n)	is the KEYSTROKE setting you would use for the Acucobol-GT KEYSTROKE environment variable. Example: “EDIT=Next TERMINATE=13 ^M” The last item in this setting is the keycode. See Acucobol-GT key codes for the list of supported key codes. The following EDIT values are supported: Backspace, Delete, Down, Erase-All, Erase-Field, Erase-To-End, First, Last, Next, Page-Down, Page-Up, Previous, Up.

This routine is implemented for compatibility with Acucobol-GT. isCOBOL offers a native syntax to set keystrokes. See Keyboard Configuration for details.

Examples:

Example - Set the exception of Control-A to 1 und F1 to 101 using ACU keystroke syntax

```
call "c$keystroke" using "EXCEPTION=1 ^A"
call "c$keystroke" using "EXCEPTION=101 k1"
```

C$LIST_DIRECTORY

The C$LIST_DIRECTORY library routine provides a number of functions to retrieve the content of a directory.

Syntax:

```
CALL "C$LIST_DIRECTORY" USING opCode
                              parameters
                        GIVING returnCode
```

Parameters:

opCode	It is the function to be executed. Valid values, defined in iscobol.def, are:	
	LISTDIR-OPEN	Open a directory to retrieve its content.
	LISTDIR-NEXT	Retrieve the next file in the directory
	LISTDIR-CLOSE	Close a previously open directory.
parameters	Parameters depend on the opcode.	

Return code:

returnCode definition and meaning depend on the opcode.

Examples:

Example - The following program displays on the system output the list of all files and directories in C:\

```
       PROGRAM-ID. dir.
       WORKING-STORAGE SECTION.
       copy "iscobol.def".
       01  fileEntry.
           05 fileType             pic x(6).
           05 fileName             pic x(74).
       77  listdirHandle           usage handle.
       PROCEDURE DIVISION.
       mainLogic.
           CALL "C$LIST-DIRECTORY" using  LISTDIR-OPEN, "C:\", "*"
                                   giving listdirHandle
           if listdirHandle not = 0
              perform until exit
                 CALL "C$LIST-DIRECTORY" using LISTDIR-NEXT
                                               listdirHandle
                                               fileName
                                               listdir-file-information
                 if fileName = spaces
                    exit perform
                 end-if
                 if listdir-file-type = "D"
                    move "<DIR>" to fileType
                 else
                    move spaces to fileType
                 end-if
                 display fileEntry upon sysout
              end-perform
              CALL "C$LIST-DIRECTORY" using LISTDIR-CLOSE, listdirHandle
           end-if
           goback
           .
```

LISTDIR-OPEN

The LISTDIR-OPEN function searches a directory for a file or subdirectory name that matches a specified name.

Syntax:

```
CALL "C$LIST_DIRECTORY" USING LISTDIR-OPEN
                              directoryName
                              pattern
                        GIVING listdirHandle
```

Parameters:

LISTDIR-OPEN	Constant	
directoryName	PIC X(n)	Specifies the name of an existing the directory.

pattern	PIC X(n)	It specifies a value representing the name to be matched. It can contain wildcard characters, for example, an asterisk (*) or a question mark (?).

Return code:

listdirHandle must be an USAGE HANDLE data item. It receives the handle of the list of matching files and will be used with the LISTDIR-NEXT and LISTDIR-CLOSE functions. It receives 0 if the directory doesn't exist.

LISTDIR-NEXT

The LISTDIR-NEXT function retrieves the next file in a list.

Syntax:

```
CALL "C$LIST_DIRECTORY" USING LISTDIR-NEXT
                              listdirHandle
                              fileName
                             [LISTDIR-FILE-INFORMATION]
```

Parameters:

LISTDIR-NEXT	Constant	
listdirHandle	USAGE HANDLE	Specifies the handle to a file list, returned by the LISTDIR-OPEN function
fileName	PIC X(n)	Receives the name of the next file in the list. When no more files are available, fileName is set to spaces.

LISTDIR-FILE-INFORMATION	Group Item	Receives the file information. This group item, defined in iscobol.def, has the following structure:
		
		`01 listdir-file-information.`
		`  03 listdir-file-type  pic x.`
		`  88 listdir-file-type-directory  value "D".`
		`  88 listdir-file-type-regular-file  value "F".`
		`  88 listdir-file-type-unknown  value "U".`
		`  03 listdir-file-creation-time.`
		`  05 ldfc-year  pic xx comp-x.`
		`  05 ldfc-month  pic x comp-x.`
		`  05 ldfc-day  pic x comp-x.`
		`  05 ldfc-hour  pic x comp-x.`
		`  05 ldfc-minute  pic x comp-x.`
		`  05 ldfc-second  pic x comp-x.`
		`  05 ldfc-hundreths  pic x comp-x.`
		`  03 listdir-file-last-access-time.`
		`  05 ldfla-year  pic xx comp-x.`
		`  05 ldfla-month  pic x comp-x.`
		`  05 ldfla-day  pic x comp-x.`
		`  05 ldfla-hour  pic x comp-x.`
		`  05 ldfla-minute  pic x comp-x.`
		`  05 ldfla-second  pic x comp-x.`
		`  05 ldfla-hundreths  pic x comp-x.`
		`  03 listdir-file-last-modification-time.`
		`  05 ldflm-year  pic xx comp-x.`
		`  05 ldflm-month  pic x comp-x.`
		`  05 ldflm-day  pic x comp-x.`
		`  05 ldflm-hour  pic x comp-x.`
		`  05 ldflm-minute  pic x comp-x.`
		`  05 ldflm-second  pic x comp-x.`
		`  05 ldflm-hundreths  pic x comp-x.`
		`  03 listdir-file-size  pic x(8) comp-x.`
		
		listdir-file-type
		File type. It can be one of the following values:
		
		"B" Block device
		"C" Character device
		"D" Directory
		"F" Standard file
		"P" Pipe
		"S" Socket
		"U" Unknown
		
		listdir-file-creation-time
		This information is not returned. Fields are reserved for future use.
		
		listdir-file-last-access-time
		This information is not returned. Fields are reserved for future use.
		
		listdir-file-last-modification-time
		File last modification time
		
		listdir-file-size
		File size

LISTDIR-CLOSE

The LISTDIR-CLOSE function closes a list of files.

Syntax:

```
CALL "C$LIST_DIRECTORY" USING LISTDIR-CLOSE
                              listdirHandle
```

Parameters:

LISTDIR-CLOSE	Constant	
listdirHandle	USAGE HANDLE	Specifies the handle to a file list, returned by the LISTDIR-OPEN function.

C$LIST_ENVIRONMENT

The C$LIST_ENVIRONMENT library routine provides a number of functions to retrieve a list of Framework properties that are currently set in the environment.

Syntax:

```
CALL "C$LIST_ENVIRONMENT" USING opCode
                                parameters
                          GIVING returnCode
```

Parameters:

opCode	It is the function to be executed. Valid values, defined in iscobol.def, are:	
	LISTENV-OPEN	Open a list of configuration properties.
	LISTENV-NEXT	Retrieve the next property in the list.
	LISTENV-CLOSE	Close a previously open list of properties.
parameters	Parameters depend on the opcode.	

Return code:

returnCode definition and meaning depend on the opcode.

Examples:

Example - The following program displays on the system output the list of all environment variables currently

set.

```
       PROGRAM-ID. listenv.
       WORKING-STORAGE SECTION.
       copy "iscobol.def".
       77 variableName  pic x any length.
       77 variableValue pic x any length.
       77 listenvHandle usage handle.
       PROCEDURE DIVISION.
       mainLogic.
           CALL "C$LIST-ENVIRONMENT" using LISTENV-OPEN
                                     giving listenvHandle
           if listenvHandle not = 0
              perform until exit
                 CALL "C$LIST-ENVIRONMENT" using LISTENV-NEXT
                                                 listenvHandle
                                                 variableName
                 if variableName = spaces
                    exit perform
                 else
                    CALL "C$GETENV" using variableName, variableValue
                    display variableName "=" variableValue upon sysout
                 end-if
              end-perform
              CALL "C$LIST-ENVIRONMENT" using LISTENV-CLOSE,
                                              listenvHandle
           end-if
           goback
           .
```

LISTENV-OPEN

The LISTENV-OPEN function opens the list of the available configuration properties.

Syntax:

```
CALL "C$LIST_ENVIRONMENT" USING LISTENV-OPEN
                          GIVING listenvHandle
```

Parameters:

LISTENV-OPEN	Constant

Return code:

listenvHandle must be an USAGE HANDLE data item. It receives the handle of the list of properties and will be used with the LISTENV-NEXT and LISTENV-CLOSE functions.

LISTENV-NEXT

The LISTENV-NEXT function retrieves the next property in a list.

Syntax:

```
CALL "C$LIST_ENVIRONMENT" USING LISTENV-NEXT
                                listenvHandle
                                variableName
```

Parameters:

LISTENV-NEXT	Constant	
listenvHandle	USAGE HANDLE	Specifies the handle of a list of properties, returned by the LISTENV-OPEN function.
variableName	PIC X(n)	Receives the name of the next property in the list. When no more properties are available, variableName is set to spaces.

LISTENV-CLOSE

The LISTENV-CLOSE function closes a list of properties.

Syntax:

```
CALL "C$LIST_ENVIRONMENT" USING LISTENV-CLOSE
                                listenvHandle
```

Parameters:

LISTENV-NEXT	Constant	
listenvHandle	USAGE HANDLE	Specifies the handle of a list of properties, returned by the LISTENV-OPEN function.

C$LOCKPID

The C$LOCKPID routine returns the c-tree task ID that is locking a record or a file.

The routine requires a c-tree version 10.4.0.39701 or greater and works only with the "ctreej" and "fscsc" file handlers. The "ctree" and "ctree2" file handlers don’t support this routine.

Calling the routine after a “file locked” error is supported only with “ctreej” file handler using c-tree version 11.2.22002 or greater. Using a previous c-tree version or using “fscsc” file handler, the routine can be called only after a “record locked” error.

Syntax:

```
CALL "C$LOCKPID" GIVING taskID
```

Parameters:

taskID	PIC 9(n)	Receives the c-tree task ID. You can obtain information about the client associated to this ID by using c-tree administration utilities like c-treeACEMonitor.

Examples:

Example - Displays the ID of the client that is locking the next record having *iscobol.file.errors_ok=1*:

```
read file1 next.
if file-status = "51"
   call "C$LOCKPID" giving taskID
   display message "Record locked by " taskID
end-if.
```

C$MAKEDIR

The C$MAKEDIR library routine creates a new directory. If the directory already exists, the routine fails.

Syntax:

```
CALL "C$MAKEDIR" USING directoryName
                 GIVING returnCode
```

Parameters:

directoryName	PIC X(n)	Specifies the name of the directory to be created. Both full and relative paths are allowed.

Return code:

returnCode can be any signed numeric data item and provides additional information:

0	Operation successful. Directory has been created.
1	Operation failed. Directory has not been created.

Examples:

Example - Create a new temporary sub-folder

```
call "c$makedir" using "c:\tmp\newtemp" giving retCode
if retCode = 0
   display message "Directory successfully created"
else
   display message "Wrong path"
end-if.
```

C$MBAR

The C$MBAR library routine displays a menu bar on the current window.

Note - This routine is supported for compatibility. If you're creating new programs with isCOBOL, you should consider using W$MENU instead.

Syntax:

```
CALL "C$MBAR" [ USING text-1 [, text-2] ... [, text-n] [, exitCode] ]
```

Parameters:

text-1 *text-2* ... *text-n*	Any alphanumeric data item or literal	These parameters define the items in the pulldown menu along with the strings that are returned to the COBOL program when an item is selected. Using the following syntax, it can specify either a value to be returned or additional sub-menu items: itemname["hint"]=menu where *itemname* is the string that is displayed in the menu bar. *hint* is an optional text string that is displayed as tool tip when the mouse hovers over the menu item. *menu* defines the items in the pulldown menu along with the key codes that are returned to the COBOL program when an item is selected. The syntax for *menu* is: menu -> [(]items[)] items -> item name=[keycode | (menu)][,items] item name -> pulldownname["menu prompt"] keycode -> key code to be sent (see RM/COBOL key codes for C$MBAR, C$RBMENU and C$TBAR routines for details) If the first character of *itemname* is a tilde (~), the menu is disabled. An ampersand (&) in *itemname* causes the next character to be underlined and used as an accelerator.
exitCode	PIC 9(n) BINARY	Optional parameter that returns the status. It will receive zero for success and non-zero for failure.

Calling the routine with no arguments turn off the menu bar.

Examples:

Example - Create a menu bar with a File menu that drops a list of four options

```
CALL "C$MBAR" USING "File=(New=\f1,Open=\f2,Save=\f5,Exit=\e)"
```

C$MONITOR

The C$MONITOR library routine returns information about the available monitors.

Syntax

```
 CALL "C$MONITOR" USING opCode
                        parameters
                  GIVING returnCode
```

Parameters:

opCode	Function to be executed. Valid values, defined in iscobol.def.	
	get-no-monitor	Add keystrokes to the end of the keyboard buffer
	get-monitor-info	Add keystrokes to the beginning of the keyboard buffer
parameter1	Parameters depend on the opcode.	

Return code

returnCode can be any signed numeric data item. The meaning depends on the opcode.

Examples:

Example - Retrieve how many monitors are available

```
       working-storage section.
       copy "iscobol.def".
       77 monitor-count pic 99.
       77 main-index    pic 99.

       procedure division.
       main.
           call "c$monitor" using cmonitor-get-no-monitor
                                  monitor-count, main-index.
           display "there are " monitor-count " monitors available".
           display "the default monitor is number " main-index.
```

CMONITOR-GET-NO-MONITOR

The CMONITOR-GET-NO-MONITOR function returns the number of available monitors and tells which is the default one.

Syntax:

```
CALL "C$MONITOR" USING CMONITOR-GET-NO-MONITOR
                       monitorCount
                      [mainMonitor]
                 GIVING returnCode
```

Parameters:

CMONITOR-GET-NO-MONITOR	Constant	
monitorCount	PIC 9(n)	Receives the number of available monitors.
mainMonitor	PIC 9(n)	Receives the index of the main monitor.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
0	Operation not supported.
-1	Operation failed due to bad arguments.

CMONITOR-GET-MONITOR-INFO

The CMONITOR-GET-MONITOR-INFO function returns information about a given monitor.

Syntax:

```
CALL "C$MONITOR" USING CMONITOR-GET-MONITOR-INFO
                       monitorIndex
                       CMONITOR-DATA
                 GIVING returnCode
```

Parameters:

CMONITOR-GET-MONITOR-INFO	Constant	
monitorIndex	PIC 9(n)	Specifies the index of the monitor to inquire.

CMONITOR-DATA	Group item	Receives the file information. This group item, defined in iscobol.def, has the following structure:
		
		`01 cmonitor-data.`
		`  03 cmonitor-usable-screen-height pic x(2) comp-x.`
		`  03 cmonitor-usable-screen-width pic x(2) comp-x.`
		`  03 cmonitor-physical-screen-height pic x(2) comp-x.`
		`  03 cmonitor-physical-screen-width pic x(2) comp-x.`
		`  03 cmonitor-start-y signed-int.`
		`  03 cmonitor-start-x signed-int.`
		
		cmonitor-usable-screen-height
		Returns the usable screen height in pixels.
		
		cmonitor-usable-screen-width
		Returns the usable screen width in pixels.
		
		cmonitor-physical-screen-height
		Returns the physical screen height in pixels.
		
		cmonitor-physical-screen-width
		Returns the physical screen width in pixels.
		
		cmonitor-start-y
		Returns the coordinate of the first usable pixel on the Y axis.
		
		cmonitor-start-x
		Returns the coordinate of the first usable pixel on the X axis.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
0	Operation not supported.
-1	Operation failed due to bad arguments.

C$MYFILE

The C$MYFILE library routine returns the full name of the physical class file or the jar library that identifies the program according to Class Path and iscobol.code_prefix settings.

If iscobol.cmyfile.classname_only (boolean) is set to true in the configuration, only the class name (without path) is returned. This kind of setting is mandatory when running inside a servlet container (e.g. Tomcat), otherwise a null result is returned.

Syntax:

```
CALL "C$MYFILE" USING fileName
                GIVING returnCode
```

Parameters:

fileName	PIC X(n)	Receives the name of the physical class file or jar library

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
-1	Operation failed.

Examples:

Example - Get the name of the current program being run

```
*> define fileName as pic x(n)
call "c$myfile" using fileName
```

C$NARG

The C$NARG library routine gets the number of parameters passed to the currently running program.

Note: This routine cannot be used in the Procedure Division of a method. It returns information only on parameters passed by CALL, not by INVOKE.

Syntax:

```
CALL "C$NARG" USING paramCount
```

Parameters:

paramCount	PIC 9(n) COMP-1	Receives the number of parameters passed to the currently running program.

Examples:

Example - Get the number of parameters passed to compute a sum with just the given parameters

```
*> These statements on the calling program
call "computesum" using num-1 num-2 num-3 giving ret-sum
call "computesum" using num-1 num-2 giving ret-sum
call "computesum" using num-1 num-2 num-3 num-4 giving ret-sum

...
*> computesum.cbl (called program)
program-id. computesum.

working-storage section.
77 the-sum    pic 9(5).
77 num-params pic 9(2) comp-1.

linkage section.
01 num-1  pic 9(3).
01 num-2  pic 9(3).
01 num-3  pic 9(3).
01 num-4  pic 9(3).

procedure division using num-1 num-2 num-3 num-4.
main.
  call "c$narg" using num-params
  evaluate num-params
  when 1
     move num-1 to the-sum
  when 2
     compute the-sum = num-1 + num-2
  when 3
     compute the-sum = num-1 + num-2 + num-3
  when 4
     compute the-sum = num-1 + num-2 + num-3 +  num-4
  when other
     move 0 to the-sum
  end-evaluate
  goback the-sum.
```

C$OPENSAVEBOX

The C$OPENSAVEBOX library routine provides a number of functions to allow the user to choose a file to be opened, a file to be saved, or a directory name. In thin client environment, this routine allows the user to

choose a directory or file on the client machine.

Note - Due to Java implementation, Windows libraries like Documents, Music, Pictures, and Videos are not shown in the C$OPENSAVEBOX dialogs.

Syntax:

```
CALL "C$OPENSAVEBOX" USING opCode
                           parameters
                    GIVING returnCode
```

Parameters:

opCode	It is the function to be executed. Valid values, defined in isopensave.def, are:	
	OPENSAVE-SUPPORTED	Check if the host system supports the C$OPENSAVEBOX library routine.
	OPENSAVE-OPEN-BOX	Open an "Open" dialog box.
	OPENSAVE-SAVE-BOX	Open a "Save as " dialog box.
	OPENSAVE-BROWSE-FOLDER	Open an "Open" dialog box showing directories only.
	OPENSAVE-OPEN-BOX-MULTI	Open an "Open" dialog box allowing to select multiple items.
	OPENSAVE-BROWSE-FOLDER-MULTI	Open an "Open" dialog box showing directories only and allowing to select multiple items.
	OPENSAVE-NEXT	Returns the next item of a multiple selection performed through OPENSAVE-OPEN-BOX-MULTI or OPENSAVE-BROWSE-FOLDER-MULTI.
	OPENSAVE-SAVE-BOX-CHECKED	Open a "Save as " dialog box that automatically performs a check on file existence.
parameters	Parameters depend on the opcode.	

Return code:

returnCode definition and meaning depend on the opcode.

Examples

Example - The following program opens a dialog box allowing the user to select multiple files and then

shows the list of selected files through a sequence of message boxes.

```
       PROGRAM-ID. multisel.

       WORKING-STORAGE SECTION.
       copy "isopensave.def".

       PROCEDURE DIVISION.
       MAIN.
           initialize opensave-data.
           call "C$OPENSAVEBOX" using opensave-open-box-multi
                                      opensave-data.
           if return-code = 1
              perform show-selected-file
              perform until exit
                 initialize opensave-data
                 call "C$OPENSAVEBOX" using opensave-next
                                            opensave-data
                 if return-code = -1
                    exit perform
                 else
                    perform show-selected-file
                 end-if
              end-perform
           end-if.
           goback.

       show-selected-file.
           display message opnsav-filename
                   title   "The user has selected".
```

Example - Create an Open File box where only txt files can be selected

```
       WORKING-STORAGE SECTION.
       copy "isopensave.def".

       PROCEDURE DIVISION.
       MAIN.
           initialize opensave-data.
           move "Text files (*.txt)|*.txt" to opnsav-filters.
           call "C$OPENSAVEBOX" using opensave-open-box
                                      opensave-data.
```

Example - Let the user find a directory where to save a file and type the filename to be saved or select an existing one. If user selects existing one the routine will ask if user wants to overwrite it.

```
*> copy "isopensave.def" on working-storage

initialize opensave-data
move "Save to file" to opnsav-title

call "c$opensavebox" using opensave-save-box-checked,
                           opensave-data
                     giving opensave-status

if opensave-status > 0
   display message "Saving to file : " opnsav-filename
else
   display message "Operation cancelled"
end-if
```

OPENSAVE-SUPPORTED

The OPENSAVE-SUPPORTED function checks if the host system supports the C$OPENSAVEBOX library routine.

Syntax:

```
 CALL "C$OPENSAVEBOX" USING OPENSAVE-SUPPORTED
                      GIVING returnCode
```

Parameters:

OPENSAVE-SUPPORTED	Constant

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	The C$OPENSAVEBOX library routine is supported by the host operating system.
-1	The C$OPENSAVEBOX library routine is not supported by the host operating system.

OPENSAVE-OPEN-BOX

The OPENSAVE-OPEN-BOX function opens an "Open" dialog box.

Syntax:

```
CALL "C$OPENSAVEBOX" USING OPENSAVE-OPEN-BOX
                           OPENSAVE-DATA
                     GIVING returnCode
```

Parameters:

OPENSAVE-OPEN-BOX	Constant	

OPENSAVE-DATA	Group Item	It is a structure that contains information used to initialize the dialog box. When the OPENSAVE-OPEN-BOX function returns successfully, this structure contains information about the user's selection. This group item, defined in isopensave.def, has the following structure:
		
		`01 opensave-data.`
		`  03 opnsav-filename pic x(256).`
		`  03 opnsav-flags pic 9(4) comp-x value 0.`
		`  03 opnsav-default-ext pic x(12).`
		`  03 opnsav-title pic x(80).`
		`  03 opnsav-filters pic x(512).`
		`  03 opnsav-default-filter pic 9(4) comp-x value 0.`
		`  03 opnsav-default-dir pic x(128).`
		`  03 opnsav-basename pic x(128).`
		
		opnsav-filename
		Specifies the default filename. When the OPENSAVE-OPEN-BOX function returns successfully, it receives the drive designator, path, file name, and extension of the selected file.
		
		opnsav-flags
		The following constants, defined in isopensave.def, can be combined to set this field:
		-*opensave-show-preview-flag*: the file chooser dialog shows image files preview on the right.
		-*opensave-pathmustexist*: the user can type only valid paths and file names.
		-*opensave-filemustexist*: the user can type only names of existing files.
		-*opensave-createprompt*: if the user specifies a file that does not exist, this flag causes the dialog box to prompt the user for permission to create the file.
		-*opensave-noreadonlyreturn:* the returned file must not be write-protected.
		
		opnsav-default-ext
		This item holds the default file name extension. The extension is the string of characters that appear after the "." in the file name. The value of OPNSAV-DEFAULT-EXT is added to the file name typed by the user, if the user does not type an extension. The default extension should not include the period ".". Set this item to spaces to avoid having a default extension.
		
		opnsav-title
		String to be placed in the title bar of the dialog box. If this member is spaces, the system uses the default title.
		
		opnsav-filters
		The value of OPNSAV-FILTERS describes the set of filters that the dialog box will use to restrict the set of files shown to the user.

	Filters make it easier for a user to navigate through a large directory by limiting the files shown at once. Each filter consists of a pair of descriptors. These descriptors are separated by a vertical bar character ("|"). Here is a sample OPNSAV-FILTERS setting that contains two filters: "isCOBOL source files (*.cbl)|*.cbl|All files (*.*)|*.*" *opnsav-default-filter* This item is used in conjunction with OPNSAV-FILTERS. The value of OPNSAV-DEFAULT-FILTER determines which of the given filters to use as the initial filter. A value of "1" selects the first filter pair, "2" selects the second pair, and so on. A value of zero also selects the first pair. This setting is not used if no filters are defined. *opnsav-default-dir* Specifies the initial directory. *opnsav-basename* When the routine returns, this item contains the base file name of the file chosen by the user. This differs from the value of OPNSAV-FILENAME in that all directory information is removed, leaving only the file name.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful. File has been chosen.
-1	The user has pressed the "Cancel" button. opensaveData contents are not updated.

OPENSAVE-SAVE-BOX

The OPENSAVE-SAVE-BOX function opens a "Save as" dialog box.

Syntax:

```
CALL "C$OPENSAVEBOX" USING OPENSAVE-SAVE-BOX
                           OPENSAVE-DATA
                     GIVING returnCode
```

Parameters:

OPENSAVE-SAVE-BOX	Constant

OPENSAVE-DATA	Group Item	It is a structure that contains information used to initialize the dialog box. When the OPENSAVE-SAVE-BOX function returns successfully, this structure contains information about the user's file selection. This group item, defined in isopensave.def, has the following structure:
		
		`01 opensave-data.`
		`  03 opnsav-filename pic x(256).`
		`  03 opnsav-flags pic 9(4) comp-x value 0.`
		`  03 opnsav-default-ext pic x(12).`
		`  03 opnsav-title pic x(80).`
		`  03 opnsav-filters pic x(512).`
		`  03 opnsav-default-filter pic 9(4) comp-x value 0.`
		`  03 opnsav-default-dir pic x(128).`
		`  03 opnsav-basename pic x(128).`
		
		opnsav-filename
		Specifies the default filename. When the OPENSAVE-SAVE-BOX function returns successfully, it receives the drive designator, path, file name, and extension of the selected file.
		
		opnsav-flags
		The following constants, defined in isopensave.def, can be combined to set this field:
		-opensave-show-preview-flag: the file chooser dialog shows image files preview on the right.
		-opensave-overwriteprompt: generate a message box if the selected file already exists.
		-opensave-pathmustexist: the user can type only valid paths and file names.
		-opensave-filemustexist: the user can type only names of existing files.
		-opensave-createprompt: if the user specifies a file that does not exist, this flag causes the dialog box to prompt the user for permission to create the file.
		-opensave-noreadonlyreturn: the returned file must not be write-protected.
		
		opnsav-default-ext
		This item holds the default file name extension. The extension is the string of characters that appear after the "." in the file name. The value of OPNSAV-DEFAULT-EXT is added to the file name typed by the user, if the user does not type an extension. The default extension should not include the period ".". Set this item to spaces to avoid having a default extension.
		
		opnsav-title
		String to be placed in the title bar of the dialog box. If this member is spaces, the system uses the default title.
		
		opnsav-filters
		The value of OPNSAV-FILTERS describes the set of filters that the dialog box will use to restrict the set of files shown to the user. Filters make it easier for a user to navigate through a large directory by limiting the files shown at once.
		
		Each filter consists of a pair of descriptors. These descriptors are separated by a vertical bar character ("|").
		
		Here is a sample OPNSAV-FILTERS setting that contains two filters:
		
		"isCOBOL source files (*.cbl)|*.cbl|All files (*.*)|*.*"
		
		opnsav-default-filter
		This item is used in conjunction with OPNSAV-FILTERS. The value of OPNSAV-DEFAULT-FILTER determines which of the given filters to use as the initial filter. A value of "1" selects the first filter pair, "2" selects the second pair, and so on. A value of zero also selects the first pair. This setting is not used if no filters are defined.

	opnsav-default-dir Specifies the initial directory. *opnsav-basename* When the routine returns, this item contains the base file name of the file chosen by the user. This differs from the value of OPNSAV-FILENAME in that all directory information is removed, leaving only the file name.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful. File has been chosen.
-1	The user has pressed the "Cancel" button. opensaveData contents are not updated.

OPENSAVE-BROWSE-FOLDER

The OPENSAVE-BROWSE-FOLDER function opens an "Open" dialog box, showing directories only.

Syntax:

```
CALL "C$OPENSAVEBOX" USING OPENSAVE-BROWSE-FOLDER
                           OPENSAVE-DATA
                     GIVING returnCode
```

Parameters:

OPENSAVE-BROWSE-FOLDER	Constant

OPENSAVE-DATA	Group Item	It is a structure that contains information used to initialize the dialog box. When the OPENSAVE-BROWSE-FOLDER function returns successfully, this structure receives information about the user's selection. This group item, defined in isopensave.def, has the following structure:
		
		`01 opensave-data.`
		`  03 opnsav-filename pic x(256).`
		`  03 opnsav-flags pic 9(4) comp-x value 0.`
		`  03 opnsav-default-ext pic x(12).`
		`  03 opnsav-title pic x(80).`
		`  03 opnsav-filters pic x(512).`
		`  03 opnsav-default-filter pic 9(4) comp-x value 0.`
		`  03 opnsav-default-dir pic x(128).`
		`  03 opnsav-basename pic x(128).`
		
		opnsav-filename
		Specifies the default filename. When the OPENSAVE-BROWSE-FOLDER function returns successfully, it receives the drive designator and path of the selected directory.
		
		opnsav-flags
		The following constants, defined in isopensave.def, can be combined to set this field:
		-opensave-browse-dontgobelowdomain: network folders below the domain level in the dialog box are not included.
		-opensave-browse-browseincludefiles: the browse dialog box displays files as well as folders.
		
		opnsav-default-ext
		Not used.
		
		opnsav-title
		String to be placed in the title bar of the dialog box. If this member is spaces, the system uses the default title.
		
		opnsav-filters
		Not used.
		
		opnsav-default-filter
		Not used.
		
		opnsav-default-dir
		Specifies the initial directory.
		
		opnsav-basename
		Not used.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful. Directory has been chosen.
-1	The user has pressed the "Cancel" button. opensaveData contents are not updated.

OPENSAVE-OPEN-BOX-MULTI

The OPENSAVE-OPEN-BOX-MULTI function opens a "Open" dialog box and allows the user to select more than one file by holding either CTRL or SHIFT while selecting the files.

Syntax:

```
CALL "C$OPENSAVEBOX" USING OPENSAVE-OPEN-BOX-MULTI
                           OPENSAVE-DATA
                    GIVING returnCode
```

Parameters:

OPENSAVE-OPEN-BOX-MULTI	Constant	

OPENSAVE-DATA	Group Item	It is a structure that contains information used to initialize the dialog box. When the OPENSAVE-OPEN-BOX-MULTI function returns successfully, this structure contains information about the user's selection. This group item, defined in isopensave.def, has the following structure:
		
		`01 opensave-data.`
		`  03 opnsav-filename pic x(256).`
		`  03 opnsav-flags pic 9(4) comp-x value 0.`
		`  03 opnsav-default-ext pic x(12).`
		`  03 opnsav-title pic x(80).`
		`  03 opnsav-filters pic x(512).`
		`  03 opnsav-default-filter pic 9(4) comp-x value 0.`
		`  03 opnsav-default-dir pic x(128).`
		`  03 opnsav-basename pic x(128).`
		
		opnsav-filename
		Specifies the default filename. When the OPENSAVE-OPEN-BOX-MULTI function returns successfully, it receives the drive designator, path, file name, and extension of the first selected file.
		
		opnsav-flags
		Not used
		
		opnsav-default-ext
		This item holds the default file name extension. The extension is the string of characters that appear after the "." in the file name. The value of OPNSAV-DEFAULT-EXT is added to the file name typed by the user, if the user does not type an extension. The default extension should not include the period ".". Set this item to spaces to avoid having a default extension.
		
		opnsav-title
		String to be placed in the title bar of the dialog box. If this member is spaces, the system uses the default title.
		
		opnsav-filters
		The value of OPNSAV-FILTERS describes the set of filters that the dialog box will use to restrict the set of files shown to the user.

	opnsav-default-dir Specifies the initial directory. *opnsav-basename* When the routine returns, this item contains the base file name of the file chosen by the user. This differs from the value of OPNSAV-FILENAME in that all directory information is removed, leaving only the file name.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful. File has been chosen.
-1	The user has pressed the "Cancel" button. *opensaveData* contents are not updated.

OPENSAVE-BROWSE-FOLDER-MULTI

The OPENSAVE-BROWSE-FOLDER-MULTI function opens an "Open" dialog box, showing directories only and allows the user to select multiple items by holding either CTRL or SHIFT while he selects the items.

Syntax:

```
CALL "C$OPENSAVEBOX" USING OPENSAVE-BROWSE-FOLDER-MULTI
                           OPENSAVE-DATA
                     GIVING returnCode
```

Parameters:

OPENSAVE-BROWSE-FOLDER-MULTI	Constant

OPENSAVE-DATA	Group Item	It is a structure that contains information used to initialize the dialog box. When the OPENSAVE-BROWSE-FOLDER-MULTI function returns successfully, this structure receives information about the user's selection. This group item, defined in isopensave.def, has the following structure:
		
		`01 opensave-data.`
		`  03 opnsav-filename pic x(256).`
		`  03 opnsav-flags pic 9(4) comp-x value 0.`
		`  03 opnsav-default-ext pic x(12).`
		`  03 opnsav-title pic x(80).`
		`  03 opnsav-filters pic x(512).`
		`  03 opnsav-default-filter pic 9(4) comp-x value 0.`
		`  03 opnsav-default-dir pic x(128).`
		`  03 opnsav-basename pic x(128).`
		
		opnsav-filename
		Specifies the default filename. When the OPENSAVE-BROWSE-FOLDER-MULTI function returns successfully, it receives the drive designator and path of the first selected directory.
		
		opnsav-flags
		Not used
		
		opnsav-default-ext
		Not used.
		
		opnsav-title
		String to be placed in the title bar of the dialog box. If this member is spaces, the system uses the default title.
		
		opnsav-filters
		Not used.
		
		opnsav-default-filter
		Not used.
		
		opnsav-default-dir
		Specifies the initial directory.
		
		opnsav-basename
		Not used.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful. Directory has been chosen.
-1	The user has pressed the "Cancel" button. opensaveData contents are not updated.

OPENSAVE-NEXT

The OPENSAVE-NEXT function returns the next item selected by the user when multiple selection was allowed. This function should be called multiple times after OPENSAVE-OPEN-BOX-MULTI and OPENSAVE-

BROWSE-FOLDER-MULTI in order to retrieve the list of selected items. When no more items are available, -1 is returned as exit status.

Syntax:

```
CALL "C$OPENSAVEBOX" USING OPENSAVE-NEXT
                           OPENSAVE-DATA
                     GIVING returnCode
```

Parameters:

OPENSAVE-NEXT	Constant	

OPENSAVE-DATA	Group Item	When the OPENSAVE-NEXT function returns successfully, this structure receives information about the user's selection. This group item, defined in isopensave.def, has the following structure:
		
		`01 opensave-data.`
		`  03 opnsav-filename pic x(256).`
		`  03 opnsav-flags pic 9(4) comp-x value 0.`
		`  03 opnsav-default-ext pic x(12).`
		`  03 opnsav-title pic x(80).`
		`  03 opnsav-filters pic x(512).`
		`  03 opnsav-default-filter pic 9(4) comp-x value 0.`
		`  03 opnsav-default-dir pic x(128).`
		`  03 opnsav-basename pic x(128).`
		
		opnsav-filename
		Specifies the default filename. When the OPENSAVE-BROWSE-FOLDER function returns successfully, it receives the drive designator and path of the first selected directory.
		
		opnsav-flags
		Not used
		
		opnsav-default-ext
		Not used.
		
		opnsav-title
		String to be placed in the title bar of the dialog box. If this member is spaces, the system uses the default title.
		
		opnsav-filters
		Not used.
		
		opnsav-default-filter
		Not used.
		
		opnsav-default-dir
		Specifies the initial directory.
		
		opnsav-basename
		Not used.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	The next item has been returned.
-1	No more items available.

OPENSAVE-SAVE-BOX-CHECKED

The OPENSAVE-SAVE-BOX-CHECKED function opens a "Save as" dialog box. If the user chooses the name of a file that already exists, the dialog box shows a message box asking what to do. The routine returns successfully only if the user provided the name of a file that doesn't exist or chose to overwrite an existing file.

Syntax:

```
CALL "C$OPENSAVEBOX" USING OPENSAVE-SAVE-BOX-CHECKED
                           OPENSAVE-DATA
                     GIVING returnCode
```

Parameters:

OPENSAVE-SAVE-BOX-CHECKED	Constant

OPENSAVE-DATA	Group Item	It is a structure that contains information used to initialize the dialog box. When the OPENSAVE-SAVE-BOX-CHECKED function returns successfully, this structure contains information about the user's file selection. This group item, defined in isopensave.def, has the following structure:
		
		`01 opensave-data.`
		`  03 opnsav-filename pic x(256).`
		`  03 opnsav-flags pic 9(4) comp-x value 0.`
		`  03 opnsav-default-ext pic x(12).`
		`  03 opnsav-title pic x(80).`
		`  03 opnsav-filters pic x(512).`
		`  03 opnsav-default-filter pic 9(4) comp-x value 0.`
		`  03 opnsav-default-dir pic x(128).`
		`  03 opnsav-basename pic x(128).`
		
		opnsav-filename
		Specifies the default filename. When the OPENSAVE-SAVE-BOX function returns successfully, it receives the drive designator, path, file name, and extension of the selected file.
		
		opnsav-flags
		Not used
		
		opnsav-default-ext
		This item holds the default file name extension. The extension is the string of characters that appear after the "." in the file name. The value of OPNSAV-DEFAULT-EXT is added to the file name typed by the user, if the user does not type an extension. The default extension should not include the period ".". Set this item to spaces to avoid having a default extension.
		
		opnsav-title
		String to be placed in the title bar of the dialog box. If this member is spaces, the system uses the default title.
		
		opnsav-filters
		The value of OPNSAV-FILTERS describes the set of filters that the dialog box will use to restrict the set of files shown to the user. Filters make it easier for a user to navigate through a large directory by limiting the files shown at once.
		
		Each filter consists of a pair of descriptors. These descriptors are separated by a vertical bar character ("|").
		
		Here is a sample OPNSAV-FILTERS setting that contains two filters:
		
		"isCOBOL source files (*.cbl)|*.cbl|All files (*.*)|*.*"
		
		opnsav-default-filter
		This item is used in conjunction with OPNSAV-FILTERS. The value of OPNSAV-DEFAULT-FILTER determines which of the given filters to use as the initial filter. A value of "1" selects the first filter pair, "2" selects the second pair, and so on. A value of zero also selects the first pair. This setting is not used if no filters are defined.

	opnsav-default-dir Specifies the initial directory. *opnsav-basename* When the routine returns, this item contains the base file name of the file chosen by the user. This differs from the value of OPNSAV-FILENAME in that all directory information is removed, leaving only the file name.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful. File has been chosen.
-1	The user has pressed the "Cancel" button. opensaveData contents are not updated.

C$PARAMSIZE

The C$PARAMSIZE library routine retrieves the LENGTH of a parameter passed by the caller.

Note: This routine cannot be used in the Procedure Division of a method. It returns information only on parameters passed by CALL, not by INVOKE.

Syntax:

```
CALL "C$PARAMSIZE" USING paramNum
                   GIVING paramSize
```

Parameters:

paramNum	any numeric data item	Specifies the ordinal number of the parameter you want to get the size. The first parameter in the USING list is identified by 1.

Return code:

paramSize can be any numeric data item. It receives the size of the parameter.

Examples:

Example - A program that may receive a variable size string to process it

```
working-storage section.
77 str-size pic 9(5).

linkage section.
01 the-string  pic x(512).

procedure division using the-string.
main.
  call "c$paramsize" using 1 giving str-size
  inspect the-string(1:str-size) replacing all "," by "|"
  goback.
```

C$PARSEEFD

The C$PARSEEFD routine is used to parse EFD files and retrieve information about them, giving you a way to map field description information to file record areas.

Syntax:

```
CALL "C$PARSEEFD" USING opCode
                        parameters
                  GIVING returnCode
```

Parameters:

opCode	It is the function to be executed. Valid values, defined in isparseefd.def, are:	
	PARSEEFD-PARSE	parse EFD file
	PARSEEFD-GET-KEY-INFO	Retrieve key information.
	PARSEEFD-GET-COND-INFO	Retrieve condition information.
	PARSEEFD-GET-FIELD-INFO	Retrieve field information.
	PARSEEFD-TEST-CONDITIONS	Test record conditions.
	PARSEEFD-RELEASE	Release EFD file from memory .
parameters	Parameters depend on the opcode.	

Return code:

returnCode definition and meaning depend on the opcode.

Examples:

Example - Display all fields defined in the *file1.xml* data dictionary:

```
working-storage section.
copy "isparseefd.def".
77 efd-handle handle.
77 i          pic 9(5).
77 xml-file   pic x(128).
77 data-file  pic x(128).
77 flags      pic 9 value 0.

procedure division.
main.
   move "file1.xml" to xml-file.
   call "c$parseefd" using parseefd-parse
                           xml-file
                           data-file
                           flags
                           parseefd-description
                    giving efd-handle.
   if efd-handle < 1
      display "Error: invalid EFD file"
      goback
   end-if.
   perform varying i from 0 by 1
                     until i =  parseefd-number-fields
      call "c$parseefd" using parseefd-get-field-info
                              efd-handle
                              i
                              parseefd-field-description
      display parseefd-field-name
   end-perform.
   call "c$parseefd" using parseefd-release
                           efd-handle.
   goback.
```

PARSEEFD-PARSE

The PARSEEFD-PARSE function parses a specified EFD file.

Syntax:

```
CALL "C$PARSEEFD" USING PARSEEFD-PARSE
                        efdName
                        fileName
                        flags
                        PARSEEFD-DESCRIPTION
                GIVING returnCode
```

Parameters:

PARSEEFD-PARSE	Constant	
efdName	PIC X(n)	Specifies the name of the EFD file to parse, with or without path information.
fileName	PIC X(n)	Specifies an indexed data file to be compared against the parsed EFD file. If the characteristics of the specified data file do not match the EFD, the parsed EFD is freed and the return-value is set to 0. If this parameter is empty, the EFD file is not compared to any file.

flags	PIC 9(n)	Specifies the type of information that will be returned from other op-codes. This parameter can be 0 (if no flags are set), or the sum of any of the following values defined in isparseefd.def:
		• PARSEEFD-FLAG-INCLUDE-COMMENTS - This option causes comments to be included in the parsed EFD. The routine cannot, however, currently retrieve those comments.
		• PARSEFD-FLAG-INCLUDE-999 - This option includes fields with a condition code of 999, which indicates group items and other fields not normally included with EFD files.
		• PARSEEFD-FLAG-EXCLUDE-ARRAYS - All table elements are normally appended with a value indicating their index. For example, for a field that occurs five times, the returned EFD includes five fields with _1, _2, _3, _4, and _5 appended to the field names. When this flag is set, such fields are returned with no suffix indicating their array index value. The information is still included, however, with the field group item (see below).
		• PARSEEFD-FLAG-DEEP-FIRST - This flag modifies the order in which fields that are sub-elements of a table are returned. For example:
		`07 file1-array occurs 3 times.`
		`09 elem-1 pic x.`
		`09 elem-2 pic x.`
		`09 elem-3 pic x.`
		Normally this is returned as elem-1(1), elem-1(2), elem-1(3), elem-2(1), elem-2(2), elem-2(3), elem-3(1), elem-3(2) and elem-3(3). If PARSEEFD-FLAG-DEEP-FIRST is specified, the items are instead returned as elem-1(1), elem-2(1), elem-3(1), elem-1(2), elem-2(2), elem-3(2), elem-1(3), elem-2(3) and elem-3(3). The same data is returned, but in a different order.

PARSEEFD-DESCRIPTION	Group Item	Structure that contains the information returned by the function. This group item, defined in isparseefd.def, has the following structure:

```
01  parseefd-description.
    03  parseefd-header-line.
        05  parseefd-version                pic x comp-n.
        05  parseefd-select-name            pic x(30).
        05  parseefd-filename               pic x(30).
        05  parseefd-filetype               pic x comp-n.
            88  parseefd-sequential-file       value 4.
            88  parseefd-relative-file         value 8.
            88  parseefd-indexed-file          value 12.
    03  parseefd-record-line.
        05  parseefd-max-rec-size           pic x(4) comp-n.
        05  parseefd-min-rec-size           pic x(4) comp-n.
        05  parseefd-num-keys               pic x comp-n.
    03  parseefd-condition-line.
        05  parseefd-number-conditions      pic xx comp-n.
    03  parseefd-fields-line.
        05  parseefd-number-fields          pic x(4) comp-n.
    03  parseefd-v6-information.
        05  parseefd-cobol-trigger          pic x(100).
        05  parseefd-compile-line.
            07  parseefd-sign-flag          pic x(2) comp-n.
                88  parseefd-sign-acu          value 0.
                88  parseefd-sign-ibm          value 4.
                88  parseefd-sign-mf           value 8.
                88  parseefd-sign-ncr          value 20.
                88  parseefd-sign-vax          value 36.
                88  parseefd-sign-mbp          value 72.
                88  parseefd-sign-rea          value 128.
            07  parseefd-max-digits         pic x(2) comp-n.
                88  parseefd-18-digits         value 40.
                88  parseefd-31-digits         value 68.
            07  parseefd-pgm-period         pic x.
            07  parseefd-pgm-comma          pic x.
            07  parseefd-encoding           pic x(2) comp-n.
                88  parseefd_ascii             value 0.
                88  parseefd_wide              value 1.
                88  parseefd_utf-8             value 2.
                88  parseefd_utf-16-le         value 3.
                88  parseefd_utf-16-be         value 5.
                88  parseefd_utf-32-le         value 4.
                88  parseefd_utf-32-be         value 6.
```

	PARSEEFD-VERSION	Version number of this EFD file
	PARSEEFD-SELECT-NAME	Logical name of the file
	PARSEEFD-FILENAME	Physical name of the file
	PARSEEFD-FILETYPE	Data file type. Valid values are 4 (sequential file), 8 (relative file), and 12 (indexed file).
	PARSEEFD-COBOL-TRIGGER	Name of the COBOL program to be executed as a trigger, if any.
	PARSEEFD-MAX-REC-SIZE and MIN-REC-SIZE	Maximum and minimum size values for a record in this file.
	PARSEEFD-NUM-KEYS	Number of keys described in the EFD.
	PARSEEFD-SIGN-FLAG	Sign compatibility
	PARSEEFD-MAX-DIGITS	Maximum numeric digits.
	PARSEEFD-PGM-PERIOD	Decimal value of the character used as the program period.
	PARSEEFD-PGM-COMMA	Decimal value of the character used as the program comma.
	PARSEEFD-NUMBER-CONDITIONS	Number of conditions described in the EFD file.
	PARSEEFD-NUMBER-FIELDS	Number of fields available in the EFD.
	PARSEEFD-ENCODING	Encoding of the EFD file.

Return code:

returnCode can be any signed numeric data item. For this function, the return value is the handle to the EFD. This handle must be used in future calls to C$PARSEEFD to get more information about the EFD, and to free the EFD when you are finished.

If *returnCode* is 0, an error occurred. You can get information about errors by examining f-errno and f-int-errno external data items, which are defined in the isfilesys.def copybook.

PARSEEFD-GET-KEY-INFO

The PARSEEFD-GET-KEY-INFO function retrieves information about a key.

Syntax:

```
CALL "C$PARSEEFD" USING PARSEEFD-GET-KEY-INFO
                        efdHandle
                        keyNum
                        PARSEEFD-KEY-DESCRIPTION
                GIVING returnCode
```

Parameters:

PARSEEFD-GET-KEY-INFO	Constant	
efdHandle	USAGE HANDLE	Handle returned by the PARSEEFD-PARSE function.
keyNum	PIC 9(n)	Specifies the key number. A value of 0 references the primary key, a value of 1 references the first alternate key, a value of 2 references the second alternate key, and so on.
PARSEEFD-KEY-DESCRIPTION	Group Item	Structure that contains the information returned by the function. This group item, defined in isparseefd.def, has the following structure:

```
01  parseefd-key-description.
    03  parseefd-number-segments              pic x comp-n.
    03  parseefd-dup-flag                     pic x comp-n.
        88  parseefd-allow-duplicates         value 1 false 0.
    03  parseefd-segment-description
            occurs max-segs times
            indexed by parseefd-seg-idx.
        05  parseefd-segment-length           pic x comp-n.
        05  parseefd-segment-offset           pic x(4) comp-n.
    03  parseefd-number-key-fields            pic x comp-n.
    03  parseefd-key-fields
            occurs parseefd-maxnumkeyfields times
            indexed by parseefd-key-field-idx.
        05  parseefd-key-field-num            pic xx comp-n.
```

PARSEEFD-NUMBER-SEGMENTS	Number of segments in this key.
PARSEEFD-DUP-FLAG	A value of 1 indicates that duplicates are allowed; a value of 0 indicates no duplicates.
PARSEEFD-SEGMENT-LENGTH and SEGMENT-OFFSET	Length and offset of each segment. The offset value is zero-based, so offset 0 is the beginning of the record. There is one SEGMENT-LENGTH and SEGMENT-OFFSET value for each segment.
PARSEEFD-NUMBER-KEY-FIELDS	Number of fields that make up this key. This is always at least as large as the number of segments, but may be larger if a segment holds multiple fields.
PARSEEFD-KEY-FIELDS	Table of key fields. This table has PARSEEFD-NUMBER-KEY-FIELDS valid elements.
PARSEEFD-KEY-FIELD-NUM	Nmber of this key field. Get information about the key field by looking at this field number.

Return code:

returnCode can be any signed numeric data item and provides additional information:

0	Operation successful.
1	Operation failed. Either the EFD handle or the key number is invalid.

PARSEEFD-GET-COND-INFO

The PARSEEFD-GET-COND-INFO function retrieves information about conditions that use the WHEN Directive within the EFD file

Syntax:

```
CALL "C$PARSEEFD" USING PARSEEFD-GET-COND-INFO
                        efdHandle
                        condIndex
                        PARSEEFD-CONDITION-DESCRIPTION
                GIVING returnCode
```

Parameters:

PARSEEFD-GET-COND-INFO	Constant	
efdHandle	USAGE HANDLE	Handle returned by the PARSEEFD-PARSE function.
condIndex	PIC 9(n)	Specifies the condition number. A value of 0 references the first condition, a value of 1 references the second condition, a value of 2 references the third condition, and so on.
PARSEEFD-CONDITION-DESCRIPTION	Group Item	Structure that contains the information returned by the function. This group item, defined in isparseefd.def, has the following structure:

```
01  parseefd-condition-description.
    03  parseefd-condition-type                pic x comp-n.
        88  parseefd-equal-condition            value 1.
        88  parseefd-and-condition              value 2.
        88  parseefd-other-condition            value 3.
        88  parseefd-gt-condition               value 4.
        88  parseefd-ge-condition               value 5.
        88  parseefd-lt-condition               value 6.
        88  parseefd-le-condition               value 7.
        88  parseefd-ne-condition               value 8.
        88  parseefd-or-condition               value 9.
        88  parseefd-comparison-cond    values 1, 4 through 8.
    03  parseefd-condition-flag                pic x.
        88  parseefd-true-condition       value 'Y' false 'N'.
    03  parseefd-comparison-conditions.
        05  parseefd-comp-fieldnum             pic xx comp-n.
        05  parseefd-comp-fieldname            pic x(30).
        05  parseefd-comp-field-val            pic x(50).
    03  parseefd-other-conditions
                redefines parseefd-comparison-conditions.
        05  parseefd-other-fieldnum            pic xx comp-n.
        05  parseefd-other-fieldname           pic x(30).
    03  parseefd-and-or-conditions
                redefines parseefd-comparison-conditions.
        05  parseefd-condition-1               pic xx comp-n.
        05  parseefd-condition-2               pic xx comp-n.
    03  parseefd-condition-tablename           pic x(30).
```

PARSEEFD-CONDITION-TYPE	Condition type: EQUAL condition, AND condition, etc.
PARSEEFD-CONDITION-FLAG	Tells whether this condition is TRUE. This is only valid after PARSEEFD-TEST-CONDITIONS has been called.
PARSEEFD-CONDITION-TABLENAME	Table name specified in the TABLENAME clause of the WHEN Directive.
For EQUAL, GT (greater than), GE (greater than or equal to), LT (less than), LE (less than or equal to), and NE (not equal to) conditions, the following fields are valid:	
PARSEEFD-COMP-FIELDNUM	Field number of the field whose value will be tested against the value of the condition.
PARSEEFD-COMP-FIELDNAME	Name of that field.
PARSEEFD-COMP-FIELD-VAL	Value to be tested. This is the value specified in the WHEN Directive of the FD used to create this EFD.
For OTHER conditions, the following fields are valid:	
PARSEEFD-OTHER-FIELDNUM	Field number of the field whose value will be different than all the other conditions which use this field.
PARSEEFD-OTHER-FIELDNAME	Name of that field.
For AND and OR conditions, the following fields are valid:	
PARSEEFD-CONDITION-1 and PARSEEFD-CONDITION-2	conditions tested to determine whether this condition is true. For AND, both conditions must be true. For OR, one or both conditions must be true.

Return code:

returnCode can be any signed numeric data item and provides additional information:

0	Operation successful.
1	Operation failed. Either the EFD handle or the condition number is invalid.

PARSEEFD-GET-FIELD-INFO

The PARSEEFD-GET-FIELD-INFO function retrieves information about the field.

Syntax:

```
CALL "C$PARSEEFD" USING PARSEEFD-GET-FIELD-INFO
                        efdHandle
                        fieldNum
                        PARSEEFD-FIELD-DESCRIPTION
                 GIVING returnCode
```

Parameters:

PARSEEFD-GET-FIELD-INFO	Constant	
efdHandle	USAGE HANDLE	Handle returned by the PARSEEFD-PARSE function.
fieldNum	PIC 9(n)	Specifies the field number. A value of 0 references the first field, a value of 1 references the second field, a value of 2 references the third field, and so on.
PARSEEFD-FIELD-DESCRIPTION	Group Item	Structure that contains the information returned by the function. This group item, defined in isparseefd.def, has the following structure:

```
01  parseefd-field-description.
    03  parseefd-field-offset                pic x(4) comp-n.
    03  parseefd-field-length                pic x(4) comp-n.
    03  parseefd-field-type                  pic x comp-n.
        88  parseefd-signed-field  values parseefd-numsignsep
                                          parseefd-numsigned
                                          parseefd-numseplead
                                          parseefd-numleading
                                          parseefd-compsigned
                                          parseefd-packedsigned
                                         parseefd-binarysigned
                                         parseefd-nativesigned.
        88  parseefd-num-field     values
              parseefd-numedited thru parseefd-nativeunsigned.
        88  parseefd-float-field   value  parseefd-flt.
        88  parseefd-ascii-field   values
                        parseefd-alphanum thru parseefd-group.
        88  parseefd-nat-field     values
                  parseefd-nat-type thru parseefd-natedited.
        88  parseefd-wide-field    values
                 parseefd-wide-type thru parseefd-wideedited.
    03  parseefd-field-digits                pic x comp-n.
    03  parseefd-field-scale                 signed-short.
    03  parseefd-field-user-type             pic xx comp-n.
    03  parseefd-field-condition             pic xx comp-n.
    03  parseefd-field-level                 pic x comp-n.
    03  parseefd-field-name                  pic x(30).
    03  parseefd-field-format                pic x(30).
    03  parseefd-field-occurs-depth          pic x comp-n.
    03  parseefd-field-occurs-table
        occurs parseefd-maxnumkeyfields times
        indexed by parseefd-field-occurs-level.
        05  parseefd-field-occ-max-idx       pic xx comp-n.
        05  parseefd-field-occ-this-idx      pic xx comp-n.
    03  parseefd-field-in-key-flag           pic x.
        88  parseefd-field-is-in-key      value 'Y' false 'N'.
    03  parseefd-field-secondary-flag        pic x.
        88  parseefd-field-is-secondary   value 'Y' false 'N'.
    03  parseefd-field-hidden-flag           pic x.
        88  parseefd-field-is-hidden      value 'Y' false 'N'.
    03  parseefd-field-read-only-flag        pic x.
        88  parseefd-field-is-read-only   value 'Y' false 'N'.
    03  parseefd-field-dbtype                pic x(32).
    03  parseefd-field-type-description      pic x(30).
```

PARSEEFD-FIELD-OFFSET	Offset of the field in the record buffer (zero-based).
PARSEEFD-FIELD-LENGTH	Number of bytes this field requires.
PARSEEFD-FIELD-TYPE	Numeric representation of the field type. The types are defined in isparseefd.def.
PARSEEFD-FIELD-DIGITS	Either the number of digits in this numeric field, or the length if the field is non-numeric.
PARSEEFD-FIELD-SCALE	Either the scale of the numeric field or 0 if the field is non-numeric. The scale is defined as the power of ten by which the numeric value must be multiplied in order to get the actual value. For example, if the scale is -2, then there are two digits to the right of the decimal point.
PARSEEFD-USER-TYPE	Describes some of the EFD directives, as listed in isparseefd.def.
PARSEEFD-FIELD-CONDITION	Condition that the field depends on. A condition of 0 means that the field is always included; 999 means that the field will never be included.
PARSEEFD-FIELD-LEVEL	Level number of the field in the FD used to create this EFD.
PARSEEFD-FIELD-NAME	Name of the field. If EXCLUDE-ARRAYS was NOT used when parsing the EFD, and the field is part of a table, then the field name may include array indices.
PARSEEFD-FIELD-FORMAT	Date format specified in the DATE Directive
PARSEEFD-FIELD-OCCURS-DEPTH	Number of valid elements in the OCCURS-TABLE.
PARSEEFD-FIELD-OCCURS-TABLE	Information about the element of a table. The OCC-MAX-IDX is the maximum index allowed. The OCC-THIS-IDX is the index of this element.
PARSEEFD-FIELD-IN-KEY-FLAG	Indicates whether this field is part of a key. The value is Y if this field is a part of one or more keys, or N if not.
PARSEEFD-FIELD-SECONDARY-FLAG	Reserved for future use.
PARSEEFD-FIELD-HIDDEN-FLAG	Indicates whether the HIDDEN Directive was used on this field. The value is Y if so or N if not.
PARSEEFD-FIELD-READ-ONLY-FLAG	Indicates whether the READ-ONLY Directive was used on this field.
PARSEEFD-FIELD-DBTYPE	Alphanumeric representation of the type that will be used on the database for this field. This value is space if no EFD directives were used for this field.
PARSEEFD-FIELD-TYPE-DESCRIPTION	Alphanumeric representation of the field type.

Return code:

returnCode can be any signed numeric data item and provides additional information:

0	Operation successful.
1	Operation failed. Either the EFD handle or the field number is invalid.

PARSEEFD-TEST-CONDITIONS

The PARSEEFD-TEST-CONDITIONS function tests the conditions of a particular record.

Syntax:

```
CALL "C$PARSEEFD" USING PARSEEFD-TEST-CONDITIONS
                        efdHandle
                        recordBuffer
```

Parameters:

PARSEEFD-TEST-CONDITIONS	Constant	
efdHandle	USAGE HANDLE	Handle returned by the PARSEEFD-PARSE function.
recordBuffer	USAGE POINTER	Pointer to the record area on which to test conditions.

After calling this function, you can get each condition with PARSEEFD-GET-COND-INFO and tell whether fields that depend on that condition should be included in this record (PARSEEFD-CONDITION-FLAG will be "Y" if fields that depend on that condition should be included in this record).

PARSEEFD-RELEASE

The PARSEEFD-RELEASE function frees all memory associated with the EFD.

Syntax:

```
CALL "C$PARSEEFD" USING PARSEEFD-RELEASE
                        efdHandle
```

Parameters:

PARSEEFD-RELEASE	Constant	
efdHandle	USAGE HANDLE	Handle returned by the PARSEEFD-PARSE function.

C$PRELOAD

The C$PRELOAD routine runs asynchronously and instantiates all the COBOL programs contained in a given jar library or folder. This will speed up subsequent calls to these programs.

Note - only programs with a PROGRAM-ID are considered by this routine. Programs with a CLASS-ID are discarded, instead.

Syntax:

```
CALL "C$PRELOAD" USING classesContainer
                 GIVING returnCode
```

Parameters:

classesContainer	PIC X(n)	Specifies the location of the class files to be preloaded. It can be either a folder of classes or one jar file name. If a folder is passed, • The folder must be available either in the CLASSPATH or in the iscobol.code_prefix. If a jar file file is passed, • Use only the name of the file. If you include the path information, an error will be returned. • The CLASSPATH is parsed to find the jar, iscobol.code_prefix is not considered. Both the path and the name of the jar must be in the CLASSPATH.

Return code:

returnCode can be any signed numeric data item and provides additional information:

0	Error occurred. The classes container is either not found or invalid.
1	The classes are going to be preloaded.
2	The preloading is in progress.
3	The classes have been successfully preloaded.

The following constants are available in the iscobol.def copybook:

```
78  cprel-error                                 value 0.
78  cprel-starting                              value 1.
78  cprel-in-progress                           value 2.
78  cprel-completed                             value 3.
```

Examples:

Example - Load myApp.jar in memory to speed up subsequent calls to myApp classes. Note: The path to the jar with the jar name must be found in the Classpath

```
call "c$preload" using "myApp.jar".
```

C$PROGINMEM

This routine returns if a cobol program is already loaded in memory or not.

Syntax:

```
CALL "C$PROGINMEM" USING progName
                   GIVING returnCode
```

Parameters:

progName	PIC X(n)	Specifies the program to check.

Return code:

returnCode can be any signed numeric data item and provides additional information:

-1	wrong parameters.
0	program not loaded.
1	program already loaded.

Examples:

Example - Check if a program is loaded in memory

```
call "c$proginmem" using "computetax"
if return-code not = 1
   display message "ComputeTax is not in memory"
end-if
```

C$RBMENU

The C$RBMENU displays a pop-up menu in the current window when the right mouse button is pressed.

Note - This routine is supported for compatibility. If you're creating new programs with isCOBOL, you should consider using W$MENU instead.

Syntax:

```
CALL "C$RBMENU" [ USING text-1 [, text-2] ... [, text-n] [, exitCode]]
```

Parameters:

text-1 *text-2* ... *text-n*	Any alphanumeric data item or literal	These parameters define the items in the pulldown menu along with the strings that are returned to the COBOL program when an item is selected. Using the following syntax, it can specify either a value to be returned or additional sub-menu items: itemname["hint"]=menu where *itemname* is the string that is displayed in the menu bar. *hint* is an optional text string that is displayed as tool tip when the mouse hovers over the menu item. *menu* defines the items in the pulldown menu along with the key codes that are returned to the COBOL program when an item is selected. The syntax for *menu* is: menu -> [(]items[)] items -> item name=[keycode | (menu)][,items] item name -> pulldownname["menu prompt"] keycode -> key code to be sent (see RM/COBOL key codes for C$MBAR, C$RBMENU and C$TBAR routines for details) If the first character of *itemname* is a tilde (~), the menu is disabled. An ampersand (&) in *itemname* causes the next character to be underlined and used as an accelerator.
exitCode	PIC 9(n) BINARY	Optional parameter that returns the status. It will receive zero for success and non-zero for failure.

Calling the routine with no arguments turn off the pop-up menu.

Examples:

Example - Create a pop-up menu with three options

```
CALL "C$RBMENU" USING "Edit=(Cut=\f1,Copy=\f2,Paste=\f3)"
```

C$REPLACE_ALL

The C$REPLACE_ALL library routine allows to replace parts of text in an alphanumeric variable according to a regular expression.

Syntax

```
CALL "C$REPLACE_ALL" USING variable
                           regExp
                           replacement
                          [options]
                          [errorDesc]
                    GIVING returnCode
```

Parameters

variable	PIC X(n)	Variable subjected to replacement
regExp	PIC X(n)	Regular expression
replacement	PIC X(n)	Text to use as replacement. This text is not trimmed, so trailing spaces will be used as replacement as well.
options	PIC 9(n)	Optional parameter. It's the sum between two or more of the following values defined in iscobol.def: CREP_CASE_INSENSITIVE CREP_LEFT_TRIMMED CREP_RIGHT_TRIMMED These options affect only *variable*.
errorDesc	PIC X(n)	Optional parameter. It holds the error description in case of illegal regular expression.

Return code:

returnCode can be any signed numeric data item and provides additional information:

0	operation successful
1	invalid regular expression
2	insufficient parameters
3	malformed replacement

Examples:

Example - Replace the beginning of a string with a prefix

```
*> copy "iscobol.def" in working-storage
*> it's good practice to define the involved alphanumeric data items as pic x any
length.

move " second third fourth" to wrk-str
move "^" to search-reg-exp | Search for the beginning of the string
move "first" to new-text
call "c$replace-all" using wrk-str
                           search-reg-exp
                           new-text
                           crep-right-trimmed
                           W-error
*> wrk-str new value will be: first second third fourth
```

C$RERR

The C$RERR library routine retrieves extended information about the status of a file after an I/O operation.

Syntax:

```
CALL "C$RERR" USING extendStat
                    [textMessage]
```

Parameters:

extendStat	any alphanumeric data item	Specifies the extended file status. The first two characters of the extended file status are identical to the normal FILE STATUS value returned by the runtime for a file operation. The last two characters further clarify the reason for the particular FILE STATUS value. The values used here are listed in File Status Codes. In thin client environment, having *iscobol.file.lock_manager=com.iscobol.as.locking.InternalLockManager*, when a lock timeout error occurs, *extendStat* is set to a concatenation of the file status and the thread ID of the client holding the lock. The thread ID is stored in *extendStat(3:)*. Note that a thread ID can be up to 10 digits long. If a "file locked" error occurs during a OPEN I-O WITH LOCK due to another client that opened the file for input, then TID=0 is returned, because the OPEN INPUT doesn't put a real lock and it's also possible to open for input the same file from many clients, so the runtime doesn't know which TID to return. Due to the variable lenght of this value, you should consider to define this parameter as PIC X ANY LENGTH.
textMessage	any alphanumeric data item	Optional. It contains a text message coming from the host system if available.

Examples:

Example - Display extended information on a file error

```
*> define crerr-status and rerrname as pic x(n)

display-file-error.
  call "C$RERR"     using crerr-status
  call "C$RERRNAME" using rerrname

  display message "Error " file-status crerr-status " on " rerrname
```

C$RERRNAME

The C$RERRNAME library routine retrieves the name of the latest file used by a COBOL I/O statement.

Syntax:

```
CALL "C$RERRNAME" USING fileName
```

Parameters:

fileName	PIC X(n)	Receives the full pathname of the latest file used by a COBOL I/O statement, followed by its logical name enclosed in parentheses.

Examples:

Example - Display extended information on a file error

```
*> define crerr-status and rerrname as pic x(n)

display-file-error.
  call "C$RERR"     using crerr-status
  call "C$RERRNAME" using rerrname

  display message "Error " file-status crerr-status " on " rerrname
```

C$RUN

The library routine runs a program asynchronously. If you want to run a program synchronously, use the SYSTEM library routine.

Syntax:

```
CALL "C$RUN" USING commandLine
             GIVING returnCode
```

Parameters:

commandLine	PIC X(n)	Specifies the operating system command to be executed.

Return code:

returnCode can be any signed numeric data item and provides additional information:

-1	Operation failed.
0	Operation succeeded.

Examples:

Example - Run a external program asynchronously

```
call "c$run" using "notepad.exe"
```

C$SBAR

The C$SCRD library routine displays a status bar made by only 1 panel in which the program can display text. The status bar doesn't use the last line of the window's client area, but it's appended to the bottom of the window increasing the window's height.

Note - This routine is supported for compatibility. If you're creating new programs with isCOBOL, you should consider using the STATUS-BAR control instead.

Syntax:

```
CALL "C$SBAR" [ USING statusText [, exitCode] ]
```

Parameters:

statusText	Any alphanumeric data item or literal	Specifies the text to be displayed in the status bar.
exitCode	PIC 9(n) BINARY	Optional parameter that returns the status. It will receive zero for success and non-zero for failure.

Calling this routine with no arguments turns off the status bar.

Examples:

Example - Display the message "Ready" in the status bar

```
call "c$sbar" using "Ready".
```

C$SCRD

The C$SCRD library routine allows the contents of the screen to be read into an alphanumeric data item.

Syntax:

```
CALL "C$SCRD" USING screenBufer
                    [bufferSize]
                    [screenLine]
                    [screenPos]
```

Parameters:

screenBuffer	PIC X(n)	Receives the characters read from the terminal display screen
bufferSize	PIC 9 COMP-1	Specifies the number of characters to be read. If the value is 0 or the parameter is omitted, the actual size of *screenBuffer* is used
screenLine	PIC 9 COMP-1	Specifies the line where the cursor is to be placed prior to the screen read. If omitted, a value of 1 is used.
screenPos	PIC 9 COMP-1	Specifies the position where the cursor is to be placed prior to the screen read. If omitted, a value of 1 is used.

Examples:

Example - Read 3 characters from the screen at line 2 position 2

```
move 3 to buffer-size.
move 2 to w-line.
move 2 to w-col.
call "c$scrd" using buffer,
                    buffer-size,
                    w-line, w-col.
```

C$SCWR

The C$SWR library routine allows to display quickly a large amount of information containing multiple display attributes.

Syntax:

```
CALL "C$SCWR" USING displayDesc
                    textCharacters
```

Parameters:

displayDesc	Group item	Group data item defined as follows:
		
		`01 DISPLAY-DESCRIPTION BINARY(2).`
		`  03 DISPLAY-VERSION PIC 9(4) VALUE 0.`
		`  03 DISPLAY-UNIT PIC 9(4) VALUE 0.`
		`  03 DISPLAY-LINE PIC 9(4).`
		`  03 DISPLAY-POSITION PIC 9(4).`
		`  03 DISPLAY-LENGTH PIC 9(4).`
		`  03 DISPLAY-EXCEPTION-CODE PIC 9(4).`
		`  03 DISPLAY-EXCEPTION-VALUE PIC 9(4).`
		
		DISPLAY-VERSION must be set to a value of 0.
		DISPLAY-UNIT is the unit number of the terminal to which the display is directed. This value must be set to 0.
		DISPLAY-LINE is the one-relative line number in the current window where the text is to be displayed. If set to a value of 0, the display begins on the current line.
		DISPLAY-POSITION is the one-relative column number in the current window where the text is to be displayed. If set to a value of 0, the display begins at the current column.
		DISPLAY-LENGTH is the number of characters of text to display. If set to a value of 0 or omitted, the actual length of text-characters is used.
		DISPLAY-EXCEPTION-CODE is set to a value of 0 if this function succeeds. Otherwise, one of the below exception codes is returned.
		DISPLAY-EXCEPTION-VALUE is set to a value of 0 if this function succeeds. Otherwise, one of the below exception codes is returned.
		
		Exception codes:
		
		1: Invalid DISPLAY-VERSION. This data item must be set to a value of 0.
		2: Invalid or missing parameters.
		3: DISPLAY-LINE is greater than the number of lines on the window or screen.
		4: DISPLAY-POSITION is greater than the number of columns on the screen.

textCharacters	PIC X(n)	Specifies the characters to be displayed.

Examples:

Example - Display “hello” at line 2 position 2

```
move 2 to display-line.
move 2 to display-position.
move "hello" to text-buffer.
call "c$scwr" using display-description,
                    text-buffer.
```

C$SETDEVELOPMENTMODE

The C$SETDEVELOPMENTMODE library routine is supported for RM/COBOL compatibility but it has currently no effect.

Syntax:

```
CALL "C$SETDEVELOPMENTMODE"
```

C$SETENV

The C$SETENV library routine sets a configuration property.

Syntax:

```
CALL "C$SETENV" USING propertyName
                      propertyValue
```

Parameters:

propertyName	PIC X(n) or string literal.	Name of the property.
propertyValue	PIC X(n) or string literal	Value of the property.

Examples:

Example - Set the value of the file.prefix configuration property

```
*> define varname and varvalue as pic x(n)

move "file.prefix" to varname
move "c:/app1/dat1;c:/app1/dat2"
call "c$setenv" using varname varvalue
```

C$SHOW

The C$SHOW library routine allows to show or hide the current window.

Syntax:

```
CALL "C$SHOW" USING flag
```

Parameters:

flag	any numeric data item or numeric literal	Specifies the visible state. Set this parameter to zero to hide the current window. Set this parameter to a value greater than zero to make the current window visible.

Examples:

Example - Hide the current window

```
call "c$show" using 0.
```

C$SLEEP

The C$SLEEP library routine suspends the running program without using CPU resources.

Syntax:

```
CALL "C$SLEEP" USING seconds
```

Parameters:

seconds	Unsigned fixed-point numeric parameter, or an alphanumeric data item containing an unsigned fixed-point number.	Specifies the time to wait before the program execution is resumed. Decimal values are allowed. If the parameter is alphanumeric, both comma and dot decimal separators are accepted. If the parameter is numeric, the decimal separator specified in the SPECIAL-NAMES should be used, instead.

Examples:

Example - Put the program to sleep for half second

```
call "c$sleep" using 0.5
```

C$SOCKET

The C$SOCKET library routine provides a number of functions to manage sockets.

Syntax:

```
CALL "C$SOCKET" USING opCode
                      parameters
                GIVING returnCode
```

Parameters:

opCode	Function to be executed. Valid values, defined in issocket.def, are:	
	CSOCKET-CREATE-SERVER	Creates a new server socket on specific port
	CSOCKET-ACCEPT	Accepts connections
	CSOCKET-CREATE-CLIENT	Creates a new client connecting to a server socket
	CSOCKET-CLOSE	Closes a socket
	CSOCKET-WRITE	Writes data into socket
	CSOCKET-READ	Read data from socket
	CSOCKET-READ	Flush socket data
	CSOCKET-EMPTY	Empty socket
	CSOCKET-GETHOSTNAME	Returns the name of the pc hosting the socket
	CSOCKET-REMOTE-NAME	Returns the machine name associated to the socket
	CSOCKET-REMOTE-ADDR	Returns the IP address associated to the socket
	CSOCKET-GETREMOTEADDRESS	Returns machine name, IP and port associated to the socket
	CSOCKET-LAST-ERROR	Returns the error-code of last operation
	CSOCKET-NEXT-READ	Returns the next readable socket
parameters	Parameters depend on the opcode.	

Note: Acucobol-GT op-codes names having the prefix "AGS-" instead of "CSOCKET-" are supported for compatibility.

Return code:

returnCode can be any signed numeric data item. The meaning depends on the opcode.

Examples:

Example - Socket server program to listen messages from different clients and respond to them

```
       program-id. sockserver.

       working-storage section.
       copy "issocket.def".

       78  data-len         value 1024.
       77  sock-hnd-1       usage handle.
       77  sock-hnd-2       usage handle.
       77  client-data      pic x(data-len).
       77  bytes-read       pic s9999.
       77  sock-timeout     signed-int value -1.
       77  sock-thread      pic 9(4).

       procedure division chaining sock-thread.
       main.
           call "c$socket" using csocket-create-server, sock-thread
                           giving sock-hnd-1.
           if sock-hnd-1 = null
              perform exit-program
           end-if.

           perform until client-data(1:9) = "sockclose"
               call "c$socket" using csocket-next-read, sock-hnd-1,
                                     sock-timeout giving sock-hnd-2
               move sock-hnd-2 to return-code
               if return-code = -1
                   call "c$socket" using csocket-close, sock-hnd-1
                   call "c$sleep" using 4
                   go to main
               end-if
               if return-code = 0
                   exit perform
               end-if
               if sock-hnd-2 = sock-hnd-1
                   call "c$socket" using csocket-accept, sock-hnd-1
                   exit perform
               end-if
               call "c$socket" using csocket-read, sock-hnd-2,
                                     client-data, data-len
                               giving bytes-read
               if bytes-read = data-len
                   if client-data(1:9) not = "sockclose"
                       perform process-request
                   end-if
               else
                   if bytes-read = -1
                       call "c$socket" using csocket-close, sock-hnd-2
                   else
                       move "01 Resend data" to client-data
```

```
                    call "c$socket" using csocket-write, sock-hnd-2,
                                          client-data, data-len
                end-if
            end-if
        end-perform.
        perform exit-program.

    exit-program.
        goback.

    process-request.
         move "00 data ok" to client-data
         call "c$socket" using csocket-write, sock-hnd-2,
                               client-data, data-len.
```

CSOCKET-CREATE-SERVER

This operation creates a server-side socket.

Syntax:

```
CALL "C$SOCKET" USING CSOCKET-CREATE-SERVER
                      portNumber
                GIVING serverHandle
```

Parameters:

CSOCKET-CREATE-SERVER	Constant	
portNumber	PIC 9 (n)	Numeric value specifying the port on which the socket is created.
serverHandle	USAGE HANDLE	Handle of the server or zero if operation fails.

Note: the SO_REUSEADDR attribute is configurable through the property iscobol.csocket.reuseaddr (boolean) .

CSOCKET-ACCEPT

This operation waits for a connection from a client. It blocks other calls while waiting, and returns only after a client has attempted to connect.

Syntax:

```
CALL "C$SOCKET" USING CSOCKET-ACCEPT
                      serverHandle
                GIVING socketHandle
```

Parameters:

CSOCKET-ACCEPT	Constant	
serverHandle	USAGE HANDLE	Handle returned by a call from CSOCKET-CREATE-SERVER.

socketHandle	USAGE HANDLE	Handle of the first client that connects.

CSOCKET-CREATE-CLIENT

This operation attempts to connect to a server.

Syntax:

```
CALL "C$SOCKET" USING CSOCKET-CREATE-CLIENT
                      portNumber
                      serverName
               GIVING clientHandle
```

Parameters:

CSOCKET-CREATE-CLIENT	Constant	
portNumber	PIC 9 (n)	Numeric value specifying the port on which the socket is created.
serverName	PIC X(n)	The machine name or ip address of the server.
clientHandle	USAGE HANDLE	Handle of the client or zero if operation fails.

Note: the SO_KEEPALIVE attribute for the socket between client and server is configurable through the property iscobol.csocket.keepalive (boolean) .

CSOCKET-CLOSE

This operation closes a socket handle.

Syntax:

```
CALL "C$SOCKET" USING CSOCKET-CLOSE
                      socketHandle
```

Parameters:

CSOCKET-CLOSE	Constant	
socketHandle	USAGE HANDLE	Identifies an existing socket created by CSOCKET-CREATE-SERVER or CSOCKET-CREATE-CLIENT functions.

CSOCKET-WRITE

This operation writes data to a socket, either from the client to the server, or from the server to the client.

Syntax:

```
CALL "C$SOCKET" USING CSOCKET-WRITE
                      socketHandle
                      buffer
                      length
               GIVING returnCode
```

Parameters:

CSOCKET-WRITE	Constant	
socketHandle	USAGE HANDLE	Identifies the socket in which data will be written.
buffer	PIC X(n)	Data to write into socket.
length	PIC 9(n)	Optional. Number of bytes to write. A length of zero causes the entire buffer to be written into the socket.

Return code:

returnCode can be any signed numeric data item and provides additional information:

<0	Operation failed.
>0	Number of written bytes.

CSOCKET-READ

This operation reads data from a socket. It blocks other calls until all the data requested is actually read, an error occurs or the read timeout expires.

If the socket is closed by the other peer before the entire buffer is filled, C$SOCKET will return the number of bytes read to that point, which will be less than the amount requested. The next time CSOCKET-READ is called, -1 will be returned.

Syntax:

```
CALL "C$SOCKET" USING CSOCKET-READ
                      socketHandle
                      buffer
                      length
                      [timeout]
                GIVING return-code
```

Parameters:

CSOCKET-READ	Constant	
socketHandle	USAGE HANDLE	Identifies the socket to be read.
buffer	PIC X(n)	Data to read from socket.
length	PIC 9(n)	When set to zero, return cod e is set to the number of bytes available on the socket. After calling CSOCKET-READ with a length of 0, you can call CSOCKET-READ again with a length equal to the previous return value and be guaranteed not to block. When set to a value greater than zero, it specifies the number of bytes to be read. If the buffer passed is smaller than the value of this parameter, or if the number of bytes available on the socket is less than the value of this parameter, an error will result. Due to this rule, *buffer* picture can't be ANY LENGTH
timeout	PIC 9(n)	Optional parameter. It specifies the amount of milliseconds that C$SOCKET has to wait before returning if less data than the amount specified by *length* is available on the socket. When the timeout expires, *buffer* is set to the available data. If the parameter is omitted, then C$SOCKET waits until the number of bytes specified by *length* has been read.

Return code:

returnCode can be any signed numeric data item and provides additional information:

<0	Operation failed.
>0	Number of read bytes.

CSOCKET-READ-LINE

This operation reads a line of data from a socket. A line is defined as a block of characters terminated by either new line (0x0A) or carriage return (0x0D). It blocks other calls until all the data requested is actually read, an error occurs or the read timeout expires.

Syntax:

```
CALL "C$SOCKET" USING CSOCKET-READ-LINE
                      socketHandle
                      buffer
                      length
                      [timeout]
                GIVING return-code
```

Parameters:

CSOCKET-READ	Constant	
socketHandle	USAGE HANDLE	Identifies the socket to be read.
buffer	PIC X(n)	Data to read from socket.
length	PIC 9(n)	Number of bytes to be read. Setting this parameter to zero allows to know how many bytes are available on the socket. If the buffer passed is smaller than the value of this parameter, or if the number of bytes available on the socket is less than the value of this parameter, an error will result. Due to this rule, *buffer* picture can't be *x any length*.
timeout	PIC 9(n)	Optional parameter. It specifies the amount of milliseconds that C$SOCKET has to wait before returning if less data than the amount specified by *length* is available on the socket. When the timeout expires, *buffer* is set to the available data. If the parameter is omitted, then C$SOCKET waits until the number of bytes specified by *length* has been read.

Return code:

returnCode can be any signed numeric data item and provides additional information:

<0	Operation failed.
>0	Number of read bytes, not including the carriage return and line feed.

CSOCKET-FLUSH

This operation flushes any data in the socket.

Syntax:

```
CALL "C$SOCKET" USING CSOCKET-FLUSH
                      socketHandle
```

Parameters:

CSOCKET-FLUSH	Constant

socketHandle	USAGE HANDLE	Identifies the socket to be flushed.

CSOCKET-EMPTY

This operation is similar to CSOCKET-READ, except that the number of bytes is thrown away, rather than being stored.

Syntax:

```
CALL "C$SOCKET" USING CSOCKET-EMPTY
                      socketHandle
                      length
               GIVING return-code
```

Parameters:

CSOCKET-EMPTY	Constant	
socketHandle	USAGE HANDLE	Identifies the socket to be emptied.
length	PIC 9(n)	Number of bytes to be thrown away.

Return code:

returnCode can be any signed numeric data item and provides additional information:

<0	Operation failed.
>0	Number of skipped bytes.

CSOCKET-GETHOSTNAME

This operation allows the COBOL program to get the name of the host machine on which the COBOL program is executing.

Syntax:

```
CALL "C$SOCKET" USING CSOCKET-GETHOSTNAME
                      hostName
```

Parameters:

CSOCKET-GETHOSTNAME	Constant	
hostName	PIC X(n)	Returns the name of the host machine.

CSOCKET-REMOTE-NAME

This operation returns the name of a remote machine.

Syntax:

```
CALL "C$SOCKET" USING CSOCKET-REMOTE-NAME
                      socketHandle
                      remoteName
```

Parameters:

CSOCKET-REMOTE-NAME	Constant	
socketHandle	USAGE HANDLE	Identifies the socket to be inquired.
remoteName	PIC X(n)	Returns the name of the host machine.

CSOCKET-REMOTE-ADDR

This operation returns the IP address of a remote machine.

Syntax:

```
CALL "C$SOCKET" USING CSOCKET-REMOTE-ADDR
                      socketHandle
                      remoteName
```

Parameters:

CSOCKET-REMOTE-ADDR	Constant	
socketHandle	USAGE HANDLE	Identifies the socket to be inquired.
remoteName	PIC X(n)	Returns the IP address of the host machine.

CSOCKET-GETREMOTEADDRESS

This operation returns the hostname, IP address and port of the endpoint to which this socket is connected.

Syntax:

```
CALL "C$SOCKET" USING CSOCKET-GETREMOTEADDRESS
                      socketHandle
                      CSOCKET-REMOTE-ADDRESS
                GIVING returnCode
```

Parameters:

CSOCKET-GETREMOTEADDRESS	Constant	
socketHandle	USAGE HANDLE	Identifies the socket to be inquired.

CSOCKET-REMOTE-ADDRESS	Group item	Structure that receives the information. This group item, defined in issocket.def, has the following structure:
		
		`01 csocket-remote-address.`
		`  03 csocket-remote-hostname pic x(64).`
		`  03 csocket-remote-ip-address pic x(15).`
		`  03 csocket-remote-port pic 9(5).`

Return code:

returnCode can be any signed numeric data item and provides additional information:

0	Operation succeeded
1	Operation failed

CSOCKET-LAST-ERROR

This operation allows the COBOL program to determine the last error on a socket.

Syntax:

```
CALL "C$SOCKET" USING CSOCKET-LAST-ERROR
                      socketHandle
                     [textDescription]
                GIVING return-code
```

Parameters:

CSOCKET-LAST-ERROR	Constant	
socketHandle	USAGE HANDLE	Identifies the socket to inquire for errors. It can be NULL.
textDescription	PIC X(n)	Optional parameter. It receives the textual description of the error, if available.

Return code:

returnCode can be any signed numeric data item and contains the socket error number. To interpret this number, see third-party documentation about sockets:

0	No errors
10004	socket error EINTR
10009	socket error EBADF
10013	socket error EACCES
10014	socket error EFAULT
10022	socket error EINVAL
10024	socket error EMFILE
10035	socket error EWOULDBLOCK
10036	socket error EINPROGRESS
10037	socket error EALREADY
10038	socket error ENOTSOCK
10039	socket error EDESTADDRREQ
10040	socket error EMSGSIZE
10041	socket error EPROTOTYPE
10042	socket error ENOPROTOOPT
10043	socket error EPROTONOSUPPORT
10044	socket error ESOCKTNOSUPPORT
10045	socket error EOPNOTSUPP
10046	socket error EPFNOSUPPORT
10047	socket error EAFNOSUPPORT
10048	socket error EADDRINUSE

10049	socket error EADDRNOTAVAIL
10050	socket error ENETDOWN
10051	socket error ENETUNREACH
10052	socket error ENETRESET
10053	socket error ECONNABORTED
10054	socket error ECONNRESET
10055	socket error ENOBUFS
10056	socket error EISCONN
10057	socket error ENOTCONN
10058	socket error ESHUTDOWN
10059	socket error ETOOMANYREFS
10060	socket error ETIMEDOUT
10061	socket error ECONNREFUSED
10062	socket error ELOOP
10063	socket error ENAMETOOLONG
10064	socket error EHOSTDOWN
10065	socket error EHOSTUNREACH
10066	socket error ENOTEMPTY
10067	socket error EPROCLIM
10068	socket error EUSERS
10069	socket error EDQUOT
10070	socket error ESTALE
10071	socket error EREMOTE
10091	socket error SYSNOTREADY
10092	socket error VERNOTSUPPORTED
10093	socket error NOTINITIALIZED
10101	socket error EDISCON
10102	socket error EUNKNOWN

CSOCKET-NEXT-READ

This operation allows you to write multi-client servers. This operation waits until data is ready to be read from one of the sockets your server has created.

Syntax:

```
CALL "C$SOCKET" USING CSOCKET-NEXT-READ
                      serverHandle
                      timeout
                GIVING socketHandle
```

Parameters:

CSOCKET-NEXT-READ	Constant			
serverHandle	USAGE HANDLE	Handle of a server created by CSOCKET-CREATE-SERVER.		
timeout	PIC S9(n)	Valid values:	-1	returns only when the socket is available.
			0	check and returns event if a socket is not available.
			>0	number of milliseconds to wait before returning.
socketHandle	USAGE HANDLE	Handle of the next readable socket, or zero if no socket is available.		

C$SORT

The C$SORT library routine sorts indexed, relative, sequential and line sequential files. It takes only one parameter that matches with the instructions you would pass to the ISSORT (External Sort) utility on the command line.

The routine internally uses the SORT verb, so it's affected by the configuration settings whose name starts with "iscobol.sort" (e.g. iscobol.sort.memsize). The routine uses the file handler specified in the configuration to sort a specific kind of file. For example, when sorting indexed files, the routine uses the file handler specified by the iscobol.file.index property. The activity of this routine is traced in the isCOBOL log if iscobol.tracelevel includes the value 8 (trace file activity).

Syntax:

```
CALL "C$SORT" USING sortInstructions
              GIVING returnCode
```

Parameters:

sortInstructions	PIC X(n) or string literal	Specifies the instructions for the sort. Refer to ISSORT (External Sort) documentation for details about the allowed syntax.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful
2	Unsupported feature
15	Command statement error(s) detected
100	I/O error

Examples:

Example - Sort the indexed file named *idxfile* by reading records on the primary key in descending order, discard records whose data at offset 37 length 4 bytes represents a numeric value less than 902. Save the result to a line sequential file named *output.txt*.

```
call "c$sort" using "sort fields (1, 6, ch, d) "
                 & "use idxfile org ix record f 40 key (1, 6, p, 7, 15, c, 22, 15, ad) "
                 & "give output.txt org ls record f 40 "
                 & "include cond = 37,4,ge,902".
if return-code not = 0
   display message "Sort failed"
end-if
```

C$SYSINFO

The C$SYSINFO library routine returns information about the current operating system. It produces the same result of ACCEPT SYSTEM-INFORMATION FROM SYSTEM-INFO Statement.

The advantage of using this routine instead of the ACCEPT Statement is that the routine can return client system information if called by CALL CLIENT in Thin Client environment.

Syntax:

```
CALL "C$SYSINFO" USING SYSTEM-INFORMATION
```

Parameters:

SYSTEM-INFORMATION	Group-item

Structure described in the iscobol.def copybook as follows:

```
01  system-information.
    03  operating-system                        pic x(10).
        88  os-is-msdos                         value "ms-dos".
        88  os-is-os2                           value "OS/2".
        88  os-is-
vms                             values "vms", "vax/vms".
        88  os-is-
unix                            value "Linux", "AIX", "HP-
UX", "SunOS", "Solaris".
        88  os-is-linux                         value "Linux".
        88  os-is-aos                           value "aos/vs".
        88  os-is-
windows                         values "Windows 95", "Windows 98",
"Windows Me".
        88  os-is-win-
nt                          values "Windows 20", "Windows XP", "Win
dows Vi", "Windows 7", "WINDOWS", "Windows Se", "Windows 8".
        88  os-is-win-
family                      values "Windows 95", "Windows 98", "Win
dows Me", "Windows 20", "Windows XP", "Windows Vi", "WINDOWS", "Wi
ndows 7", "Windows Se", "Windows 8".
        88  os-is-amos                          value "amos".
        88  os-is-mpe                           value "MPE/iX".
        88  os-is-mpeix                         value "MPE/iX".
        88  os-is-
mac                             value "Mac OS", "Mac OS X".
    03  user-id                                 pic x(12).
    03  station-id                              pic x(12).
    03  filler                                  pic x.
        88  has-indexed-read-previous           value "Y".
    03  filler                                  pic x.
        88  has-relative-read-previous          value "Y".
    03  filler                                  pic x.
        88  can-test-input-status               value "Y".
    03  filler                                  pic x.
        88  is-multi-tasking                    value "Y".
    03  runtime-version.
        05  runtime-major-version               pic 99.
        05  runtime-minor-version               pic 99.
        05  runtime-release                     pic 99.
    03  filler                                  pic x.
        88  is-plugin                           value "Y".
    03  serial-number                           pic x(20).
    03  filler                                  pic x.
        88  has-large-file-support              value "Y".
    03  filler                                  pic x.
    03  filler                                  pic x.
        88  is-64-bit                           value "Y".
```

Examples:

Example - Get the client system information in a thin-client environment

```
*> copy "iscobol.def" working-storage

call "c$sysinfo" using system-information
display operating-system
```

C$SYSTEM

The C$SYSTEM library routine executes a program from an isCOBOL application.

On Windows the routine invokes the CreateProcessA API function via the jna libraries (installed along with isCOBOL). Calling the Windows API function allows to support the CSYS-MAXIMIZED, the CSYS-MINIMIZED and the CSYS-HIDDEN flags.

On Linux/Unix platforms, or when jna is not available in the Classpath, the routine uses the exec() method of the java.lang.Runtime class. In this scenario CSYS-MAXIMIZED, CSYS-MINIMIZED and CSYS-HIDDEN are not supported.

Syntax:

```
CALL "C$SYSTEM" USING commandLine
                      [flags]
                GIVING systemStatus
```

Parameters:

commandLine	PIC X(n)	Specifies the operating system command to be executed.
flags	any numeric data item	It specifies a value that affect the behavior of the library routine. The value is calculated by combining one or more of the following values, defined in iscobol.def:

	CSYS-ASYNC	The command specified in *commandLine* is executed asynchronously.
	CSYS-MAXIMIZED	The command window, if any, is shown maximized. It works only on Windows.
	CSYS-MINIMIZED	The command window, if any, is shown minimized. It works only on Windows.
	CSYS-HIDDEN	The command window, if any, is not shown. It works only on Windows.
	CSYS-SHELL	Only on Windows, the system command interpreter *cmd.exe* is used to run the command specified in *commandLine*. Note that on Linux/Unix the command interpreter *sh* is always used, instead, even without this flag.
	CSYS-DESKTOP	The command is executed client side in thin client environment. This flag is supported for compatibility. It's preferable to call the routine using CALL CLIENT in order to obtain the same result.

Return code:

systemStatus can be any signed numeric data item. It receives the exit status of *commandLine*.

Examples:

Example - Run notepad.exe and wait for it to finish

```
call "c$system" using "notepad.exe"
```

C$TBAR

The C$TBAR displays a toolbar in the current window.

Note - This routine is supported for compatibility. If you're creating new programs with isCOBOL, you should consider using the TOOL-BAR instead.

Syntax:

```
CALL "C$TBAR" [ USING button-1 [, button-2] ... [, button-n] [, exitCode]]
```

Parameters:

button-1 *button-2* ... *button-n*	Any alphanumeric data item or literal	These parameters define the buttons in the toolbar. The syntax is as follows: buttonname["hint"]=keycode where *buttonname* is the name of the icon. The runtime will look for a file named <buttonname>.png (lower case) in the Classpath. For example, if buttonname is 'exit', then 'exit.png' is required. If this file is not found, then no icon will be shown on the button. It's good practice to gather all the icon files in a library (e.g. tbar.jar) and add this library to the Classpath. *hint* is an optional text string that is displayed as tool tip when the mouse hovers over the button. *keycode* is the key code to be sent (see RM/COBOL key codes for C$MBAR, C$RBMENU and C$TBAR routines for details)
exitCode	PIC 9(n) BINARY	Optional parameter that returns the status. It will receive zero for success and non-zero for failure.

Calling the routine with no arguments turn off the toolbar.

Examples:

Example - Create a toolbar with three buttons

```
*>exit.png, left.png and right.png image files required in the Classpath
CALL "C$TBAR" USING "exit=\e" "left=\zl" "right=\zr"
```

C$TOLOWER

The C$TOLOWER library routine converts a data item's content to lower-case.

Syntax:

```
CALL "C$TOLOWER" USING textItem
                       [textLen]
```

Parameters:

textItem	any alphanumeric data item	Specifies the text to be converted.
textLen	any numeric data item or numeric literal	Specifies the number of characters to be converted. Conversion occurs from left to right and only the first textLen characters are converted. If this parameter is omitted the whole string is translated.

Examples:

Example - Change case to lowercase for a string

```
*> define str1 as pic x(n)

move "Hello COBOL World" to str1
call "c$tolower" using str1
*> new value for str1 will be: hello cobol world
```

C$TOUPPER

The C$TOUPPER library routine converts a data item's content to upper-case.

Syntax:

```
CALL "C$TOUPPER" USING textItem
                       [textLen]
```

Parameters:

textItem	any alphanumeric data item	Specifies the text to be converted.
textLen	any numeric data item or numeric literal	Specifies the number of characters to be converted. Conversion occurs from left to right and only the first textLen characters are converted. If this parameter is omitted the whole string is translated.

Examples:

Example - Change case to uppercase for a string

```
*> define str1 as pic x(n)

move "Hello COBOL World" to str1
call "c$toupper" using str1
*> new value for str1 will be: HELLO COBOL WORLD
```

C$TRIM

The C$TRIM library routine performs a trim operation on alphanumeric variables.

Note: Since TRIM, TRIML and TRIMR intrinsic functions are available, this routine is deprecated.

Syntax:

```
CALL "C$TRIM" USING var
```

Parameters:

var	PIC X(n)	If *var* is defined as pic x any length, the trim removes leading and trailing spaces and the variable is resized. If *var* is defined as pic x(n), the trim only removes leading spaces, the result is the same of a left justification of the variable made by calling C$JUSTIFY. Variables that are not alphanumeric produce undefined effects.

Examples:

Example - Remove leading spaces on a string

```
*> define str1 as pic x(n)

move "   Hello COBOL World" to str1
call "c$trim" using str1
*> the first 3 spaces will be removed
```

C$UNLOAD

The C$UNLOAD library routine removes COBOL programs class definition from memory so they're reloaded from disc the next time they're called. In order to work, iscobol.code_prefix.reload (boolean)* must be set to false.

Syntax:

```
CALL "C$UNLOAD" USING programNames
                GIVING returnCode
```

Parameters:

programNames	PIC X(n)	Specifies the name of the programs to be unloaded. If this parameter is omitted, set to spaces or set to "*", then all programs are unloaded. Multiple program names can be specified. They must be separated by a line feed (X"0A"). The "*" wildcard character is supported at the end of the program name to unload all the programs whose name begins with the same characters. You should pass the same name used in the CALL statement. For example, if you used *CALL "dir1\PROG1"*, pass "dir1\PROG1" to C$UNLOAD, not just "PROG1".

Return code:

returnCode can be any signed numeric data item. It returns the number of programs that were actually unloaded.

Examples:

Example - Unload some programs from memory:

```
       working-storage section.
       77 unload-list pic x any length.

       procedure division.
      *the following programs will be unloaded:
      * CUSTADM
      * ORD1A
      * ORD1B
           initialize unload-list.
           string "CUSTADM" x"0a" "ORD1*"
                  delimited by size
                  into unload-list.
           call "c$unload" using unload-list.
```

C$UNLOAD_NATIVE

The C$UNLOAD_NATIVE library routine removes a shared library (e.g. a DLL) from memory.

The routine fails on purpose in thin client and J2EE environments in order to avoid unloading a library that other connected clients may need.

Syntax:

```
CALL "C$UNLOAD_NATIVE" USING libraryName
                       GIVING returnCode
```

Parameters:

libraryName	PIC X(n)	Specifies the name of the shared library to unload.

Return code:

returnCode can be any signed numeric data item. It contains 0 if the unload was successful, else it contains 1.

Examples:

Example - Unload the KERNEL32 library when the provided functions are no more needed

```
call "c$unload_native" using "kernel32.dll"
```

C$UNSET

The C$UNSET library routine removes a variable from the environment.

It can be used to remove any configuration setting except for keystrokes. Keystrokes cannot be initialized to their defaults by calling C$UNSET, they must be reset using a Format 6 SET statement, setting them to the proper value as described in Default Keyboard Configuration.

Syntax:

```
CALL "C$UNSET" USING envVarName
```

Parameters:

envVarName	PIC X(n)	Specifies the environment variable to be removed.

Examples:

Example - Remove custom property my.report.path

```
call "c$unset" using "my.report.path"
```

C$VERSION

The C$VERSION retrieves the isCOBOL framework version.

Syntax:

```
CALL "C$VERSION" USING version
```

Parameters:

version	PIC X(n)	After this routine has been called this data item contains the isCOBOL version framework.

The same result is reachable through the following statement:

```
ACCEPT version FROM ENVIRONMENT "runtime.version"
```

Examples:

Example - Get the isCOBOL runtime version

```
*> define str1 as pic x(n)

call "c$version" using str1
*> the value of str1 will be like: isCOBOL release 2019 R1 build#977.10-20190516-27629
```

C$WRITELOG

The C$WRITELOG library routine allows to add custom information to the isCOBOL log file. The iscobol.tracelevel property must be set to a value greater than 0 or the routine will have no effect.

Every call to this routine generates a new line in the log.

All the messages written to the log by this routine are messages of type INFO. If you wish to write messages of different type, consider using the Logger (com.iscobol.logger.Logger) internal object instead.

Syntax:

```
CALL "C$WRITELOG" USING text-1 [, text-2] ... [, text-n]
                  GIVING returnCode
```

Parameters:

text-1 *text-2* ... *text-n*	any data item or literal	Specifies the text to be written into the log file. At least one parameter must be passed. The others are optional. Multiple parameters values are combined together in the same log line.

Return code:

returnCode can be any numeric data item and provides additional information:

1	Operation successful
0	Operation failed, probably iscobol.tracelevel is set to 0

Examples:

Example - Write some custom information to the log

```
*> define trace-text as pic x(n)
initialize trace-text
string "Customer in debt "
       cust-code
       ". Debt Amount : "
       cust-amount-ed
       into trace-text
end-string
call "c$writelog" using trace-text
```

C$WRU

The C$WRU library routine returns the name of the program that has called the currently running program.

Syntax:

```
CALL "C$WRU" USING programName,
                   programLine, programIntraLine.
```

Parameters:

programName	PIC X(n)	Receives the name of the calling program. If this routine is called by the main program, "RUNCOBOL" is returned.
programLine	PIC 9(n)	Receives 0.
programIntraLine	PIC 9(n)	Receives 0.

Examples:

Example - Retrieve the name of the calling program

```
*> define calling-program as pic x(n)
*> define foo-1 and foo-2 as pic 9

call "c$wru" using calling-program, foo-1, foo-2.
```

C$XML

The C$XML routine allows to deal with XML documents. It allows to parse documents and change their content as well as create new documents from scratch.

When parsing the XML document, referenced DTD (Document Type Definition) and XSD (XML Schema definition) are considered. If the referenced DTD can't be found due to invalid or unreachable location, then it's ignored.

When writing content to the XML document, the following data conversion rules are applied:

- Text is encoded using the character set specified by the iscobol.encoding * property
- Trailing spaces in alphanumeric parameters are trimmed
- Leading zeros are kept in numeric values

Syntax:

```
CALL "C$XML" USING opCode
                   parameters
             GIVING returnCode
```

Parameters:

opCode	Is the function to be executed. Valid values, defined in ..., are:	
	CXML-PARSE-FILE	Opens a XML file and returns the XML tree
	CXML-RELEASE-PARSER	Releases memory used by the XML tree
	CXML-GET-FIRST-CHILD	Returns the first child of a given element
	CXML-GET-NEXT-SIBLING	Returns the next sibling of a given element
	CXML-GET-PARENT	Returns the parent element of a given element
	CXML-GET-DATA	Returns the name and the CDATA of a given element
	CXML-GET-ATTRIBUTE-COUNT	Returns the number of attributes of a given element
	CXML-GET-ATTRIBUTE	Returns the name and the value of a given attribute
	CXML-GET-LAST-ERROR	Returns the last error
	CXML-OPEN-FILE	Opens a XML file without parsing it
	CXML-PARSE-STRING	Parses a string as XML
	CXML-PARSE-NEXT-RECORD	Parses the next record in the element
	CXML-GET-PREV-SIBLING	Returns the previous sibling of a given element
	CXML-NEW-PARSER	Creates an empty XML document in memory
	CXML-GET-ATTRIBUTE-BY-NAME	Returns the attribute with the given name along with its value
	CXML-GET-CHILD-BY-NAME	Returns the child element with the given name
	CXML-GET-CHILD-BY-CDATA	Returns the child element with the given CDATA
	CXML-GET-CHILD-BY-ATTR-NAME	Returns the child element with the given attribute
	CXML-GET-CHILD-BY-ATTR-VALUE	Returns the child element with the given attribute value
	CXML-GET-SIBLING-BY-NAME	Returns the sibling element with the given name
	CXML-GET-SIBLING-BY-CDATA	Returns the sibling element with the given CDATA
	CXML-GET-SIBLING-BY-ATTR-NAME	Returns the sibling element with the given attribute

	CXML-GET-SIBLING-BY-ATTR-VALUE	Returns the sibling element with the given attribute value
	CXML-GET-COMMENT	Returns the comment associated to a given element or document
	CXML-SET-DATA	Changes the CDATA of a given element
	CXML-MODIFY-ATTRIBUTE-VALUE	Changes the value of a given attribute
	CXML-ADD-CHILD	Adds a new child item to the given element
	CXML-ADD-SIBLING	Adds a new sibling item to the given element
	CML-ADD-ATTRIBUTE	Adds an attribute to a given element
	CXML-ADD-COMMENT	Adds a comment to a given element or document
	CXML-APPEND-COMMENT	Appends a new comment to a given element or document
	CXML-DELETE-ATTRIBUTE	Deletes an attribute from an element
	CXML-DELETE-ELEMENT	Deletes an element
	CXML-DELETE-COMMENT	Deletes comments from an element
	CXML-WRITE-FILE	Writes a XML document to file
	CXML-WRITE-STRING	Writes a String
	CXML-GET-PROC-INSTR-COUNT	Retrieves the number of processing instructions
	CXML-GET-PROC-INSTR	Retrieves processing instructions
	CXML-SET-PROC-INSTR	Sets processing instructions
	CXML-GET-VERSION	Returns the version of the XML document
	CXML-SET-VERSION	Changes the version of the XML document
	CXML-GET-ENCODING	Returns the encoding of the XML document
	CXML-SET-ENCODING	Changes the encoding of the XML document
	CXML-GET-STANDALONE	Returns the value of the standalone pseudo-attribute
	CXML-SET-STANDALONE	Changes the value of the standalone pseudo-attribute

Return code:

The definition and meaning of the *returnCode* depend on the opcode.

Examples:

Example - Read the content of file.xml and show it on the console.

```
       working-storage section.
       copy "iscobol.def".

       77  xml-handle                          handle.
       77  elem-handle                         handle.
       77  next-elem-handle                    handle.
       77  xml-item-name                       pic x(32).
       77  xml-item-value                      pic x(32).
       77  attr-count                          pic 99.
       77  i                                   pic 99.
       77  attr-name                           pic x(32).
       77  attr-value                          pic x(32).

       procedure division.
       read-xml.
           call "c$xml" using cxml-parse-file
                              "file.xml"
                        giving xml-handle.

           call "c$xml" using cxml-get-first-child
                              xml-handle
                        giving elem-handle.

           perform until exit
             initialize xml-item-name, xml-item-value
             call "c$xml" using cxml-get-data
                                elem-handle
                                xml-item-name
                                xml-item-value
             display "elem. name: " xml-item-name
             display "elem. value: " xml-item-value

             call "c$xml" using cxml-get-attribute-count
                                elem-handle
                          giving attr-count
             display "attr. count: " attr-count
             if attr-count > 0
                perform varying i from 1 by 1 until i > attr-count
                  call "c$xml" using cxml-get-attribute
                                     elem-handle
                                     i
                                     attr-name
                                     attr-value
                  display "attr. name: " attr-name
                  display "attr. value: " attr-value
                end-perform
             end-if
             call "c$xml" using cxml-get-next-sibling
                                elem-handle
                          giving next-elem-handle
             destroy elem-handle
```

```
            if next-elem-handle = 0
               exit perform
            else
               move next-elem-handle to elem-handle
            end-if
         end-perform.
         display ""
         call "c$xml" using cxml-release-parser
                            xml-handle.
         accept omitted.
         goback.
```

CXML-PARSE-FILE

The CXML-PARSE-FILE function parses a XML file, returning a handle to the entire XML tree as the return-code.

Syntax:

```
CALL "C$XML" USING CXML-PARSE-FILE
                   fileName
             GIVING xmlHandle
```

Parameters:

CXML-PARSE-FILE	Constant	
fileName	PIC X(n)	Specifies the name of the file to open.

Return code:

xmlHandle must be a USAGE HANDLE data item. It receives the handle of the XML tree and will be used by other C$XML op-codes.

CXML-RELEASE-PARSER

The CXML-RELEASE-PARSER releases memory allocated by parsing.

Syntax:

```
CALL "C$XML" USING CXML-RELEASE-PARSER
                   xmlHandle
```

Parameters:

CXML-RELEASE-PARSER	Constant	
xmlHandle	USAGE HANDLE	Specifies the handle of a XML tree returned by CXML-PARSE-FILE.

Return code:

This operation has no return value.

CXML-GET-FIRST-CHILD

The CXML-GET-FIRST-CHILD function retrieves the handle of the first child element of the handle passed.

Syntax:

```
CALL "C$XML" USING CXML-GET-FIRST-CHILD
                   handle
             GIVING childHandle
```

Parameters:

CXML-GET-FIRST-CHILD	Constant	
handle	USAGE HANDLE	A XML element handle.

Return code:

childHandle must be a USAGE HANDLE data item. It receives the handle of the first child element, or zero if there are no children. It's good practice to free this handle with a DESTROY statement before reusing it in order to avoid memory leaks.

CXML-GET-NEXT-SIBLING

The CXML-GET-NEXT-SIBLING function retrieves the handle of the next sibling element of the handle passed.

Syntax:

```
CALL "C$XML" USING CXML-GET-NEXT-SIBLING
                   handle
             GIVING siblingHandle
```

Parameters:

CXML-GET-NEXT-SIBLING	Constant	
handle	USAGE HANDLE	A XML element handle.

Return code:

siblingHandle must be a USAGE HANDLE data item. It receives the handle of the next sibling element, or zero if there are no sibling items. It's good practice to free this handle with a DESTROY statement before reusing it in order to avoid memory leaks.

CXML-GET-PARENT

The CXML-GET-PARENT function retrieves the handle of the parent element of the handle passed.

Syntax:

```
CALL "C$XML" USING CXML-GET-PARENT
                   handle
             GIVING parentHandle
```

Parameters:

CXML-GET-PARENT	Constant	
handle	USAGE HANDLE	A XML element handle.

Return code:

parentHandle must be a USAGE HANDLE data item. It receives the handle of the parent element, or zero if you're processing the root element. It's good practice to free this handle with a DESTROY statement before reusing it in order to avoid memory leaks.

CXML-GET-DATA

The CXML-GET-DATA function retrieves name and value of a given element.

Syntax:

```
CALL "C$XML" USING CXML-GET-DATA
                   handle
                   itemName
                   itemValue
                  [valueLength]
```

Parameters:

CXML-GET-DATA	Constant	
handle	USAGE HANDLE	A XML element handle.
itemName	PIC X(n)	Receives the name of the element.
itemValue	PIC X(n)	Receives the value of the element. This value is trimmed.
valueLength	PIC 9(n)	Optional. Receives the length in bytes of the value after the trim.

Return code:

This operation has no return value.

CXML-GET-ATTRIBUTE-COUNT

The CXML-GET-ATTRIBUTE-COUNT function retrieves the number of attributes of an element.

Syntax:

```
CALL "C$XML" USING CXML-GET-ATTRIBUTE-COUNT
                   handle
             GIVING attrCount
```

Parameters:

CXML-GET-ATTRIBUTE-COUNT	Constant	
handle	USAGE HANDLE	A XML element handle.

Return code:

If the return-code is zero, it can mean both that there are no elements, or that the passed handle is not a valid handle. If it's greater than zero, it is the attribute count.

CXML-GET-ATTRIBUTE

The CXML-GET-DATA function retrieves name and value of the attributes of an element.

Syntax:

```
CALL "C$XML" USING CXML-GET-ATTRIBUTE
                   handle
                   attrNum
                   attrName
                   attrValue
                  [valueLength]
```

Parameters:

CXML-GET-ATTRIBUTE	Constant	
handle	USAGE HANDLE	A XML element handle.
attrNum	PIC 9(n)	The attribute to get, starting at 1.
itemName	PIC X(n)	Receives the name of the attribute.
attrValue	PIC X(n)	Receives the value of the attribute.
valueLength	PIC 9(n)	Optional. Receives the lenght in bytes of the attribute value.

Return code:

This operation has no return value.

CXML-GET-LAST-ERROR

The CXML-GET-LAST-ERROR function returns the last error.

Syntax:

```
CALL "C$XML" USING CXML-GET-LAST-ERROR
                   errorText
```

Parameters:

CXML-GET-LAST-ERROR	Constant	
errorText	PIC X(n)	Receives a textual description of the error.

Return code:

This operation has no return value.

CXML-OPEN-FILE

The CXML-OPEN-FILE function opens a named XML file without parsing it. Once the file is open, you can read individual records by calling CXML-PARSE-NEXT-RECORD.

Syntax:

```
CALL "C$XML" USING CXML-OPEN-FILE
                   fileName
             GIVING xmlHandle
```

Parameters:

CXML-OPEN-FILE	Constant	
fileName	PIC X(n)	Specifies the name of the file to open.

Return code:

xmlHandle must be a USAGE HANDLE data item. It receives the handle of the XML file.

CXML-PARSE-STRING

The CXML-PARSE-STRING function parses the specified string as XML, returning a parser handle.

Syntax:

```
CALL "C$XML" USING CXML-PARSE-STRING
                   string
             GIVING xmlHandle
```

Parameters:

CXML-PARSE-STRING	Constant	
string	PIC X(n)	Specifies the string to parse.

Return code:

xmlHandle must be a USAGE HANDLE data item. It receives the handle of the XML tree and will be used by other C$XML op-codes.

CXML-PARSE-NEXT-RECORD

The CXML-PARSE-NEXT-RECORD function allows to parse individual records of a file instead of parsing the entire file all at once.

Syntax:

```
CALL "C$XML" USING CXML-PARSE-NEXT-RECORD
                   xmlHandle
             GIVING recordHandle
```

Parameters:

CXML-PARSE-NEXT-RECORD	Constant	
xmlHandle	USAGE HANDLE	A parser handle returned by CXML-OPEN-FILE.

Return code:

recordHandle must be a USAGE HANDLE data item. It receives the handle of record read. It's good practice to free this handle with a DESTROY statement before reusing it in order to avoid memory leaks.

CXML-GET-PREV-SIBLING

The CXML-GET-PREV-SIBLING function allows to move backwards in an XML file.

Syntax:

```
CALL "C$XML" USING CXML-GET-PREV-SIBLING
                   handle
             GIVING siblingHandle
```

Parameters:

CXML-GET-PREV-SIBLING	Constant	
handle	USAGE HANDLE	A XML element handle.

Return code:

siblingHandle must be a USAGE HANDLE data item. It receives the handle of the previous sibling element. It's good practice to free this handle with a DESTROY statement before reusing it in order to avoid memory leaks.

CXML-NEW-PARSER

The CXML-NEW-PARSER function creates a new XML document.

Syntax:

```
CALL "C$XML" USING CXML-NEW-PARSER
             GIVING xmlHandle
```

Parameters:

CXML-NEW-PARSER	Constant

Return code:

xmlHandle must be a USAGE HANDLE data item. It receives the handle of the XML document.

CXML-GET-ATTRIBUTE-BY-NAME

The CXML-GET-ATTRIBUTE-BY-NAME function returns the content of the named attribute.

Syntax:

```
CALL "C$XML" USING CXML-GET-ATTRIBUTE-BY-NAME
                   handle
                   attrName
                   attrFlags
                   attrIdx
                   attrActualName
                   attrValue
                   attrLen
```

Parameters:

CXML-GET-ATTRIBUTE-BY-NAME	Constant	
handle	USAGE HANDLE	A XML element handle.
attrName	PIC X(n)	It specifies the name of the attribute to search.
attrFlags	PIC 9(n)	It specifies if the search should consider case. Valid values are: **0** - case sensitive search **1** - case insensitive search
attrIdx	PIC 9(n)	On entry, it specifies the starting index of attributes to search; use zero to start at the first attribute. On exit, it returns the index of the attribute found.
attrActualName	PIC X(n)	It returns the name of the attribute found.
attrValue	PIC X(n)	It returns the value of the attribute found.
attrLen	PIC 9(n)	It returns the lenght of the attribute value.

CXML-GET-CHILD-BY-NAME

The CXML-GET-CHILD-BY-NAME function returns the named child element directly, avoiding you to parse elements one by one.

Syntax:

```
CALL "C$XML" USING CXML-GET-CHILD-BY-NAME
                   handle
                   childName
                   childFlags
             GIVING childHandle
```

Parameters:

CXML-GET-CHILD-BY-NAME	Constant	
handle	USAGE HANDLE	A XML element handle.
childName	PIC X(n)	It specifies the name of the child element to search.
childFlags	PIC 9(n)	It specifies if the search should consider case. Valid values are: **0** - case sensitive search **1** - case insensitive search

Return code:

childHandle must be a USAGE HANDLE data item. It receives the handle of the found child element or 0 if no child was found. It's good practice to free this handle with a DESTROY statement before reusing it in order to avoid memory leaks.

CXML-GET-CHILD-BY-CDATA

The CXML-GET-CHILD-BY-CDATA function returns the handle of the child element that includes the specified CDATA.

Syntax:

```
CALL "C$XML" USING CXML-GET-CHILD-BY-CDATA
                   handle
                   cData
                   cDataFlags
             GIVING childHandle
```

Parameters:

CXML-GET-CHILD-BY-CDATA	Constant	
handle	USAGE HANDLE	A XML element handle.
cData	PIC X(n)	It specifies the CDATA to search.
cDataFlags	PIC 9(n)	It specifies if the search should consider case. Valid values are: **0** - case sensitive search **1** - case insensitive search

Return code:

childHandle must be a USAGE HANDLE data item. It receives the handle of the found child element or 0 if no child was found. It's good practice to free this handle with a DESTROY statement before reusing it in order to avoid memory leaks.

CXML-GET-CHILD-BY-ATTR-NAME

The CXML-GET-CHILD-BY-ATTR-NAME function returns the handle of the child element that includes the specified attribute.

Syntax:

```
CALL "C$XML" USING CXML-GET-CHILD-BY-ATTR-NAME
                   handle
                   attrName
                   attrFlags
             GIVING childHandle
```

Parameters:

CXML-GET-CHILD-BY-ATTR-NAME	Constant	
handle	USAGE HANDLE	A XML element handle.
attrName	PIC X(n)	It specifies the attribute name to search.
attrFlags	PIC 9(n)	It specifies if the search should consider case. Valid values are: **0** - case sensitive search **1** - case insensitive search

Return code:

childHandle must be a USAGE HANDLE data item. It receives the handle of the found child element or 0 if no child was found. It's good practice to free this handle with a DESTROY statement before reusing it in order to avoid memory leaks.

CXML-GET-CHILD-BY-ATTR-VALUE

The CXML-GET-CHILD-BY-ATTR-VALUE function returns the handle of the child element that includes an attribute with a given value.

Syntax:

```
CALL "C$XML" USING CXML-GET-CHILD-BY-ATTR-VALUE
                   handle
                   attrValue
                   attrFlags
             GIVING childHandle
```

Parameters:

CXML-GET-CHILD-BY-ATTR-VALUE	Constant	
handle	USAGE HANDLE	A XML element handle.
attrValue	PIC X(n)	It specifies the attribute value to search.
attrFlags	PIC 9(n)	It specifies if the search should consider case. Valid values are: **0** - case sensitive search **1** - case insensitive search

Return code:

childHandle must be a USAGE HANDLE data item. It receives the handle of the found child element or 0 if no child was found. It's good practice to free this handle with a DESTROY statement before reusing it in order to avoid memory leaks.

CXML-GET-SIBLING-BY-NAME

The CXML-GET-SIBLING-BY-NAME function returns the named sibling element directly, avoiding you to parse elements one by one.

Syntax:

```
CALL "C$XML" USING CXML-GET-SIBLING-BY-NAME
                   handle
                   childName
                   childFlags
             GIVING siblingHandle
```

Parameters:

CXML-GET-SIBLING-BY-NAME	Constant	
handle	USAGE HANDLE	A XML element handle.
siblingName	PIC X(n)	It specifies the name of the sibling element to search.
siblingFlags	PIC 9(n)	It specifies if the search should consider case. Valid values are: **0** - case sensitive search **1** - case insensitive search

Return code:

siblingHandle must be a USAGE HANDLE data item. It receives the handle of the found sibling element or 0 if such element wasn't found. It's good practice to free this handle with a DESTROY statement before reusing it in order to avoid memory leaks.

CXML-GET-SIBLING-BY-CDATA

The CXML-GET-SIBLING-BY-CDATA function returns the handle of the sibling element that includes the specified CDATA.

Syntax:

```
CALL "C$XML" USING CXML-GET-SIBLING-BY-CDATA
                   handle
                   cData
                   cDataFlags
             GIVING siblingHandle
```

Parameters:

CXML-GET-SIBLING-BY-CDATA	Constant	
handle	USAGE HANDLE	A XML element handle.
cData	PIC X(n)	It specifies the CDATA to search.
cDataFlags	PIC 9(n)	It specifies if the search should consider case. Valid values are: **0** - case sensitive search **1** - case insensitive search

Return code:

siblingHandle must be a USAGE HANDLE data item. It receives the handle of the found sibling element or 0 if such element wasn't found. It's good practice to free this handle with a DESTROY statement before reusing it in order to avoid memory leaks.

CXML-GET-SIBLING-BY-ATTR-NAME

The CXML-GET-SIBLING-BY-ATTR-NAME function returns the handle of the sibling element that includes the specified attribute.

Syntax:

```
CALL "C$XML" USING CXML-GET-SIBLING-BY-ATTR-NAME
                   handle
                   attrName
                   attrFlags
             GIVING siblingHandle
```

Parameters:

CXML-GET-SIBLING-BY-ATTR-NAME	Constant	
handle	USAGE HANDLE	A XML element handle.
attrName	PIC X(n)	It specifies the attribute name to search.
attrFlags	PIC 9(n)	It specifies if the search should consider case. Valid values are: **0** - case sensitive search **1** - case insensitive search

Return code:

siblingHandle must be a USAGE HANDLE data item. It receives the handle of the found sibling element or 0 if such element wasn't found. It's good practice to free this handle with a DESTROY statement before reusing it in order to avoid memory leaks.

CXML-GET-SIBLING-BY-ATTR-VALUE

The CXML-GET-SIBLING-BY-ATTR-VALUE function returns the handle of the sibling element that includes an attribute with a given value.

Syntax:

```
CALL "C$XML" USING CXML-GET-SIBLING-BY-ATTR-VALUE
                   handle
                   attrValue
                   attrFlags
             GIVING siblingHandle
```

Parameters:

CXML-GET-SIBLING-BY-ATTR-VALUE	Constant	
handle	USAGE HANDLE	A XML element handle.
attrValue	PIC X(n)	It specifies the attribute value to search.
siblingFlags	PIC 9(n)	It specifies if the search should consider case. Valid values are: **0** - case sensitive search **1** - case insensitive search

Return code:

siblingHandle must be a USAGE HANDLE data item. It receives the handle of the found sibling element or 0 if such element wasn't found. It's good practice to free this handle with a DESTROY statement before reusing it in order to avoid memory leaks.

CXML-GET-COMMENT

The CXML-GET-COMMENT function returns the comments associated to a given element or document.

Syntax:

```
CALL "C$XML" USING CXML-GET-COMMENT
                   handle
                   commentData
                  [commentLength]
```

Parameters:

CXML-GET-COMMENT	Constant	
handle	USAGE HANDLE	A XML element or parser handle.
commentData	PIC X(n)	It receives the comment text. If more comments are available, they're returned as substrings separated by a x"00" digit.
commentLength	PIC 9(n)	It receives the length of the comment text.

CXML-SET-DATA

The CXML-SET-DATA function modifies the CDATA of an element.

CXML-SET-DATA is the new name of a renamed function. The old name CXML-MODIFY-CDATA can still be used, though.

Syntax:

```
CALL "C$XML" USING CXML-SET-DATA
                   handle
                   data
                  [length]
                  [startPos]
                  [endPos]
```

Parameters:

CXML-SET-DATA	Constant	
handle	USAGE HANDLE	A XML element handle.
data	PIC X(n)	Specifies the new CDATA.
length	PIC 9(n)	Specifies the number of bytes in *data* to consider. If omitted, the whole *data* content is used.
startPos	PIC 9(n)	Specifies the starting position of the data to modify. If omitted, it defaults to the start of the data.
endPos	PIC 9(n)	Specifies the ending position of the data to modify. If omitted, it defaults to the end of the data.

CXML-MODIFY-ATTRIBUTE-VALUE

The CXML-MODIFY-ATTRIBUTE-VALUE function modifies and attribute value.

Syntax:

```
CALL "C$XML" USING CXML-MODIFY-ATTRIBUTE-VALUE
                   handle
                   attrNum
                   attrValue
                  [attrLength]
```

Parameters:

CXML-MODIFY-ATTRIBUTE-VALUE	Constant	
handle	USAGE HANDLE	A XML element handle.
attrNum	PIC 9(n)	Specifies the ordinal number of the attribute to modify.
attrValue	PIC X(n)	Specifies the new value of the attribute.
attrLength	PIC 9(n)	Specifies the length of the new attribute value. If omitted, it defaults to the length of *attrValue*.

CXML-ADD-CHILD

The CXML-ADD-CHILD function adds a child item to the given element.

Syntax:

```
CALL "C$XML" USING CXML-ADD-CHILD
                   handle
                   elemName
                  [elemData]
                  [dataLength]
            GIVING childHandle
```

Parameters:

CXML-ADD-CHILD	Constant	
handle	USAGE HANDLE	A XML element handle.
elemName	PIC X(n)	Specifies the name of the child element that you're going to add.
elemData	PIC X(n)	Specifies the value of the new child element. If omitted, blank is used.
attrLength	PIC 9(n)	Specifies the length of the element value. If omitted, it defaults to the length of *elemData*.

Return code:

childHandle must be a USAGE HANDLE data item. It receives the handle of the new child element. It's good practice to free this handle with a DESTROY statement before reusing it in order to avoid memory leaks.

CXML-ADD-SIBLING

The CXML-ADD-SIBLING function adds a sibling item to the given element.

Syntax:

```
CALL "C$XML" USING CXML-ADD-SIBLING
                   handle
                   elemName
                  [elemData]
                  [dataLength]
             GIVING siblingHandle
```

Parameters:

CXML-ADD-SIBLING	Constant	
handle	USAGE HANDLE	A XML element handle.
elemName	PIC X(n)	Specifies the name of the sibling element that you're going to add.
elemData	PIC X(n)	Specifies the value of the new element. If omitted, blank is used.
attrLength	PIC 9(n)	Specifies the length of the element value. If omitted, it defaults to the length of *elemData*.

Return code:

siblingHandle must be a USAGE HANDLE data item. It receives the handle of the new element. It's good practice to free this handle with a DESTROY statement before reusing it in order to avoid memory leaks.

CML-ADD-ATTRIBUTE

The CXML-ADD-ATTRIBUTE function adds an attribute to the given element.

Syntax:

```
CALL "C$XML" USING CXML-ADD-ATTRIBUTE
                   handle
                   attrName
                  [attrValue]
                  [valueLength]
```

Parameters:

CXML-ADD-ATTRIBUTE	Constant	
handle	USAGE HANDLE	A XML element handle.
attrName	PIC X(n)	Specifies the name of the attribute that you're going to add.
attrValue	PIC X(n)	Specifies the value of the attribute. If omitted, blank is used.
valueLength	PIC 9(n)	Specifies the length of the attribute value. If omitted, it defaults to the length of *attrValue*.

CXML-ADD-COMMENT

The CXML-ADD-COMMENT function adds a comment into the XML.

Syntax:

```
CALL "C$XML" USING CXML-ADD-COMMENT
                   handle
                   data
                  [length]
```

Parameters:

CXML-ADD-COMMENT	Constant	
handle	USAGE HANDLE	A XML element or parser handle.
data	PIC X(n)	Specifies the comment text.
length	PIC 9(n)	Specifies the length of the comment text. If omitted, it defaults to the length of *data*.

CXML-APPEND-COMMENT

The CXML-APPEND-COMMENT function appends a comment to a given element or document.

Syntax:

```
CALL "C$XML" USING CXML-APPEND-COMMENT
                   handle
                   data
                  [length]
```

Parameters:

CXML-APPEND-COMMENT	Constant	
handle	USAGE HANDLE	A XML element or parser handle.
data	PIC X(n)	Specifies the comment text.
length	PIC 9(n)	Specifies the length of the comment text. If omitted, it defaults to the length of *data*.

CXML-DELETE-ATTRIBUTE

The CXML-DELETE-ATTRIBUTE function deletes an attribute from an element.

Syntax:

```
CALL "C$XML" USING CXML-DELETE-ATTRIBUTE
                   handle
                   attrNum
```

Parameters:

CXML-DELETE-ATTRIBUTE	Constant	
handle	USAGE HANDLE	A XML element handle.
attrNum	PIC 9(n)	Specifies the ordinal number of the attribute to delete.

CXML-DELETE-ELEMENT

The CXML-DELETE-ELEMENT function deletes an element.

Syntax:

```
CALL "C$XML" USING CXML-DELETE-ELEMENT
                   handle
```

Parameters:

CXML-DELETE-ELEMENT	Constant	
handle	USAGE HANDLE	A XML element handle. It specifies the element to delete.

CXML-DELETE-COMMENT

The CXML-DELETE-COMMENT function removes comments from XML elements or document.

Syntax:

```
CALL "C$XML" USING CXML-DELETE-COMMENT
                   handle
```

Parameters:

CXML-DELETE-COMMENT	Constant	
handle	USAGE HANDLE	A XML element or parser handle.

CXML-WRITE-FILE

The CXML-WRITE-FILE function writes the content of a XML document to file. After this operation, the CXML-RELEASE-PARSER op-code should be used to release the XML document from memory.

The indentation of elements in the generated XML is controlled by the configuration property iscobol.cxml.indent_number.

Syntax:

```
CALL "C$XML" USING CXML-WRITE-FILE
                   handle
                   fileName
```

Parameters:

CXML-WRITE-FILE	Constant	
handle	USAGE HANDLE	A XML parser handle.
fileName	PIC X(n)	Specifies the name of the file to write.

CXML-WRITE-STRING

The CXML-WRITE-STRING function copies the content of a XML document to an alphanumeric data item.

The indentation of elements in the generated XML is controlled by the configuration property iscobol.cxml.indent_number.

Syntax:

```
CALL "C$XML" USING CXML-WRITE-STRING
                   handle
                   dataArea
             GIVING returnCode
```

Parameters:

CXML-WRITE-STRING	Constant	
handle	USAGE HANDLE	A XML parser handle.
dataArea	PIC X(n)	Receives the XML content.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
>0	Data truncated. *dataArea* wasn't large enough. The value of *returnCode* is the number of bytes discarded.
<0	Operation failed.

CXML-GET-PROC-INSTR-COUNT

The CXML-GET-PROC-INSTR-COUNT function returns the number of processing instructions that exist.

Syntax:

```
CALL "C$XML" USING CXML-GET-PROC-INSTR-COUNT
                   handle
             GIVING procInstrCount
```

Parameters:

CXML-GET-PROC-INSTR-COUNT	Constant	
handle	USAGE HANDLE	A XML parser handle.

Return code:

procInstrCount can be any numeric data item and provides additional information:

0	An error occurred.
>0	The number of processing instructions.

CXML-GET-PROC-INSTR

The CXML-GET-PROC-INSTR function retrieves processing instructions (i.e. the stylesheet) from the XML file.

Syntax:

```
CALL "C$XML" USING CXML-GET-PROC-INSTR
                   handle
                   idx
                   target
                   data
```

Parameters:

CXML-GET-PROC-INSTR	Constant	
handle	USAGE HANDLE	A XML parser handle.
idx	PIC 9(n)	Specifies the index of the processing instruction. Use 1 to get the first processing instruction, 2 to get the second, and so on.
target	PIC X(n)	Returns the target. For example, if the following instruction is processed: <?xml-stylesheet type="text/css" href="veryant.css"?> *target* will be set to: xml-stylesheet.
data	PIC X(n)	Returns the data. For example, if the following instruction is processed: <?xml-stylesheet type="text/css" href="veryant.css"?> *data* will be set to: type="text/css" href="veryant.css".

CXML-SET-PROC-INSTR

The CXML-SET-PROC-INSTR function sets processing instructions (i.e. the stylesheet) in the XML file.

Syntax:

```
CALL "C$XML" USING CXML-SET-PROC-INSTR
                   handle
                   idx
                   target
                   data
```

Parameters:

CXML-GET-PROC-INSTR	Constant	
handle	USAGE HANDLE	A XML parser handle.
idx	PIC 9(n)	Specifies the index of the processing instruction. Use 1 to set the first processing instruction, 2 to set the second, and so on.
target	PIC X(n)	Specifies the target. For example, to generate the following instruction: <?xml-stylesheet type="text/css" href="veryant.css"?> set *target* to: xml-stylesheet.
data	PIC X(n)	Specifies the data. For example, to generate the following instruction: <?xml-stylesheet type="text/css" href="veryant.css"?> set *data* to: type="text/css" href="veryant.css". Set this item to spaces to remove the instruction.

CXML-GET-VERSION

The CXML-GET-VERSION function returns the version of the XML document.

Syntax:

```
CALL "C$XML" USING CXML-GET-VERSION
                   handle
                   version
```

Parameters:

CXML-GET-VERSION	Constant	
handle	USAGE HANDLE	A XML parser handle.
version	PIC X(n)	Receives the XML version.

CXML-SET-VERSION

The CXML-SET-VERSION function changes the version of the XML document.

Syntax:

```
CALL "C$XML" USING CXML-SET-VERSION
                   handle
                   version
```

Parameters:

CXML-SET-VERSION	Constant	
handle	USAGE HANDLE	A XML parser handle.
version	PIC X(n)	Specifies the new XML version.

CXML-GET-ENCODING

The CXML-GET-ENCODING function returns the encoding of the XML document.

Syntax:

```
CALL "C$XML" USING CXML-GET-ENCODING
                   handle
                   encoding
```

Parameters:

CXML-GET-ENCODING	Constant	
handle	USAGE HANDLE	A XML parser handle.
encoding	PIC X(n)	Receives the XML encoding.

CXML-SET-ENCODING

The CXML-SET-ENCODING function changes the encding of the XML document. Note that this operation affects only the document header, the actual encoding used inside the document depends by how the program writes data.

Syntax:

```
CALL "C$XML" USING CXML-SET-ENCODING
                   handle
                   encoding
```

Parameters:

CXML-SET-ENCODING	Constant	
handle	USAGE HANDLE	A XML parser handle.
encoding	PIC X(n)	Specifies the new XML encoding. All the canonical names listed in the following Java documentation can be used: http://java.sun.com/javase/6/docs/technotes/guides/intl/encoding.doc.html. If a invalid encoding or no encoding are specified, then UTF-8 is assumed.

CXML-GET-STANDALONE

The CXML-GET-STANDALONE function returns the standalone pseudo-attribute of the XML document.

Syntax:

```
CALL "C$XML" USING CXML-GET-STANDALONE
                   handle
                   standalone
```

Parameters:

CXML-GET-STANDALONE	Constant	
handle	USAGE HANDLE	A XML parser handle.
standalone	PIC X(n)	Receives the standalone pseudo-attribute value. Possible values are "yes" or "no".

CXML-SET-STANDALONE

The CXML-SET-STANDALONE function changes the standalone pseudo-attribute of the XML document.

Syntax:

```
CALL "C$XML" USING CXML-SET-STANDALONE
                   handle
                   standalone
```

Parameters:

CXML-SET-STANDALONE	Constant	
handle	USAGE HANDLE	A XML parser handle.
standalone	PIC X(n)	Specifies the new standalone pseudo-attribute value. Valid values are "yes" or "no".

CBL_AND

The CBL_AND library routine compares each bit of the first operand to the corresponding bit of the second operand. If both bits are 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.

Source	Target	Result
0	0	0
0	1	0
1	0	0
1	1	1

Syntax:

```
CALL "CBL_AND" USING source
                     destination
                    [length]
              GIVING returnCode
```

Parameters:

source	PIC X(n)	Specifies the first operand.
destination	PIC X(n)	Specifies the second operand and receives the result of the operation.
length	any numeric data item or numeric literal	Specifies the number of bytes to be considered when executing the routine. When this parameter is omitted, all bytes in source are used.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
1	Operation failed.

Examples:

Example - Get the bit level AND result between characters 3 and 4

```
move "3" to char1 *> internal bit representation 00110011
move "4" to char2 *> internal bit representation 00110100
            char-and-result
call "cbl_and" using char1 char-and-result

if return-code = 0
   display message "AND operation between " char1 " and " char2
                   " is " char-and-result
*> char-and-result will be 0 (with internal bit representation 00110000)
end-if
```

CBL_CHANGE_DIR

The CBL_CHANGE_DIR library routine changes the current directory for the COBOL program.

This routine changes the working directory only for files opened by the COBOL program.

Syntax:

```
CALL "CBL_CHANGE_DIR" USING pathName
                      GIVING returnCode
```

Parameters:

pathName	PIC X(n)	Specifies the new current directory. Must be terminated by space or low-value.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
14605	Not found.
14613	Directory.
14628	Exists.
14629	No permission.

Examples:

Example - Change current directory to a temporary reports one

```
*> define new-dir as pic x(n)
move "c:\tmp\reports" to new-dir
call "cbl_change_dir" using new-dir
if return-code = 0
   display message "Current Directory changed to " new-dir
else
   display message "Error : " return-code
end-if
```

CBL_CHECK_FILE_EXIST

The CBL_CHECK_FILE_EXIST library routine checks if a file exists. If the file exists, the routine returns information about the file.

Syntax:

```
CALL "CBL_CHECK_FILE_EXIST" USING fileName
                                  fileDetails
                            GIVING returnCode
```

Parameters:

fileName	PIC X(n)	Specifies the name of the file to check. The name can be partially or entirely changed through configuration properties if iscobol.file.env_naming (boolean) is set to true. If the name is a relative path and iscobol.file.prefix is set, then the first FILE-PREFIX path is used to locate the file. The ISF protocol is not supported, it will invalidate the file path, if used.

fileDetails	Group Item	Receives the file information. It must have the following structure:
		
		`01 cblt-fileexist-buf.`
		`  03 cblte-fe-filesize  pic x(8) comp-x.`
		`  03 cblte-fe-date.`
		`  05 cblte-fe-day  pic x comp-x.`
		`  05 cblte-fe-month  pic x comp-x.`
		`  05 cblte-fe-year  pic x(2) comp-x.`
		`  03 cblte-fe-time.`
		`  05 cblte-fe-hours  pic x comp-x.`
		`  05 cblte-fe-minutes  pic x comp-x.`
		`  05 cblte-fe-seconds  pic x comp-x.`
		`  05 cblte-fe-hundredths pic x comp-x.`

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
>0	Operation failed. See Interpreting the return code as a file status code.

Examples:

Example - Check if file exist and return information about it

```
working-storage section.
01 cbl-fileexist-buf.
   03 cblte-fe-filesize       pic x(8) comp-x.
   03 cblte-fe-date.
      05 cblte-fe-day         pic x comp-x.
      05 cblte-fe-month       pic x comp-x.
      05 cblte-fe-year        pic x(2) comp-x.
   03 cblte-fe-time.
      05 cblte-fe-hours       pic x comp-x.
      05 cblte-fe-minutes     pic x comp-x.
      05 cblte-fe-seconds     pic x comp-x.
      05 cblte-fe-hundredths  pic x comp-x.
...
procedure division.
...
check-file-exist.
   call "cbl_check_file_exist"
                        using "c:\app1\config\settings.txt"
                              cbl-fileexist-buf
   if return-code = 0
      display message "File size : " cblte-fe-filesize
   else
      display message "File not found"
   end-if.
```

CBL_CLEAR_SCR

The CBL_CLEAR_SCR library routine clears the entire screen using a specified character and attribute.

This routine is implemented for compatibility with other COBOLs in order to facilitate migrations. Using this routine in new programs is not suggested; you should rely on DISPLAY and ACCEPT verbs instead.

Syntax:

```
CALL "CBL_CLEAR_SCR" USING character
                           attribute
                     GIVING returnCode
```

Parameters:

character	PIC X COMP-X	Specifies the character to write.
attribute	PIC X COMP-X	Specifies the attribute to write.

Return code:

returnCode can be any numeric data item and provides additional information:

1	Operation successful.

0	Operation failed.

CBL_CLOSE_FILE

The CBL_CLOSE_FILE library routine closes a file opened for byte-stream operations.

Syntax:

```
CALL "CBL_CLOSE_FILE" USING fileHandle
                      GIVING returnCode
```

Parameters:

fileHandle	PIC X(4) COMP-X	A handle returned from CBL_OPEN_FILE or CBL_CREATE_FILE.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
>0	Operation failed, the return code must be processed as file status.

Examples:

Example - Close a previously opened byte-stream

```
      ...
      working-storage section.
      ...
      77 file-handle pic x(4) comp-x.
      ...
      procedure division.
      ...
          call "cbl_close_file" using file-handle.
      ...
```

CBL_COPY_FILE

The CBL_COPY_FILE library routine copies a file.

Syntax:

```
CALL "CBL_COPY_FILE" USING sourceFile
                           destFile
                          [flagPreserve]
                    GIVING returnCode
```

Parameters:

sourceFile	PIC X(n)	Specifies the filename to be copied. The name can be partially or entirely changed through configuration properties if iscobol.file.env_naming (boolean) is set to true. If the name is a relative path and iscobol.file.prefix is set, then the first FILE-PREFIX path is used to locate the file. The ISF protocol is not supported, it will invalidate the file path, if used.
destFile	PIC X(n)	Specifies the destination filename. The name can be partially or entirely changed through configuration properties if iscobol.file.env_naming (boolean) is set to true. If the name is a relative path and iscobol.file.prefix is set, then the first FILE-PREFIX path is used to locate the file. The ISF protocol is not supported, it will invalidate the file path, if used.
flagPreserve	PIC 9	Optional parameter. If set to "1" applies the source file last modification date and time to the destination file. If set to "0" or omitted, the destination file is created with the current date and time.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
>0	Operation failed. See Interpreting the return code as a file status code.

Examples:

Example - Copy a report from main reports folder to users folder

```
*> Define src-file and dest-file as pic x(n)
move "c:\app1\reports\repcustomers.lst" to src-file
move "c:\users\adam\reports\repcustomers.lst" to dest-file
call "cbl_copy_file" using src-file dest-file
if return-code not = 0
   display message "Copy failed"
end-if
```

CBL_CREATE_DIR

The CBL_CREATE_DIR library routine creates a new directory. If the directory already exists, the routine fails.

Syntax:

```
CALL "CBL_CREATE_DIR" USING directoryName
                      GIVING returnCode
```

Parameters:

directoryName	PIC X(n)	Specifies the name of the directory to be created. Both full and relative paths are allowed.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
>0	Operation failed. See Interpreting the return code as a file status code.

Examples:

Example - Create user's report directory

```
*> define dir-name as pic x(n)
move "c:\users\adam\reports" to dir-name
call "cbl_create_dir" using dir-name
if return-code not = 0
   display message "Directory creation failed"
end-if
```

CBL_CREATE_FILE

The CBL_CREATE_FILE library routine creates a new file and leaves it open for byte-stream operations.

Syntax:

```
CALL "CBL_CREATE_FILE" USING fileName
                             accessMode
                             denyMode
                             device
                             fileHandle
                       GIVING returnCode
```

Parameters:

fileName	PIC X(n)	Specifies the name of the file to create.
accessMode	PIC X COMP-X	Specifies the access mode. Possible values are: **1...** Read only **2...** Write only (*denyMode* must be 0) **3...** Read/write **64...** Read/write files greater than 4Gb
denyMode	PIC X COMP-X	Specifies the deny mode. Possible values are: **0...** Deny both read and write (exclusive) **1...** Deny write **2...** Deny read **3...** Deny neither read nor write
device	PIC X COMP-X	This item must be set to zero.
fileHandle	PIC X(4) COMP-X	Returnsanhandletothecreatedfileunlessanerroroccurs.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
>0	Operation failed. See Interpreting the return code as a file status code.

Examples:

Example - Create a file and put the return code into file status if the operation fails

```
working-storage section.
...
01  file-status-group.
    03  file-status     pic xx comp-x.
    03  redefines file-status.
        05  fs-byte-1  pic x.
        05  fs-byte-2  pic x comp-x.
01 file-name   pic x(32) value "test".
01 access-mode pic x comp-x value 3.
01 deny-mode   pic x comp-x value 3.
01 device      pic x comp-x value 0.
01 file-handle pic x(4) comp-x.

...
procedure division.
...
call "cbl_create_file" using file-name, access-mode,
                             deny-mode, device, file-handle.
if return-code not = 0
    move return-code to file-status
...
```

CBL_DELETE_DIR

The CBL_DELETE_DIR library routine removes an empty directory.

Syntax:

```
CALL "CBL_DELETE_DIR" USING directoryName
                      GIVING returnCode
```

Parameters:

directoryName	PIC X(n)	Specifies the name of the directory to be removed. Both full and relative paths are allowed.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
>0	Operation failed. See Interpreting the return code as a file status code.

Examples:

Example - Delete temporary sorts directory

```
*> define dir-name as pic x(n)
move "c:\tmp\sorts1" to dir-name
call "cbl_delete_dir" using dir-name
if return-code not = 0
   display message "Directory deletion failed"
end-if
```

CBL_DELETE_FILE

The CBL_DELETE_FILE library routine deletes a file.

Syntax:

```
 CALL "CBL_DELETE_FILE" USING fileName
                        GIVING returnCode
```

Parameters:

fileName	PIC X(n)	Specifies the name of the file to be deleted. The name can be partially or entirely changed through configuration properties if iscobol.file.env_naming (boolean) is set to true. If the name is a relative path and iscobol.file.prefix is set, then the first FILE-PREFIX path is used to locate the file. The ISF protocol is not supported, it will invalidate the file path, if used.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
>0	Operation failed. See Interpreting the return code as a file status code.

Examples:

Example - Delete temporary sort file

```
*> define file-name as pic x(n)
move "c:\tmp\sorts1\sortcust" to file-name
call "cbl_delete_file" using file-name
if return-code not = 0
   display message "File deletion failed"
end-if
```

CBL_DIR_SCAN_END

The CBL_DIR_SCAN_END library routine closes a list of files.

Syntax:

```
CALL "CBL_DIR_SCAN_END" USING dirHandle
                        GIVING returnCode
```

Parameters:

dirHandle	Handle	It must point to a valid handle returned by CBL_DIR_SCAN_START.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
1	An error occurred.

Examples:

Example - Open a directory, list its contents and then close the list with this routine

```
working-storage section.
77  hDir                    handle.

01  pattern.
    03 pattern-length       pic x(2) comp-n.
    03 pattern-content      pic x(128).

77  search-attribute        pic x(4) comp-n.
77  search-flags            pic x(4) comp-n.

01  dir-entry.
    03  dir-attribute       pic x(4) comp-n.
    03  dir-date-stamp.
        05 dir-year         pic x(4) comp-n.
        05 dir-month        pic x(2) comp-n.
        05 dir-day          pic x(2) comp-n.
        05 dir-hour         pic x(2) comp-n.
        05 dir-minute       pic x(2) comp-n.
        05 dir-second       pic x(2) comp-n.
        05 dir-millisec     pic x(2) comp-n.
        05 dir-dst          pic x(1) comp-n.
        05 dir-size         pic x(8) comp-n.
        05 dir-name.
           07 dir-name-len  pic x(2) comp-n value 32.
           07 dir-entry-name pic x(32).
...
procedure division.
...
list-directory.
   initialize  pattern
   move "./*"  to pattern-content
   move 3      to pattern-length
   move 1      to search-attribute
   move 3      to search-flags
   call "cbl_dir_scan_start" using hDir
        pattern
        search-attribute
        search-flags
   if return-code not = 0
      display message "Invalid directory"
      exit paragraph
   end-if
   perform until exit
      initialize dir-entry-name
      call "cbl_dir_scan_read" using hDir, dir-entry
      if return-code = 0
         display dir-entry-name
      else
         exit perform
      end-if
   end-perform
   call "cbl_dir_scan_end" using hDir.
```

CBL_DIR_SCAN_READ

The CBL_DIR_SCAN_READ library routine returns the next item in a list of files.

Syntax:

```
CALL "CBL_DIR_SCAN_READ" USING dirHandle
                               entry
                         GIVING returnCode
```

Parameters:

dirHandle	Handle	It must point to a valid handle returned by CBL_DIR_SCAN_START.

entry	Group Item	A group item defined as follows:
		
		`01 entry.`
		`  03 attribute  pic x(4) comp-n.`
		`  03 date-stamp.`
		`  05 year  pic x(4) comp-n.`
		`  05 month  pic x(2) comp-n.`
		`  05 day  pic x(2) comp-n.`
		`  05 hour  pic x(2) comp-n.`
		`  05 minute  pic x(2) comp-n.`
		`  05 second  pic x(2) comp-n.`
		`  05 millisec  pic x(2) comp-n.`
		`  05 dst  pic x comp-n.`
		`  05 size  pic x(8) comp-n.`
		`  05 name.`
		`  07 max-len  pic x(2) comp-n.`
		`  07 entry-name pic x(max-len).`

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
1	An error occurred.

Examples:

Example - Open a directory, read all contents with CBL_DIR_SCAN_READ and then close it

```
working-storage section.
77  hDir                    handle.

01  pattern.
    03 pattern-length       pic x(2) comp-n.
    03 pattern-content      pic x(128).

77  search-attribute        pic x(4) comp-n.
77  search-flags            pic x(4) comp-n.

01  dir-entry.
    03  dir-attribute       pic x(4) comp-n.
    03  dir-date-stamp.
        05 dir-year         pic x(4) comp-n.
        05 dir-month        pic x(2) comp-n.
        05 dir-day          pic x(2) comp-n.
        05 dir-hour         pic x(2) comp-n.
        05 dir-minute       pic x(2) comp-n.
        05 dir-second       pic x(2) comp-n.
        05 dir-millisec     pic x(2) comp-n.
        05 dir-dst          pic x(1) comp-n.
        05 dir-size         pic x(8) comp-n.
        05 dir-name.
           07 dir-name-len  pic x(2) comp-n value 32.
           07 dir-entry-name pic x(32).
...
procedure division.
...
list-directory.
   initialize  pattern
   move "./*"  to pattern-content
   move 3      to pattern-length
   move 1      to search-attribute
   move 3      to search-flags
   call "cbl_dir_scan_start" using hDir
        pattern
        search-attribute
        search-flags
   if return-code not = 0
      display message "Invalid directory"
      exit paragraph
   end-if
   perform until exit
      initialize dir-entry-name
      call "cbl_dir_scan_read" using hDir, dir-entry
      if return-code = 0
         display dir-entry-name
      else
         exit perform
      end-if
   end-perform
   call "cbl_dir_scan_end" using hDir.
```

CBL_DIR_SCAN_START

The CBL_DIR_SCAN_START library routine opens a list of files.

Syntax:

```
CALL "CBL_DIR_SCAN_START" USING dirHandle
                                pattern
                                searchAttribute
                                flags
                          GIVING returnCode
```

Parameters:

dirHandle	Handle	Receives the handle of the directory.

pattern	Group Item	Group item defined as follows:
		
		`01 pattern`
		`  03 pattern-length   pic x(2) comp-n.`
		`  03 pattern-content  pic x(pattern-length).`
		
		pattern-content can contain a full or partial directory path with or without a filename or just a filename.

searchAttribute	PIC 9(4) COMP-5	Valid values are: **1** - Search for a file **2** - Search for a directory **4** - Search for any entry that is neither a file or a directory
flags	PIC 9(4) COMP-5	Valid values are: **1** - Turns on processing of escape sequences in *pattern* **2** - Turns on the use of wildcards in *pattern*

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
1	An error occurred.

Examples:

Example - Open a directory with CBL_DIR_SCAN_START and then get the contents of it

```
working-storage section.
77  hDir                    handle.

01  pattern.
    03 pattern-length       pic x(2) comp-n.
    03 pattern-content      pic x(128).

77  search-attribute        pic x(4) comp-n.
77  search-flags            pic x(4) comp-n.

01  dir-entry.
    03  dir-attribute       pic x(4) comp-n.
    03  dir-date-stamp.
        05 dir-year         pic x(4) comp-n.
        05 dir-month        pic x(2) comp-n.
        05 dir-day          pic x(2) comp-n.
        05 dir-hour         pic x(2) comp-n.
        05 dir-minute       pic x(2) comp-n.
        05 dir-second       pic x(2) comp-n.
        05 dir-millisec     pic x(2) comp-n.
        05 dir-dst          pic x(1) comp-n.
        05 dir-size         pic x(8) comp-n.
        05 dir-name.
           07 dir-name-len  pic x(2) comp-n value 32.
           07 dir-entry-name pic x(32).
...
procedure division.
...
list-directory.
   initialize  pattern
   move "./*"  to pattern-content
   move 3      to pattern-length
   move 1      to search-attribute
   move 3      to search-flags
   call "cbl_dir_scan_start" using hDir
        pattern
        search-attribute
        search-flags
   if return-code not = 0
      display message "Invalid directory"
      exit paragraph
   end-if
   perform until exit
      initialize dir-entry-name
      call "cbl_dir_scan_read" using hDir, dir-entry
      if return-code = 0
         display dir-entry-name
      else
         exit perform
      end-if
   end-perform
   call "cbl_dir_scan_end" using hDir.
```

CBL_EQ

The CBL_EQ library routine compares each bit of the first operand to the corresponding bit of its second operand. If both bits are the same, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.

Source	Target	Result
0	0	1
0	1	0
1	0	0
1	1	1

Syntax:

```
CALL "CBL_EQ" USING source
                    destination
                    [length]
              GIVING returnCode
```

Parameters:

source	PIC X(n)	Specifies the first operand.
destination	PIC X(n)	Specifies the second operand and receives the result of the operation.
length	any numeric data item or numeric literal	Specifies the number of bytes to be considered when executing the routine. When this parameter is omitted, all bytes in source are used.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
1	Operation failed.

Examples:

Example - Get the bit level EQ result between characters 3 and 4

```
move "3" to char1 *> internal bit representation 00110011
move "4" to char2 *> internal bit representation 00110100
            char-eq-result
call "cbl_eq" using char1 char-eq-result

if return-code = 0
   display message "EQ operation between " char1 " and " char2
                   " is " char-eq-result
*> char-eq-result will be 0xF8 (with internal bit representation 11111000)
end-if
```

CBL_ERROR_PROC

The CBL_ERROR_PROC library routine installs or removes error procedures to be called automatically if and when the current run unit generates any of certain runtime errors. This implementation calls error procedures only when a run unit generates what is called an intermediate runtime error.

In order to receive the error information, the error procedure program should include the following Linkage Section parameter:

```
       LINKAGE SECTION.
       77 ERROR-DESC PIC X(325)

       PROCEDURE DIVISION USING ERROR-DESC.
```

Note: only errors of the kind java.lang.Exception cause the error procedure to be invoked. This kind of exception is raised for most of the COBOL errors though.

Syntax:

```
 CALL "CBL_ERROR_PROC" USING installFlag
                             programName
                  [RETURNING statusCode]
```

Parameters:

installFlag	any numeric data item or numeric literal	Zero if the error procedure is to be installed; nonzero if it is to be removed.
programName	PIC X(n)	Name of the error procedure to be installed or removed.

Return code:

statusCode can be any numeric data item and it is always zero.

Examples:

Example - Use a specific program to catch runtime errors during the execution of a program, disable it at the end

```
...
set-error-proc.
   call "cbl_error_proc" using 0, "myerrtrap".
...
finish-program.
   call "cbl_error_proc" using 1, "myerrtrap"
   goback.
```

CBL_EXEC_RUN_UNIT

The CBL_EXEC_RUN_UNIT library routine creates a synchronous or asynchronous run unit that inherits the environment variables set by the caller program.

Syntax:

```
CALL "CBL_EXEC_RUN_UNIT" USING commandLine
                               commandLineLength
                               runUnitId
                               stackSize
                               flags
                     [RETURNING statusCode]
```

Parameters:

commandLine	PIC X(n)	The command passed to the new run unit. This should be the name of a executable file followed by any parameters.
commandLineLength	PIC X(4) COMP-5	Specifies the length of the *commandLine* parameter.
runUnitId	PIC X(8) COMP-5	Returns the unique handle identifying the new run-unit if bit 0 of *flags* is not set. If bit 0 of *flags* is set, this value is unchanged.
stackSize	PIC X(4) COMP-5	Ignored.
flags	PIC X(4) COMP-5	This parameter is ignored under Windows. On Linux/Unix this 32-bit word indicates how the new run unit is created as follows: Bit 0: if set to 0, the routine returns to the caller immediately after creating the child process. If set to 1, the routine waits for the new run unit to complete before returning to the caller. Bit 1: ignored Bit 2: if set to 0, the messages that the new run unit prints on stdout and stderr are lost. If set to 1, the messages that the new run unit prints on stdout and stderr are shown in the caller program's console. Bit 3: reserved, it must be set to 0.

Return code:

statusCode can be any numeric data item and provides additional information:

0	Operation successful.
181	Invalid parameters.
200	Internal error.
255	Program not found.
Other non-zero	Returned error code from the executed program.

Examples:

Example - Run the program “sub” in a separate run unit

```
working-storage section.
...
01  cmd-line            pic x(512).
01  cmd-line-len        pic x(04) comp-5 value 512.
01  run-unit-id         pic x(08) comp-5 value 01.
01  stack-size          pic x(04) comp-5 value zero.
01  run-flags           pic x(04) comp-5.
...
procedure division.
...
    move h'0007' to run-flags.
    move "iscrun sub"   to cmd-line.
    call "CBL_EXEC_RUN_UNIT" using cmd-line
                                   cmd-line-len
                                   run-unit-id
                                   stack-size
                                   run-flags.
```

CBL_EXIT_PROC

The CBL_EXIT_PROC library routine installs and removes exit procedures to be called automatically when the current run unit terminates normally.

Syntax:

```
CALL "CBL_EXIT_PROC" USING installFlag
                           programName
                 [RETURNING statusCode]
```

Parameters:

installFlag	any numeric data item or numeric literal	Zero if the exit procedure is to be installed; nonzero if it is to be removed.
programName	PIC X(n)	Name of the exit procedure to be installed or queried.

Return code:

statusCode can be any numeric data item and it is always zero.

Examples:

Example - Make a cleaning routine to be called when the current program exits normally

```
call "cbl_exit_proc" using 0, "myclean".
```

CBL_FLUSH_FILE

The CBL_FLUSH_FILE library routine ensures all file buffers for a file are written to disk.

Syntax:

```
CALL "CBL_FLUSH_FILE" USING fileHandle
                      GIVING returnCode
```

Parameters:

fileHandle	PIC X(4) COMP-X	A handle returned from CBL_OPEN_FILE or CBL_CREATE_FILE.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
>0	Operation failed. See Interpreting the return code as a file status code.

Examples:

Example - Flush a previously opened byte-stream

```
      ...
      working-storage section.
      ...
      77 file-handle pic x(4) comp-x.
      ...
      procedure division.
      ...
          call "cbl_flush_file" using file-handle.
      ...
```

CBL_GET_SCR_SIZE

The CBL_GET_SCR_SIZE library routine returns the screen size.

Syntax:

```
CALL "CBL_GET_SCR_SIZE" USING screenDepth
                              screenWidth
                        GIVING returnCode
```

Parameters:

screenDepth	PIC X COMP-X	Returns the number of lines in the screen.
screenWidth	PIC X COMP-X	Returns the number of columns in the screen.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
1	Operation failed.

Examples:

Example - Get the screen size

```
call "cbl_get_scr_size" using depth, width.
```

CBL_IMP

The CBL_IMP library routine compares each bit of the first operand to the corresponding bit of its second operand. If the bit of the second operand is 0 and the bit of the first operand is 1, the corresponding result bit is set to 0. Otherwise, the corresponding result bit is set to 1.

Source	Target	Result
0	0	1
0	1	1
1	0	0
1	1	1

Syntax:

```
CALL "CBL_IMP" USING source
                     destination
                    [length]
              GIVING returnCode
```

Parameters:

source	PIC X(n)	Specifies the first operand.
destination	PIC X(n)	Specifies the second operand and receives the result of the operation.
length	any numeric data item or numeric literal	Specifies the number of bytes to be considered when executing the routine. When this parameter is omitted, all bytes in source are used.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
1	Operation failed.

Examples:

Example - Get the bit level IMP result between characters 3 and 4

```
move "3" to char1 *> internal bit representation 00110011
move "4" to char2 *> internal bit representation 00110100
            char-imp-result
call "cbl_imp" using char1 char-imp-result

if return-code = 0
   display message "IMP operation between " char1 " and " char2
                   " is " char-imp-result
*> char-imp-result will be 0xFC (with internal bit representation 11111100)
end-if
```

CBL_JOIN_FILENAME

The CBL_JOIN_FILENAME library routine forms a filename by joining together its component parts; that is, the pathname, basename and extension.

Syntax:

```
CALL "CBL_JOIN_FILENAME" USING cbltSplitjoinBuf
                               joinBuffer
                               pathBuffer
                               basenameBuffer
                               extensionBuffer
                         GIVING returnCode
```

Parameters:

cbltSplitjoinBuf	Group Item	Group item defined as follows:
		
		`01 cblt-splitjoin-buf.`
		`  03 param-length pic x(2) comp-x.`
		`  03 split-join-flag1 pic x comp-x.`
		`  03 split-join-flag2 pic x comp-x.`
		`  03 device-offset pic x(2) comp-x.`
		`  03 device-length pic x(2) comp-x.`
		`  03 basename-offset pic x(2) comp-x.`
		`  03 basename-length pic x(2) comp-x.`
		`  03 extension-offset pic x(2) comp-x.`
		`  03 extension-length pic x(2) comp-x.`
		`  03 total-length pic x(2) comp-x.`
		`  03 split-buf-len pic x(2) comp-x.`
		`  03 join-buf-len pic x(2) comp-x.`
		`  03 first-component-length pic x(2) comp-x.`
		
		On entry:
		param-lenght is the length of the structure in bytes. It should be set to 24.
		splitjoin-flg1's bit 1 says that the strings are null-terminated if set to 1, while they are space-terminated if set to 0.
		splitjoin-flg1's bit 2 says that the filename is folded to upper case if set to 1, while the original case is preserved if set to 0.
		split-buf-len specifies the length in bytes of *joinBuffer*.
		splitjoin-flg2's bit 2 is set if there is a significant space in the filename.
		splitjoin-flg2's bit 1 is set if there is a wildcard in the path.
		splitjoin-flg2's bit 0 is set if there is a wildcard in basename or extension.
		device-offset specifies the start of pathname in *joinBuffer*, from one.
		device-length specifies the length of the pathname.
		basename-offset specifies the start of basename in *joinBuffer*, from one.
		basename-length specifies the length of the basename.
		extension-offset specifies the start of extension in *joinBuffer*, from one.
		extension-length specifies the length of the extension.
		first-component-length specifies the number of characters up to and including the first backslash or slash or colon in *joinBuffer*.
		On exit:
		total-length specifies the total number of characters in *joinBuffer*.

joinBuffer	PIC X(n)	Specifies the area to store the joined file name.
pathBuffer	PIC X(n)	Specifies the pathname.

basenameBuffer	PIC X(n)	Specifies the basename.
extensionBufffer	PIC X(n)	Specifies the extension.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
1	Buffer overflow.
4	File name is invalid.

Examples:

Example - Build the file name “C:\myfile.txt”.

```
WORKING-STORAGE SECTION.
77  fi-base-name            pic x(30) value "myfile".
77  fi-extension            pic x(10) value "txt".
77  fi-path                 pic x(256) value "C:\Temp".
01  join-buffer             pic x(256).

 PROCEDURE DIVISION.
 MAIN.
      move 1 to device-offset
                basename-offset
                extension-offset

      move length of fi-path      to device-length
      move length of fi-base-name to basename-length
      move length of fi-extension to extension-length
      move length of join-buffer  to join-buf-len

      move 0 to split-join-flag1
      move 24 to param-length

      call "CBL_JOIN_FILENAME" using cblt-splitjoin-buf
                                     join-buffer
                                     fi-path
                                     fi-base-name
                                     fi-extension
```

CBL_NOT

The CBL_NOT library routine examines each bit of the operand. If the bit is 1, the corresponding result bit is set to 0. Otherwise, the corresponding result bit is set to 1.

Before	After
0	1

Before	After
1	0

Syntax:

```
CALL "CBL_NOT" USING destination
                     [length]
               GIVING returnCode
```

Parameters:

destination	PIC X(n)	Specifies the operand and receives the result of the operation.
length	any numeric data item or numeric literal	Specifies the number of bytes to be considered when executing the routine. When this parameter is omitted, all bytes in destination are used.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
1	Operation failed.

Examples:

Example - Get the bit level NOT result of a character

```
move "3" to char1 *> internal bit representation 00110011
            char-result
call "cbl_not" using char-result
if return-code = 0
   display message "NOT " char1 " = " char-result
*> char-result will have internal bit representation 11001100
end-if
```

CBL_OPEN_FILE

The CBL_OPEN_FILE library routine opens a file for byte-stream operations.

Syntax:

```
CALL "CBL_OPEN_FILE" USING    fileName
                              accessMode
                              denyMode
                              device
                              fileHandle
                     GIVING   returnCode
```

Parameters:

fileName	PIC X(n)	Specifies the name of the file to open. The name can be partially or entirely changed through configuration properties if iscobol.file.env_naming (boolean) is set to true. If the name is a relative path and iscobol.file.prefix is set, then the first FILE-PREFIX path is used to locate the file. The ISF protocol is not supported, it will invalidate the file path, if used.
accessMode	PIC X COMP-X	Specifies the access mode. Possible values are: **1...** Read only **2...** Write only (*denyMode* must be 0) **3...** Read/write **64...** Read/write files greater than 4Gb
denyMode	PIC X COMP-X	Specifies the deny mode. Possible values are: **0...** Deny both read and write (exclusive) **1...** Deny write **2...** Deny read **3...** Deny neither read nor write
device	PIC X COMP-X	This item must be set to zero.
fileHandle	PIC X(4) COMP-X	Returnsanhandletotheopenedfileunlessanerroroccurs.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
>0	Operation failed. See Interpreting the return code as a file status code.

Examples:

Example - Open a file and put the return code into file status if the operation fails

```
working-storage section.
...
01  file-status-group.
    03  file-status     pic xx comp-x.
    03  redefines file-status.
        05  fs-byte-1  pic x.
        05  fs-byte-2  pic x comp-x.
01 file-name   pic x(32) value "test".
01 access-mode pic x comp-x value 3.
01 deny-mode   pic x comp-x value 3.
01 device      pic x comp-x value 0.
01 file-handle pic x(4) comp-x.

...
procedure division.
...
call "cbl_open_file" using file-name, access-mode,
                           deny-mode, device, file-handle.
if return-code not = 0
    move return-code to file-status
...
```

CBL_OR

The CBL_OR library routine compares each bit of the first operand to the corresponding bit of its second operand. If either bit is 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.

Source	Target	Result
0	0	0
0	1	1
1	0	1
1	1	1

Syntax:

```
CALL "CBL_OR" USING source
                    destination
                   [length]
             GIVING returnCode
```

Parameters:

source	PIC X(n)	Specifies the first operand

destination	PIC X(n)	Specifies the second operand and receives the result of the operation.
length	any numeric data item or numeric literal	Contains the number of bytes to be considered when executing the routine. When this parameter is omitted, all bytes in source are used.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
1	Operation failed.

Examples:

Example - Get the bit level OR result between characters 3 and 4

```
move "3" to char1 *> internal bit representation 00110011
move "4" to char2 *> internal bit representation 00110100
            char-result
call "cbl_or" using char1 char-result

if return-code = 0
   display message "OR operation between " char1 " and " char2
                   " is " char-result
*> char-result will be 7 (with internal bit representation 00110111)
end-if
```

CBL_READ_FILE

The CBL_READ_FILE library routine reads bytes from a file.

Syntax:

```
CALL "CBL_READ_FILE" USING fileHandle
                           offset
                           byteCount
                           flags
                           buffer
                     GIVING returnCode
```

Parameters:

fileHandle	PIC X(4) COMP-X	A handle returned from CBL_OPEN_FILE.
offset	PIC X(8) COMP-X	The offset in the file at which to read. This field is limited to a maximum value of 4294967295 unless *accessMode* is set to 64 when the file is opened using CBL_OPEN_FILE or CBL_CREATE_FILE.
byteCount	PIC X(4) COMP-X	Specifies the number of bytes to read.
flags	PIC X COMP-X	Possible values are: **0...** Returns read content in the *buffer* field **128...** Returns the file size in the *offset* field
buffer	PIC X(n)	Receives the read bytes.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
>0	Operation failed. See Interpreting the return code as a file status code.

Examples:

Example - Read the first three bytes of an opened file

```
working-storage section.
...
01 file-handle pic x(4) comp-x.
01 ofs         pic x(8) comp-x.
01 cnt         pic x(4) comp-x.
01 flg         pic x comp-x value 0.
01 buff        pic x(3).
...
procedure division.
...
move 1 to ofs.
move 3 to cnt.
call "cbl_read_file" using file-handle, ofs,
                           cnt, flg, buff.
```

CBL_READ_SCR_CHARS

The CBL_READ_SCR_CHARS library routine reads a string of characters from the screen.

This routine is implemented for compatibility with other COBOLs in order to facilitate migrations. Using this routine in new programs is not suggested; you should rely on DISPLAY and ACCEPT verbs instead.

Syntax:

```
CALL "CBL_READ_SCR_CHARS" USING screenPosition
                                characterBuffer
                                stringLength
                          GIVING returnCode
```

Parameters:

screenPosition	Group Item	Group item is defined as follows:
		
		`01 screenPosition.`
		`  03 rowNumber PIC X COMP-X.`
		`  03 columnNumber PIC X COMP-X.`
		
		It specifies the screen position at which to start reading. The top left corner is row 0, column 0.

characterBuffer	PIC X(n)	It returns the characters read from the screen. It must be at least the length specified by *stringLength*. Positions in it beyond that length are unchanged.
stringLength	PIC XX COMP-X	On entry, contains the length of the string to read. On exit, contains the length of the string read when the end of the screen is reached

Return code:

returnCode can be any numeric data item and provides additional information:

1	Operation successful.
0	Operation failed.

CBL_READ_SCR_CHATTRS

The CBL_READ_SCR_CHATTRS library routine reads a string of characters and their corresponding attributes from the screen.

This routine is implemented for compatibility with other COBOLs in order to facilitate migrations. Using this routine in new programs is not suggested; you should rely on DISPLAY and ACCEPT verbs instead.

Syntax:

```
CALL "CBL_READ_SCR_CHATTRS" USING screenPosition
                                  characterBuffer
                                  attributeBuffer
                                  stringLength
                            GIVING returnCode
```

Parameters:

screenPosition	Group Item	Group item is defined as follows:
		
		`01 screenPosition.`
		`  03 rowNumber PIC X COMP-X.`
		`  03 columnNumber PIC X COMP-X.`
		
		It specifies the screen position at which to start reading. The top left corner is row 0, column 0.

characterBuffer	PIC X(n)	It returns the characters read from the screen. It must be at least the length specified by *stringLength*. Positions in it beyond that length are unchanged.
attributeBuffer	PIC X(n)	On exit, this data item contains the attributes read from the screen. It must be at least the length specified by *stringLength*. Positions in the data item beyond that length are unchanged.
stringLength	PIC XX COMP-X	On entry, contains the length of the string to read. On exit, contains the length of the string read when the end of the screen is reached

Return code:

returnCode can be any numeric data item and provides additional information:

1	Operation successful.
0	Operation failed.

CBL_RENAME_FILE

The CBL_RENAME_FILE library routine renames a file.

Syntax:

```
CALL "CBL_RENAME_FILE" USING oldName
                             newName
                       GIVING returnCode
```

Parameters:

oldName	PIC X(n)	Specifies the name of the file that you want to rename. If no path is given, then the current directory is assumed. The name can be partially or entirely changed through configuration properties if iscobol.file.env_naming (boolean) is set to true. If the name is a relative path and iscobol.file.prefix is set, then the first FILE-PREFIX path is used to locate the file. The ISF protocol is not supported, it will invalidate the file path, if used.
newName	PIC X(n)	Specifies the new name for the file. If no path is given, then the current directory is assumed. The name can be partially or entirely changed through configuration properties if iscobol.file.env_naming (boolean) is set to true. If the name is a relative path and iscobol.file.prefix is set, then the first FILE-PREFIX path is used to locate the file. The ISF protocol is not supported, it will invalidate the file path, if used.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
>0	Operation failed. See Interpreting the return code as a file status code.

Examples:

Example - Rename file to append the date it was generated

```
*> define curr-date as pic x(6)
*> new-name and old-name as pic x(n)
accept curr-date from date
initialize new-name
string "c:\app1\reps\cust-report-"
       curr-date
       ".lst"
  into new-name
end-string
move "c:\app1\reps\cust-report.lst" to old-name
call "cbl_rename_file" old-name new-name
if return-code not = 0
   display message "Rename operation failed"
end-if
```

CBL_SPLIT_FILENAME

The CBL_SPLIT_FILENAME library routine splits a filename into its component parts; that is, the pathname, basename and extension.

Syntax:

```
CALL "CBL_SPLIT_FILENAME" USING cbltSplitjoinBuf
                                splitBuffer
                          GIVING returnCode
```

Parameters:

cbltSplitjoinBuf	Group Item	Group item defined as follows:
		
		`01 cblt-splitjoin-buf.`
		`  03 param-length pic x(2) comp-x.`
		`  03 split-join-flag1 pic x comp-x.`
		`  03 split-join-flag2 pic x comp-x.`
		`  03 device-offset pic x(2) comp-x.`
		`  03 device-length pic x(2) comp-x.`
		`  03 basename-offset pic x(2) comp-x.`
		`  03 basename-length pic x(2) comp-x.`
		`  03 extension-offset pic x(2) comp-x.`
		`  03 extension-length pic x(2) comp-x.`
		`  03 total-length pic x(2) comp-x.`
		`  03 split-buf-len pic x(2) comp-x.`
		`  03 join-buf-len pic x(2) comp-x.`
		`  03 first-component-length pic x(2) comp-x.`
		
		On entry:
		param-lenght is the length of the structure in bytes. It should be set to 24. The value 0 is treated as if it were the structure dimension.
		splitjoin-flag1's bit 1 says that the strings are null-terminated if set to 1, while they are space-terminated if set to 0.
		splitjoin-flag1's bit 2 says that the filename is folded to upper case if set to 1, while the original case is preserved if set to 0.
		split-buf-len specifies the length in bytes of splitBuffer.
		
		On exit:
		splitjoin-flag2's bit 2 is set if there is a significant space in the filename.
		splitjoin-flag2's bit 1 is set if there is a wildcard in the path.
		splitjoin-flag2's bit 0 is set if there is a wildcard in basename or extension.
		device-offset returns the start of pathname in splitBuffer, from one.
		device-length returns the length of the pathname.
		basename-offset returns the start of basename in splitBuffer, from one.
		basename-length returns the length of the basename.
		extension-offset returns the start of extension in splitBuffer, from one.
		extension-length returns the length of the extension.
		total-length returns the total number of characters in splitBuffer.
		first-component-length returns the number of characters up to and including the first backslash or slash or colon in splitBuffer.

splitBuffer	PIC X(n)	It specifies the string to split.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
4	Invalid filename.

Examples:

Example - Get the extension in a full pathname

```
move "C:\temp\print_01.pdf" to the-path. |the-path is a pic x(n) data-item
call "CBL_SPLIT_FILENAME" using cblt-splitjoin-buf, the-path.
display the-path(extension-offset:extension-length)|it will display 'pdf'
```

CBL_TOLOWER

The CBL_TOLOWER library routine converts a string of letters to lower case.

Syntax:

```
CALL "CBL_TOLOWER" USING textString
                   BY VALUE length
                     GIVING returnCode
```

Parameters:

textString	PIC X(n)	Specifies the string to convert.
length	PIC X(4) COMP-5	Specifies the number of bytes of string to change; positions beyond this are unchanged.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
-1	Operation failed.

Examples:

Example - Convert the content of wrk-string to lower case

```
       working-storage section.
       77 wrk-string pic x(12).
       77 wrk-len    pic x(4) comp-5.

       procedure division.
       main.
           move "Test String" to wrk-string.
           move 12 to wrk-len.
           call "CBL_TOLOWER" using wrk-string, wrk-len.
```

CBL_TOUPPER

The CBL_TOUPPER library routine converts a string of letters to upper case.

Syntax:

```
CALL "CBL_TOUPPER" USING textString
                 BY VALUE length
                   GIVING returnCode
```

Parameters:

textString	PIC X(n)	Specifies the string to convert.
length	PIC X(4) COMP-5	Specifies the number of bytes of string to change; positions beyond this are unchanged.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
-1	Operation failed.

Examples:

Example - Convert the content of wrk-string to upper case

```
       working-storage section.
       77 wrk-string pic x(12).
       77 wrk-len    pic x(4) comp-5.

       procedure division.
       main.
           move "Test String" to wrk-string.
           move 12 to wrk-len.
           call "CBL_TOUPPER" using wrk-string, wrk-len.
```

CBL_WRITE_FILE

The CBL_WRITE_FILE library routine writes bytes from a file.

Syntax:

```
CALL "CBL_WRITE_FILE" USING fileHandle
                            offset
                            byteCount
                            flags
                            buffer
                      GIVING returnCode
```

Parameters:

fileHandle	PIC X(4) COMP-X	AhandlereturnedfromCBL_OPEN_FILEorCBL_CREATE_FILE.
offset	PIC X(8) COMP-X	The offset in the file at which to write. This field is limited to a maximum value of 4294967295 unless *accessMode* is set to 64 when the file is opened using CBL_OPEN_FILE or CBL_CREATE_FILE.
byteCount	PIC X(4) COMP-X	Specifies the number of bytes to write. Setting this parameter to zero causes the file to be truncated or extended to the size specified in the file-offset field.
flags	PIC X COMP-X	Must be zero.
buffer	PIC X(n)	Specifies the bytes to be written.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
>0	Operation failed. See Interpreting the return code as a file status code.

Examples:

Example - Rewrite the first three bytes of an opened file

```
working-storage section.
...
01 file-handle pic x(4) comp-x.
01 ofs         pic x(8) comp-x.
01 cnt         pic x(4) comp-x.
01 flg         pic x comp-x value 0.
01 buff        pic x(3).
...
procedure division.
...
move 1 to ofs.
move 3 to cnt.
move "xxx" to buff.
call "cbl_write_file" using file-handle, ofs,
                            cnt, flg, buff.
```

CBL_WRITE_SCR_CHARS

The CBL_WRITE_SCR_CHARS library routine writes a string of characters to the screen.

This routine is implemented for compatibility with other COBOLs in order to facilitate migrations. Using this routine in new programs is not suggested; you should rely on DISPLAY and ACCEPT verbs instead.

Syntax:

```
CALL "CBL_WRITE_SCR_CHARS" USING screenPosition
                                 characterBuffer
                                 stringLength
                           GIVING returnCode
```

Parameters:

screenPosition	Group Item	Group item defined as follows:
		
		`01 screenPosition.`
		`  03 rowNumber PIC X COMP-X.`
		`  03 columnNumber PIC X COMP-X.`
		
		It specifies the screen position at which to start writing. The top left corner is row 0, column 0.

characterBuffer	PIC X(n)	It specifies the characters to write to the screen.
stringLength	PIC XX COMP-X	On entry, contains the length of the string to write.

Return code:

returnCode can be any numeric data item and provides additional information:

1	Operation successful.
0	Operation failed.

CBL_WRITE_SCR_CHATTRS

The CBL_WRITE_SCR_CHATTRS library routine writes a string of characters and their attributes to the screen.

This routine is implemented for compatibility with other COBOLs in order to facilitate migrations. Using this routine in new programs is not suggested; you should rely on DISPLAY and ACCEPT verbs instead.

Syntax:

```
CALL "CBL_WRITE_SCR_CHATTRS" USING screenPosition
                                   characterBuffer
                                   attributeBuffer
                                   stringLength
                             GIVING returnCode
```

Parameters:

screenPosition	Group Item	Group item defined as follows:
		
		`01 screenPosition.`
		`  03 rowNumber PIC X COMP-X.`
		`  03 columnNumber PIC X COMP-X.`
		
		It specifies the screen position at which to start writing. The top left corner is row 0, column 0.

characterBuffer	PIC X(n)	It specifies the characters to write to the screen.
attributeBuffer	PIC X(n)	It specifies the character attributes.
stringLength	PIC XX COMP-X	On entry, contains the length of the string to write.

Return code:

returnCode can be any numeric data item and provides additional information:

1	Operation successful.
0	Operation failed.

CBL_WRITE_SCR_N_CHAR

The CBL_WRITE_SCR_N_CHAR library routine writes a specified character to a string of positions on the screen.

This routine is implemented for compatibility with other COBOLs in order to facilitate migrations. Using this routine in new programs is not suggested; you should rely on DISPLAY and ACCEPT verbs instead.

Syntax:

```
CALL "CBL_WRITE_SCR_N_CHAR" USING screenPosition
                                  character
                                  stringLength
                            GIVING returnCode
```

Parameters:

screenPosition	Group Item	Group item defined as follows:
		
		`01 screenPosition.`
		`  03 rowNumber PIC X COMP-X.`
		`  03 columnNumber PIC X COMP-X.`
		
		It specifies the screen position at which to start writing. The top left corner is row 0, column 0.

character	PIC X COMP-X	It specifies the character to write to the screen.
stringLength	PIC XX COMP-X	On entry, contains the length of the character string.

Return code:

returnCode can be any numeric data item and provides additional information:

1	Operation successful.
0	Operation failed.

CBL_WRITE_SCR_N_CHATTR

The CBL_WRITE_SCR_N_CHATTR library routine writes a specified character and attribute to a string of positions on the screen.

This routine is implemented for compatibility with other COBOLs in order to facilitate migrations. Using this routine in new programs is not suggested; you should rely on DISPLAY and ACCEPT verbs instead.

Syntax:

```
CALL "CBL_WRITE_SCR_N_CHATTR" USING screenPosition
                                    character
                                    attributeBuffer
                                    stringLength
                              GIVING returnCode
```

Parameters:

screenPosition	Group Item	Group item defined as follows:
		
		`01 screenPosition.`
		`  03 rowNumber PIC X COMP-X.`
		`  03 columnNumber PIC X COMP-X.`
		
		It specifies the screen position at which to start writing. The top left corner is row 0, column 0.

character	PIC X COMP-X	It specifies the character to write to the screen.
attributeBuffer	PIC X(n)	It specifies the character attributes.
stringLength	PIC XX COMP-X	On entry, contains the length of the character string.

Return code:

returnCode can be any numeric data item and provides additional information:

1	Operation successful.
0	Operation failed.

CBL_XOR

The CBL_XOR library routine compares each bit of its first operand to the corresponding bit of its second operand. If one bit is 0 and the other bit is 1, the corresponding result bit is set to 1. Otherwise, the corresponding result bit is set to 0.

Source	Target	Result
0	0	0
0	1	1
1	0	1
1	1	0

Syntax:

```
CALL "CBL_XOR" USING source
                     destination
                    [length]
              GIVING returnCode
```

Parameters:

source	PIC X(n)	Specifies the first operand.
destination	PIC X(n)	Specifies the second operand and receives the result of the operation.
length	any numeric data item or numeric literal	Specifies the number of bytes to be considered when executing the routine. When this parameter is omitted, all bytes in source are used.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
1	Operation failed.

Examples:

Example - Get the bit level XOR result between characters 3 and 4

```
move "3" to char1 *> internal bit representation 00110011
move "4" to char2 *> internal bit representation 00110100
            char-result
call "cbl_xor" using char1 char-result

if return-code = 0
   display message "XOR operation between " char1 " and " char2
                   " is " char-result
*> char-result will have internal bit representation 00000111
end-if
```

DCI

isCOBOL supports the following DCI routines:

- DCI_BLOB_ERROR
- DCI_BLOB_GET
- DCI_BLOB_PUT
- DCI_DISCONNECT
- DCI_FREE_XFD
- DCI_GETENV
- DCI_GET_TABLE_NAME
- DCI_GET_TABLE_SERIAL_VALUE
- DCI_SETENV
- DCI_SET_TABLE_CACHE
- DCI_SET_WHERE_CONSTRAINT

Refer to the DCI Manual from Casemaker for more information about the usage of DCI routines.

DCI routines work on the active file handler between DCI and The DCI File Connector. The file handler is activated at the first OPEN. If DCI routines are called before the first OPEN, they work on the file handler pointed by iscobol.file.index setting. If such setting is neither 'dci' nor 'dcic', then DCI routines work on the DCI file handler.

Due to the above rule, if you have a configuration like this:

```
# all files managed by jisam except few files managed by dcic

iscobol.file.index=jisam
iscobol.file.index.file1=dcic
iscobol.file.index.file2=dcic
```

and you wish to call DCI_SETENV (e.g. to set the database) before opening file1 or file2, then you need to act like this:

```
      *temporarily set file.index to dcic
           set environment "file.index" to "dcic"
      *call dci_setenv
           call "dci_setenv" using "dci_database" "mydb"
      *restore file.index to jisam
           set environment "file.index" to "jisam"
```

DELETE

The DELETE library routine deletes a file.

Syntax:

```
CALL "DELETE" USING fileName
                    [exitCode]
```

Parameters:

fileName	PIC X(n)	Specifies the full or relative pathname of the file to be deleted.
exitCode	PIC S9(4) BINARY	Receives the exit code of the command upon return from the operating system. The value is dependent on the underlying operating system. A value of 0, however, indicates success and a non-zero value indicates an error.

Examples:

Example - Delete the temporary file foo:

```
call "delete" using "C:\Temp\foo".
```

EDBI_DISCONNECT

The EDBI_DISCONNECT library routine disconnects from a database while working with isCOBOL Database Bridge. When called without parameters, it disconnects the current connection.

Syntax:

```
CALL "EDBI_DISCONNECT" [USING opCode
                               parameters]
                       GIVING returnCode
```

Parameters:

opCode	It is the function to be executed. Valid values, defined in iscobol.def, are:	
	omitted	Disconnects from the current connection.
	EDBI-DISCONNECT-CONNECTION	Disconnects from a named connection.
	EDBI-DISCONNECT-ALL	Disconnects from all connections.
parameters	Parameters are used only by EDBI-DISCONNECT-CONNECTION opCode.	

Return code:

returnCode is zero if the disconnection was successful, 1 if the disconnection failed.

Examples:

Example - Disconnects from current connection

```
call "edbi_disconnect"
if return-code not = 0
   display "Failed to disconnect from current connection"
end-if
```

Example - Disconnects from specific connection to Oracle with custom name "OraConn1" when using multiple connections

```
call "edbi_disconnect" using edbi-disconnect-connection
                             "OraConn1"
if return-code not = 0
   display "Failed to disconnect from OraConn1"
end-if
```

Example - Disconnects from all connections when using multiple connections

```
call "edbi_disconnect" using edbi-disconnect-all
if return-code not = 0
   display "Failed to disconnect from all connections"
end-if
```

EDBI-DISCONNECT-CONNECTION

The EDBI-DISCONNECT-CONNECTION function allows to disconnect from a named connection.

Syntax:

```
 CALL "EDBI_DISCONNECT" USING EDBI-DISCONNECT-CONNECTION
                              connectionName
                        GIVING returnCode
```

Parameters:

EDBI-DISCONNECT-CONNECTION	Constant	
connectionName	PIC X(n)	Specifies the name of the connection to close. Please consult Database Bridge documentation, chapter Working with multiple connections, for details about connection names.

Return code:

returnCode is zero if the disconnection was successful, 1 if the disconnection failed.

EDBI-DISCONNECT-ALL

The EDBI-DISCONNECT-ALL function allows to disconnect from all the active connections open by isCOBOL Database Bridge.

Syntax:

```
CALL "EDBI_DISCONNECT" USING EDBI-DISCONNECT-ALL
                       GIVING returnCode
```

Parameters:

EDBI-DISCONNECT-ALL	Constant

Return code:

returnCode is zero if the disconnection was successful, 1 if the disconnection failed.

ESQL$BLOB

The ESQL$BLOB library routine provides a number of functions to handle BLOB data in ESQL programs.

Syntax:

```
CALL "ESQL$BLOB" USING opCode
                       parameters
                 GIVING returnCode
```

Parameters:

opCode	It is the function to be executed. Valid values, defined in iscobol.def, are:	
	GET-BLOB-FROM-FILE	Read BLOB data from a file on disc.
	PUT-BLOB-INTO-FILE	Write the content of a BLOB into a file on disc.
	FREE-BLOB-HANDLE	Free memory initializing the BLOB handle.
parameters	Parameters depend on the opcode.	

Return code:

returnCode definition and meaning depend on the opcode.

Examples:

Example - The following program creates a test table and stores the file *img1.bmp* inside it as a blob. After it,

the program reads the previously inserted record and writes the blob to a file named *blob.bmp*.

```
       PROGRAM-ID. readwriteblob.

       WORKING-STORAGE SECTION.
       copy "SQLCA".
       copy "iscobol.def".

       01 W-REC.
        03 W-KEY   pic 9(4).
        03 W-DATA  pic x(30).
        03 W-BLOB  HANDLE.

       PROCEDURE DIVISION.
       Main.
           CALL "ESQL$BLOB" USING GET-BLOB-FROM-FILE, W-BLOB,"img1.bmp".

           EXEC SQL
                CONNECT
           END-EXEC

           EXEC SQL
                DROP TABLE IS_TABLE
           END-EXEC

           EXEC SQL
                CREATE TABLE IS_TABLE
                             (IS_KEY INT NOT NULL,
                              IS_DATA CHAR(6),
                              IS_BLOB BLOB)
           END-EXEC

           EXEC SQL
                ALTER TABLE IS_TABLE ADD PRIMARY KEY (IS_KEY)
           END-EXEC

           EXEC SQL INSERT INTO IS_TABLE VALUES (1, 'row1',
                                                 :W-BLOB)
           END-EXEC

           CALL "ESQL$BLOB" USING FREE-BLOB-HANDLE, W-BLOB.

           EXEC SQL
                SELECT * INTO :W-KEY, :W-DATA, :W-BLOB
                             FROM IS_TABLE
                             WHERE IS_KEY = 1
           END-EXEC

           CALL "ESQL$BLOB" USING PUT-BLOB-INTO-FILE, W-BLOB, "blob.bmp".

           EXEC SQL
                DISCONNECT
           END-EXEC

           GOBACK.
```

GET-BLOB-FROM-FILE

The GET-BLOB-FROM-FILE function reads a binary file on the disc and generates a BLOB providing a handle to it. The handle can be used as host variable in INSERT and UPDATE ESQL statements.

Syntax:

```
CALL "ESQL$BLOB" USING GET-BLOB-FROM-FILE
                       blobHandle
                       fileName
                 GIVING returnCode
```

Parameters:

GET-BLOB-FROM-FILE	Constant	
blobHandle	USAGE HANDLE	It's the handle in which the reference to the BLOB will be stored.
fileName	PIC X(n)	It's the name of the file that has to be stored as BLOB data in the database table.

Return code:

returnCode is set to the size in bytes of the file if the function succeeds, otherwise it's set to zero.

PUT-BLOB-INTO-FILE

The PUT-BLOB-INTO-FILE function writes the binary content of a BLOB into a file on disc.

Syntax:

```
CALL "ESQL$BLOB" USING PUT-BLOB-INTO-FILE
                       blobHandle
                       fileName
                 GIVING returnCode
```

Parameters:

PUT-BLOB-INTO-FILE	Constant	
blobHandle	USAGE HANDLE	It's the handle that identifies the BLOB to be written.
fileName	PIC X(n)	It's the name of the disc file that will host the BLOB data.

Return code:

returnCode is set to the size in bytes of the written file if the function succeeds, otherwise it's set to zero.

FREE-BLOB-HANDLE

The FREE-BLOB-HANDLE function initializes a BLOB handle freeing the memory allocated for the BLOB data.

This function should be used after the BLOB has been stored into the database table and its handle has become useless.

Syntax:

```
CALL "ESQL$BLOB" USING FREE-BLOB-HANDLE
                       blobHandle
                 GIVING returnCode
```

Parameters:

ESQL-BLOB-FREE	Constant	
blobHandle	USAGE HANDLE	It's the handle that identifies the BLOB to be initialized.

Return code:

returnCode is set to the size in bytes of the BLOB data if the function succeeds, otherwise it's set to zero.

HEX2ASCII

The HEX2ASCII library routine converts a hexadecimal number to a string according to the ASCII table.

Syntax:

```
CALL "HEX2ASCII" USING asciiValue
                       hexValue
```

Parameters:

asciiValue	PIC X(n)	Receives the converted ASCII string.
hexValue	PIC X(n) It should be twice the size of *asciiValue*	Specifies the hexadecimal number to be converted.

Examples:

Example - Convert HEX representation to Ascii value

```
*> define ascii-value and hex-value as pic x(n) and
*> hex-value should be double the lenght of ascii-value
move "a1" to hex-value
call "hex2ascii" using ascii-value, hex-value
*> ascii-value will contain "i"
```

I$IO

The I$IO routine provides a low-level interface to indexed files allowing them to be managed without knowing their FD.

The configuration properties iscobol.file.index and iscobol.file.index.FileName specify which file handler is used.

The following configuration properties are ignored: iscobol.file.case, iscobol.file.prefix and iscobol.file.indexed_file_prefix.

Syntax:

```
CALL "I$IO" USING opCode
                  parameters
           GIVING returnCode
```

Parameters:

opCode	Specifies the file handling function to be performed.	
	Valid values, defined in isfilesys.def are:	
	OPEN-FUNCTION	Opens an existing file
	CLOSE-FUNCTION	Closes an opened file
	MAKE-FUNCTION	Creates an empty file
	INFO-FUNCTION	Returns file information
	READ-FUNCTION	Reads a specific record of a file
	NEXT-FUNCTION	Reads the next record of a file
	PREVIOUS-FUNCTION	Reads the previous record of a file
	START-FUNCTION	Sets the file pointer on a specific record
	WRITE-FUNCTION	Writes data into file
	REWRITE-FUNCTION	Rewrites data into file
	DELETE-FUNCTION	Deletes data from file
	UNLOCK-FUNCTION	Unlocks all locked records in a file
	REMOVE-FUNCTION	Removes file from disc
	FLUSH-FUNCTION	Flushes all data to disc
	START-TRANSACTION-FUNCTION	Begins a transaction
	COMMIT-TRANSACTION-FUNCTION	Commits a transaction
	ROLLBACK-FUNCTION	Rollbacks a transaction
	RECOVER-FUNCTION	Rolls forward a transaction
	IN-TRANSACTION-FUNCTION	Tests if transaction is finished
parameters	Parameters depend on the opcode.	

Return code:

returnCode contains useful information such as file handles and record sizes or zero if an error occurs. Check the external variables F_ERRNO, F_SYSERR and F_ERRMSG for additional information on the error.

Possible F_ERRNO (listed below) values are available as 88 level items in the isfilesys.def copybook.

Value	Meaning
0	Operation successful.
1	System error.
2	Parameter error.
3	Too many files opened.
4	Mode clash.
5	Record locked.
6	File broken.
7	Duplicate record.
8	Record not found or end of file.
9	Undefined record.
10	Disk full.
11	File locked.
12	Record size changed.
13	Mismatch between the physical file and its description in the program.
14	Out of memory.
15	Missing file.
16	Permission denied.
17	Unsupported operation.
18	No more locks available.
19	Interface error.
20	License error.
21	Unknown error.

F_SYSERR and F_ERRMSG are set to different values depending on the current file handler.

Examples:

Example - Create an empty file with 1 key and 22 characters records lenght

```
      *> 01 record.
      *>  03 rec-key    pic 99.  *> This is the record key
      *>  03 rec-data   pic x(20).

working-storage section.
copy "isgui.def".
copy "isfilesys.def".
77  f                        handle .
77  file-io                  pic x(128).
77  key-io                   pic x(10).

procedure division.
create-file.
   move "iss-file-io" to file-io
   move zero to block-multiple pre-allocation-amount
                extension-amount compression-factor encrypted-flag
   move 22 to max-rec-size
   move 22 to min-rec-size
   move 1 to num-keys
   move "1,0,2,0" to key-io
   inspect file-io replacing trailing spaces by low-value
   inspect key-io  replacing trailing spaces by low-value
   inspect logical-info  replacing trailing spaces by low-value
   set make-function to true
   call "i$io" using io-function, file-io, 0, physical-info,
                                          logical-info, key-io, 0

   if f_errno not = 0
      display message "I$IO Error: make : " f_errno
   end-if
   goback.
```

Example - Open a file on I-O mode, get the record count, then close it

```
working-storage section.
copy "isgui.def".
copy "isfilesys.def".
77  f                          handle .
77  file-io                    pic x(128).
77  key-io                     pic x(10).

procedure division.
...
open-io.
   move "iss-file-io" to file-io
   move 22 to max-rec-size
   move 22 to min-rec-size
   move  2 to num-keys
   set open-function to true
   move fio to open-mode
   call "i$io" using io-function, file-io,
                             open-mode, logical-info

   if return-code = 0
      display message "I$IO Error: open : " F_ERRNO
   else
      move return-code to f
   end-if.
get-record-count.
   set info-function to true
   set get-record-count to true
   call "I$IO" using io-function, f, info-mode,
                     record-count-info
close-file.
   set close-function to true
   call "i$io" using io-function  f
        giving returnCode
   if returnCode = 0
      display message "I$IO Error: close : " F_ERRNO
   end-if.
```

OPEN-FUNCTION

This function opens an existing indexed file.

If it is successful, the value in RETURN-CODE should be moved to a data item that is USAGE HANDLE. This data item is passed as the open file handle to the other file handling functions.

If it fails, RETURN-CODE is set to ZERO.

After the file is opened, the primary key is set as the current key of reference and it is positioned at the beginning of the file.

This function only opens already existing files. If the file does not exist, the function fails, even when opening for output.

Syntax

```
SET OPEN-FUNCTION TO TRUE

CALL "I$IO" USING IO-FUNCTION
                  name
                  mode
                  lParms
            GIVING returnCode
```

Parameters:

name	the name of the file to open.
mode	one of the following values (defined in isfilesys.def): Finput — Open for input only. Foutput — Open for output only. Fio — Open for input and output. Fextend — Same as Foutput. mode may also have one of the following flags (defined in isfilesys.def) added to it to indicate file locking options: Fmulti_lock — Keep locks on multiple records. Fread_lock — Locks file against other updaters. Fwrite_lock — Locks file against all others. Fmass_update — Same as Fwrite_lock. FEncrypt — Consider the file as encrypted. Ftrans — Extends locking rules for transaction management. FBulk_addition — Same as Fwrite_lock.
lParms	The l_parms parameter is the same as the l_parms parameter passed when using the MAKE-FUNCTION opcode. This parameter is a string that contains three comma-separated numbers. Valid values are (in order): • the maximum record size, • the minimum record size, • the number of keys for the file, If the maximum record size does not match the minimum record size, then variable sized records are implied. If the parameter is initialized, the runtime retrieves the values from the file during the open. If the parameter is set, some file handlers may return an error if it doesn't match with the actual characteristics of the file.

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	File handle.
0	Operation failed.

CLOSE-FUNCTION

This function closes an opened file.

It also removes currently held locks on the file.

Syntax

```
SET CLOSE-FUNCTION TO TRUE

CALL "I$IO" USING IO-FUNCTION
                  f
            GIVING returnCode
```

Parameters:

f	the file handle returned by OPEN-FUNCTION.

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Operation successful.
0	Operation failed.

MAKE-FUNCTION

This function is used to create a new indexed file. If the file already exists, it will be overwritten.

Syntax

```
SET MAKE-FUNCTION TO TRUE

CALL "I$IO" USING IO-FUNCTION
                  name
                  comment
                  pParms
                  lParms
                  keys
                  trans
            GIVING returnCode
```

Parameters:

name	points to the name of the file.
comment	may be NULL or may point to comment string that describes the file.
pParms	points to a string that describes various physical characteristics of the file. A structure called PHYSICAL-INFO is provided in the "isfilesys.def" copybook. These fields should be set to zero
lParms	points to a string that describes various logical characteristics of the file The "isfilesys.def" copy file has a data item containing these fields: • Maximum record size, the size of the largest record to be placed in the file. • Minimum record size, the size of the smallest record to be placed in the file. • Number of keys, the number of keys in the file, including the primary key.

keys	points to a null-terminated string that describes the key structure for the file. keys is a string of numbers separated by commas. The first key described is the primary key. It may not allow duplicate values. The primary key is called key "0". The next key described is key "1" and so on. There should be as many keys described as the "number of keys" field of l_parms indicates. The "isfilesys.def" copy file has a data item containing these fields for each key: • Number of segments, the number of segments in this key. • Duplicates flag. If this value is "1", then duplicate keys are allowed. If "0", then duplicate values are not allowed. • Segment size, the number of bytes in the first segment. • Segment offset, the offset from the beginning of the record to the first byte of the segment. • Remaining segments. The segment size and segment offset fields are repeated for each additional segment in the key. For example, a file with two keys, the first one having two segments (offset zero, length 10 and offset 50, length 5) and the second one with one segment (offset 20, length 15) and allowing duplicates would be written: 2,0,10,0,5,50,1,1,15,20
trans	This parameter specifies an alternate collating sequence for the keys. If this parameter is NULL or omitted, then keys are ordered in ascending sequence based on their native unsigned value. Be aware that compiling with -ca option causes NULL to be translated to 0, that is an invalid collating sequence, therefore omitting the parameter is safer than setting it to NULL. If it is not NULL, it must point to a 256 byte region of memory. Unlike other strings, this need not be null-terminated and is likely to contain nulls within it. This 256 byte region is used as a translation table on the bytes of each key to arrive at a new key-ordering. Each byte is used as an index into this table, and the resulting value is used to order the keys.

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Operation successful.
0	Operation failed.

INFO-FUNCTION

This function returns information, depending on the value of mode parameter, about an opened indexed file

The "isfilesys.def" copy file contains layouts for each kind of information that can be retrieved with this function.

Note - some information might not be returned by the current file handler.

Syntax

```
SET INFO-FUNCTION TO TRUE

CALL "I$IO" USING IO-FUNCTION
                  f
                  mode
                  result
            GIVING returnCode
```

Parameters:

f	the file handle returned by OPEN-FUNCTION.	
mode	determines which result is returned.	
	-1	returns the same information as the l_parms parameter of the MAKE-FUNCTION. result is in the format of "11111,22222,333" where: 1 maximum record size
		2 minimum record size
		3 number of keys
	-4	returns the number of records in the file.
	-5	returns the 256-byte key translation table. If no key translation table was specified, then the E_NO_SUPPORT error is set. In this case, this should be simply taken to mean that the native key ordering was used.
	0 or greater	a value of zero or greater indicates that information about a particular key is desired. '0' indicates the primary key, '1' indicates the first alternate key and so on. That key information is returned as "11,2,333,44444" where: (third and fourth fields are repeated for each additional segment in the key) 1 number of segments in key.
		2 "1" if duplicates are allowed.
		3 size of first segment.
		4 byte offset of first segment.
result	depends by mode parameter.	

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Operation successful.
0	Operation failed.

READ-FUNCTION

This function reads a record out of an indexed file.

The record locking is controlled by the *f-no-lock* external data-item. Move 1 to *f-no-lock* to read without locking. Move 0 to *f-no-lock* to read with lock. The default value of *f-no-lock* is 0.

If this function has success, RETURN-CODE is set to the size of the record read. RETURN-CODE is set to zero on failure.

Syntax

```
SET READ-FUNCTION TO TRUE

CALL "I$IO" USING IO-FUNCTION
                  f
                  record
                  keyNum
            GIVING returnCode
```

Parameters:

f	must be a valid file handle returned by OPEN-FUNCTION.
record	points to the area to hold the record read. **Note**: in order to gain performance, in isCOBOL File Server environment the previous content of this parameter is discarded before reading. After reading, the bytes exceeding the number of read bytes are initialized to low-values.
keyNum	the key number to read from. The value can be 0 or greater than 0, where 0 is the primary key, 1 is the first alternate key and so on.

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Number of read bytes.
0	Operation failed.

NEXT-FUNCTION

This function reads the next record in the sequence of records specified by the current key of reference.

The record locking is controlled by the *f-no-lock* external data-item. Move 1 to *f-no-lock* to read without locking. Move 0 to *f-no-lock* to read with lock. The default value of *f-no-lock* is 0.

If this function succeeds, RETURN-CODE is set to the size of the record read. RETURN-CODE is set to zero on failure.

Syntax

```
SET NEXT-FUNCTION TO TRUE

CALL "I$IO" USING IO-FUNCTION
                  f
                  record
            GIVING returnCode
```

Parameters:

f	must be a valid file handle returned by OPEN-FUNCTION.
record	points to the area to hold the record read. **Note**: in order to gain performance, in isCOBOL File Server environment the previous content of this parameter is discarded before reading. After reading, the bytes exceeding the number of read bytes are initialized to low-values.

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Number of read bytes.
0	Operation failed.

PREVIOUS-FUNCTION

This function reads the previous record in the sequence of records specified by the current key of reference.

The record locking is controlled by the *f-no-lock* external data-item. Move 1 to *f-no-lock* to read without locking. Move 0 to *f-no-lock* to read with lock. The default value of *f-no-lock* is 0.

If this function succeeds, RETURN-CODE is set to the size of the record read. RETURN-CODE is set to zero on failure.

Syntax

```
SET PREVIOUS-FUNCTION TO TRUE

CALL "I$IO" USING IO-FUNCTION
                  f
                  record
           GIVING returnCode
```

Parameters:

f	must be a valid file handle returned by OPEN-FUNCTION.
record	points to the area to hold the record read. **Note**: in order to gain performance, in isCOBOL File Server environment the previous content of this parameter is discarded before reading. After reading, the bytes exceeding the number of read bytes are initialized to low-values.

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Number of read bytes.
0	Operation failed.

START-FUNCTION

This function selects the current key of reference and positions the file pointer for the next NEXT or PREVIOUS function.

If this function fails, RETURN-CODE is set to zero and the current key of reference is placed in the "undefined" state.

Syntax

```
SET START-FUNCTION TO TRUE

CALL "I$IO" USING IO-FUNCTION
                  f
                  record
                  keyNum
                  keySize
                  mode
            GIVING returnCode
```

Parameters:

f	must be a valid file handle returned by OPEN-FUNCTION.	
record	points to the area to hold the record read.	
keyNum	selects which key to use. The value can be 0 or greater than 0, where 0 is the primary key, 1 is the first alternate key and so on. The corresponding key area in record must contain the key value that will be used to position the file.	
keySize	indicates the size of the key. If keysize is zero, the entire key is used. Otherwise, only the first keysize bytes of the key will be used.	
mode	selects how the file is to be positioned with respect to the key value defined in record. It can be one of the following values (defined in "isfilesys.def"):	
	F_EQUALS	The file is positioned at the record that matches the key value.
	F_NOT_LESS	The file is positioned at the record that matches the key value, or the next greater one if no one matches.
	F_GREATER	The file is positioned at the first record greater than the key value specified.
	F_LESS	The file is positioned at the last record smaller than the key value specified.
	F_NOT_GREATER	The file is positioned at the record that matches the key value, or the last record smaller than the key value if no one matches.

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Operation successful.
0	Operation failed.

WRITE-FUNCTION

This function adds a new record to the passed file.

If this function succeeds, RETURN-CODE is set to the size of record written. RETURN-CODE is set to zero on failure.

Syntax

```
SET WRITE-FUNCTION TO TRUE

CALL "I$IO" USING IO-FUNCTION
                  f
                  record
                  size
            GIVING returnCode
```

Parameters:

f	must be a valid file handle returned by OPEN-FUNCTION.
record	points to the record to add.
size	the size of the record. If size is zero, then the maximum record size for the file is used.

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Operation successful.
0	Operation failed.

REWRITE-FUNCTION

This function replaces an existing record in the file.

If this function succeeds, RETURN-CODE is set to the size of record written. RETURN-CODE is set to zero on failure.

Syntax

```
SET REWRITE-FUNCTION TO TRUE

CALL "I$IO" USING IO-FUNCTION
                  f
                  record
                  size
            GIVING returnCode
```

Parameters:

f	must be a valid file handle returned by OPEN-FUNCTION.
record	points to the new record to place in the file.
size	may be zero to indicate the maximum record size for the file.
	The record replaced is specified by the primary key value found in record.
	The size of the new record need not match the size of the existing record.

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Operation successful.
0	Operation failed.

DELETE-FUNCTION

This function deletes the record identified by the value found in the primary key area of the record. It does not affect the current file position or key of reference.

RETURN-CODE is set to zero on failure.

Syntax

```
SET DELETE-FUNCTION TO TRUE

CALL "I$IO" USING IO-FUNCTION
                  f
                  record
            GIVING returnCode
```

Parameters:

f	must be a valid file handle returned by OPEN-FUNCTION.
record	points to the area to hold the record read.

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Operation successful.
0	Operation failed.

UNLOCK-FUNCTION

This function unlocks any locked records held by the current process in the passed file. It does not affect the current file position or key of reference.

This function will not unlock any records if it is called during a transaction. COMMIT-TRANSACTION-FUNCTION should be used instead.

RETURN-CODE is set to zero on failure.

Syntax

```
SET UNLOCK-FUNCTION TO TRUE

CALL "I$IO" USING IO-FUNCTION
                  f
            GIVING returnCode
```

Parameters:

f	must be a valid file handle returned by OPEN-FUNCTION.

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Operation successful.

0	Operation failed.

REMOVE-FUNCTION

This function removes the indexed file from disk.

RETURN-CODE is set to ZERO on failure.

Syntax

```
SET REMOVE-FUNCTION TO TRUE

CALL "I$IO" USING IO-FUNCTION
                  name
            GIVING returnCode
```

Parameters:

name	the name of the file to be removed.

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Operation successful.
0	Operation failed.

FLUSH-FUNCTION

This function causes all file buffers to be flushed to disk.

Syntax

```
SET FLUSH-FUNCTION TO TRUE

CALL "I$IO" USING IO-FUNCTION
                  allFiles
            GIVING returnCode
```

Parameters:

allFiles	bit value. Can be set to one of the following values (defined in isfilesys.def):	
	FA-MASS-UPDATE	if all_files and FA-MASS-UPDATE are not 0, then MASS-UPDATE files should be synced.
	FA-REMOTE	if all_files and FA-REMOTE are not 0, then remote files should be synced.

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Operation successful.
0	Operation failed.

START-TRANSACTION-FUNCTION

This function initiates a transaction.

The START-TRANSACTION function has no parameters.

Syntax

```
SET START-TRANSACTION-FUNCTION TO TRUE

CALL "I$IO" USING IO-FUNCTION
            GIVING returnCode
```

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Operation successful.
0	Operation failed.

COMMIT-TRANSACTION-FUNCTION

This function commits all changes and releases all locks, ending a transaction.

Syntax

```
SET COMMIT-TRANSACTION-FUNCTION TO TRUE

CALL "I$IO" USING IO-FUNCTION
                  context
            GIVING returnCode
```

Parameters:

context	PIC 9(n)	Indicates the transaction context. Refer to the file handler documentation for the list of possible values. **Note -** c-treeRTG ignores this parameter.

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Operation successful.
0	Operation failed.

ROLLBACK-FUNCTION

This function rolls back all files affected to the state that they were in after the last completed transaction.

The ROLLBACK function has no parameters.

Syntax

```
SET ROLLBACK-FUNCTION TO TRUE

CALL "I$IO" USING IO-FUNCTION
            GIVING returnCode
```

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Operation successful.
0	Operation failed.

RECOVER-FUNCTION

This function rolls forward all files affected to the state that they were in after the last completed transaction.

The RECOVER function has no parameters.

Syntax

```
SET RECOVER-FUNCTION TO TRUE

CALL "I$IO" USING IO-FUNCTION
            GIVING returnCode
```

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Operation successful.
0	Operation failed.

IN-TRANSACTION-FUNCTION

This function returns a value indicating whether or not the program is currently in an unfinished transaction.

The IN-TRANSACTION function has no parameters.

RETURN-CODE is set to "1" if there is current and unfinished transaction or "0" otherwise.

Note - The isCOBOL implementation is independent from the current file handler. Every time the START TRANSACTION operation is invoked, then an environmental flag is set to true, while every time a COMMIT or ROLLBACK operation is called, the flag is set to false. No check on errors is performed. The function returns the value of the flag.

Syntax

```
SET IN-TRANSACTION-FUNCTION TO TRUE

CALL "I$IO" USING IO-FUNCTION
            GIVING returnCode
```

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	In transaction.
0	Not in transaction.

J$GETFROMLAF

The J$GETFROMLAF library routine returns a font or a color by inquiring the current Look and Feel (LAF). In thin client environment it automatically works client side.

Syntax:

```
CALL "J$GETFROMLAF" USING opCode
                          parameters
                    GIVING returnCode
```

Parameters:

opCode	Function to be executed. Valid values, defined in iscobol.def are:	
	JGET-LAF-COLOR	Returns the RGB representation of a given color retrieved from the current LAF in the form of a negative number
	JGET-LAF-FONT	Returns the handle of a given font retrieved from the current LAF
parameters	Parameters depend on the opcode.	

Return code:

returnCode can be any signed numeric data item. The meaning depends on the opcode.

Examples:

Example - display a screen where LAF font and colors are used

```
working-storage section.
copy "iscrt.def".
copy "isgui.def".
copy "iscobol.def".
copy "isresize.def".
77 crt-status             special-names crt status pic 9(5).
77 close-win              pic 9 value 0.
77 result-getfromlaf      pic s9.
77 label-font             handle of font.
77 lb-bg-color            pic s9(9).
77 lb-fg-color            pic s9(9).
77 hWin                   handle of window.

screen section.
01  screen-01.
    03 label
       line                 2
       lines                2
       col                  2
       size                 68 cells
       title "The font and the color of the controls of this screen"
       foreground-color     lb-fg-color
       background-color     lb-bg-color
       font                 label-font.
    03 label
       line                 3
       lines                2
       col                  2
       size                 68 cells
       title "are loaded by 'J$GETFROMLAF' Routine"
       foreground-color     lb-fg-color
       background-color     lb-bg-color
       font                 label-font.
    03 Pb-exit
       push-button
       line                 20
       col                  62
       size                 8 cells
       title                "Exit"
       exception-value      27.

procedure division.
main.
    perform get-label-font
    perform get-label-color
```

```
    display standard graphical window
            control-font label-font
            background-low
            resizable
            line 2
            col 65
            title  "J$GETFROMLAF Routine"
            lines 21
            min-lines 21
            size 70
            min-size 70
            handle hWin
            event  win-evt.
    display screen-01
    accept  screen-01 until crt-status = 27 or close-win = 1
                    on exception continue
    end-accept
    destroy screen-01
    destroy hWin
    destroy label-font
    goback.

get-label-font.
    call "J$GETFROMLAF" using jget-laf-font
                              "Label.font"
                              label-font
                        giving result-getfromlaf.
    perform check-laf-result.

get-label-color.
    call "J$GETFROMLAF" using jget-laf-color
                              "Label.background"
                              lb-bg-color
                        giving result-getfromlaf.
    perform check-laf-result.
    call "J$GETFROMLAF" using jget-laf-color
                              "Label.foreground"
                              lb-fg-color
                        giving result-getfromlaf.
    perform check-laf-result.

check-laf-result.
    if result-getfromlaf = -1
       display message "Invalid Laf Entry"
       goback
    end-if.

win-evt.
    if event-type = cmd-close
       move 1 to close-win
    end-if.
```

JGET-LAF-COLOR

The JGET-LAF-COLOR function returns the RGB representation of a given color retrieved from the current LAF in the form of a negative value.

This is the list of the known color entries that you can inquire:

- Button.background
- Button.darkShadow
- Button.disabledForeground
- Button.disabledShadow
- Button.focus
- Button.foreground
- Button.highlight
- Button.light
- Button.shadow
- CheckBox.background
- CheckBox.darkShadow
- CheckBox.focus
- CheckBox.foreground
- CheckBox.highlight
- CheckBox.interiorBackground
- CheckBox.light
- CheckBox.shadow
- CheckBoxMenuItem.acceleratorForeground
- CheckBoxMenuItem.acceleratorSelectionForeground
- CheckBoxMenuItem.background
- CheckBoxMenuItem.foreground
- CheckBoxMenuItem.selectionBackground
- CheckBoxMenuItem.selectionForeground
- ColorChooser.background
- ColorChooser.foreground
- ColorChooser.swatchesDefaultRecentColor
- ComboBox.background
- ComboBox.buttonBackground
- ComboBox.buttonDarkShadow
- ComboBox.buttonHighlight
- ComboBox.buttonShadow
- ComboBox.disabledBackground
- ComboBox.disabledForeground
- ComboBox.foreground
- ComboBox.selectionBackground
- ComboBox.selectionForeground
- Desktop.background
- EditorPane.background
- EditorPane.caretForeground

- EditorPane.disabledBackground
- EditorPane.foreground
- EditorPane.inactiveBackground
- EditorPane.inactiveForeground
- EditorPane.selectionBackground
- EditorPane.selectionForeground
- FileChooser.listViewBackground
- FormattedTextField.background
- FormattedTextField.caretForeground
- FormattedTextField.disabledBackground
- FormattedTextField.foreground
- FormattedTextField.inactiveBackground
- FormattedTextField.inactiveForeground
- FormattedTextField.selectionBackground
- FormattedTextField.selectionForeground
- InternalFrame.activeBorderColor
- InternalFrame.activeTitleBackground
- InternalFrame.activeTitleForeground
- InternalFrame.activeTitleGradient
- InternalFrame.borderColor
- InternalFrame.borderDarkShadow
- InternalFrame.borderHighlight
- InternalFrame.borderLight
- InternalFrame.borderShadow
- InternalFrame.inactiveBorderColor
- InternalFrame.inactiveTitleBackground
- InternalFrame.inactiveTitleForeground
- InternalFrame.inactiveTitleGradient
- InternalFrame.minimizeIconBackground
- InternalFrame.resizeIconHighlight
- InternalFrame.resizeIconShadow
- Label.background
- Label.disabledForeground
- Label.disabledShadow
- Label.foreground
- List.background
- List.dropLineColor
- List.foreground
- List.selectionBackground
- List.selectionForeground

- Menu.acceleratorForeground
- Menu.acceleratorSelectionForeground
- Menu.background
- Menu.foreground
- Menu.selectionBackground
- Menu.selectionForeground
- MenuBar.background
- MenuBar.foreground
- MenuBar.highlight
- MenuBar.shadow
- MenuItem.acceleratorForeground
- MenuItem.acceleratorSelectionForeground
- MenuItem.background
- MenuItem.disabledForeground
- MenuItem.foreground
- MenuItem.selectionBackground
- MenuItem.selectionForeground
- OptionPane.background
- OptionPane.foreground
- OptionPane.messageForeground
- Panel.background
- Panel.foreground
- PasswordField.background
- PasswordField.caretForeground
- PasswordField.disabledBackground
- PasswordField.foreground
- PasswordField.inactiveBackground
- PasswordField.inactiveForeground
- PasswordField.selectionBackground
- PasswordField.selectionForeground
- PopupMenu.background
- PopupMenu.foreground
- ProgressBar.background
- ProgressBar.foreground
- ProgressBar.highlight
- ProgressBar.selectionBackground
- ProgressBar.selectionForeground
- ProgressBar.shadow
- RadioButton.background
- RadioButton.darkShadow

- RadioButton.focus
- RadioButton.foreground
- RadioButton.highlight
- RadioButton.interiorBackground
- RadioButton.light
- RadioButton.shadow
- RadioButtonMenuItem.acceleratorForeground
- RadioButtonMenuItem.acceleratorSelectionForeground
- RadioButtonMenuItem.background
- RadioButtonMenuItem.disabledForeground
- RadioButtonMenuItem.foreground
- RadioButtonMenuItem.selectionBackground
- RadioButtonMenuItem.selectionForeground
- ScrollBar.background
- ScrollBar.foreground
- ScrollBar.thumb
- ScrollBar.thumbDarkShadow
- ScrollBar.thumbHighlight
- ScrollBar.thumbShadow
- ScrollBar.track
- ScrollBar.trackForeground
- ScrollBar.trackHighlight
- ScrollBar.trackHighlightForeground
- ScrollPane.background
- ScrollPane.foreground
- Separator.background
- Separator.foreground
- Separator.highlight
- Separator.shadow
- Slider.background
- Slider.focus
- Slider.foreground
- Slider.highlight
- Slider.shadow
- Slider.tickColor
- Spinner.background
- Spinner.foreground
- SplitPane.background
- SplitPane.darkShadow
- SplitPane.highlight

- SplitPane.shadow
- SplitPaneDivider.draggingColor
- TabbedPane.background
- TabbedPane.darkShadow
- TabbedPane.focus
- TabbedPane.foreground
- TabbedPane.highlight
- TabbedPane.light
- TabbedPane.shadow
- Table.background
- Table.darkShadow
- Table.dropLineColor
- Table.dropLineShortColor
- Table.focusCellBackground
- Table.focusCellForeground
- Table.foreground
- Table.gridColor
- Table.highlight
- Table.light
- Table.selectionBackground
- Table.selectionForeground
- Table.shadow
- Table.sortIconColor
- Table.sortIconHighlight
- Table.sortIconLight
- TableHeader.background
- TableHeader.foreground
- TextArea.background
- TextArea.caretForeground
- TextArea.disabledBackground
- TextArea.foreground
- TextArea.inactiveBackground
- TextArea.inactiveForeground
- TextArea.selectionBackground
- TextArea.selectionForeground
- TextField.background
- TextField.caretForeground
- TextField.darkShadow
- TextField.disabledBackground
- TextField.foreground

- TextField.highlight
- TextField.inactiveBackground
- TextField.inactiveForeground
- TextField.light
- TextField.selectionBackground
- TextField.selectionForeground
- TextField.shadow
- TextPane.background
- TextPane.caretForeground
- TextPane.disabledBackground
- TextPane.foreground
- TextPane.inactiveBackground
- TextPane.inactiveForeground
- TextPane.selectionBackground
- TextPane.selectionForeground
- TitledBorder.titleColor
- ToggleButton.background
- ToggleButton.darkShadow
- ToggleButton.focus
- ToggleButton.foreground
- ToggleButton.highlight
- ToggleButton.light
- ToggleButton.shadow
- ToolBar.background
- ToolBar.darkShadow
- ToolBar.dockingBackground
- ToolBar.dockingForeground
- ToolBar.floatingBackground
- ToolBar.floatingForeground
- ToolBar.foreground
- ToolBar.highlight
- ToolBar.light
- ToolBar.shadow
- ToolTip.background
- ToolTip.foreground
- Tree.background
- Tree.dropLineColor
- Tree.foreground
- Tree.hash
- Tree.selectionBackground

- Tree.selectionBorderColor
- Tree.selectionForeground
- Tree.textBackground
- Tree.textForeground
- Viewport.background
- Viewport.foreground
- activeCaption
- activeCaptionBorder
- activeCaptionText
- control
- controlDkShadow
- controlHighlight
- controlLtHighlight
- controlShadow
- controlText
- desktop
- inactiveCaption
- inactiveCaptionBorder
- inactiveCaptionText
- info
- infoText
- menu
- menuPressedItemB
- menuPressedItemF
- menuText
- scrollbar
- text
- textHighlight
- textHighlightText
- textInactiveText
- textText
- window

Syntax:

```
CALL "J$GETFROMLAF" USING JGET-LAF-COLOR
                          lafEntry
                          colorValue
                    GIVING returnCode
```

Parameters:

JGET-LAF-COLOR	Constant	
lafEntry	PIC X(n) or string literal	Specifies the name of the color resource you want to inquire.
colorValue	PIC S9(9)	Receives the RGB representation of the color

Return code:

returnCode can be any signed numeric data item.

-1	Operation failed.
0	Operation successful.

JGET-LAF-FONT

The JGET-LAF-FONT function returns the handle of a given font retrieved from the current LAF. It's program duty to destroy the font handle when it's not more necessary.

This is the list of the known font entries that you can inquire:

- Button.font
- CheckBox.font
- CheckBoxMenuItem.acceleratorFont
- CheckBoxMenuItem.font
- ColorChooser.font
- ComboBox.font
- EditorPane.font
- FileChooser.listFont
- FormattedTextField.font
- InternalFrame.titleFont
- Label.font
- List.font
- Menu.acceleratorFont
- Menu.font
- MenuBar.font
- MenuItem.acceleratorFont
- MenuItem.font

- OptionPane.buttonFont
- OptionPane.font
- OptionPane.messageFont
- Panel.font
- PasswordField.font
- PopupMenu.font
- ProgressBar.font
- RadioButton.font
- RadioButtonMenuItem.acceleratorFont
- RadioButtonMenuItem.font
- ScrollPane.font
- Slider.font
- Spinner.font
- TabbedPane.font
- Table.font
- TableHeader.font
- TextArea.font
- TextField.font
- TextPane.font
- TitledBorder.font
- ToggleButton.font
- ToolBar.font
- ToolTip.font
- Tree.font
- Viewport.font

Syntax:

```
CALL "J$GETFROMLAF" USING JGET-LAF-FONT
                          lafEntry
                          fontHandle
                    GIVING returnCode
```

Parameters:

JGET-LAF-FONT	Constant	
lafEntry	PIC X(n) or string literal	Specifies the name of the font resource you want to inquire.
fontHandle	USAGE HANDLE OF FONT	Receives handle of the requested font

Return code:

returnCode can be any signed numeric data item.

-1	Operation failed.
0	Operation successful.

J$NETADDRESS

The J$NETADDRESS library routine retrieves the name and the IP address of the computer where this routine is executed.

Syntax:

```
CALL "J$NETADDRESS" USING computerName
                          computerIpAddress
```

Parameters:

computerName	PIC X(n)	Receives the name of the computer where this routine has been executed.[*]
computerIpAddress	PIC X(n)	Receives the IP address of the computer where this routine has been executed.[*]

[*] A computer may have multiple IPs and multiple alias name for the same IP too, so you might not receive the expected IP and name. Usually a safe method to get the desired information is to change the hosts configuration file (/etc/hosts on Linux/Unix, %SystemRoot%\System32\drivers\etc\hosts on Windows) appropriately.

Examples:

Example - Get the current computer name and ip address

```
*> define arguments as pic x(n)
CALL "J$NETADDRESS" USING comp-Name comp-IpAddress
```

KEISEN

The KEISEN routine allows to draw a line in a Japanese character-based program. It invokes KEISEN1 or KEISEN2 depending on the value of iscobol.keisen.method.

Syntax:

```
 CALL "KEISEN" USING { KEISEN  }
                     { KEISEN2 }
              GIVING returnCode
```

Parameters:

KEISEN	Group Item	Structure defined in iskeisen.def.
		
		`01 KEISEN.`
		`  02 KEI-CMD   PIC 9(1) COMP-X.`
		`  02 KEI-LINE  PIC 9(2) COMP-X.`
		`  02 KEI-COL   PIC 9(2) COMP-X.`
		`  02 KEI-LNG1  PIC 9(2) COMP-X.`
		`  02 KEI-LNG2  PIC 9(2) COMP-X.`
		`  02 KEI-COLOR PIC 9(2) COMP-X.`
		`  02 KEI-PTN   PIC 9(2) COMP-X.`
		
		Use this structure if you set iscobol.keisen.method to 1 or you didn't set the property to any value. Refer to KEISEN1 documentation for details about the meaning of these data items.

KEISEN2	Group Item	Structure defined in iskeisen.def.
		
		`01 KEISEN2.`
		`  02 KEI2-CMD        PIC 9(1) COMP-X.`
		`  02 KEI2-START-LINE PIC 9(2) COMP-X.`
		`  02 KEI2-START-COL  PIC 9(2) COMP-X.`
		`  02 KEI2-END-LINE   PIC 9(2) COMP-X.`
		`  02 KEI2-END-COL    PIC 9(2) COMP-X.`
		`  02 KEI2-PRN        PIC 9(2) COMP-X.`
		`  02 KEI2-COLOR      PIC 9(2) COMP-X.`
		
		Use this structure if you set iscobol.keisen.method to 2. Refer to KEISEN2 documentation for details about the meaning of these data items.

Return Code

returnCode can be any signed numeric data item and provides additional information:

-1	Operation failed.
0	Operation successful.

Examples:

Example - Draw a rectangle with KAISEN1 or KAISEN2

```
       program-id. keisen_sample.

       working-storage section.
       copy "isgui.def".
       copy "iscrt.def".
       copy "iskeisen.def".
       77  crt-status              special-names crt status pic 9(5).
       77  hwin                    handle of window.
       77  close-win               pic 9 value 0.
       77  k-method                pic 9.
       77  lk-method               pic 9.


       procedure division chaining lk-method.
       main.
           display independent graphical window
                   color  65793
                   with   system menu
                   title  "KEISEN Routines"
                   handle hwin
                   event  win-evt
           display window erase.

           perform draw-keisen


           destroy hwin
           goback
           .

       draw-keisen.
           if lk-method = 1 or lk-method = 2
              move lk-method to kei-param
              call "keisen_select" using kei-param
           end-if
           accept k-method from environment "keisen.method"
              on  exception
                  move 1 to k-method
           end-accept
           evaluate k-method
           when 1
                perform keisen1
           when 2
                perform keisen2
           end-evaluate
           .
```

```
keisen1.
    move 5  to kei-cmd
    move 3  to kei-line
    move 3  to kei-col
    move 70 to kei-lng1
    move 20 to kei-lng2
    move 2  to kei-color
    move 4  to kei-ptn
    call "keisen" using keisen
    display message
    "The lines on the screen have been drawn with KEISEN"
    .

keisen2.
    move 5  to kei2-cmd
    move 3  to kei2-start-line
    move 3  to kei2-start-col
    move 70 to kei2-end-col
    move 20 to kei2-end-line
    move 2  to kei2-color
    move 4  to kei2-prn
    call "keisen2" using keisen2
    display message
    "The lines on the screen have been drawn with KEISEN2"
    .

win-evt.
    if event-type = cmd-close
       move 1 to close-win
    end-if
    .
```

KEISEN1

The KEISEN1 routine allows to draw a line of type 1 in a Japanese character-based program.

Syntax:

```
CALL "KEISEN1" USING KEISEN
               GIVING returnCode
```

Parameters:

KEISEN	Group Item	Structure defined in iskeisen.def.
		
		`01 KEISEN.`
		`  02 KEI-CMD  PIC 9(1) COMP-X.`
		`  02 KEI-LINE  PIC 9(2) COMP-X.`
		`  02 KEI-COL  PIC 9(2) COMP-X.`
		`  02 KEI-LNG1  PIC 9(2) COMP-X.`
		`  02 KEI-LNG2  PIC 9(2) COMP-X.`
		`  02 KEI-COLOR PIC 9(2) COMP-X.`
		`  02 KEI-PTN  PIC 9(2) COMP-X.`
		
		KEI-CMD:
		0 - default (clear screen)
		1 - underline (bottom)
		2 - over the line (top)
		3 - batikarurain (left)
		4 - batikarurain (right)
		5 - box
		6 - vertical (left) and underline (below)
		9 - termination
		
		KEI-LINE:
		opening lines, values range: 1 to 24
		
		KEI-COL:
		start column, values range: 1 to 80
		
		KEI-LNG1:
		wire, if KEI-CMD is 1,2 or 5 (Horizontal), values from 1 to 80 are allowed. if KEI-CMD is 3 or 4 (Vertical), values from 1 to 24 are allowed.
		
		KEI-LNG2:
		wire, if KEI-CMD is 5, values from 1 to 24 are allowed.
		
		KEI-COLOR:
		0 - black
		1 - blue
		2 - green
		3 - blue, green
		4 - red
		5 - scarlet
		6 - brown
		7 - white
		
		KEI-PTN:
		linetype
		1 - solid
		2 - dashed
		3 - dotted line
		4 - dashed line
		5 - two-dot chain

Return Code

returnCode can be any signed numeric data item and provides additional information:

-1	Operation failed.
0	Operation successful.

Examples:

See KEISEN for examples.

KEISEN2

The KEISEN2 routine allows to draw a line of type 2 in a Japanese character-based program.

Syntax:

```
CALL "KEISEN2" USING KEISEN2
               GIVING returnCode
```

Parameters:

KEISEN2	Group Item	Structure defined in iskeisen.def.
		
		`01 KEISEN2.`
		`  02 KEI2-CMD  PIC 9(1) COMP-X.`
		`  02 KEI2-START-LINE PIC 9(2) COMP-X.`
		`  02 KEI2-START-COL  PIC 9(2) COMP-X.`
		`  02 KEI2-END-LINE  PIC 9(2) COMP-X.`
		`  02 KEI2-END-COL  PIC 9(2) COMP-X.`
		`  02 KEI2-PRN  PIC 9(2) COMP-X.`
		`  02 KEI2-COLOR  PIC 9(2) COMP-X.`
		
		KEI2-CMD:
		0 - default
		1 - clear screen
		2 - Line
		3 - Boxes
		9 - Termination
		
		KEI2-START-LINE:
		opening lines, values range: 1 to 24
		
		KEI2-START-COL:
		start column, values range: 1 to 80
		
		KEI2-END-LINE:
		line termination, values range: 1 to 24
		
		KEI2-END-COL
		column end, values range: 1 to 80
		
		KEI2-PRN:
		linetype
		1 - solid
		2 - dashed
		3 - dotted line
		4 - dashed line
		5 - two-dot chain
		
		KEI2-COLOR:
		0 - black
		1 - blue
		2 - green
		3 - blue, green
		4 - red
		5 - scarlet
		6 - brown
		7 - white

Return Code

returnCode can be any signed numeric data item and provides additional information:

-1	Operation failed.
0	Operation successful.

Examples:

See KEISEN for examples.

KEISEN_SELECT

The KEISEN_SELECT routine allows to specify the method for the next call to KEISEN. It updates the value of the iscobol.keisen.method configuration property.

Syntax:

```
CALL "KEISEN_SELECT" USING KEI-PARAM
                     GIVING returnCode
```

Parameters:

KEI-PARAM	PIC 9 (1) COMP-X	This item, defined in iskeisen.def, can be set either to 1 or 2.

Return Code

returnCode can be any signed numeric data item and provides additional information:

-1	Operation failed.
0	Operation successful.

Examples:

See KEISEN for examples.

M$ALLOC

The M$ALLOC library routine dynamically allocates memory.

Syntax:

```
CALL "M$ALLOC" USING memSize
                     memAddress
```

Parameters:

memSize	any numeric data item or numeric literal	Specifies the number of bytes to be allocated.
memAddress	USAGE HANDLE	Receives the handle to the allocated memory. If the allocation fails the this item is set to NULL. **Note -** for compatibility reasons this item may also be defined as USAGE POINTER. In this case the program must be compiled either with -ca or -cp options. With -ca POINTER is translated to HANDLE. With -cp POINTER is a real pointer that can be shared with external C routines.

Examples:

Example - Allocate 8 bytes of memory on each handle and put the words Hello World in them

```
*> define mem1 and mem2 usage handle

call "m$alloc" using 8, mem1
call "m$alloc" using 8, mem2
call "m$put" using mem1, "Hello", 8, 1
call "m$put" using mem2, "World", 8, 1
```

M$COPY

The M$COPY library routine copies memory from the source memory area to the destination memory area.

The involved memory regions are usually allocated with the M$ALLOC library routine. However, programs compiled with `-cp` option may operate also on a memory regions allocated by external C functions or by a Format 7 SET statement.

Syntax:

```
CALL "M$COPY" USING destination
                    source
                    length
```

Parameters:

destination	USAGE HANDLE	Specifies the handle to the destination memory region. **Note -** for compatibility reasons this item may also be defined as USAGE POINTER. In this case the program must be compiled either with -ca or -cp options. With -ca POINTER is translated to HANDLE. With -cp POINTER is a real pointer that can be shared with external C routines.
source	USAGE HANDLE	Specifies the handle to the source memory region. **Note -** for compatibility reasons this item may also be defined as USAGE POINTER. In this case the program must be compiled either with -ca or -cp options. With -ca POINTER is translated to HANDLE. With -cp POINTER is a real pointer that can be shared with external C routines.
length	any numeric data item or numeric literal	Specifies the number of bytes to be copied.

Examples:

Example - Allocate 8 bytes of memory on each handle, fill 1 with X characters and then copy first memory contents to second memory area

```
*> define mem1 and mem2 usage handle

call "m$alloc" using 8, mem1
call "m$alloc" using 8, mem2
call "m$fill" using mem1, "X", 8
call "m$copy" using mem1, mem2, 8
```

M$FILL

The M$FILL library routine fills a previously allocated memory region with a specific value.

The memory region is usually allocated with the M$ALLOC library routine. However, programs compiled with `-cp` option may operate also on a memory regions allocated by external C functions or by a Format 7 SET statement.

Syntax:

```
CALL "M$FILL" USING destination
                    value
                    length
```

Parameters:

destination	USAGE HANDLE	Specifies the handle to the destination memory region. **Note -** for compatibility reasons this item may also be defined as USAGE POINTER. In this case the program must be compiled either with -ca or -cp options. With -ca POINTER is translated to HANDLE. With -cp POINTER is a real pointer that can be shared with external C routines.
value	PIC X	Specifies the value to be copied to destination.
length	any numeric data item or numeric literal	Specifies the number of bytes to be copied.

Examples:

Example - Allocate 8 bytes of memory, fill the memory area with all "A" characters

```
*> define mem1 usage handle
call "m$alloc" using 8, mem1
call "m$fill" using mem1, "A", 8
```

M$FREE

The M$FREE library routine releases a previously allocated memory region

The memory region is usually allocated with the M$ALLOC library routine. However, programs compiled with `-cp` option may operate also on a memory regions allocated by external C functions or by a Format 7 SET statement.

Syntax:

```
CALL "M$FREE" USING memAddress
```

Parameters:

memAddress	USAGE HANDLE	Specifies the handle of the memory region to be released. **Note -** for compatibility reasons this item may also be defined as USAGE POINTER. In this case the program must be compiled either with -ca or -cp options. With -ca POINTER is translated to HANDLE. With -cp POINTER is a real pointer that can be shared with external C routines.

Examples:

Example - Allocate 8 bytes of memory on each of 2 different handles and release that memory after using it in the program.

```
*> define mem1 and mem2 usage handle
call "m$alloc" using 8, mem1
call "m$alloc" using 8, mem2
...
call "m$free" using mem1
call "m$free" using mem2
```

M$GET

The M$GET library routine copies the content of a memory region to a data item.

The memory region is usually allocated with the M$ALLOC library routine. However, programs compiled with `-cp` option may operate also on a memory regions allocated by external C functions or by a Format 7 SET statement.

Syntax:

```
CALL "M$GET" USING memAddress
                   dataItem
                  [dataSize]
                  [dataOffset]
```

Parameters:

memAddress	USAGE HANDLE	Specifies the handle to the source memory region. **Note -** for compatibility reasons this item may also be defined as USAGE POINTER. In this case the program must be compiled either with -ca or -cp options. With -ca POINTER is translated to HANDLE. With -cp POINTER is a real pointer that can be shared with external C routines.
dataItem	PIC X(n)	Receives the content of memory region pointed by memAddress.
dataSize	any numeric data item or numeric literal	Specifies the number of bytes to be moved to dataItem. If this parameter is omitted, all the allocated memory is moved to dataItem.
dataOffset	any numeric data item or numeric literal	Specifies the memory offset from which the data will be copied. The default value is 1.

Examples:

Example - Allocate 8 bytes of memory, put a value in it and then query the value putting it into a pic x(8) variable

```
*> define mem1 usage handle
*> define str1 as pic x(8)
call "m$alloc" using 8, mem1
call "m$put" using mem1, "Hello", 8, 1
call "m$get" using mem1, str1, 8, 1
```

M$PUT

The M$PUT library routine copies the content of data item to a memory region.

The memory region is usually allocated with the M$ALLOC library routine. However, programs compiled with `-cp` option may operate also on a memory regions allocated by external C functions or by a Format 7 SET statement.

Syntax:

```
CALL "M$PUT" USING  memAddress
                    dataItem
                   [dataSize]
                   [dataOffset]
```

Parameters:

memAddress	USAGE HANDLE	Specifies the handle to the destination memory region. **Note -** for compatibility reasons this item may also be defined as USAGE POINTER. In this case the program must be compiled either with -ca or -cp options. With -ca POINTER is translated to HANDLE. With -cp POINTER is a real pointer that can be shared with external C routines.
dataItem	PIC X(n)	Specifies the content of memory region pointed by memAddress.
dataSize	any numeric data item or numeric literal	Specifies the number of bytes to be moved from dataItem. If this parameter is omitted, the entire content of dataItem is copied.
dataOffset	any numeric data item or numeric literal	Specifies the memory offset from which the data will be copied from. The default value is 1.

Examples:

Example - Allocate 8 bytes of memory, put a value in it

```
*> define mem1 usage handle
call "m$alloc" using 8, mem1
call "m$put" using mem1, "Hello", 8, 1
```

M$SIZE

The M$SIZE library routine retrieves the size of a memory region.

The memory region is usually allocated with the M$ALLOC library routine. However, programs compiled with `-cp` option may operate also on a memory regions allocated by external C functions or by a Format 7 SET statement.

Syntax:

```
CALL "M$SIZE" USING memAddress
              GIVING memSize
```

Parameters:

memAddress	USAGE HANDLE	Specifies the handle to the memory region. **Note -** for compatibility reasons this item may also be defined as USAGE POINTER. In this case the program must be compiled either with -ca or -cp options. With -ca POINTER is translated to HANDLE. With -cp POINTER is a real pointer that can be shared with external C routines.

Return code:

memSize can be any numeric data item. It receives the size of the memory region pointed by memAddress.

Examples:

Example - Allocate 8 bytes of memory, check if the memory size is 8

```
*> define msize as pic 9(2) and mem1 usage handle

call "m$alloc" using 8, mem1
call "m$size" using mem1 giving msize
if msize not = 8
   display "Error allocating memory"
end-if
```

OCTAL2ASCII

The OCTAL2ASCII library routine converts an octal number to a string according to the ASCII table.

Syntax:

```
CALL "OCTAL2ASCII" USING octalValue
                         asciiValue
```

Parameters:

octalValue	PIC 9(8)	Specifies the octal number to be converted. Valid values range from 0 to 177777.
asciiValue	PIC X(2)	Receives the converted ASCII string.

Examples:

Example - Convert Octal value to ascii characters

```
*> define oct-value as pic 9(8)
*> define asc-value as pic x(2)

move 00040501 to oct-value
move spaces   to asc-value

call "octal2ascii" using oct-value asc-value
*> asc-value will contain "AA"
```

P$

The P$ library routines allow access to printing features.

Routine	Feature
P$CLEARDIALOG	resets the *Choose Printer* dialog
P$CLEARFONT	clears font description
P$DISABLEDIALOG	disables automatic *Choose Printer* dialog
P$DISPLAYDIALOG	shows the *Choose Printer* dialog
P$DRAWBITMAP	prints a bitmap
P$DRAWBOX	draws a box
P$DRAWLINE	draws a line
P$DRAWROUNDBOX	draws a rounded box
P$ENABLEDIALOG	enables automatic *Choose Printer* dialog
P$GETDEVICECAPABILITIES	retrieves device capabilities
P$GETDIALOG	retrieves current *Choose Printer* dialog fields values
P$GETFONT	retrieves current font description
P$GETTEXTMETRICS	retrieves characteristics of the current font
P$NEWPAGE	forces the next printer output to a new page.
P$SETDEFAULTMODE	changes default mode for positions and sizes
P$SETDEFAULTUNITS	changes default measurement unit for positions and sizes
P$SETDIALOG	sets values for the *Choose Printer* dialog
P$SETDOCUMENTNAME	sets the print job name
P$SETFONT	changes the current font

Routine	Feature
P$SETPEN	sets the style, width and color of the pen
P$SETPOSITION	sets the position for the next print operation
P$SETTEXTCOLOR	sets the text color
P$SETTEXTPOSITION	sets the position for the next print operation adjusted from the top or bottom of the current font
P$SETTOPMARGIN	sets the top margin of the paper
P$TEXTOUT	prints text

Note - These routines are supported for RM/COBOL compatibility. In order to take advantage of every print feature provided by isCOBOL, refer to WIN$PRINTER.

Examples:

Example - Create a graphical document

```
       program-id. pprinter.

       input-output section.
       file-control.
       select print-job assign to printer spooler-name
           organization line sequential.

       file section.
       fd  print-job.
       01  print-record            pic x(80).

       working-storage section.
       copy "isgui.def".
       copy "iscrt.def".
       copy "isopensave.def".
       77  crt-status              is special-names crt status pic 9(5).
       77  hWin                    handle of window.
       77  close-win               pic 9 value 0.
       77  spooler-name            pic x(128).

       screen section.
       01  mask.
           03 push-button
              line                 2
              col                  2
              title                "&Print"
              exception-value      101
              .
           03 push-button
              line                  2
              col                   + 2
              title                 "Pre&view"
              exception-value       102
              .
           03 push-button
              line                 2
              col                  + 2
              title                "PD&F"
              exception-value      103
              .

       procedure division.
       main.
           call "c$guicfg" using "Printer Dialog Always=False"

           call "c$setdevelopmentmode"
```

```
           display independent graphical window
                   color  65793
                   with   system menu
                   title  "P$ Routines"
                   handle hWin
                   event  win-evt

           display Mask

           perform until crt-status = 27 or close-win = 1
             accept Mask
                    on exception
                       continue
             end-accept
             evaluate crt-status
                      when 101
                           perform normal-print
                      when 102
                           perform print-preview
                      when 103
                           perform print-pdf
             end-evaluate
           end-perform

           destroy mask
           destroy hwin
           goback
           .


       print-preview.
           move "-p preview" to spooler-name
           perform print-procedure
           .


       print-pdf.
           initialize opensave-data, spooler-name.
           accept opnsav-default-dir from environment "user-path"
           move "PDF Files (*.pdf)|*.pdf" to opnsav-filters
           move "pdf"                     to opnsav-default-ext
           call "c$opensavebox" using opensave-save-box
                                      opensave-data
           if return-code < 0
              exit paragraph
           end-if
```

```
           string "-P PDF "       delimited by size
                  opnsav-filename delimited by trailing spaces
                                  into spooler-name

           perform print-procedure
           .

       normal-print.
           move "-p spooler" to spooler-name
           perform print-procedure
           .


       print-procedure.
           open output print-job
      *print of bitmap pictures
           call "p$drawbitmap" using "files/img.png",
                                     3, 3, "Absolute", "Metric"
      *print of colored strings (a red text in this case)
           call "p$settextcolor" using "Red"
           call "p$textout" using "colored string", 2, 9,
                                  "Absolute", "Metric"
      *print of graphical shapes (how to create a table)
           call "p$drawbox"  using 2, 10, "Absolute", "Metric"
                                  16,  5, "Metric"
           call "p$drawline" using 5, 10, "Absolute", "Metric"
                                   5, 15, "Absolute", "Metric"
           call "p$drawline" using 2, 12, "Absolute", "Metric"
                                  18, 12, "Absolute", "Metric"

           close print-job
           .


       win-evt.
           if event-type = cmd-close
              move 1 to close-win
           end-if.
```

P$CLEARDIALOG

The P$CLEARDIALOG library routine clears the standard *Choose Printer* dialog box values back to their default (unset) state.

Syntax

```
CALL "P$CLEARDIALOG"
```

P$CLEARFONT

The P$CLEARFONT library routine clears the font description values that were set using P$SETFONT and returns them to their default (unset) state.

Syntax

```
CALL "P$CLEARFONT"
```

P$DISABLEDIALOG

The P$DISABLEDIALOG library routine causes the *Choose Printer* dialog box not to be displayed the next time the predefined dynamic printer device is opened.

Syntax

```
CALL "P$DISABLEDIALOG"
```

P$DISPLAYDIALOG

The P$DISPLAYDIALOG library routine invokes the standard *Choose Printer* dialog box. After choosing a printer with this dialog box, the next open of a dynamic printer device will use the selected printer.

Syntax

```
CALL "P$DISPLAYDIALOG" GIVING dialogReturn
```

Return code

dialogReturn	PIC 9(n)	Receives the exit status: zero if the user selects a printer and non-zero if the user cancels the operation.

P$DRAWBITMAP

The P$DRAWBITMAP library routine prints a bitmap.

If you need to print text over the bitmap, do it after this call. If you do it before, the bitmap covers the text.

Syntax

```
CALL "P$DRAWBITMAP" USING fileName
                          [xPosition, yPosition]
                          [positionMode]
                          [positionUnits]
                          [sizeWidth, sizeHeight]
                          [sizeUnits]
                    [GIVING returnCode]
```

Parameters

fileName	PIC X(n)	Specifies the bitmap file. The following extensions are supported: BMP, JPG, PNG, GIF
xPosition	any numeric data item	Optional. X coordinate of the bitmap.
yPosition	any numeric data item	Optional. Y coordinate of the bitmap.
positionMode	PIC X(n)	Optional. Must contain “Absolute”
positionUnits	PIC X(n)	Optional. Can be either "Inches", "Metric", "Characters", or "Device Units". This value should match with the value of *SizeUnits*.
sizeWidth	any numeric data item	Optional. Width of the bitmap.
sizeHeight	any numeric data item	Optional. Height of the bitmap.
sizeUnits	PIC X(n)	Optional. Can be either "Inches", "Metric", "Characters", or "Device Units". This value should match with the value of *PositionUnits*.

Return code

returnCode	PIC 9(n)	Receives the exit status: zero for failure, non-zero for success

P$DRAWBOX

The P$DRAWBOX library routine draws a box.

Syntax

```
CALL "P$DRAWBOX" USING [xPosition, yPosition]
                       [positionMode]
                       [positionUnits]
                       [sizeWidth, sizeHeight]
                       [sizeUnits]
```

Parameters

xPosition	any numeric data item	Optional. X coordinate of the box.
yPosition	any numeric data item	Optional. Y coordinate of the box.
positionMode	PIC X(n)	Optional. Must contain “Absolute”
positionUnits	PIC X(n)	Optional. Can be either "Inches", "Metric", "Characters", or "Device Units". This value should match with the value of *SizeUnits*.
sizeWidth	any numeric data item	Optional. Width of the box.
sizeHeight	any numeric data item	Optional. Height of the box.
sizeUnits	PIC X(n)	Optional. Can be either "Inches", "Metric", "Characters", or "Device Units". This value should match with the value of *PositionUnits*.

P$DRAWLINE

The P$DRAWLINE library routine draws a line.

Syntax

```
CALL "P$DRAWLINE" USING [x1Point, y1Point]
                        [point1Mode]
                        [point1Units]
                        [x2Point, y2Point]
                        [point2Mode]
                        [point2Units]
```

Parameters

x1Point	any numeric data item	Optional. X coordinate where the line starts.
y1Point	any numeric data item	Optional. Y coordinate where the line starts.
point1Mode	PIC X(n)	Optional. Must contain “Absolute”
point1Units	PIC X(n)	Optional. Can be either "Inches", "Metric", "Characters", or "Device Units". This value should match with the value of *Point2Units*.
x2Point	any numeric data item	Optional. X coordinate where the line ends.
y2Point	any numeric data item	Optional. Y coordinate where the line ends.
point2Mode	PIC X(n)	Optional. Must contain “Absolute”
point2Units	PIC X(n)	Optional. Can be either "Inches", "Metric", "Characters", or "Device Units". This value should match with the value of *Point1Units*.

P$DRAWROUNDBOX

The P$DRAWROUNDBOX library routine draws a box with rounded corners.

Syntax

```
CALL "P$DRAWROUNDBOX" USING [xPosition, yPosition]
                            [positionMode]
                            [positionUnits]
                            [sizeWidth, sizeHeight]
                            [sizeUnits]
```

Parameters

xPosition	any numeric data item	Optional. X coordinate of the box.
yPosition	any numeric data item	Optional. Y coordinate of the box.
positionMode	PIC X(n)	Optional. Must contain “Absolute”
positionUnits	PIC X(n)	Optional. Can be either "Inches", "Metric", "Characters", or "Device Units". This value should match with the value of *SizeUnits*.
sizeWidth	any numeric data item	Optional. Width of the box.
sizeHeight	any numeric data item	Optional. Height of the box.
sizeUnits	PIC X(n)	Optional. Can be either "Inches", "Metric", "Characters", or "Device Units". This value should match with the value of *PositionUnits*.

P$ENABLEDIALOG

The P$ENABLEDIALOG library routine causes the *Choose Printer* dialog box to display automatically the next time the predefined dynamic printer device is opened.

Syntax

```
CALL "P$ENABLEDIALOG"
```

P$GETDEVICECAPABILITIES

The P$GETDEVICECAPABILITIES returns some information about the current printing device.

Syntax 1

```
CALL "P$GETDEVICECAPABILITIES" USING deviceCaps
```

Parameters

deviceCaps	Group Item	Group Item defined as follows: 01 DeviceCapabilities. 02 DC-DriverVersion pic 9(10) Binary(4). 02 DC-TechnologyValue pic 9 Binary(4). 02 DC-HorzSize pic 9(10) Binary(4). 02 DC-VertSize pic 9(10) Binary(4). 02 DC-HorzRes pic 9(10) Binary(4). 02 DC-VertRes pic 9(10) Binary(4). 02 DC-LogPixelsX pic 9(10) Binary(4). 02 DC-LogPixelsY pic 9(10) Binary(4). 02 DC-AspectX pic 9(10) Binary(4). 02 DC-AspectY pic 9(10) Binary(4). 02 DC-AspectXY pic 9(10) Binary(4). 02 DC-PhysicalWidth pic 9(10) Binary(4). 02 DC-PhysicalHeight pic 9(10) Binary(4). 02 DC-PhysicalOffsetX pic 9(10) Binary(4). 02 DC-PhysicalOffsetY pic 9(10) Binary(4). 02 DC-ScalingFactorX pic 9(10) Binary(4). 02 DC-ScalingFactorY pic 9(10) Binary(4). **Note -** DC-DriverVersion, Dc-ScalingFactorX and DC-ScalingFactorY are always zero, while DC-TechnologyValue is always 2.

Syntax 2

```
CALL "P$GETDEVICECAPABILITIES" USING settingName1, settingValue1
                                     [settingName2, settingValue2]
                                      ...
                                     [settingNameN, settingValueN]
```

Parameters

settingName1 *settingName2* ... *settingNameN*	PIC X(n)	Specifies the setting name. Possible values are: "Driver Version" "Technology" "Horizontal Size" "Vertical Size" "Horizontal Resolution" "Vertical Resolution" "Logical Pixels X" "Logical Pixels Y" "Aspect X" "Aspect Y" "Aspect XY" "Physical Width" "Physical Height" "Physical Offset X" "Physical Offset Y" "Scaling Factor X" "Scaling Factor Y"
settingValue1 *settingValue2* ... *settingValueN*	PIC X(n) or PIC 9(n) depending on the value type	Receives the setting value.

P$GETDIALOG

The P$GETDIALOG library routine retrieves fields from the *Choose Printer* dialog box.

Syntax 1

```
CALL "P$GETDIALOG" USING printDialog
```

Parameters

printDialog	Group Item	Group Item defined as follows:

```
01  PrintDialog.
  02  PD-ReturnValue                pic x.
  02  PD-ExtendedErrorValue pic 9(5) binary(2).
  02  PD-Flags.
       03  PD-AllPagesFlagValue      pic x.
       03  PD-SelectionFlagValue     pic x.
       03  PD-PageNumbersFlagValue   pic x.
       03  PD-NoSelectionFlagValue   pic x.
       03  PD-NoPageNumbersFlagValue pic x.
       03  PD-CollateFlagValue       pic x.
       03  PD-PrintSetupFlagValue    pic x.
       03  PD-PrintToFileFlagValue   pic x.
       03  PD-NoWarningFlagValue     pic x.
       03  PD-UseDevModeCopiesFlagValue pic x.
       03  PD-DisablePrintToFileFlagValue pic x.
       03  PD-HidePrintToFileFlagValue pic x.
       03  PD-NoNetworkButtonFlagValue pic x.
  02  PD-FromPage           pic 9(5) binary(2).
  02  PD-ToPage             pic 9(5) binary(2).
  02  PD-MinPage            pic 9(5) binary(2).
  02  PD-MaxPage             pic 9(5) binary(2).
  02  PD-Copies              pic 9(5) binary(2).
  02  DM-DeviceName                 pic X(31).
  02  DM-Fields.
       03  DM-OrientationFieldValue  pic x.
       03  DM-PaperSizeFieldValue    pic x.
       03  DM-PaperLengthFieldValue  pic x.
       03  DM-PaperWidthFieldValue   pic x.
       03  DM-ScaleFieldValue        pic x.
       03  DM-CopiesFieldValue       pic x.
       03  DM-PaperSourceFieldValue  pic x.
       03  DM-PrintQualityFieldValue pic x.
       03  DM-ColorFieldValue        pic x.
       03  DM-DuplexFieldValue       pic x.
       03  DM-YResolutionFieldValue  pic x.
       03  DM-TrueTypeOptionFieldValue pic x.
       03  DM-CollateFieldValue      pic x.
       03  DM-ICMMethodFieldValue    pic x.
       03  DM-ICMIntentFieldValue    pic x.
       03  DM-MediaTypeFieldValue    pic x.
       03  DM-DitherTypeFieldValue   pic x.
  02  DM-OrientationValue   pic 9 binary(2).
  02  DM-PaperSizeValue      pic 9(2) binary(2).
  02  DM-PaperLength        pic 9(5) binary(2).
  02  DM-PaperWidth         pic 9(5) binary(2).
```

```
02  DM-Scale               pic 9(5) binary(2).
02  DM-Copies              pic 9(5) binary(2).
02  DM-PaperSourceValue    pic 9(2) binary(2).
02  DM-ResolutionValue     pic S9 binary(2).
02  DM-ColorValue          pic 9 binary(2).
02  DM-DuplexValue         pic 9 binary(2).
02  DM-YResolution         pic 9(5) binary(2).
02  DM-TrueTypeValue       pic 9 binary(2).
02  DM-CollateValue        pic 9 binary(2).
02  DM-ICMMethodValue      pic 9 binary(4).
02  DM-ICMIntentValue      pic 9 binary(4).
02  DM-MediaTypeValue      pic 9 binary(4).
02  DM-DitherTypeValue     pic 99 binary(4).
```

Note - only the following fields are supported:
- PD-CollateFlagValue
- PD-Copies
- DM-DeviceName
- DM-Fields
- DM-OrientationValue
- DM-PaperSizeValue
- DM-Copies
- DM-PaperSourceValue
- DM-ColorValue

Syntax 2

```
CALL "P$GETDIALOG" USING settingName1, settingValue1
                         [settingName2, settingValue2]
                         ...
                         [settingNameN, settingValueN]
```

Parameters

settingName1 *settingName2* ... *settingNameN*	PIC X(n)	Specifies the setting name. Possible values are: “Collate Flag” “Print Dialog Copies” “Device Name” “Orientation” “Paper Size” “Device Mode Copies” “Default Source” “Print Quality” “Color”
settingValue1 *settingValue2* ... *settingValueN*	PIC X(n) or PIC 9(n) depending on the value type	Receives the setting value.

P$GETFONT

The P$GETFONT library routine retrieves information on the current font.

Syntax 1

```
CALL "P$GETFONT" USING logicalFont
```

Parameters

logicalFont	Group Item	Group item with the following structure:
		
		`01 LogicalFont.`
		`  02  LF-Height  pic S9(5) Binary(2).`
		`  02  LF-Width  pic 9(5) Binary(2).`
		`  02  LF-Escapement  pic 9(5) Binary(2).`
		`  02  LF-Orientation  pic 9(5) Binary(2).`
		`  02  LF-WeightValue  pic 9(3) Binary(2).`
		`  02  LF-ItalicValue  pic x.`
		`  02  LF-UnderlineValue  pic x.`
		`  02  LF-StrikeoutValue  pic x.`
		`  02  LF-CharSetValue  pic 9(3) Binary(2).`
		`  02  LF-OutPrecisValue  pic 9 Binary(2).`
		`  02  LF-ClipPrecisValue pic 9(3) Binary(2).`
		`  02  LF-QualityValue  pic 9 Binary(2).`
		`  02  LF-PitchValue  pic 9 Binary(2).`
		`  02  LF-FamilyValue  pic 9 Binary(2).`
		`  02  LF-FaceName  pic x(31).`
		
		Note - only these fields are supported:
		- LF-Height
		- LF-Escapement
		- LF-WeightValue
		- LF-ItalicValue
		- LF-UnderlineValue
		- LF-StrikeoutValue
		- LF-PitchValue
		- LF-FaceName

Syntax 2

```
CALL "P$GETFONT" USING settingName1, settingValue1
                       [settingName2, settingValue2]
                       ...
                       [settingNameN, settingValueN]
```

Parameters

settingName1 *settingName2* ... *settingNameN*	PIC X(n)	Specifies the setting name. Possible values are: "Height" "Escapement" "Weight" "Italic" "Underline" "Strike Out" "Pitch" "Face Name"
settingValue1 *settingValue2* ... *settingValueN*	PIC X(n) or PIC 9(n) depending on the value type	Receives the setting value.

P$GETTEXTMETRICS

The P$GETTEXTMETRICS library routine retrieves characteristics of the current font.

Syntax 1

```
CALL "P$GETTEXTMETRICS" USING textMetrics
```

Parameters

textMetrics	Group Item	Group item with the following structure:
		
		`01 TextMetrics.`
		`  02  TM-Height  pic 9(10) Binary(4).`
		`  02  TM-Ascent  pic 9(10) Binary(4).`
		`  02  TM-Descent  pic 9(10) Binary(4).`
		`  02  TM-InternalLeading  pic 9(10) Binary(4).`
		`  02  TM-ExternalLeading  pic 9(10) Binary(4).`
		`  02  TM-AveCharWidth  pic 9(10) Binary(4).`
		`  02  TM-MaxCharWidth  pic 9(10) Binary(4).`
		`  02  TM-WeightValue  pic 9(3) Binary(4).`
		`  02  TM-Overhang  pic 9(10) Binary(4).`
		`  02  TM-DigitizedAspectX pic 9(10) Binary(4).`
		`  02  TM-DigitizedAspectY pic 9(10) Binary(4).`
		`  02  TM-ItalicValue  pic x.`
		`  02  TM-UnderlinedValue  pic x.`
		`  02  TM-StruckOutValue  pic x.`
		`  02  TM-FirstChar  pic x.`
		`  02  TM-LastChar  pic x.`
		`  02  TM-DefaultChar  pic x.`
		`  02  TM-BreakChar  pic x.`
		`  02  TM-PitchValue  pic 9 Binary(2).`
		`  02  TM-FamilyValue  pic 9 Binary(2).`
		`  02  TM-CharSetValue  pic 9(3) Binary(2).`
		
		Note - only these fields are supported:
		- TM-Height
		- TM-Ascent
		- TM-Descent
		- TM-InternalLeading
		- TM-ExternalLeading
		- TM-AveCharWidth
		- TM-MaxCharWidth
		- TM-WeightValue
		- TM-ItalicValue
		- TM-UnderlinedValue
		- TM-StruckOutValue
		- TM-PitchValue

Syntax 2

```
CALL "P$GETTEXTMETRICS" USING settingName1, settingValue1
                              [settingName2, settingValue2]
                               ...
                              [settingNameN, settingValueN]
```

Parameters

settingName1 *settingName2* ... *settingNameN*	PIC X(n)	Specifies the setting name. Possible values are: "Height" "Ascent" "Descent" "Internal Leading" "External Leading" "Average Character Width" "Maximum Character Width" "Weight" "Italic" "Underlined" "Struck Out" "Pitch"
settingValue1 *settingValue2* ... *settingValueN*	PIC X(n) or PIC 9(n) depending on the value type	Receives the setting value.

P$NEWPAGE

The P$NEWPAGE library routine forces the next printer output to a new page.

Syntax

```
CALL "P$NEWPAGE"
```

P$SETDEFAULTMODE

The P$SETDEFAULTMODE library routine changes the default mode for positions and sizes.

Syntax

```
CALL "P$SETDEFAULTMODE" USING mode
```

Parameters

mode	PIC X(n)	Must contain “Absolute”.

P$SETDEFAULTUNITS

The P$SETDEFAULTUNITS library routine changes the default measurement unit for positions and sizes.

Syntax

```
CALL "P$SETDEFAULTUNITS" USING units
```

Parameters

units	PIC X(n)	Can be either "Inches", "Metric", "Characters", or "Device Units".

P$SETDIALOG

The P$SETDIALOG library routine initializes fields for the *Choose Printer* dialog box.

Syntax 1

```
CALL "P$SETDIALOG" USING printDialog
```

Parameters

printDialog	Group Item	Group Item defined as follows:

```
01  PrintDialog.
 02  PD-ReturnValue                pic x.
 02  PD-ExtendedErrorValue pic 9(5) binary(2).
 02  PD-Flags.
  03  PD-AllPagesFlagValue       pic x.
  03  PD-SelectionFlagValue      pic x.
  03  PD-PageNumbersFlagValue    pic x.
  03  PD-NoSelectionFlagValue    pic x.
  03  PD-NoPageNumbersFlagValue pic x.
  03  PD-CollateFlagValue        pic x.
  03  PD-PrintSetupFlagValue     pic x.
  03  PD-PrintToFileFlagValue    pic x.
  03  PD-NoWarningFlagValue      pic x.
  03  PD-UseDevModeCopiesFlagValue pic x.
  03  PD-DisablePrintToFileFlagValue pic x.
  03  PD-HidePrintToFileFlagValue pic x.
  03  PD-NoNetworkButtonFlagValue pic x.
 02  PD-FromPage           pic 9(5) binary(2).
 02  PD-ToPage             pic 9(5) binary(2).
 02  PD-MinPage            pic 9(5) binary(2).
 02  PD-MaxPage            pic 9(5) binary(2).
 02  PD-Copies             pic 9(5) binary(2).
 02  DM-DeviceName                 pic X(31).
 02  DM-Fields.
  03  DM-OrientationFieldValue  pic x.
  03  DM-PaperSizeFieldValue    pic x.
  03  DM-PaperLengthFieldValue  pic x.
  03  DM-PaperWidthFieldValue   pic x.
  03  DM-ScaleFieldValue        pic x.
  03  DM-CopiesFieldValue       pic x.
  03  DM-PaperSourceFieldValue  pic x.
  03  DM-PrintQualityFieldValue pic x.
  03  DM-ColorFieldValue        pic x.
  03  DM-DuplexFieldValue       pic x.
  03  DM-YResolutionFieldValue  pic x.
  03  DM-TrueTypeOptionFieldValue Pic x.
  03  DM-CollateFieldValue      pic x.
  03  DM-ICMMethodFieldValue    pic x.
  03  DM-ICMIntentFieldValue    pic x.
  03  DM-MediaTypeFieldValue    pic x.
  03  DM-DitherTypeFieldValue   pic x.
 02  DM-OrientationValue      pic 9 binary(2).
 02  DM-PaperSizeValue     pic 9(2) binary(2).
 02  DM-PaperLength        pic 9(5) binary(2).
```

```
02  DM-PaperWidth          pic 9(5) binary(2).
02  DM-Scale               pic 9(5) binary(2).
02  DM-Copies              pic 9(5) binary(2).
02  DM-PaperSourceValue    pic 9(2) binary(2).
02  DM-ResolutionValue     pic S9 binary(2).
02  DM-ColorValue          pic 9 binary(2).
02  DM-DuplexValue         pic 9 binary(2).
02  DM-YResolution         pic 9(5) binary(2).
02  DM-TrueTypeValue       pic 9 binary(2).
02  DM-CollateValue        pic 9 binary(2).
02  DM-ICMMethodValue      pic 9 binary(4).
02  DM-ICMIntentValue      pic 9 binary(4).
02  DM-MediaTypeValue      pic 9 binary(4).
02  DM-DitherTypeValue     pic 99 binary(4).
```

Note - only the following fields are supported:
- PD-Collate-Flag-Value
- PD-Copies
- DM-DeviceName
- DM-Fields
- DM-OrientationValue
- DM-PaperSizeValue
- DM-Copies
- DM-PaperSourceValue
- DM-ColorValue

Syntax 2

```
CALL "P$SETDIALOG" USING settingName1, settingValue1
                         [settingName2, settingValue2
                          ...
                         [settingNameN, settingValueN]
```

Parameters

settingName1 *settingName2* ... *settingNameN*	PIC X(n)	Specifies the setting name. Possible values are: "Collate Flag" "Print Dialog Copies" "Device Name" "Orientation" "Paper Size" "Device Mode Copies" "Default Source" "Print Quality" "Color"
settingValue1 *settingValue2* ... *settingValueN*	PIC X(n) or PIC 9(n) depending on the value type	Specifies the setting value. Possible values are: Print Dialog Copies -> any numeric value Device Name -> any alphanumeric value Orientation -> 1=Portrait, 2=Landscape Paper Size -> see *winprint-curr-papersize* constants in isprint.def Device Mode Copies -> any numeric value Default Source -> see *winprint-curr-tray* constants in isprint.def Print Quality" -> from -1 (highest) to -4 (lowest) or 0 for default Color -> 1=Monochrome, 2=Color

P$SETDOCUMENTNAME

The P$SETDOCUMENTNAME sets the name of the print job.

Syntax

```
CALL "P$SETDOCUMENTNAME" USING documentName
                               [destination]
```

Parameters

documentName	PIC X(n)	Specifies the job name.
destination	PIC X(n)	Optional. Specifies the job output. Possible value are: "Pdf" -> the document is saved as a PDF file "View" -> the document is shown as print preview "Spool" -> the document is sent to the system Spooler

P$SETFONT

The P$SETFONT library routine changes the current font.

Syntax 1

```
CALL "P$SETFONT" USING logicalFont
```

Parameters

logicalFont	Group Item	Group item with the following structure:
		
		`01 LogicalFont.`
		`  02  LF-Height  pic S9(5) Binary(2).`
		`  02  LF-Width  pic 9(5) Binary(2).`
		`  02  LF-Escapement  pic 9(5) Binary(2).`
		`  02  LF-Orientation  pic 9(5) Binary(2).`
		`  02  LF-WeightValue  pic 9(3) Binary(2).`
		`  02  LF-ItalicValue  pic x.`
		`  02  LF-UnderlineValue  pic x.`
		`  02  LF-StrikeoutValue  pic x.`
		`  02  LF-CharSetValue  pic 9(3) Binary(2).`
		`  02  LF-OutPrecisValue  pic 9 Binary(2).`
		`  02  LF-ClipPrecisValue pic 9(3) Binary(2).`
		`  02  LF-QualityValue  pic 9 Binary(2).`
		`  02  LF-PitchValue  pic 9 Binary(2).`
		`  02  LF-FamilyValue  pic 9 Binary(2).`
		`  02  LF-FaceName  pic x(31).`
		
		Note - only these fields are supported:
		- LF-Height
		- LF-Escapement
		- LF-WeightValue
		- LF-ItalicValue
		- LF-UnderlineValue
		- LF-StrikeoutValue
		- LF-PitchValue
		- LF-FaceName

Syntax 2

```
CALL "P$SETFONT" USING settingName1, settingValue1
                      [settingName2, settingValue2]
                       ...
                      [settingNameN, settingValueN]
```

Parameters

settingName1 *settingName2* ... *settingNameN*	PIC X(n)	Specifies the setting name. Possible values are: "Height" "Escapement" "Weight" "Italic" "Underline" "Strike Out" "Pitch" "Face Name"
settingValue1 *settingValue2* ... *settingValueN*	PIC X(n) or PIC 9(n) depending on the value type	Specifies the setting value. Possible values are: Height -> any numeric value Escapement -> any numeric value Weight -> any numeric value Italic -> “Y”=italic, “N”=not italic Underline -> “Y”=underline, “N”=not underline Strike Out -> “Y”=strike out, “N”=not strike out Pitch -> 1= fixed pitch, 2= variable pitch Face Name -> any alphanumeric value

P$SETPEN

The P$SETPEN library routine sets the style, width and color of the pen.

Syntax

```
CALL "P$SETPEN" USING  style
                      [width]
                      [color]
```

Parameters

style	any numeric data item	Specifies the pen style. Possible values are: 0 -> solid 1 -> dash 2 -> dot 3 -> dashdot 4 -> dashdotdot 5 -> null
width	any numeric data item	Optional. Width of the pen in logical units.
color	PIC X(n)	Optional. Specifies the color name. Possible values are: "Black" "Dark Blue" "Dark Green" "Dark Cyan" "Dark Red" "Dark Magenta" "Brown" "Dark Gray" "Light Gray" "Blue" "Green" "Cyan" "Red" "Magenta" "Yellow" "White"

P$SETPOSITION

The P$SETPOSITION library routine sets the position for the next print operation.

Syntax

```
CALL "P$SETPOSITION" USING [xPosition]
                           [yPosition]
                           [mode]
                           [units]
```

Parameters

xPosition	any numeric data item	Optional. X coordinate.
yPosition	any numeric data item	Optional. Y coordinate.
mode	PIC X(n)	Optional. Must contain “Absolute”.
units	PIC X(n)	Optional. Can be either "Inches", "Metric", "Characters", or "Device Units".

P$SETTEXTCOLOR

The P$SETTEXTCOLOR library routine sets text color.

Syntax

```
CALL "P$SETTEXTCOLOR" USING color
```

Parameters

color	PIC X(n)	Specifies the color name. Possible values are: "Black" "Dark Blue" "Dark Green" "Dark Cyan" "Dark Red" "Dark Magenta" "Brown" "Dark Gray" "Light Gray" "Blue" "Green" "Cyan" "Red" "Magenta" "Yellow" "White"

P$SETTEXTPOSITION

The P$SETTEXTPOSITION library routine sets the position for the next print operation adjusted from the top or bottom of the current font.

Syntax

```
CALL "P$SETTEXTPOSITION" USING [xPosition]
                               [yPosition]
                               [alignment]
                               [mode]
                               [units]
```

Parameters

xPosition	any numeric data item	Optional. X coordinate.
yPosition	any numeric data item	Optional. Y coordinate.
alignment	PIC X(n)	Optional. Can be either “Top” or “Bottom”.
mode	PIC X(n)	Optional. Must contain “Absolute”.
units	PIC X(n)	Optional. Can be either "Inches", "Metric", "Characters", or "Device Units".

P$SETTOPMARGIN

The P$SETTOPMARGIN library routine sets the top margin of the paper.

Syntax

```
CALL "P$SETTOPMARGIN" USING sizeHeight
                            [units]
```

Parameters

sizeHeight	any numeric data item	Specifies the margin size
Units	PIC X(n)	Optional. Can be either "Inches", "Metric", or "Device Units".

P$TEXTOUT

The P$TEXTOUT library routine prints some text.

Syntax

```
CALL "P$TEXTOUT" USING text
                       [xPosition]
                       [yPosition]
                       [mode]
                       [units]
```

Parameters

text	PIC X(n)	Specifies the text to print.
xPosition	any numeric data item	Optional. X coordinate of the text.
yPosition	any numeric data item	Optional. Y coordinate of the text.
mode	PIC X(n)	Optional. Must contain “Absolute”.
units	PIC X(n)	Optional. Can be either "Inches", "Metric", "Characters", or "Device Units".

R$IO

The R$IO routine provides a low-level interface to relative files allowing them to be managed without knowing their FD.

The configuration properties iscobol.file.relative and iscobol.file.relative.FileName specify which file handler is used.

The following configuration properties are ignored: iscobol.file.case, iscobol.file.prefix and iscobol.file.relative_file_prefix.

Syntax:

```
CALL "R$IO" USING opCode
                  parameters
           GIVING returnCode
```

Parameters:

opCode	Specifies the file handling function to be performed.	
	Valid values, defined in isfilesys.def are:	
	R-OPEN-FUNCTION	Opens an existing file
	R-CLOSE-FUNCTION	Closes an opened file
	R-MAKE-FUNCTION	Creates an empty file
	R-READ-FUNCTION	Reads a record from a file
	R-NEXT-FUNCTION	Reads the next record in a file
	R-PREVIOUS-FUNCTION	Reads the previous record in a file
	R-START-FUNCTION	Sets the file pointer on a specific record
	R-WRITE-FUNCTION	Writes data into file
	R-REWRITE-FUNCTION	Rewrites data into file
	R-DELETE-FUNCTION	Deletes data from file
	R-UNLOCK-FUNCTION	Unlocks all locked records in a file
parameters	Parameters depend on the opcode.	

Return code:

returnCode contains useful information such as file handles and record sizes or zero if an error occurs. Check the external variable F_ERRNO for additional information on the error.

Examples:

Example - Create a relative file, write and read records from it

```
working-storage section.
copy "isfilesys.def".

77  f                       handle .
77  file-io                 pic x(128).
77  rec-buffer              pic x(22).

01  rio-lparms.
    03 max-rec-sz           pic 999.
    03 filler               pic x value ",".
    03 min-rec-sz           pic 999.

77  key-val                 pic 9(5) value 0.

procedure division.
main.
*> creates the file
*> this is the FD
*>  01 record.
*>   03 r-key    pic 99.
*>   03 r-data   pic x(20).

  display "Creating file..."

  move "c:\tmp\myrelfile" to file-io
  move 22 to max-rec-sz, min-rec-sz
  set r-make-function to true
  call "r$io" using rio-function, file-io, rio-lparms

  if return-code = 0
     display message "R$IO Error: make : " F_ERRNO
  end-if

  *> opening
  display "Opening file..."

  set r-open-function to true
  move fio to open-mode
  move 22  to max-rec-size, min-rec-size
  call "r$io" using rio-function, file-io, open-mode
                    max-rec-size, min-rec-size
  if return-code > 0
     move return-code to f
  else
     display message "R$IO Error: open : " F_ERRNO
  end-if
```

```
*> record writing
move 0 to key-val.
display "Writing into file..."
set r-write-function to true
perform 3 times
   add 1 to key-val
   move "00xxx" to rec-buffer
   call "r$io" using rio-function, f, rec-buffer,
                             max-rec-size, key-val
   if return-code = 0
      display message "R$IO Error: write : " F_ERRNO
   end-if
end-perform

*> start
display "Getting the first record..."
set r-start-function to true
set f-equals       to true
move 1             to key-val
call "r$io" using rio-function, f, key-val, start-mode
if return-code = 0
   display message "R$IO Error: start : " F_ERRNO
end-if

*> read next
display "Reading next record..."
set r-next-function to true
call "r$io" using rio-function, f, rec-buffer
if return-code = 0
   display message "R$IO Error: next : " F_ERRNO
end-if

*> close
display "Closing file..."
set r-close-function to true
call "r$io" using rio-function, f

goback.
```

R-OPEN-FUNCTION

This function opens an existing relative file. If it is successful, the value in RETURN-CODE should be moved to a data item that is USAGE HANDLE. This data item is passed as the open file handle to the other file handling functions.

If it fails, RETURN-CODE is set to ZERO.

This function only opens already existing files. If the file does not exist, the function fails, even when opening for output.

Syntax:

```
SET R-OPEN-FUNCTION TO TRUE

CALL "R$IO" USING RIO-FUNCTION
                  name
                  mode
                  maxSize
                  minSize
            GIVING returnCode
```

Parameters:

name	the name of the file to open.
mode	one of the following values (defined in isfilesys.def): Finput — Open for input only. Foutput — Open for output only. Fio — Open for input and output. Fextend — Same as Foutput. mode may also have one of the following flags (defined in isfilesys.def) added to it to indicate file locking options: Fread_lock — Locks file against other updaters. Fwrite_lock — Locks file against all others. Fmass_update — Same as Fwrite_lock.
maxSize	the maximum record size.
minSize	the minimum record size. It should be equal to max-size, since relative files can't be variable lenght.

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	File handle.
0	Operation failed.

R-CLOSE-FUNCTION

This function closes an open file. It also removes currently held locks on the file.

Syntax:

```
SET R-CLOSE-FUNCTION TO TRUE

CALL "R$IO" USING RIO-FUNCTION
                  f
            GIVING returnCode
```

Parameters:

f	the file handle returned by R-OPEN-FUNCTION

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Operation successful.
0	Operation failed.

R-MAKE-FUNCTION

This function creates a new relative file. It overwrites any existing file of the same name unless they're in use. If the file is in use, the function fails.

Syntax:

```
SET R-MAKE-FUNCTION TO TRUE

CALL "R$IO" USING RIO-FUNCTION
                  name
                  lParms
           GIVING returnCode
```

Parameters:

name	the name of the file to create.
lParms	a string that describes the record size by providing two numbers separated by comma in the form: *max-rec-size,min-rec-size* Variable length files are not supported, so max-rec-size should be equal to min-rec-size.

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Operation successful.
0	Operation failed.

R-READ-FUNCTION

This function reads a record out of the relative file.

The record locking is controlled by the *f-no-lock* external data-item. Move 1 to *f-no-lock* to read without locking. Move 0 to *f-no-lock* to read with lock. The default value of *f-no-lock* is 0.

Syntax:

```
SET R-READ-FUNCTION TO TRUE

CALL "R$IO" USING RIO-FUNCTION
                  f
                  recBuffer
                  keyVal
            GIVING returnCode
```

Parameters:

f	the file handle returned by R-OPEN-FUNCTION.
recBuffer	an alphanumeric data-item that receives the record read.
keyVal	the record number of the record to read.

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Number of bytes read, plus one.
0	Operation failed.

R-NEXT-FUNCTION

This function reads the next record in the sequence of records in a relative file.

The record locking is controlled by the *f-no-lock* external data-item. Move 1 to *f-no-lock* to read without locking. Move 0 to *f-no-lock* to read with lock. The default value of *f-no-lock* is 0.

Syntax:

```
SET R-NEXT-FUNCTION TO TRUE

CALL "R$IO" USING RIO-FUNCTION
                  f
                  recBuffer
            GIVING returnCode
```

Parameters:

f	the file handle returned by R-OPEN-FUNCTION.
recBuffer	an alphanumeric data-item that receives the record read.

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Number of bytes read, plus one.
0	Operation failed.

R-PREVIOUS-FUNCTION

This function reads the previous record in the sequence of records in a relative file.

The record locking is controlled by the *f-no-lock* external data-item. Move 1 to *f-no-lock* to read without locking. Move 0 to *f-no-lock* to read with lock. The default value of *f-no-lock* is 0.

Syntax:

```
SET R-PREVIOUS-FUNCTION TO TRUE

CALL "R$IO" USING RIO-FUNCTION
                  f
                  recBuffer
            GIVING returnCode
```

Parameters:

f	the file handle returned by R-OPEN-FUNCTION.
recBuffer	an alphanumeric data-item that receives the record read.

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Number of bytes read, plus one.
0	Operation failed.

R-START-FUNCTION

This function positions the file pointer for the next R-NEXT-FUNCTION or R-PREVIOUS-FUNCTION.

Syntax:

```
SET R-START-FUNCTION TO TRUE

CALL "R$IO" USING RIO-FUNCTION
                  f
                  keyVal
                  mode
            GIVING returnCode
```

Parameters:

f	the file handle returned by R-OPEN-FUNCTION.
keyVal	the record number of the record at which to start.
mode	one of the following values (defined in isfilesys.def): F_EQUALS start at the specified key F_NOT_LESS start at the specified key, or the one after F_GREATER start at the record beyond the specified key F_LESS start at the record before the specified key F_NOT_GREATER start at the specified key, or the one before

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Operation successful.
0	Operation failed.

R-WRITE-FUNCTION

This function adds a new record to the relative file.

Syntax:

```
SET R-WRITE-FUNCTION TO TRUE

CALL "R$IO" USING RIO-FUNCTION
                  f
                  recBuffer
                  lenght
                  keyVal
            GIVING returnCode
```

Parameters:

f	the file handle returned by R-OPEN-FUNCTION.
recBuffer	an alphanumeric data-item that contains the record to write.
length	number of bytes to write.
keyVal	record number to write.

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Operation successful.

0	Operation failed.

R-REWRITE-FUNCTION

This function rewrites a record in the relative file. It doesn't change the file position.

Syntax:

```
SET R-REWRITE-FUNCTION TO TRUE

CALL "R$IO" USING RIO-FUNCTION
                  f
                  recBuffer
                  lenght
                  keyVal
            GIVING returnCode
```

Parameters:

f	the file handle returned by R-OPEN-FUNCTION.
recBuffer	an alphanumeric data-item that contains the new record data.
length	number of bytes to write.
keyBal	record number to write

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Operation successful.
0	Operation failed.

R-DELETE-FUNCTION

This function deletes the specified record. It does not affect the current file position.

Syntax:

```
SET R-DELETE-FUNCTION TO TRUE

CALL "R$IO" USING RIO-FUNCTION,
                  f
                  keyVal
            GIVING returnCode
```

Parameters:

f	the file handle returned by R-OPEN-FUNCTION.

key-val	record number to delete

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Operation successful.
0	Operation failed.

R-UNLOCK-FUNCTION

This function unlocks any locked records held by the current process in the specified file.

Syntax:

```
SET R-UNLOCK-FUNCTION TO TRUE

CALL "R$IO" USING RIO-FUNCTION
                  f
           GIVING ReturnCode
```

Parameters:

f	the file handle returned by R-OPEN-FUNCTION.

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Operation successful.
0	Operation failed.

REG

The REG library routines allow access to Windows registry.

Routine	Feature
REG_CLOSE_KEY, DISPLAY_REG_CLOSE_KEY	closes a key
REG_CREATE_KEY, DISPLAY_REG_CREATE_KEY	creates a key
REG_CREATE_KEY_EX, DISPLAY_REG_CREATE_KEY_EX	creates a key

Routine	Feature
REG_DELETE_KEY, DISPLAY_REG_DELETE_KEY	deletes a key
REG_DELETE_VALUE, DISPLAY_REG_DELETE_VALUE	deletes a value
REG_ENUM_KEY, DISPLAY_REG_ENUM_KEY	lists subkeys of a key
REG_ENUM_VALUE, DISPLAY_REG_ENUM_VALUE	lists values of a key
REG_OPEN_KEY, DISPLAY_REG_OPEN_KEY	opens a key
REG_OPEN_KEY_EX, DISPLAY_REG_OPEN_KEY_EX	opens a key
REG_QUERY_VALUE, DISPLAY_REG_QUERY_VALUE	retrieve the default value of a key
REG_QUERY_VALUE_EX, DISPLAY_REG_QUERY_VALUE_EX	retrieves a specific value of a key
REG_SET_VALUE, DISPLAY_REG_SET_VALUE	sets the default value of a key
REG_SET_VALUE_EX, DISPLAY_REG_SET_VALUE_EX	sets a specific value of a key

REG_CLOSE_KEY, DISPLAY_REG_CLOSE_KEY

The REG_CLOSE_KEY library routine closes a registry key.

To perform this action on the Windows client machine in an Application Server architecture, use DISPLAY_REG_CLOSE_KEY instead of REG_CLOSE_KEY.

Syntax:

```
CALL "REG_CLOSE_KEY" USING openKey
                     GIVING returnCode
```

Parameters:

openKey	USAGE UNSIGNED-LONG	Specifies the handle to the open key to be closed. The handle must have been opened by the REG_CREATE_KEY, REG_CREATE_KEY_EX, REG_OPEN_KEY or REG_OPEN_KEY_EX library routine.

Return code:

returnCode can be any signed numeric data item and provides additional information:

-1	Invalid or missing parameters, or not running on Windows
0	Operation successful.
Non zero	Operation failed. Click here for a list of error codes.

Examples:

Example - Close an opened key

```
working-storage section.
copy "isreg.def".
01  open-key-handle         usage unsigned-long.
01  subkey-handle           usage unsigned-long.
77  status-code             pic 9(3).
...
procedure division.
...
open-key.
   move hkey_local_machine to open-key-handle
   call "reg_open_key"  using open-key-handle
                              "SOFTWARE"
                              subkey-handle
                        giving status-code.

close-key.
   call "reg_close_key" using open-key-handle
                              giving status-code.
```

REG_CREATE_KEY, DISPLAY_REG_CREATE_KEY

The REG_CREATE_KEY library routine creates the specified registry key. If the key already exists in the registry, it is opened.

To perform this action on the Windows client machine in an Application Server architecture, use DISPLAY_REG_CREATE_KEY instead of REG_CREATE_KEY.

Syntax:

```
CALL "REG_CREATE_KEY" USING openKey
                            subKey
                            resultKey
                      GIVING returnCode
```

Parameters:

openKey	USAGE UNSIGNED-LONG	Specifies the handle to an open registry key. The calling process must have KEY_CREATE_SUB_KEY access to the key. This handle is returned by the REG_CREATE_KEY, REG_CREATE_KEY_EX, REG_OPEN_KEY or REG_OPEN_KEY_EX library routine, or it can be one of the following predefined keys, defined in isreg.def: HKEY_CLASSES_ROOT HKEY_CURRENT_CONFIG HKEY_CURRENT_USER HKEY_LOCAL_MACHINE HKEY_USERS HKEY_DYN_DATA
subKey	PIC X(n)	Specifies the name of a key that this library routine opens or creates. This key must be a subkey of the key identified by the openKey parameter. For more information on key names, see Structure of the Registry. If openKey is one of the predefined keys, subKey may be NULL. In that case, the handle returned in resultKey is the same openKey handle passed in to the function.
resultKey	USAGE UNSIGNED-LONG	Receives the handle to the opened or created key. If the key is not one of the predefined registry keys, call the REG_CLOSE_KEY library routine after you have finished using the handle.

Return code:

returnCode can be any signed numeric data item and provides additional information:

-1	Invalid or missing parameters, or not running on Windows
0	Operation successful.
Non zero	Operation failed. Click here for a list of error codes.

Examples:

Example - Create a subkey

```
working-storage section.
copy "isreg.def".
01  open-key-handle         usage unsigned-long.
01  subkey-handle           usage unsigned-long.
01  subkey-to-be-created    pic x(40).
77  status-code             pic 9(3).
...
procedure division.
...
subkey-creation.
   move "iscobol-test-key" to subkey-to-be-created.
   call "reg_create_key" using open-key-handle
                               subkey-to-be-created
                               subkey-handle
                       giving status-code.
```

REG_CREATE_KEY_EX, DISPLAY_REG_CREATE_KEY_EX

The REG_CREATE_KEY_EX library routine creates the specified registry key. If the key already exists, it is opened. Note that key names are not case sensitive.

To perform this action on the Windows client machine in an Application Server architecture, use DISPLAY_REG_CREATE_KEY_EX instead of REG_CREATE_KEY_EX.

Syntax:

```
 CALL "REG_CREATE_KEY_EX" USING openKey
                                subKey
                                class
                                options
                                sam
                                resultKey
                                disposition
                          GIVING returnCode
```

Parameters:

openKey	USAGE UNSIGNED-LONG	Specifies the handle to an open registry key. The calling process must have KEY_CREATE_SUB_KEY access to the key. This handle is returned by the REG_CREATE_KEY, REG_CREATE_KEY_EX, REG_OPEN_KEY or REG_OPEN_KEY_EX library routine, or it can be one of the following predefined keys, defined in isreg.def: HKEY_CLASSES_ROOT HKEY_CURRENT_CONFIG HKEY_CURRENT_USER HKEY_LOCAL_MACHINE HKEY_USERS HKEY_DYN_DATA

subKey	PIC X(n)	Specifies the name of a key that this library routine opens or creates. This key must be a subkey of the key identified by the openKey parameter. For more information on key names, see Structure of the Registry. If openKey is one of the predefined keys, subKey may be NULL. In that case, the handle returned in resultKey is the same openKey handle passed in to the function.	
class	PIC X(n)	Specifies the class (object type) of this key. It can be NULL.	
options	USAGE UNSIGNED-LONG	This parameter can be one of the following values, defined in isreg.def:	
		REG_OPTION_NON_VOLATILE	This key is not volatile; this is the default. The information is stored in a file and is preserved when the system is restarted.
		REG_OPTION_VOLATILE	All keys created by the function are volatile. The information is stored in memory and is not preserved when the corresponding registry hive is unloaded. For HKEY_LOCAL_MACHINE, this occurs when the system is shut down.
sam	USAGE UNSIGNED-LONG	It specifies a mask that specifies the access rights for the key. It must be the combination of the following values, defined in isreg.def:	

		KEY_ALL_ACCESS	Combines the STANDARD_RIGHTS_REQUIRED, KEY_QUERY_VALUE, KEY_SET_VALUE, KEY_CREATE_SUB_KEY, KEY_ENUMERATE_SUB_KEYS, KEY_NOTIFY, and KEY_CREATE_LINK access rights.
		KEY_CREATE_LINK	Reserved for system use.
		KEY_CREATE_SUB_KEY	Required to create a subkey of a registry key.
		KEY_ENUMERATE_SUB_KEYS	Required to enumerate the subkeys of a registry key.
		KEY_NOTIFY	Required to request change notifications for a registry key or for subkeys of a registry key.
		KEY_QUERY_VALUE	Required to query the values of a registry key.
		KEY_READ	Combines the STANDARD_RIGHTS_READ, KEY_QUERY_VALUE, KEY_ENUMERATE_SUB_KEYS, and KEY_NOTIFY values.
		KEY_SET_VALUE	Required to create, delete, or set a registry value.
		KEY_WRITE	Combines the STANDARD_RIGHTS_WRITE, KEY_SET_VALUE, and KEY_CREATE_SUB_KEY access rights.
resultKey	USAGE UNSIGNED-LONG	It receives a handle to the opened or created key. If the key is not one of the predefined registry keys, call the REG_CLOSE_KEY library routine after you have finished using the handle.	
disposition	USAGE UNSIGNED-LONG	It receives one of the following disposition values, defined in isreg.def:	
		REG_CREATED_NEW_KEY	The key did not exist and was created.
		REG_OPENED_EXISTING_KEY	The key existed and was simply opened without being changed.

Return code:

returnCode can be any signed numeric data item and provides additional information:

-1	Invalid or missing parameters, or not running on Windows
0	Operation successful.
Non zero	Operation failed. Click here for a list of error codes.

Examples:

Example - Create a subkey

```
working-storage section.
copy "isreg.def".
01  open-key-handle         usage unsigned-long.
01  subkey-handle           usage unsigned-long.
01  subkey-to-be-created    pic x(40).
77  status-code             pic 9(3).
01  key-class               pic x(10) value spaces.
01  key-options             usage unsigned-long.
01  key-sam                 usage unsigned-long.
01  key-disposition         usage unsigned-long.
...
procedure division.
...
subkey-creation.
   move "iscobol-test-key" to subkey-to-be-created.
   move REG_OPTION_NON_VOLATILE to key-options
   move KEY_ALL_ACCESS          to key-sam
   call "reg_create_key_ex" using open-key-handle
                                  subkey-to-be-created
                                  key-class
                                  key-options
                                  key-sam
                                  subkey-handle
                                  key-disposition
                            giving status-code.
```

REG_DELETE_KEY, DISPLAY_REG_DELETE_KEY

The REG_DELETE_KEY library routine deletes the specified registry key.

To perform this action on the Windows client machine in an Application Server architecture, use DISPLAY_REG_DELETE_KEY instead of REG_DELETE_KEY.

Syntax:

```
CALL "REG_DELETE_KEY" USING openKey
                            subKey
                      GIVING returnCode
```

Parameters:

openKey	USAGE UNSIGNED-LONG	Specifies the handle to an open registry key. The key must have been opened with the DELETE access right. This handle is returned by the REG_CREATE_KEY, REG_CREATE_KEY_EX, REG_OPEN_KEY or REG_OPEN_KEY_EX library routine, or it can be one of the following predefined keys, defined in isreg.def: HKEY_CLASSES_ROOT HKEY_CURRENT_CONFIG HKEY_CURRENT_USER HKEY_LOCAL_MACHINE HKEY_USERS HKEY_DYN_DATA
subKey	PIC X(n)	Specifies the name of the key to be deleted. It must be a subkey of the key that openKey identifies, but it cannot have subkeys. This parameter cannot be NULL. Key names are not case sensitive.

Return code:

returnCode can be any signed numeric data item and provides additional information:

-1	Invalid or missing parameters, or not running on Windows
0	Operation successful.
Non zero	Operation failed. Click here for a list of error codes.

Examples:

Example - Delete a key

```
working-storage section.
copy "isreg.def".
01  open-key-handle         usage unsigned-long.
01  subkey-handle           usage unsigned-long.
01  subkey-to-be-created    pic x(40).
77  status-code             pic 9(3).
...
procedure division.
...
subkey-creation.
   move "iscobol-test-key" to subkey-to-be-created.
   call "reg_create_key" using open-key-handle
                               subkey-to-be-created
                               subkey-handle
                       giving status-code.
...
delete-key.
    call "reg_delete_key" using open-key-handle
                                "iscobol-test-key"
                         giving status-code.
```

REG_DELETE_VALUE, DISPLAY_REG_DELETE_VALUE

The REG_DELETE_VALUE removes a named value from the specified registry key. Note that value names are not case sensitive.

To perform this action on the Windows client machine in an Application Server architecture, use DISPLAY_REG_DELETE_VALUE instead of REG_DELETE_VALUE.

Syntax:

```
CALL "REG_DELETE_VALUE" USING openKey
                              valueName
                        GIVING returnCode
```

Parameters:

openKey	USAGE UNSIGNED-LONG	Specifies the handle to an open registry key. This handle is returned by the REG_CREATE_KEY, REG_CREATE_KEY_EX, REG_OPEN_KEY or REG_OPEN_KEY_EX library routine, or it can be one of the following predefined keys, defined in isreg.def: HKEY_CLASSES_ROOT HKEY_CURRENT_CONFIG HKEY_CURRENT_USER HKEY_LOCAL_MACHINE HKEY_USERS HKEY_DYN_DATA
valueName	PIC X(n)	The registry value to be removed. If this parameter is NULL or an empty string, the value set by the REG_SET_VALUE library routine is removed.

Return code:

returnCode can be any signed numeric data item and provides additional information:

-1	Invalid or missing parameters, or not running on Windows
0	Operation successful.
Non zero	Operation failed. Click here for a list of error codes.

Examples:

Example - Delete registry value

```
working-storage section.
copy "isreg.def".
01  open-key-handle         usage unsigned-long.
77  status-code             pic 9(3).
...
procedure division.
...
delete-value.
   call "reg_delete_value" using open-key-handle
                                 "iscobol-key-value"
                           giving status-code.
```

REG_ENUM_KEY, DISPLAY_REG_ENUM_KEY

The REG_ENUM_KEY library routine enumerates the subkeys of the specified open registry key. It retrieves the name of one subkey each time it is called.

To perform this action on the Windows client machine in an Application Server architecture, use DISPLAY_REG_ENUM_KEY instead of REG_ENUM_KEY.

Syntax:

```
CALL "REG_ENUM_KEY" USING openKey
                          index
                          name
                          nameSize
                    GIVING returnCode
```

Parameters:

openKey	USAGE UNSIGNED-LONG	Specifies the handle to an open registry key. This handle is returned by the REG_CREATE_KEY, REG_CREATE_KEY_EX, REG_OPEN_KEY or REG_OPEN_KEY_EX library routine, or it can be one of the following predefined keys, defined in isreg.def: HKEY_CLASSES_ROOT HKEY_CURRENT_CONFIG HKEY_CURRENT_USER HKEY_LOCAL_MACHINE HKEY_USERS HKEY_DYN_DATA
index	USAGE UNSIGNED-LONG	Specifies the index of the subkey of openKey to be retrieved. This value should be zero for the first call to the REG_ENUM_KEY library routine and then increased incrementally for subsequent calls. Because subkeys are not ordered, any new subkey will have an arbitrary index. This means that the function may return subkeys in any order.
name	PIC X(n)	Receives the name of the subkey, including the terminating null character. This function copies only the name of the subkey, not the full key hierarchy, to the buffer.
nameSize	USAGE UNSIGNED-LONG	Specifies the size of the name parameter

Return code:

returnCode can be any signed numeric data item and provides additional information:

-1	Invalid or missing parameters, or not running on Windows
0	Operation successful.

Non zero	Operation failed. Click here for a list of error codes.

Examples:

Example - Get subkeys list

```
working-storage section.
copy "isreg.def".
01  open-key-handle         usage unsigned-long.
77  ndx                     pic 9(3).
77  status-code             pic 9(3).
01  subkey-name             pic x(40).
01  name-size               usage unsigned-long.
...
procedure division.
...
get-subkeys.
    set name-size to size of subkey-name
    perform varying ndx from 1 by 1 until 1 = 2
       call "reg_enum_key" using  open-key-handle
                                  ndx
                                  subkey-name
                                  name-size
                          giving  status-code

       if status-code not = 0
          exit perform
       end-if
       display subkey-name
    end-perform.
```

REG_ENUM_VALUE, DISPLAY_REG_ENUM_VALUE

The REG_ENUM_VALUE library routine removes a named value from the specified registry key. Note that value names are not case sensitive.

To perform this action on the Windows client machine in an Application Server architecture, use DISPLAY_REG_ENUM_VALUE instead of REG_ENUM_VALUE.

Syntax:

```
CALL "REG_ENUM_VALUE" USING openKey
                            index
                            valueName
                            valueNameSize
                            type
                            data
                            dataSize
                      GIVING returnCode
```

Parameters:

openKey	USAGE UNSIGNED-LONG	Specifies the handle to an open registry key. This handle is returned by the REG_CREATE_KEY, REG_CREATE_KEY_EX, REG_OPEN_KEY or REG_OPEN_KEY_EX library routine, or it can be one of the following predefined keys, defined in isreg.def: HKEY_CLASSES_ROOT HKEY_CURRENT_CONFIG HKEY_CURRENT_USER HKEY_LOCAL_MACHINE HKEY_USERS HKEY_DYN_DATA
index	USAGE UNSIGNED-LONG	Specifies the index of the value to be retrieved. This parameter should be zero for the first call to the REG_ENUM_VALUE library routine and then be increased incrementally for subsequent calls. Because values are not ordered, any new value will have an arbitrary index. This means that the function may return values in any order.
valueName	PIC X(n)	Receives the name of the value, including the terminating null character.
valueNameSize	USAGE UNSIGNED-LONG	Specifies the size of the valueName parameter. This size should include the terminating null character. When the function returns, valueNameSize contains the number of characters stored in valueName. The count returned does not include the terminating null character.
type	USAGE UNSIGNED-LONG	It receives a code indicating the type of data stored in the specified value. It can be one of the following values, defined in isreg.def:

		REG_BINARY	Binary data in any form.
		REG_DWORD	A 32-bit number.
		REG_DWORD_LITTLE_ENDIAN	A 32-bit number in little-endian format.
		REG_DWORD_BIG_ENDIAN	A 32-bit number in big-endian format.
		REG_EXPAND_SZ	A null-terminated string that contains unexpanded references to environment variables (for example, "%PATH%").
		REG_LINK	Reserved for system use.
		REG_MULTI_SZ	A sequence of null-terminated strings. The following is an example: `String1\0String2\0String3\0LastString\0\0` The first \0 terminates the first string, the second to the last \0 terminates the last string, and the final \0 terminates the sequence. Note that the final terminator must be factored into the length of the string.
		REG_NONE	No defined value type.
		REG_QWORD	A 64-bit number.
		REG_QWORD_LITTLE_ENDIAN	A 64-bit number in little-endian format.
		REG_SZ	A string.
data	PIC X(n)	Receives the data for the value entry. This parameter can be NULL if the data is not required. If data is NULL and dataSize is non-NULL, the function stores the size of the data, in bytes, in the variable pointed to by dataSize. This enables an application to determine the best way to allocate a buffer for the data.	
dataSize	USAGE UNSIGNED-LONG	Specifies the size of the data parameter. When the function returns, dataSize contains the number of bytes stored in data. This parameter can be NULL only if data is NULL. If the data has the REG_SZ, REG_MULTI_SZ or REG_EXPAND_SZ type, this size includes any terminating null character or characters. If data is not large enough to hold the data, the function returns ERROR_MORE_DATA and stores the required buffer size in dataSize. In this case, the contents of data are undefined. Registry value names are limited to 32767 bytes. Therefore, if you specify a value greater than 32767 bytes, there is an overflow and the function may return ERROR_MORE_DATA.	

Return code:

returnCode can be any signed numeric data item and provides additional information:

-1	Invalid or missing parameters, or not running on Windows
0	Operation successful.
Non zero	Operation failed. Click here for a list of error codes.

Examples:

Example - Enumarate the values of a key

```
working-storage section.
copy "isreg.def".
01  open-key-handle         usage unsigned-long.
01  subkey-key-handle       usage unsigned-long.
77  ndx                     pic 9(3).
77  status-code             pic 9(3).
01  subkey-name             pic x(40).
01  name-size               usage unsigned-long.
01  data-type               usage unsigned-long.
01  value-data              pic x(40).
01  data-size               usage unsigned-long.
...
procedure division.
...
enum-values.
    set name-size to size of value-name
    set data-size to size of value-data
    perform varying ndx from 1 by 1 until 1 = 2
       call "reg_enum_value" using subkey-handle
            ndx
            value-name
            name-size
            data-type
            value-data
            data-size
            giving status-code
       if status-code not = 0
          exit perform
       end-if
       display value-name
    end-perform.
```

REG_OPEN_KEY, DISPLAY_REG_OPEN_KEY

The REG_OPEN_KEY library routine opens the specified registry key.

To perform this action on the Windows client machine in an Application Server architecture, use DISPLAY_REG_OPEN_KEY instead of REG_OPEN_KEY.

Syntax:

```
CALL "REG_OPEN_KEY" USING openKey
                          subKey
                          resultKey
                    GIVING returnCode
```

Parameters:

openKey	USAGE UNSIGNED-LONG	Specifies the handle to an open registry key. This handle is returned by the REG_CREATE_KEY, REG_CREATE_KEY_EX, REG_OPEN_KEY or REG_OPEN_KEY_EX library routine, or it can be one of the following predefined keys, defined in isreg.def: HKEY_CLASSES_ROOT HKEY_CURRENT_CONFIG HKEY_CURRENT_USER HKEY_LOCAL_MACHINE HKEY_USERS HKEY_DYN_DATA
subKey	PIC X(n)	Specifies the name of the registry key to be opened. This key must be a subkey of the key identified by the openKey parameter. Key names are not case sensitive. If this parameter is NULL or an empty string, the function returns the same handle that was passed in.
resultKey	USAGE UNSIGNED-LONG	It receives a handle to the opened key. If the key is not one of the predefined registry keys, call the REG_CLOSE_KEY library routine after you have finished using the handle.

Return code:

returnCode can be any signed numeric data item and provides additional information:

-1	Invalid or missing parameters, or not running on Windows
0	Operation successful.
Non zero	Operation failed. Click here for a list of error codes.

Examples:

Example - Open a key

```
working-storage section.
copy "isreg.def".
01  open-key-handle         usage unsigned-long.
01  subkey-handle           usage unsigned-long.
77  status-code             pic 9(3).
...
procedure division.
...
open-key.
   move hkey_local_machine to open-key-handle
   call "reg_open_key"  using open-key-handle
                              "SOFTWARE"
                              subkey-handle
                        giving status-code.
```

REG_OPEN_KEY_EX, DISPLAY_REG_OPEN_KEY_EX

The REG_OPEN_KEY_EX library routine opens the specified registry key. Note that key names are not case sensitive.

To perform this action on the Windows client machine in an Application Server architecture, use DISPLAY_REG_OPEN_KEY_EX instead of REG_OPEN_KEY_EX.

Syntax:

```
 CALL "REG_OPEN_KEY_EX" USING openKey
                              subKey
                              sam
                              resultKey
                        GIVING returnCode
```

Parameters:

openKey	USAGE UNSIGNED-LONG	Specifies the handle to an open registry key. The calling process must have KEY_CREATE_SUB_KEY access to the key. This handle is returned by the REG_CREATE_KEY, REG_CREATE_KEY_EX, REG_OPEN_KEY or REG_OPEN_KEY_EX library routine, or it can be one of the following predefined keys, defined in isreg.def: HKEY_CLASSES_ROOT HKEY_CURRENT_CONFIG HKEY_CURRENT_USER HKEY_LOCAL_MACHINE HKEY_USERS HKEY_DYN_DATA
subKey	PIC X(n)	Specifies the name of the registry subkey to be opened. Key names are not case sensitive. If this parameter is NULL or an empty string, the function will open a new handle to the key identified by the openKey parameter.
sam	USAGE UNSIGNED-LONG	It specifies a mask that specifies the desired access rights to the key. The function fails if the security descriptor of the key does not permit the requested access for the calling process. It must be the combination of the following values, defined in isreg.def:

		KEY_ALL_ACCESS	Combines the STANDARD_RIGHTS_REQUIRED, KEY_QUERY_VALUE, KEY_SET_VALUE, KEY_CREATE_SUB_KEY, KEY_ENUMERATE_SUB_KEYS, KEY_NOTIFY, and KEY_CREATE_LINK access rights.
		KEY_CREATE_LINK	Reserved for system use.
		KEY_CREATE_SUB_KEY	Required to create a subkey of a registry key.
		KEY_ENUMERATE_SUB_KEYS	Required to enumerate the subkeys of a registry key.
		KEY_NOTIFY	Required to request change notifications for a registry key or for subkeys of a registry key.
		KEY_QUERY_VALUE	Required to query the values of a registry key.
		KEY_READ	Combines the STANDARD_RIGHTS_READ, KEY_QUERY_VALUE, KEY_ENUMERATE_SUB_KEYS, and KEY_NOTIFY values.
		KEY_SET_VALUE	Required to create, delete, or set a registry value.
		KEY_WRITE	Combines the STANDARD_RIGHTS_WRITE, KEY_SET_VALUE, and KEY_CREATE_SUB_KEY access rights.
resultKey	USAGE UNSIGNED-LONG	It receives a handle to the opened or created key. If the key is not one of the predefined registry keys, call the REG_CLOSE_KEY library routine after you have finished using the handle.	

Return code:

returnCode can be any signed numeric data item and provides additional information:

-1	Invalid or missing parameters, or not running on Windows
0	Operation successful.
Non zero	Operation failed. Click here for a list of error codes.

Examples:

Example - Open a key

```
working-storage section.
copy "isreg.def".
01  open-key-handle         usage unsigned-long.
01  subkey-handle           usage unsigned-long.
77  status-code             pic 9(3).
01  key-sam                 usage unsigned-long.
...
procedure division.
...
open-key.
   move hkey_local_machine to open-key-handle
   move KEY_ALL_ACCESS to key-sam
   call "reg_open_key_ex"  using open-key-handle
                                 "SOFTWARE"
                                 key-sam
                                 subkey-handle
                           giving status-code.
```

REG_QUERY_VALUE, DISPLAY_REG_QUERY_VALUE

The REG_QUERY_VALUE library routine retrieves the data associated with the default or unnamed value of a specified registry key. The data must be a null-terminated string.

To perform this action on the Windows client machine in an Application Server architecture, use DISPLAY_REG_QUERY_VALUE instead of REG_QUERY_VALUE.

Syntax:

```
CALL "REG_QUERY_VALUE" USING openKey
                             value
                             valueSize
                            [subKey]
                      GIVING returnCode
```

Parameters:

openKey	USAGE UNSIGNED-LONG	Specifies the handle to an open registry key. The calling process must have KEY_QUERY_VALUE access to the key. This handle is returned by the REG_CREATE_KEY, REG_CREATE_KEY_EX, REG_OPEN_KEY or REG_OPEN_KEY_EX library routine, or it can be one of the following predefined keys, defined in isreg.def: HKEY_CLASSES_ROOT HKEY_CURRENT_CONFIG HKEY_CURRENT_USER HKEY_LOCAL_MACHINE HKEY_USERS HKEY_DYN_DATA

value	PIC X(n)	Receives the default value of the specified key. If value is NULL, and valueSize is non-NULL, the function returns ERROR_SUCCESS, and stores the size of the data, in bytes, in valueSize. This enables an application to determine the best way to allocate a buffer for the value's data.
valueSize	USAGE UNSIGNED-LONG	Specifies the size of the value parameter, in bytes. When the function returns, this variable contains the size of the data copied to value, including any terminating null characters. If the data has the REG_SZ, REG_MULTI_SZ or REG_EXPAND_SZ type, this size includes any terminating null character or characters. If value is not large enough to hold the data, the function returns ERROR_MORE_DATA and stores the required buffer size in valueSize. In this case, the contents of value are undefined.
subKey	PIC X(n)	Specifies the name of the subkey of the openKey parameter for which the default value is retrieved. Key names are not case sensitive. If this parameter is omitted, the function retrieves the default value for the key identified by openKey.

Return code:

returnCode can be any signed numeric data item and provides additional information:

-1	Invalid or missing parameters, or not running on Windows
0	Operation successful.
Non zero	Operation failed. Click here for a list of error codes.

Examples:

Example - query a value

```
working-storage section.
copy "isreg.def".
01 subkey-handle    usage unsigned-long.
77 status-code      pic 9(3).
77 value-name       pic x(20).
77 data-size        usage unsigned-long.
...
procedure division.
...
query-value.
    set data-size to size of value-name
    move "iscobol-value"  to value-name
    call "reg_query_value" using subkey-handle
                                 value-name
                                 data-size
                           giving status-code.
```

REG_QUERY_VALUE_EX, DISPLAY_REG_QUERY_VALUE_EX

The REG_QUERY_VALUE_EX library routine retrieves the type and data for the specified value name associated with an open registry key.

To perform this action on the Windows client machine in an Application Server architecture, use DISPLAY_REG_QUERY_VALUE_EX instead of REG_QUERY_VALUE_EX.

Syntax:

```
CALL "REG_QUERY_VALUE_EX" USING openKey
                                valueName
                                type
                                data
                                dataSize
                          GIVING returnCode
```

Parameters:

openKey	USAGE UNSIGNED-LONG	Specifies the handle to an open registry key. The calling process must have KEY_QUERY_VALUE access to the key. This handle is returned by the REG_CREATE_KEY, REG_CREATE_KEY_EX, REG_OPEN_KEY or REG_OPEN_KEY_EX library routine, or it can be one of the following predefined keys, defined in isreg.def: HKEY_CLASSES_ROOT HKEY_CURRENT_CONFIG HKEY_CURRENT_USER HKEY_LOCAL_MACHINE HKEY_USERS HKEY_DYN_DATA
valueName	PIC X(n)	Specifies the name of the registry value. If valueName is NULL or an empty string, "", the function retrieves the type and data for the key's unnamed or default value, if any.
type	USAGE UNSIGNED-LONG	It receives a code indicating the type of data stored in the specified value. It can be one of the following values, defined in isreg.def:

		REG_BINARY	Binary data in any form.
		REG_DWORD	A 32-bit number.
		REG_DWORD_LITTLE_ENDIAN	A 32-bit number in little-endian format.
		REG_DWORD_BIG_ENDIAN	A 32-bit number in big-endian format.
		REG_EXPAND_SZ	A null-terminated string that contains unexpanded references to environment variables (for example, "%PATH%").
		REG_LINK	Reserved for system use.
		REG_MULTI_SZ	A sequence of null-terminated strings. The following is an example: `String1\0String2\0String3\0LastString\0\0` The first \0 terminates the first string, the second to the last \0 terminates the last string, and the final \0 terminates the sequence. Note that the final terminator must be factored into the length of the string.
		REG_NONE	No defined value type.
		REG_QWORD	A 64-bit number.
		REG_QWORD_LITTLE_ENDIAN	A 64-bit number in little-endian format.
		REG_SZ	A string.
data	PIC X(n)	Receives the data for the value entry. This parameter can be NULL if the data is not required.	

dataSize	USAGE UNSIGNED-LONG	Specifies the size of the data parameter, in bytes. When the function returns, this variable contains the size of the data copied to data. The dataSize parameter can be NULL only if data is NULL. If the data has the REG_SZ, REG_MULTI_SZ or REG_EXPAND_SZ type, this size includes any terminating null character or characters. If the data parameter is not large enough to hold the data, the function returns ERROR_MORE_DATA and stores the required buffer size in the dataSize. In this case, the contents of data are undefined. If data is NULL, and dataSize is non-NULL, the function returns ERROR_SUCCESS and stores the size of the data, in bytes, in dataSize. This enables an application to determine the best way to allocate a buffer for the value's data. If openKey specifies HKEY_PERFORMANCE_DATA and dataSize is not large enough to contain all of the returned data, REG_QUERY_VALUE_EX returns ERROR_MORE_DATA and the value returned through the dataSize parameter is undefined. This is because the size of the performance data can change from one call to the next. In this case, you must increase the buffer size and call REG_QUERY_VALUE_EX again passing the updated buffer size in the dataSize parameter. Repeat this until the function succeeds. You need to maintain a separate variable to keep track of the buffer size, because the value returned by dataSize is unpredictable.

Return code:

returnCode can be any signed numeric data item and provides additional information:

-1	Invalid or missing parameters, or not running on Windows
0	Operation successful.
Non zero	Operation failed. Click here for a list of error codes.

Examples:

Example - Query a value

```
working-storage section.
copy "isreg.def".
01 subkey-handle    usage unsigned-long.
77 status-code      pic 9(3).
77 value-name       pic x(20).
77 value-data       pic x(50).
77 data-size        usage unsigned-long.
...
procedure division.
...
query-value.
   move "iscobol-value"  to value-name
   set data-size to size of value-data
   call "reg_query_value_ex" using subkey-handle
                                   value-name
                                   data-type
                                   value-data
                                   data-size
                         giving status-code.
```

REG_SET_VALUE, DISPLAY_REG_SET_VALUE

The REG_SET_VALUE library routine sets the data for the default or unnamed value of a specified registry key. The data must be a text string.

To perform this action on the Windows client machine in an Application Server architecture, use DISPLAY_REG_SET_VALUE instead of REG_SET_VALUE.

Syntax:

```
CALL "REG_SET_VALUE" USING openKey
                           value
                          [subKey]
                    GIVING returnCode
```

Parameters:

openKey	USAGE UNSIGNED-LONG	Specifies the handle to an open registry key. The calling process must have KEY_SET_VALUE access to the key. This handle is returned by the REG_CREATE_KEY, REG_CREATE_KEY_EX, REG_OPEN_KEY or REG_OPEN_KEY_EX library routine, or it can be one of the following predefined keys, defined in iscoblib.def: HKEY_CLASSES_ROOT HKEY_CURRENT_CONFIG HKEY_CURRENT_USER HKEY_LOCAL_MACHINE HKEY_USERS HKEY_DYN_DATA

value	PIC X(n)	Specifies the value to be stored.
subKey	PIC X(n)	Specifies the name of a subkey of the openKey parameter. The function sets the default value of the specified subkey. Key names are not case sensitive. If this parameter is omitted, the function sets the default value of the key identified by openKey.

Return code:

returnCode can be any signed numeric data item and provides additional information:

-1	Invalid or missing parameters, or not running on Windows
0	Operation successful.
Non zero	Operation failed. Click here for a list of error codes.

Examples:

Example - Set a registry value

```
working-storage section.
01 subkey-handle    usage unsigned-long.
77 status-code      pic 9(3).
77 value-data       pic x(50).
...
procedure division.
...
set-key-value.
    move "iscobol-value" to value-data
    call "reg_set_value" using subkey-handle
                               value-data
                         giving status-code.
```

REG_SET_VALUE_EX, DISPLAY_REG_SET_VALUE_EX

The REG_SET_VALUE_EX library routine sets the data and type of a specified value under a registry key.

To perform this action on the Windows client machine in an Application Server architecture, use DISPLAY_REG_SET_VALUE_EX instead of REG_SET_VALUE_EX.

Syntax:

```
 CALL "REG_SET_VALUE_EX" USING openKey
                               type
                               data
                               dataSize
                              [valueName]
                         GIVING returnCode
```

Parameters:

openKey	USAGE UNSIGNED-LONG	Specifies the handle to an open registry key. The calling process must have KEY_SET_VALUE access to the key. This handle is returned by the REG_CREATE_KEY, REG_CREATE_KEY_EX, REG_OPEN_KEY or REG_OPEN_KEY_EX library routine, or it can be one of the following predefined keys, defined in isreg.def: HKEY_CLASSES_ROOT HKEY_CURRENT_CONFIG HKEY_CURRENT_USER HKEY_LOCAL_MACHINE HKEY_USERS HKEY_DYN_DATA	
type	USAGE UNSIGNED-LONG	Specifies the type of data pointed to by the data parameter. Valid values, defined in isreg.def are:	
		REG_BINARY	Binary data in any form.
		REG_DWORD	A 32-bit number.
		REG_DWORD_LITTLE_ENDIAN	A 32-bit number in little-endian format.
		REG_DWORD_BIG_ENDIAN	A 32-bit number in big-endian format.
		REG_EXPAND_SZ	A null-terminated string that contains unexpanded references to environment variables (for example, "%PATH%").
		REG_LINK	Reserved for system use.
		REG_MULTI_SZ	A sequence of null-terminated strings. The following is an example: `String1\0String2\0String3\0LastString\0\0` The first \0 terminates the first string, the second to the last \0 terminates the last string, and the final \0 terminates the sequence. Note that the final terminator must be factored into the length of the string.
		REG_NONE	No defined value type.
		REG_QWORD	A 64-bit number.
		REG_QWORD_LITTLE_ENDIAN	A 64-bit number in little-endian format.
		REG_SZ	A string.

data	PIC X(n)	Specifies the data to be stored. For string-based types, such as REG_SZ, the string must be null-terminated. With the REG_MULTI_SZ data type, the string must be terminated with two null characters.
dataSize	USAGE UNSIGNED-LONG	Specifies the size of the information pointed to by the data parameter, in bytes. If the data is of type REG_SZ, REG_EXPAND_SZ, or REG_MULTI_SZ, dataSize must include the size of the terminating null character or characters.
valueName	PIC X(n)	Specifies the name of the value to be set. If a value with this name is not already present in the key, the function adds it to the key. If valueName is omitted, the function sets the type and data for the key's unnamed or default value.

Return code:

returnCode can be any signed numeric data item and provides additional information:

-1	Invalid or missing parameters, or not running on Windows
0	Operation successful.
Non zero	Operation failed. Click here for a list of error codes.

Examples:

Example - Set a registry value

```
working-storage section.
01 subkey-handle    usage unsigned-long.
77 status-code      pic 9(3).
01 data-type        usage unsigned-long.
01 data-size        usage unsigned-long.
01 value-name       pic x(40).
01 value-data       pic x(40).
...
procedure division.
...
set-value.
   move reg_sz to data-type
   move "new-iscobol-value" to value-data
   inspect value-data replacing trailing spaces by low-value
   move 1 to data-size
   inspect value-data tallying data-size
           for characters before initial x"00"
   call "reg_set_value_ex" using subkey-handle
                           data-type
                           value-data
                           data-size
                           value-name
        giving status-code.
```

RENAME

The RENAME library routine renames a file.

Many aspects of the behavior of this routine are inherently platform-dependent: The rename operation might not be able to move a file from one filesystem to another, it might not be atomic, and it might not succeed if a file with the destination abstract pathname already exists. The return value should always be checked to make sure that the rename operation was successful.

A full path is built according to the working directory before processing the file. This full path may not be valid in c-tree environment where the c-tree server working directory doesn't match with the runtime working directory; in this case, the C$FSRENAME should be used.

Syntax:

```
CALL "RENAME" USING sourceFile
                    destFile
                    renameStatus
                   [fileType]
```

Parameters:

sourceFile	PIC X(n)	Specifies the name of the file you want to rename. If the file name starts with "isf://", the file will be searched via the File Server specified in the name. See The ISF protocol for more information.
destFile	PIC X(n)	Specifies the new file name.
renameStatus	any numeric data item	Receives the status of the operation: 0 the operation has been executed successfully. 1 the operation failed.
fileType	PIC X(1)	Specifies the file type. Valid values are: "I" File is Indexed. "R" File is Relative. "S" File is binary Sequential, the default. The default type "S" is suitable to rename generic disk files like PDFs. File type "I" is useful in cases where the original file is held in more than one physical disk file (for example, JIsam and c-tree files are physically held in two separate files). With File type "I" the file name is passed to the indexed file handler APIs so it must be suitable for them. For example, if the file handler is JIsam, avoid the "dat" extension and use a server side path.

Examples:

Example - Rename an ISAM file

```
call "rename" using "CUST", "CUST2015", renameStatus, "I"
if renameStatus not = 0
   display message "Rename failed!"
end-if
```

Example - Rename an sequential file

```
call "rename" using "cust.list" "custJan14.list" renameStatus
if renameStatus not = 0
   display message "Rename failed!"
end-if
```

S$IO

The S$IO routine provides a low-level interface to sequential files allowing them to be managed without knowing their FD.

The configuration properties iscobol.file.linesequential and iscobol.file.linesequential.FileName specify which file handler is used for line sequential files.

The configuration properties iscobol.file.sequential and iscobol.file.sequential.FileName specify which file handler is used for binary sequential files.

The following configuration properties are ignored: iscobol.file.case, iscobol.file.prefix, iscobol.file.sequential_file_prefix and iscobol.file.binary_file_prefix.

Syntax:

```
CALL "S$IO" USING opCode
                  parameters
           GIVING returnCode
```

Parameters:

opCode	Specifies the file handling function to be performed.	
	Valid values, defined in isfilesys.def are:	
	S-OPEN-FUNCTION	Opens an existing file
	S-CLOSE-FUNCTION	Closes an opened file
	S-MAKE-FUNCTION	Creates an empty file
	S-READ-FUNCTION	Reads a record from a file
	S-WRITE-FUNCTION	Writes data into file
	S-REWRITE-FUNCTION	Rewrites data into file
parameters	Parameters depend on the opcode.	

Return code:

returnCode contains useful information such as file handles and record sizes or zero if an error occurs. Check the external variable F_ERRNO for additional information on the error.

Examples:

Example - create a sequential file, write some records and then close it

```
       working-storage section.
       ...
       copy "isfilesys.def".
       77  f                       handle .
       77  file-io                 pic x(128).


       01  sio-lparms.
           03 max-rec-sz pic 9.
           03 filler     pic x value ",".
           03 file-type  signed-short.
           03 filler     pic x value ",".
           03 block-sz   pic 9 value 0.

       01  rec-buffer.
           03 rec-val    pic 9(5).
           03 filler     pic x value x"00".
       ...
       procedure division.
       ...
           set s-make-function to true
           call "s$io" using sio-function,
                             file-io,
                             sio-lparms
           if return-code = 0
              display message F_ERRNO
                      icon    mb-error-icon
                      title   "S$IO Error: make"
           end-if

      *opening
           display label line 4 title "Opening file..."
           set s-open-function to true
           move foutput to open-mode
           call "s$io" using sio-function,
                             file-io,
                             open-mode,
                             max-rec-sz,
                             seq-type,
                             0,
                             0
```

```
           if return-code > 0
              move return-code to f
           else
              display message F_ERRNO
                      icon    mb-error-icon
                      title   "S$IO Error: open"
           end-if

      *record writing
           move 0 to rec-val.
           display label line 6 title "Writing into file..."
           set s-write-function to true
           perform 5 times
              add 1 to rec-val
              call "s$io" using sio-function,
                                f,
                                rec-buffer,
                                0,
                                0
              if return-code = 0
                 display message F_ERRNO
                         icon    mb-error-icon
                         title   "S$IO Error: write"
              end-if
           end-perform

      *close file
           display label line 8 title "Closing file..."
           set s-close-function to true.
           call "s$io" using sio-function,
                             f.

           if return-code > 0
              move return-code to f
           else
              display message F_ERRNO
                      icon    mb-error-icon
                      title   "S$IO Error: close"
           end-if
...
```

S-OPEN-FUNCTION

This function opens an existing sequential file. If it is successful, the value in RETURN-CODE should be moved to a data item that is USAGE HANDLE. This data item is passed as the open file handle to the other file handling functions.

If it fails, RETURN-CODE is set to ZERO.

This function only opens already existing files. If the file does not exist, the function fails, even when opening for output.

Syntax:

```
SET S-OPEN-FUNCTION TO TRUE

CALL "S$IO" USING SIO-FUNCTION
                  name
                  mode
                  recSize
                  type
                 [blocking]
                 [padding ]
                 [pipeName]
           GIVING returnCode
```

Parameters:

name	the name of the file to open.
mode	one of the following values (defined in isfilesys.def): Finput — Open for input only. Foutput — Open for output only. Fio — Open for input and output. Fextend — Same as Foutput. mode may also have one of the following flags (defined in isfilesys.def) added to it to indicate file locking options: Fread_lock — Locks file against other updaters. Fwrite_lock — Locks file against all others. Fmass_update — Same as Fwrite_lock.
recSize	the maximum record size.
type	one of the following values (defined in isfilesys.def): S-FIXED — fixed record binary sequential file S-VAR-COUNT — variable record length binary sequential file S-LINE — line sequential file
blocking	size of a block, in bytes. This parameter is optional. If omitted or set to 0, a block is the size of a record.
padding	value of the pad character for filling short blocks. This parameter is optional. If omitted or set to 0, any short blocks are padded with 0x00.
pipeName	name of the pipe to open instead of a file. This parameter is optional.

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	File handle.
0	Operation failed.

S-CLOSE-FUNCTION

This function closes an open file. It also removes currently held locks on the file.

Syntax:

```
SET S-CLOSE-FUNCTION TO TRUE

CALL "S$IO" USING SIO-FUNCTION
                  f
           GIVING returnCode
```

Parameters:

f	the file handle returned by S-OPEN-FUNCTION

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Operation successful.
0	Operation failed.

S-MAKE-FUNCTION

This function creates a new sequential file. It overwrites any existing file of the same name unless they're in use. If the file is in use, the function fails.

Syntax:

```
SET S-MAKE-FUNCTION TO TRUE

CALL "S$IO" USING SIO-FUNCTION
                  name
                 [lParms]
          GIVING returnCode
```

Parameters:

name	the name of the file to create.

lParms	a string that describes various logical characteristics of the file. The string consists of three numeric fields separated by commas: • *Maximum record size*. This is the size in bytes of the largest record to be placed in the file. This can range from 1 to 67,108,864. • *File type*. This must be a single byte containing a binary value that indicates the type of the file. You should use the same values as described in S-OPEN-FUNCTION. • *Block size*. This is the size of a block of records. This parameter is optional. If omitted, the values are not known to the runtime.

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Operation successful.
0	Operation failed.

S-READ-FUNCTION

This function reads the next record in the sequence of records

Syntax:

```
SET S-READ-FUNCTION TO TRUE

CALL "S$IO" USING SIO-FUNCTION
                  f
                  recBuffer
            GIVING returnCode
```

Parameters:

f	the file handle returned by S-OPEN-FUNCTION.
recBuffer	an alphanumeric data-item that receives the record read.

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Number of bytes read, plus one.
0	Operation failed.

S-WRITE-FUNCTION

This function adds a new record to the sequential file.

Syntax:

```
SET S-WRITE-FUNCTION TO TRUE

CALL "S$IO" USING SIO-FUNCTION
                  f
                  recBuffer
                  lenght
            GIVING returnCode
```

Parameters:

f	the file handle returned by S-OPEN-FUNCTION.
recBuffer	an alphanumeric data-item that contains the record to write.
length	number of bytes to write. If zero, then the maximum record size for the file is used.

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Operation successful.
0	Operation failed.

S-REWRITE-FUNCTION

This function rewrites a record in the sequential file. It does not change the current file position.

Syntax:

```
SET S-REWRITE-FUNCTION TO TRUE

CALL "S$IO" USING SIO-FUNCTION
                  f
                  recBuffer
                  lenght
            GIVING returnCode
```

Parameters:

f	the file handle returned by R-OPEN-FUNCTION.
recBuffer	an alphanumeric data-item that contains the new record data.
length	number of bytes to write. It may be zero to indicate the maximum record size for the file. The size of the new record need not match the size of the existing record.

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Operation successful.
0	Operation failed.

SYSTEM

This routine allows to execute an operating system command.

The SYSTEM library routine has been implemented in isCOBOL using Java API. For this reason the implementation differs from the one available in other COBOLs, that usually reflects the system() C library function behavior. The SYSTEM library routine behavior can be configured by setting the iscobol.system.exec property.

Syntax:

```
CALL "SYSTEM" USING commandLine
             GIVING systemStatus
```

Parameters:

commandLine	PIC X(n)	Specifies the operating system command to be executed.

Return code:

systemStatus can be any signed numeric data item and receives the system exit status.

Examples:

Example - Execute an external batch file

```
move "c:\myapp\bats\bkp1.bat" to commandLine
call "system" using commandLine
```

W$BITMAP

The W$BITMAP library routine provides a number of functions to manage images. It recognizes and works with the following image formats: BMP, JPG, GIF, ICO and PNG.

With this library routine it is possible to load and unload (release memory) images, in addition to displaying them. Dealing with too many images can be inefficient. For this reason it is preferable to work with a file known as *imagelist*. This is a file containing a strip of images. The bitmap is then divided into several images with the size specified and the single image can be referenced by an index representing its position in the bitmap.

The CMYK (Cyan-Magenta-Yellow-Black four inks) color model is not supported. Bitmaps including this model may fail to be loaded, display bad or not be printed correctly. 16-bit bitmaps are also not supported.

Syntax:

```
CALL "W$BITMAP" USING opCode
                      parameters
                GIVING bitmapHandle
```

Parameters:

opCode	Function to be executed. Valid values, defined in isgui.def, are:	
	WBITMAP-DESTROY	Destroy an image and releases the memory.
	WBITMAP-DISPLAY	Load an image into memory and then displays it on the screen.
	WBITMAP-LOAD	Load an image into memory. The image can then be shown on the GUI or used in print jobs.
	WBITMAP-LOAD-FROM-CLIENT	Load an image from the client machine into memory. The image can then be shown on the GUI or used in client-side print jobs.
parameters	Parameters depend on the opcode.	

Examples:

Example - Load a bitmap from disk and display it

```
working-storage section.
copy "isgui.def".
77  h-bitmap                pic s9(9) comp-4.

screen section.
01  Mask.
    03 Bmp1
       bitmap line 2 col 2
       lines 10 cells size 21 cells
       .
    03 Ef1
       entry-field read-only
       line 13 col 2 size 15 cells
       .


procedure division.
main.
   call "w$bitmap" using wbitmap-load, "images/img.png"
                                giving h-bitmap
   if h-bitmap < 0
      display message "W$BITMAP Error: " h-bitmap
   else
      modify Bmp1 bitmap-handle h-bitmap
   end-if.
   ...
```

WBITMAP-DESTROY

The WBITMAP-DESTROY function destroys an image previously loaded is destroyed and releases its associated memory. If the image was displayed with the WBITMAP-DISPLAY function, it is removed from the screen and he space previously occupied by the bitmap is filled with the window's background color.

Note: Using the DESTROY statement instead of this op-code will not remove the handle from memory.

Syntax:

```
CALL "W$BITMAP" USING WBITMAP-DESTROY
                      bitmapHandle
```

Parameters:

WBITMAP-DESTROY	Constant	
bitmapHandle	USAGE HANDLE	It specifies a valid handle returned by the WBITMAP-DISPLAY or the WBITMAP-LOAD function or the W$CAPTURE, the W$ROTATE or the W$SCALE library routines.

WBITMAP-DISPLAY

The WBITMAP-DISPLAY function loads an image into memory and then displays it on the screen. The position of the image is expressed in cells. It is not possible to define a custom size or a "zoom level" for the image. By default, the screen cells occupied by the image are filled with the window's background color, unless the WBITMAP-NO-FILL flag is specified.

The bitmap shown by this op-code is not erased if it’s overlapped by another control.

Syntax:

```
CALL "W$BITMAP" USING WBITMAP-DISPLAY
                      name
                      row
                      column
                      flags
                GIVING bitmapHandle
```

Parameters:

WBITMAP-DISPLAY	Constant	
name	PIC X(n)	Specifies the name of a regular disk file containing the image to be displayed.
row	any numeric data item or numeric literal	Specifies the row, measured in window's cells, where the upper-left image corner is positioned. Decimal values are allowed.
column	any numeric data item or numeric literal	Specifies the column, measured in window's cells, where the upper-left image corner is positioned. Decimal values are allowed.
flags	any numeric data item or numeric literal	Its value affects the way the image is displayed. At present it is possible to assign to it the following symbolic value only:
		WBITMAP-NO-FILL — The screen cells occupied by the image are not filled with the window's background color.

Return code:

bitmapHandle must be declared as PIC S9(9) COMP-4 and provides additional information:

> 0	Receives the handle to the bitmap
-1	Error: file not found or not readable
-2	Error: out of memory loading the bitmap
-3	Error: not a valid bitmap
-4	Error: format not supported

WBITMAP-LOAD

The WBITMAP-LOAD function loads an image into memory.

Syntax:

```
CALL "W$BITMAP" USING WBITMAP-LOAD
                      name
                GIVING bitmapHandle
```

Parameters:

WBITMAP-LOAD	Constant	
name	PIC X(n)	Specifies the name of a regular disk file containing the image to be loaded.

Return code:

bitmapHandle must be declared as PIC S9(9) COMP-4 and provides additional information:

> 0	Receives the handle to the bitmap
-1	Error: file not found or not readable
-2	Error: out of memory loading the bitmap
-3	Error: not a valid bitmap
-4	Error: format not supported

WBITMAP-LOAD-FROM-CLIENT

The WBITMAP-LOAD-FROM-CLIENT function loads an image into memory. In a thin client environment, the bitmap is searched on the client side. In the other environments, calling this function is the same as calling WBITMAP-LOAD.

The loaded bitmap can be used on the GUI and on client-side print jobs, but not on server-side print jobs (see WINPRINT-SET-PRINTER-AS for more information about client-side and server-side print jobs).

Syntax:

```
CALL "W$BITMAP" USING WBITMAP-LOAD-FROM-CLIENT
                      name
                GIVING bitmapHandle
```

Parameters:

WBITMAP-LOAD-FROM-CLIENT	Constant	
name	PIC X(n)	Specifies the name of a regular disk file containing the image to be loaded.

Return code:

bitmapHandle must be declared as PIC S9(9) COMP-4 and provides additional information:

> 0	Receives the handle to the bitmap
-1	Error: file not found or not readable
-2	Error: out of memory loading the bitmap
-3	Error: not a valid bitmap
-4	Error: format not supported

W$CAPTURE

The W$CAPTURE library routine allows to take a screenshot of a window or the whole screen. Different actions are performed depending on the number of parameters.

The routine creates a new image resource and returns its handle.

Note - The newly created resource is not released automatically. It must be destroyed by the programmer with the WBITMAP-DESTROY function when it is no longer needed.

Syntax:

```
CALL "W$CAPTURE"  USING [windowHandle ]
                        [fileName ]
                        [fileFormat ]
                  GIVING  bitmapHandle
```

Parameters:

windowHandle	USAGE HANDLE	Specifies the handle of the window to be captured. The window decorations are included in the screenshot. When this parameter is omitted or zero, the whole screen is captured.
fileName	PIC X(n) or alphanumeric literal	Specifies the name of the file where the image has to be stored. When this parameter is omitted, the image is just loaded into memory and it is not stored on disk. In thin client environment, the file is created on the client disk.
fileFormat	PIC X(n) or alphanumeric literal	Specifies the format used to store the image. Possible values are: "bmp" Windows and OS/2 bitmap format "gif" CompuServe Graphics Interchange Format "png" Portable Network Graphics format "jpg" JPEG (Joint Photographic Experts Group) format

Return code:

bitmapHandle must be an USAGE HANDLE data item and receives the handle of the captured image. A value

of 0 means that the routine is not supported in the current runtime environment.

Examples:

Example - Capture full desktop and save to a .png file

```
call "w$capture" using 0, "c:\tmp\screenshot.png", "png"

Example - Capture main program screen and save to a .jpg file

working-storage section.
copy "isgui.def".
copy "iscrt.def".
77  crt-status              special-names crt status pic 9(5).
77  hWin                    handle of window.
77  hBmp                    pic s9(9) comp-4.


screen section.
01  main-screen.
    03 push-button
       line 2 col 3 size 15 cells
       title "Capture Window"
       exception-value 102
       .
    03 push-button
       line 6 col 3
       title "Exit"
       exception-value 27
       .

procedure division.
main.
  display independent graphical window
          color  65793
          with   system menu
          title  "W$CAPTURE Routine"
          handle hWin
          event  WIN-EVT

  display main-screen

  accept main-screen on exception continue
  end-accept

  if crt-status = 102
       call "w$capture" using hWin "c:\tmp\mainscr.jpg", "jpg"
            giving hBmp
  end-if.
```

W$CENTER_WINDOW

The W$CENTER_WINDOW library routine moves the window on the screen in order to have the center of the window matching with the center of the screen.

Syntax:

```
CALL "W$CENTER_WINDOW" USING windowHandle, [screenIndex]
```

Parameters:

windowHandle	HANDLE OF WINDOW	Specifies the window to be centered.
screenIndex	PIC 9(n)	Optional parameter. Specifies in which monitor the window should be centered in case that multiple monitors are available. In a multi-monitor environment, the window would be centered in the monitor where it was displayed. Set this parameter to move the window in another monitor and center it there. Note - the SCREEN-INDEX property of the window will be altered according to this parameter.

Examples:

Example - Display a independent window and center it on the screen

```
working-storage section.
77  hWin                  handle of window.

procedure division.
main.
  display independent graphical window
          color  65793
          with   system menu
          title  "Centered window"
          handle hWin
          event  WIN-EVT
  call "w$center_window" using hWin.
```

W$CREATEFONT

The W$CREATEFONT library routine loads a font into memory even if it is not installed on the host system.

Note - Fonts used on video by controls titles and more generally by controls where user input is not possible (such as grid headings) require the font to be installed in the system as their text is rendered through HTML by the Framework.

Syntax:

```
CALL "W$CREATEFONT" USING fontName
                          fontLogicalName
                    GIVING returnCode
```

Parameters:

fontName	PIC X(n)	Specifies the font name followed by the suffix ".ttf".
fontLogicalName	PIC X(n)	Logical name used with the W$FONT and WFONT-GET-FONT functions.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation failed.
1	Operation successful.

Examples:

Example - Enable a font from its disk .ttf file

```
working-storage section.
copy "isfonts.def".
77  wfont-status            pic s99.
77  h-font                  handle of font.

procedure division.
main.
  call "w$createfont" using "files/fsex2p00_public.ttf"
                            "Fixedsys Excelsior 2.00"
                      giving wfont-status
  if wfont-status = 0
     display message "w$createfont error: " wfont-status
     goback
  end-if.
  initialize wfont-data
  move "Fixedsys Excelsior 2.00" to wfont-name
  move "11"                      to wfont-size
  call "w$font" using wfont-get-font, h-font, wfont-data
                giving wfont-status
  if wfont-status < 0
     display message "w$font error: " wfont-status
     goback
  end-if.
  ...
```

W$FLUSH

The W$FLUSH library routine allows to control the client/server traffic generated by the communication between the user interface (client) and the back end logic (server).

Syntax:

```
CALL "W$FLUSH" USING opCode
                     parameters
```

Parameters:

opCode	Function to be executed. The only valid value, defined in iscobol.def, is:	
	WFLUSH-REFRESH	refresh the screen
	WFLUSH-INHIBIT	start bufferization of INQUIRE statements
	WFLUSH-ALLOW	stop bufferization of INQUIRE statements
	WFLUSH-DISABLE-UI	start ignoring all operations on the UI
	WFLUSH-ENABLE-UI	stop ignoring all operations on the UI
parameters	Parameters depend on the opcode.	

Examples:

Example - Refresh all windows on thin-client session

```
*> On working-storage copy "iscobol.def"

call "w$flush" using wflush-refresh
```

WFLUSH-REFRESH

The WFLUSH-REFRESH op-code causes the screen to be refreshed. This is useful if you want to update the video before the client/server timeout (iscobol.gui.cstimeout *) expires or the client/sever maximum buffer size (iscobol.gui.csmaxbuffersize *) is reached.

Syntax:

```
CALL "W$FLUSH" USING WFLUSH-REFRESH
                     [windowHandle]
```

Parameters:

WFLUSH-REFRESH	Constant	
windowHandle	HANDLE OF WINDOW	Optional. Specifies the window you whish to refresh. If omitted, then all windows are refreshed.

WFLUSH-INHIBIT

The WFLUSH-INHIBIT op-code starts the bufferization of INQUIRE statements to reduce client/server traffic. Normally every INQUIRE causes client/server traffic. By bufferizing them, and send more of them at once, you can reduce the traffic.

Bufferized INQUIRE statements are sent when WFLUSH-ALLOW is called or when one of the other conditions listed in UI changes bufferization occurs. The destination variables of your inquires should be checked only after that moment.

It's important to use a separate variable for each INQUIRE (e.g. use occurs data items). If you use the same variable for multiple inquires, such variable will be set to the result of the last INQUIRE overwriting values set by previous inquires.

Syntax:

```
CALL "W$FLUSH" USING WFLUSH-INHIBIT
                     [windowHandle]
```

Parameters:

WFLUSH-INHIBIT	Constant	
windowHandle	HANDLE OF WINDOW	Optional. Specifies the window where you wish to gather INQUIRE statements. If omitted, all INQUIRE statements are bufferized regardless of the window where they are performed.

WFLUSH-ALLOW

The WFLUSH-ALLOW op-code stops the bufferization of INQUIRE statements, if this bufferization is active, and then sends all the bufferized data to the client.

Syntax:

```
CALL "W$FLUSH" USING WFLUSH-ALLOW
                     [windowHandle]
```

Parameters:

WFLUSH-ALLOW	Constant	
windowHandle	HANDLE OF WINDOW	Optional. Specifies the window where you wish to stop the INQUIRE statements bufferization and send the bufferized data.

WFLUSH-DISABLE-UI

The WFLUSH-DISABLE-UI op-code makes the runtime ignoring all statements that affect the User Interface. Every DESTROY, DISPLAY, MODIFY and INQUIRE performed after this call will be skipped. This kind of operation is useful to speed up performance during elaborations where there is no need to update the User Interface and you don't want to review the program logic to manually remove the useless UI statements.

Use WFLUSH-ENABLE-UI to restore the management of UI statements.

Syntax:

```
CALL "W$FLUSH" USING WFLUSH-DISABLE-UI
                     [windowHandle]
```

Parameters:

WFLUSH-DISABLE-UI	Constant	
windowHandle	HANDLE OF WINDOW	Optional. Specifies the window where you wish to disable the UI management

WFLUSH-ENABLE-UI

The WFLUSH-ENABLE-UI op-code restores the standard management of User Interface updates. Every DESTROY, DISPLAY, MODIFY and INQUIRE performed after this call will be managed by the runtime.

Syntax:

```
CALL "W$FLUSH" USING WFLUSH-ENABLE-UI
                     [windowHandle]
```

Parameters:

WFLUSH-ENABLE-UI	Constant	
windowHandle	HANDLE OF WINDOW	Optional. Specifies the window where you wish to enable the UI management.

W$FONT

The W$FONT library routine provides a number of functions to manage fonts.

Note: isCOBOL can handle only TrueType fonts.

Syntax:

```
CALL "W$FONT" USING opCode
                    parameters
              GIVING returnCode
```

Parameters:

opCode	Function to be executed. Valid values, defined in isfonts.def, are:	
	WFONT-CHOOSE-FONT	Show a dialog window to choose a font
	WFONT-DESCRIBE-FONT	Retrieve font characteristics
	WFONT-GET-CLOSEST-FONT	Load the font matching some characteristics
	WFONT-GET-FONT	Load a specific font
	WFONT-SUPPORTED	Check if the host system supports the W$FONT library routine
parameters	Parameters depend on the opcode.	

Return code:

returnCode definition and meaning depend on the opcode.

Examples:

Example - Load the Arial font to be used on video

```
working-storage section.
copy "isfonts.def".
77  wfont-status           pic s99.
77  h-font                 handle of font.

procedure division.
main.
  initialize wfont-data
  move "Arial" to wfont-name
  move "11"    to wfont-size
  call "w$font" using wfont-get-font, h-font, wfont-data
       giving wfont-status
  if wfont-status < 0
     display message "w$font error: " wfont-status
     goback
  end-if.
  ...
```

Example - Load the Arial font to be used for printing

```
working-storage section.
copy "isfonts.def".
77  wfont-status            pic s99.
77  h-font                  handle of font.

procedure division.
main.
  initialize wfont-data
  move "Arial"             to wfont-name
  move "11"                to wfont-size
  set wfdevice-win-printer to true
  call "w$font" using wfont-get-font, h-font, wfont-data
       giving wfont-status
  if wfont-status < 0
     display message "w$font error: " wfont-status
     goback
  end-if.
  ...
```

WFONT-CHOOSE-FONT

The WFONT-CHOOSE-FONT function shows a dialog window to choose a font.

Syntax:

```
CALL "W$FONT" USING WFONT-CHOOSE-FONT
                    fontHandle
                    WFONT-DATA
              GIVING returnCode
```

Parameters:

WFONT-CHOOSE-FONT	Constant	
fontHandle	USAGE HANDLE OF FONT	Not used. It should be 0.

WFONT-DATA
Group Item
	Structure that describes the initial configuration of the dialog box and receives the user's selection when the dialog box is closed. Note that the dialog box is initialized with the information stored in wfont-face-data only when wfont-choose-flags contains the wfchoose-initialize flag. This group item, defined in isfonts.def, has the following structure:
	
	`01 wfont-data.`
	`  03 wfont-face-data.`
	`  05 wfont-device handle, value null.`
	`  88 wfdevice-console value null.`
	`  88 wfdevice-printer value 1.`
	`  05 wfont-name pic x(33).`
	`  05 wfont-char-set pic x comp-x.`
	`  05 wfont-size pic x comp-x.`
	`  05 wfont-bold-state pic x comp-x.`
	`  88 wfont-bold value 1, false zero.`
	`  05 wfont-italic-state pic x comp-x.`
	`  88 wfont-italic value 1, false zero.`
	`  05 wfont-underline-state pic x comp-x.`
	`  88 wfont-underline value 1, false zero.`
	`  05 wfont-strikeout-state pic x comp-x.`
	`  88 wfont-strikeout value 1, false zero.`
	`  05 wfont-pitch-state pic x comp-x.`
	`  88 wfont-fixed-pitch value 1, false zero.`
	`  05 wfont-family pic x comp-x.`
	`  03 wfont-choose-data.`
	`  05 wfont-choose-flags pic x comp-x.`
	`  05 wfont-choose-min-size pic x comp-x.`
	`  05 wfont-choose-max-size pic x comp-x.`
	`  05 wfont-choose-red pic x comp-x.`
	`  05 wfont-choose-green pic x comp-x.`
	`  05 wfont-choose-blue pic x comp-x.`
	`  05 wfont-choose-color-num pic x comp-x.`
	`  03 wfont-angle pic x(2) comp-x.`
	`  03 wfont-scale-x float value 0.`
	`  03 wfont-scale-y float value 0.`
	
	Note - members not mentioned below are not used by this function.
	
	wfont-device
	It is the device the function refers to. The value of this data item should be set with the condition names provided.
	When *wfdevice-console* is set to true, the function will load a font for the current screen configuration.
	When *wfdevice-printer* is set to true, the function will load a font for the current printer configuration.
	
	wfont-name
	When the function starts and wfont-choose-flags contains the wfchoose-initialize flag, the value set here is used as default font name.
	When the function exits, it receives the name of the font chosen by the user.

wfont-size
When the function starts and wfont-choose-flags contains the wfchoose-initialize flag, the value set here is used as default font size.
When the function exits, it receives the size of the font chosen by the user.

wfont-bold-state
When the function starts and wfont-choose-flags contains the wfchoose-initialize flag, the value set here affects the initial value of the check-box "Bold".
When the function exits, it receives the value of the check-box "Bold".

wfont-italic-state
When the function starts and wfont-choose-flags contains the wfchoose-initialize flag, the value set here affects the initial value of the check-box "Italic".
When the function exits, it receives the value of the check-box "Italic".

wfont-underline-state
When the function starts and wfont-choose-flags contains the wfchoose-initialize flag, the value set here affects the initial value of the check-box "Underline".
When the function exits, it receives the value of the check-box "Underline".

wfont-strikeout-state
When the function starts and wfont-choose-flags contains the wfchoose-initialize flag, the value set here affects the initial value of the check-box "Strikethrough".
When the function exits, it receives the value of the check-box "Strikethrough".

wfont-pitch-state
When the function, the value is ignored.
When the function exits, and the font is a fixed-pitch font, wfont-fixed-pitch is set to true. If the font is a variable-pitch font, wfont-fixed-pitch is set to false.

wfont-choose-data - wfont-choose-flags
Flags affecting the behavior of the dialog box. Possible values, that can be added together, are:

wfchoose-fixed-only	Only fixed-pitch fonts are listed, therefore the user cannot choose a variable-pitch font.
wfchoose-initialize	The dialog box is initialized according to the values set in wfont-face-data.
wfchoose-effects-ok	Additional check-boxes are shown to allow the selection of special font effects, such as underline and strike-out.

wfont-choose-min-size
It is the minimum allowed size for the font.

wfont-choose-max-size
It is the maximum allowed size for the font.

	wfont-angle It is the angle at which the font will print. The value can range from the default of "0", which is the normal horizontal orientation, to "360", which is the same as "0". For example, to print at a 90-degree angle, set WFONT-ANGLE to "90". This feature works only when printing a font, not when displaying a font on screen. *wfont-scale-x* It is the scale factor on the X coordinate. A value of 0 or 1 means that no scale is performed. This setting should be used only for print fonts (*wfont-device* = *wfdevice-printer*) that are not rotated (*wfont-angle* = 0), otherwise the effects are unpredictable. *wfont-scale-y* It is the scale factor on the Y coordinate. A value of 0 or 1 means that no scale is performed. This setting should be used only for print fonts (*wfont-device* = *wfdevice-printer*) that are not rotated (*wfont-angle* = 0), otherwise the effects are unpredictable

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WFONTERR-UNSUPPORTED	The W$FONT library routine is not supported.
WFONTERR-CANCELLED	The user has pressed the Cancel button. wfontData is not updated.

WFONT-DESCRIBE-FONT

The WFONT-DESCRIBE-FONT function describes the characteristics of a loaded font.

Syntax:

```
CALL "W$FONT" USING WFONT-DESCRIBE-FONT
                    fontHandle
                    WFONT-DATA
              GIVING returnCode
```

Parameters:

WFONT-DESCRIBE-FONT	Constant	
fontHandle	USAGE HANDLE OF FONT	It may be set to a valid font handle.

WFONT-DATA	Group Item	Structure that receives the characteristics of the font. This group item, defined in isfonts.def, has the following structure:
		
		`01 wfont-data.`
		`  03 wfont-face-data.`
		`  05 wfont-device  handle, value null.`
		`  88 wfdevice-console  value null.`
		`  88 wfdevice-printer  value 1.`
		`  05 wfont-name  pic x(33).`
		`  05 wfont-char-set  pic x comp-x.`
		`  05 wfont-size  pic x comp-x.`
		`  05 wfont-bold-state  pic x comp-x.`
		`  88 wfont-bold  value 1, false zero.`
		`  05 wfont-italic-state  pic x comp-x.`
		`  88 wfont-italic  value 1, false zero.`
		`  05 wfont-underline-state  pic x comp-x.`
		`  88 wfont-underline  value 1, false zero.`
		`  05 wfont-strikeout-state  pic x comp-x.`
		`  88 wfont-strikeout  value 1, false zero.`
		`  05 wfont-pitch-state  pic x comp-x.`
		`  88 wfont-fixed-pitch  value 1, false zero.`
		`  05 wfont-family  pic x comp-x.`
		`  03 wfont-choose-data.`
		`  05 wfont-choose-flags  pic x comp-x.`
		`  05 wfont-choose-min-size  pic x comp-x.`
		`  05 wfont-choose-max-size  pic x comp-x.`
		`  05 wfont-choose-red  pic x comp-x.`
		`  05 wfont-choose-green  pic x comp-x.`
		`  05 wfont-choose-blue  pic x comp-x.`
		`  05 wfont-choose-color-num  pic x comp-x.`
		`  03 wfont-angle  pic x(2) comp-x.`
		`  03 wfont-scale-x  float value 0.`
		`  03 wfont-scale-y  float value 0.`
		
		Note - members not mentioned below are not used by this function.
		
		wfont-device
		It is the device the function refers to. The value can be one of the condition names provided.
		When *wfdevice-console* is set to true, the function will describe a font for the current screen configuration.
		When *wfdevice-printer* is set to true, the function will describe a font for the current printer configuration.
		
		wfont-name
		It is the name of the font.
		
		wfont-size
		It is the size of the font.

	wfont-bold-state It is the bold state of the font. If the font is bold, *wfont-bold* is set to true. *wfont-italic-state* It is the italic state of the font. If the font is italic, *wfont-italic* is set to true. *wfont-underline-state* It is the underline state of the font. If the font is underline, *wfont-underline* is set to true. *wfont-strikeout-state* It is the strike-out state of the font. If the font is stricken out, *wfont-strikeout* is set to true. *wfont-pitch-state* It is the pitch state of the font. If the font is a fixed-pitch font, *wfont-fixed-pitch* is set to true. If the font is a variable-pitch font, *wfont-fixed-pitch* is set to false. *wfont-angle* It is the angle at which the font will print. The value can range from the default of "0", which is the normal horizontal orientation, to "360", which is the same as "0". For example, to print at a 90-degree angle, set WFONT-ANGLE to "90". This feature works only when printing a font, not when displaying a font on screen. *wfont-scale-x* It is the scale factor on the X coordinate. A value of 0 or 1 means that no scale is performed. This setting should be used only for print fonts (*wfont-device* = *wfdevice-printer*) that are not rotated (*wfont-angle* = 0), otherwise the effects are unpredictable. *wfont-scale-y* It is the scale factor on the Y coordinate. A value of 0 or 1 means that no scale is performed. This setting should be used only for print fonts (*wfont-device* = *wfdevice-printer*) that are not rotated (*wfont-angle* = 0), otherwise the effects are unpredictable

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WFONTERR-UNSUPPORTED	The W$FONT library routine is not supported.
WFONTERR-INVALID-HANDLE	fontHandle is not valid.

WFONT-GET-CLOSEST-FONT

The WFONT-GET-CLOSEST-FONT function loads the font matching some characteristics. The function will always load a font, trying to meet the program requirements.

In thin client environment, if the font exists on the server and the server has graphical interface, then the font is loaded both client and server side. This allows to use the font also for PDF print jobs created server-side (see WINPRINT-SET-PRINTER-AS for details).

Syntax:

```
CALL "W$FONT" USING WFONT-GET-CLOSEST-FONT
                    fontHandle
                    WFONT-DATA
              GIVING returnCode
```

Parameters:

WFONT-GET-CLOSEST-FONT	Constant	
fontHandle	USAGE HANDLE OF FONT	When the function succeeds, it receives the handle of the font.

WFONT-DATA	Group Item	Structure that contains the characteristics of the font to be loaded. All data items it contains should be set properly before calling the function. For your convenience you may INITIALIZE the group item, that way, all subordinate items will be set to a valid value. This group item, defined in isfonts.def, has the following structure:
		
		`01 wfont-data.`
		`  03 wfont-face-data.`
		`  05 wfont-device handle, value null.`
		`  88 wfdevice-console value null.`
		`  88 wfdevice-printer value 1.`
		`  05 wfont-name pic x(33).`
		`  05 wfont-char-set pic x comp-x.`
		`  05 wfont-size pic x comp-x.`
		`  05 wfont-bold-state pic x comp-x.`
		`  88 wfont-bold value 1, false zero.`
		`  05 wfont-italic-state pic x comp-x.`
		`  88 wfont-italic value 1, false zero.`
		`  05 wfont-underline-state pic x comp-x.`
		`  88 wfont-underline value 1, false zero.`
		`  05 wfont-strikeout-state pic x comp-x.`
		`  88 wfont-strikeout value 1, false zero.`
		`  05 wfont-pitch-state pic x comp-x.`
		`  88 wfont-fixed-pitch value 1, false zero.`
		`  05 wfont-family pic x comp-x.`
		`  03 wfont-choose-data.`
		`  05 wfont-choose-flags pic x comp-x.`
		`  05 wfont-choose-min-size pic x comp-x.`
		`  05 wfont-choose-max-size pic x comp-x.`
		`  05 wfont-choose-red pic x comp-x.`
		`  05 wfont-choose-green pic x comp-x.`
		`  05 wfont-choose-blue pic x comp-x.`
		`  05 wfont-choose-color-num pic x comp-x.`
		`  03 wfont-angle pic x(2) comp-x.`
		`  03 wfont-scale-x float value 0.`
		`  03 wfont-scale-y float value 0.`
		
		Note - members not mentioned below are not used by this function.
		
		wfont-device
		It is the device the function refers to. The value of this data item should be set with the condition names provided.
		When *wfdevice-console* is set to true, the function will load a font for the current screen configuration.
		When *wfdevice-printer* is set to true, the function will load a font for the current printer configuration.
		
		wfont-name
		It is the name of the font that will be loaded. Font names are case-sensitive. If a font with the name here specified is not found, the function loads a font matching the remaining characteristics.
		
		wfont-size
		It is the size of the font that will be loaded. It may be greater than zero.

	wfont-bold-state It is the bold state of the font that will be loaded. The value of this data item should be set with the condition name provided. When *wfont-bold* is set to true, the font will be bold. *wfont-italic-state* It is the italic state of the font that will be loaded. The value of this data item should be set with the condition name provided. When *wfont-italic* is set to true, the font will be italic. *wfont-underline-state* It is the underline state of the font that will be loaded. The value of this data item should be set with the condition name provided. When *wfont-underline* is set to true, the font will be underlined. *wfont-strikeout-state* It is the strike-out state of the font that will be loaded. The value of this data item should be set with the condition name provided. When *wfont-strikeout i*s set to true, the font will be stricken out. *wfont-pitch-state* It is the pitch state of the font that will be loaded. The value of this data item should be set with the condition name provided. When *wfont-fixed-pitch* is set to true, the font will be loaded only if it is a fixed-pitch font. When *wfont-fixed-pitch* is set to false, the function will load fonts with both fixed or variable pitch. *wfont-angle* It is the angle at which the font will print. The value can range from the default of "0", which is the normal horizontal orientation, to "360", which is the same as "0". For example, to print at a 90-degree angle, set WFONT-ANGLE to "90". This feature works only when printing a font, not when displaying a font on screen. *wfont-scale-x* It is the scale factor on the X coordinate. A value of 0 or 1 means that no scale is performed. This setting should be used only for print fonts (*wfont-device* = *wfdevice-printer*) that are not rotated (*wfont-angle* = 0), otherwise the effects are unpredictable. *wfont-scale-y* It is the scale factor on the Y coordinate. A value of 0 or 1 means that no scale is performed. This setting should be used only for print fonts (*wfont-device* = *wfdevice-printer*) that are not rotated (*wfont-angle* = 0), otherwise the effects are unpredictable

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WFONTERR-UNSUPPORTED	The W$FONT library routine is not supported.

WFONT-GET-FONT

The WFONT-GET-FONT function loads a specific font. The font is loaded only if it matches exactly the characteristics requested.

In thin client environment, if the font exists on the server and the server has graphical interface, then the font is loaded both client and server side. This allows to use the font also for PDF print jobs created server-side (see WINPRINT-SET-PRINTER-AS for details).

Syntax:

```
CALL "W$FONT" USING WFONT-GET-FONT
                    fontHandle
                    WFONT-DATA
              GIVING returnCode
```

Parameters:

WFONT-GET-FONT	Constant	
fontHandle	USAGE HANDLE OF FONT	When the function succeeds, it receives the handle of the font.
WFONT-DATA	Group Item	Structure that contains the characteristics of the font to be loaded. All data items it contains should be set properly before calling the function. For your convenience you may INITIALIZE the group item, that way, all subordinate items will be set to a valid value. This group item, defined in isfonts.def, has the following structure:

```
01  wfont-data.
    03 wfont-face-data.
       05 wfont-device                handle, value null.
          88 wfdevice-console         value null.
          88 wfdevice-printer         value 1.
       05 wfont-name                  pic x(33).
       05 wfont-char-set              pic x comp-x.
       05 wfont-size                  pic x comp-x.
       05 wfont-bold-state            pic x comp-x.
          88 wfont-bold               value 1, false zero.
       05 wfont-italic-state          pic x comp-x.
          88 wfont-italic             value 1, false zero.
       05 wfont-underline-state       pic x comp-x.
          88 wfont-underline          value 1, false zero.
       05 wfont-strikeout-state       pic x comp-x.
          88 wfont-strikeout          value 1, false zero.
       05 wfont-pitch-state           pic x comp-x.
          88 wfont-fixed-pitch        value 1, false zero.
       05 wfont-family                pic x comp-x.
    03 wfont-choose-data.
       05 wfont-choose-flags          pic x comp-x.
       05 wfont-choose-min-size       pic x comp-x.
       05 wfont-choose-max-size       pic x comp-x.
       05 wfont-choose-red            pic x comp-x.
       05 wfont-choose-green          pic x comp-x.
       05 wfont-choose-blue           pic x comp-x.
       05 wfont-choose-color-num      pic x comp-x.
    03 wfont-angle                    pic x(2) comp-x.
    03 wfont-scale-x                  float value 0.
    03 wfont-scale-y                  float value 0.
```

Note - members not mentioned below are not used by this function.

wfont-device
It is the device the function refers to. The value of this data item should be set with the condition names provided.
When *wfdevice-console* is set to true, the function will load a font for the current screen configuration.
When *wfdevice-printer* is set to true, the function will load a font for the current printer configuration.

wfont-name
It is the name of the font that will be loaded. It may not be spaces. Font names are case-sensitive.

wfont-size
It is the size of the font that will be loaded. It may be greater than zero.

wfont-bold-state
It is the bold state of the font that will be loaded. The value of this data item should be set with the condition name provided.
When *wfont-bold* is set to true, the font will be bold.

wfont-italic-state
It is the italic state of the font that will be loaded. The value of this data item should be set with the condition name provided.
When *wfont-italic* is set to true, the font will be italic.

wfont-underline-state
It is the underline state of the font that will be loaded. The value of this data item should be set with the condition name provided.
When *wfont-underline* is set to true, the font will be underlined.

wfont-strikeout-state
It is the strike-out state of the font that will be loaded. The value of this data item should be set with the condition name provided.
When *wfont-strikeout* is set to true, the font will be stricken out.

wfont-pitch-state
It is the pitch state of the font that will be loaded. The value of this data item should be set with the condition name provided.
When wfont-fixed-pitch is set to true, the font will be loaded only if it is a fixed-pitch font.
When *wfont-fixed-pitch* is set to false, the function will load fonts with both fixed or variable pitch.

	wfont-angle It is the angle at which the font will print. The value can range from the default of "0", which is the normal horizontal orientation, to "360", which is the same as "0". For example, to print at a 90-degree angle, set WFONT-ANGLE to "90". This feature works only when printing a font, not when displaying a font on screen. *wfont-scale-x* It is the scale factor on the X coordinate. A value of 0 or 1 means that no scale is performed. This setting should be used only for print fonts (*wfont-device* = *wfdevice-printer*) that are not rotated (*wfont-angle* = 0), otherwise the effects are unpredictable. *wfont-scale-y* It is the scale factor on the Y coordinate. A value of 0 or 1 means that no scale is performed. This setting should be used only for print fonts (*wfont-device* = *wfdevice-printer*) that are not rotated (*wfont-angle* = 0), otherwise the effects are unpredictable.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WFONTERR-UNSUPPORTED	The W$FONT library routine is not supported.
WFONTERR-FONT-NOT-FOUND	The W$FONT library routine was not able to find the font. fontHandle is set to 0.

WFONT-SUPPORTED

The WFONT-SUPPORTED function checks if the host system supports the W$FONT library routine.

Syntax:

```
CALL "W$FONT" USING WFONT-SUPPORTED
              GIVING returnCode
```

Parameters:

WFONT-SUPPORTED	Constant

Return code:

returnCode can be any signed numeric data item and provides additional information:

WFONTERR-UNSUPPORTED	The W$FONT library routine is not supported.
WFONT-FONT-SUPPORT	The W$FONT library routine is supported, but the WFONT-CHOOSE-FONT is not and may not be used.
WFONT-FULL-SUPPORT	The W$FONT library routine is fully supported.

NOTE - Currently, the WFONT-SUPPORTED function always returns WFONT-FULL-SUPPORT because the

W$FONT library routine is fully supported on all platforms

W$HINT

The W$HINT library routine allows to show hints programmatically.

Syntax:

```
CALL "W$HINT" USING hintText
                    x
                    y
                   [timeout]
```

Parameters:

hintText	PIC X(n)	Specifies the text to be shown in the hint box. It can include HTML tags for text formatting.
x	any numeric data item	Specifies the x coordinate where the hint box must be shown. The value is expressed in cells.
y	any numeric data item	Specifies the y coordinate where the hint box must be shown. The value is expressed in cells.
timeout	any numeric data item	Specifies how many hundreds of seconds the hint box must stay on video. If passed, this parameter overrides the iscobol.gui.hints_on and iscobol.gui.hints_off settings. The user can close an hint before the timeout expires by pressing one of these keys: • F1 to F24, • PAGE-UP or PAGE-DOWN • arrow keys • HOME • END • ENTER • ESCAPE • TAB

Examples:

Example - Show a notification hint at line 2 column 2 and make it stay on video for 2 seconds

```
procedure division.
main.
  call "w$hint" using "Download completed", 2, 2, 200.
```

W$IMAGESIZE

The W$IMAGESIZE library routine allows to retrieve the width and the height in pixels of an image resource.

Syntax:

```
CALL "W$IMAGESIZE" USING bitmapHandle
                         imageWidth
                         imageHeight
```

Parameters:

BitmapHandle	USAGE HANDLE	Specifies the handle of a valid image resource.
ImageWidth	any numeric data item	Receives the width in pixels of the image.
ImageHeight	any numeric data item	Receives the height in pixels of the image.

Examples:

Example - Get the dimensions of an image

```
working-storage section.
copy "isgui.def".
77  hBmp pic s9(9) comp-4.
77  w-width    pic 9(5)v99.
77  w-height   pic 9(5)v99.
77  f-image-name pic x(80).


procedure division.
main.
  move "c:\tmp\img1.jpg" to f-image-name
  call "w$bitmap" using wbitmap-load, f-image-name
       giving hBmp
  call "w$imagesize" using hBmp, w-width, w-height

  display message "File : " f-image-name x"0d0a"
                  "width : " w-width " height : " w-height.
```

W$KEYBUF

The W$KEYBUF library routine allows to send keycodes to the runtime, simulating the user input as well as recording the user input to a file for playback.

Note - calling WKBUF-ADD-TO-END or WKBUF-ADD-TO-BEGINNING within control's Before Procedures is not supported.

Syntax:

```
CALL "W$KEYBUF" USING opCode
                      parameters
                GIVING returnCode
```

Parameters:

opCode	Function to be executed. Valid values, defined in iscobol.def.	
	WKBUF-ADD-TO-END	Add keystrokes to the end of the keyboard buffer
	WKBUF-ADD-TO-BEGINNING	Add keystrokes to the beginning of the keyboard buffer
	WKBUF-CLEAR-BUFFER	Clear the keyboard buffer
	WKBUF-START-RECORDING	Start recording keystrokes from the user input
	WKBUF-STOP-RECORDING	Stop recording keystrokes from the user input
	WKBUF-IS-RECORDING-ACTIVE	Checks whether or not the recording mechanism is on
	WKBUF-START-RECORDING-FILE	Start recording keystrokes from the user input into a new file
	WKBUF-START-RECORDING-FILE-APPEND	Start recording keystrokes from the user input into an existing file
	WKBUF-LOAD-FROM-FILE	Read keystrokes from a file and adds them to the kyboard buffer
parameter1	Parameters depend on the opcode.	

Return Code:

returnCode can be any signed numeric data item. The meaning depends on the opcode.

Examples:

Example - Send some keystrokes to the standard input

```
move "This is an automatic text" to txt-string
inspect txt-string replacing trailing spaces by low-values
perform varying i from 1 by 1
          until txt-string(i:1) = low-value
        call "w$keybuf" using 2, txt-string(i:1)
end-perform.
```

WKBUF-ADD-TO-END

This operation adds keystrokes to the end of the keyboard buffer.

Syntax

```
CALL "W$KEYBUF" USING WKBUF-ADD-TO-END
                      keyCodes
                GIVING returnCode
```

Parameters:

WKBUF-ADD-TO-END	Constant	
keyCodes	PIC X(n)	Keycodes to be added. The keycode string can be composed of letters, numbers and special keycodes. Special keycodes must be used with Acucobol-GT syntax and must be placed between curly brackets. For example, in order to send "A", "B" and enter to the runtime, the following string should be used "AB{^M}". Menu selections are encoded as {m#} where "#" is the numeric ID of the menu item. See Acucobol-GT key codes for W$KEYBUF routine for the lists the supported special keycodes.

Return Code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
1	Operation failed.

WKBUF-ADD-TO-BEGINNING

This operation adds keystrokes to the beginning of the keyboard buffer.

Syntax

```
CALL "W$KEYBUF" USING WKBUF-ADD-TO-BEGINNING
                      keyCodes
                GIVING returnCode
```

Parameters:

WKBUF-ADD-TO-BEGINNING	Constant	
keyCodes	PIC X(n)	Keycodes to be added. The keycode string can be composed of letters, numbers and special keycodes. Special keycodes must be used with Acucobol-GT syntax and must be placed between curly brackets. For example, in order to send "A", "B" and enter to the runtime, the following string should be used "AB{^M}". Menu selections are encoded as {m#} where "#" is the numeric ID of the menu item. See Acucobol-GT key codes for W$KEYBUF routine for the lists the supported special keycodes.

Return Code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
1	Operation failed.

WKBUF-CLEAR-BUFFER

This operation clears the keyboard buffer.

Syntax

```
CALL "W$KEYBUF" USING WKBUF-CLEAR-BUFFER
                GIVING returnCode
```

Parameters:

WKBUF-CLEAR-BUFFER	Constant

Return Code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.

1	Operation failed.

WKBUF-START-RECORDING

This operation starts recording keystrokes from the user input.

Keystrokes generated by WKBUF-ADD-TO-END and WKBUF-ADD-TO-BEGINNING, if any, are recorded along with the user input.

Syntax

```
CALL "W$KEYBUF" USING WKBUF-START-RECORDING
                      keyCodes
                     [bufferSize]
              GIVING returnCode
```

Parameters:

WKBUF-START-RECORDING	Constant	
keyCodes	PIC X(n)	Receives the recorded keystrokes.
bufferSize	PIC 9(n)	Optional parameter. It specifies the size of the recording buffer. If omitted, *keyCodes* is used entirely.

Return Code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
1	Operation failed.

WKBUF-STOP-RECORDING

This operation stops recording keystrokes from the user input.

Syntax

```
CALL "W$KEYBUF" USING WKBUF-STOP-RECORDING
                GIVING returnCode
```

Parameters:

WKBUF-STOP-RECORDING	Constant

Return Code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
1	Operation failed.

This operation stops recording keystrokes from the user input.

WKBUF-IS-RECORDING-ACTIVE

This operation inquires whether or not the keystroke recorder is active.

Syntax

```
CALL "W$KEYBUF" USING WKBUF-IS-RECORDING-ACTIVE
                GIVING returnCode
```

Parameters:

WKBUF-IS-RECORDING-ACTIVE	Constant

Return Code:

returnCode can be any numeric data item and provides additional information:

0	The recording mechanism is off.
1	The recording mechanism is on.

WKBUF-START-RECORDING-FILE

This operation causes keys typed by the user to be recorded in a file. If that file exists, it is deleted first.

Keystrokes generated by WKBUF-ADD-TO-END and WKBUF-ADD-TO-BEGINNING, if any, are recorded along with the user input.

Use WKBUF-STOP-RECORDING to stop recording and close the file.

Syntax

```
CALL "W$KEYBUF" USING WKBUF-START-RECORDING-FILE
                      fileName
                GIVING returnCode
```

Parameters:

WKBUF-START-RECORDING-FILE	Constant	
fileName	PIC X(n)	Specifies the name of the file where recorded keystrokes are saved. It can be either a full or a relative pathname.

Return Code:

returnCode can be any numeric data item and provides additional information:

0	The file has been successfully opened for recording.
1	An I/O error occurred while opening the file.

WKBUF-START-RECORDING-FILE-APPEND

This operation causes keys typed by the user to be recorded in a file. If that file exists, the recorded keystrokes are appended to the existing file.

Keystrokes generated by WKBUF-ADD-TO-END and WKBUF-ADD-TO-BEGINNING, if any, are recorded along with the user input.

Use WKBUF-STOP-RECORDING to stop recording and close the file.

Syntax

```
CALL "W$KEYBUF" USING WKBUF-START-RECORDING-FILE-APPEND
                      fileName
                GIVING returnCode
```

Parameters:

WKBUF-START-RECORDING-FILE-APPEND	Constant	
fileName	PIC X(n)	Specifies the name of the file where recorded keystrokes are saved. It can be either a full or a relative pathname.

Return Code:

returnCode can be any numeric data item and provides additional information:

0	The file has been successfully opened for recording.

1	An I/O error occurred while opening the file.

WKBUF-LOAD-FROM-FILE

This operation reads keystrokes from a file and adds them to the keyboard buffer.

Syntax

```
CALL "W$KEYBUF" USING WKBUF-LOAD-FROM-FILE
                      fileName
                GIVING returnCode
```

Parameters:

WKBUF-LOAD-FROM-FILE	Constant	
fileName	PIC X(n)	Specifies the path and the name of the file where to read keystrokes. File Handling Configuration properties don't affect the search of this file. The keycode string read from the file can be composed of letters, numbers and special keycodes. Special keycodes must be used with Acucobol-GT syntax and must be placed between curly brackets. For example, in order to send "A", "B" and enter to the runtime, the following string should be used "AB{^M}" See Acucobol-GT key codes for W$KEYBUF routine for the lists the supported special keycodes.

Return Code:

returnCode can be any numeric data item and provides additional information:

0	Operation successful.
1	Operation failed.

W$MENU

The W$MENU library routine provides a number of functions to manage menus.

Note - Every call to W$MENU performed before the first WMENU-SHOW is buffered and doesn't update the video. Every call to W$MENU performed after the first WMENU-SHOW immediately updates the video and therefore it might affect performance.

Syntax:

```
CALL "W$MENU" USING opCode
                    parameters
              GIVING returnCode
```

Parameters:

opCode	Function to be executed. Valid values, defined in isgui.def, are:	
	WMENU-ADD	Add an item to an existing menu
	WMENU-ADD-BITMAP	Display a bitmap in front of a menu item
	WMENU-BLOCK	Inhibits the user from using the menu menu
	WMENU-CHANGE	Modify a menu item
	WMENU-CHANGE-BITMAP	Modify the bitmap shown in front of a menu item
	WMENU-CHECK	Show a check mark beside a menu item
	WMENU-DELETE	Delete a menu item
	WMENU-DELETE-BITMAP	Delete a bitmap displayed in front of a menu item
	WMENU-DESTROY	Destroy a menu and remove it from the screen
	WMENU-DESTROY-DELAYED	Mark a menu, so that it will be destroyed later
	WMENU-DISABLE	Disable a menu item
	WMENU-ENABLE	Enable a menu item previously disabled
	WMENU-ENSURE-VISIBLE	Expands the necessary menus in order to show a given menu item
	WMENU-GET-BLOCK	Gets the menu blocking count
	WMENU-GET-MENU	Get the handle of the current menu
	WMENU-NEW	Create a new menu bar
	WMENU-NEW-POPUP	Create a new pop-up menu
	WMENU-NEW-TRAY	Create a new tray icon with menu
	WMENU-POPUP	Display a pop-up menu
	WMENU-RELEASE	Logically destroy the current menu, without updating the screen
	WMENU-SET-BLOCK	Sets the menu blocking count
	WMENU-SHOW	Display a menu on the screen
	WMENU-UNBLOCK	Re-enables the user to use the menu menu
	WMENU-UNCHECK	Remove a check mark shown beside a menu item
parameters	Parameters depend on the opcode.	

Return code:

returnCode definition and meaning depend on the opcode.

Examples:

Example - Building a whole menu with options, sub-options and icons to some sub-options

```
77  h-img                   pic s9(9) comp-4.

78  mnu-new                 value 101.
78  mnu-open                value 102.
78  mnu-save                value 103.
78  mnu-cut                 value 201.
78  mnu-copy                value 202.
78  mnu-paste               value 203.

procedure division.
main.
  perform build-menu
  ...

build-menu.
  call "w$bitmap" using wbitmap-load, "files/icone-menu.gif",
                        giving h-img

  call "w$menu" using wmenu-new giving menu-handle

  call "w$menu" using wmenu-new giving sub-handle-1

  call "w$menu" using wmenu-add, menu-handle, 0, 0,
                              "&File", 0, sub-handle-1
  call "w$menu" using wmenu-add, sub-handle-1, 0, 0,
                              "&New", mnu-new
  call "w$menu" using wmenu-add-bitmap
                              menu-handle mnu-new h-img 3 24
  call "w$menu" using wmenu-add, sub-handle-1, 0, 0,
                              "&Open", mnu-open
  call "w$menu" using wmenu-add-bitmap
                              menu-handle mnu-open h-img 2 24
  call "w$menu" using wmenu-add, sub-handle-1, 0, 0,
                              "&Save", mnu-save
  call "w$menu" using wmenu-add-bitmap
                              menu-handle mnu-save h-img 5 24
  call "w$menu" using wmenu-add, sub-handle-1, 0, w-separator
          call "w$menu" using wmenu-add, sub-handle-1, 0, 0,
  "E&xit", 27
```

```
call "w$menu" using wmenu-new giving sub-handle-2

call "w$menu" using wmenu-add, menu-handle, 0, 0,
                            "&Edit", 0, sub-handle-2
call "w$menu" using wmenu-add, sub-handle-2, 0, 0,
                            "&Cut", mnu-cut
call "w$menu" using wmenu-add-bitmap
                            menu-handle mnu-cut h-img 1 24
call "w$menu" using wmenu-add, sub-handle-2, 0, 0,
                            "&Copy", mnu-copy
call "w$menu" using wmenu-add-bitmap
                            menu-handle mnu-copy h-img 4 24
call "w$menu" using wmenu-add, sub-handle-2, 0, 0,
                            "&Paste", mnu-paste
call "w$menu" using wmenu-add-bitmap
                            menu-handle mnu-paste h-img 6 24
call "w$menu" using wmenu-add, sub-handle-2, 0, w-separator
call "w$menu" using wmenu-add, sub-handle-2, 0, w-disabled,
                            " w-disabled", 70
call "w$menu" using wmenu-add, sub-handle-2, 0, w-checked,
                            " w-checked", 80
.
```

WMENU-ADD

The WMENU-ADD function adds an item to an existing menu.

Syntax:

```
CALL "W$MENU" USING WMENU-ADD
                    menuHandle
                    position
                    flags
                    text
                    ID
                   [submenu]
              GIVING returnCode
```

Parameters:

WMENU-ADD	Constant		
menuHandle	USAGE HANDLE	Specifies the handle of an existing menu.	
position	any numeric data item or numeric literal	It specifies where to insert the new item. If this parameter is set to 0, then the new item is added at the end of the menu. Otherwise specify the menu ID before which to insert the new item.	
flags	any numeric data item or numeric literal	It defines the item characteristics. Possible values, that can be added together, are:	
		W-CHECKED	If this flag is applied a check mark is put beside the menu item.

		W-DISABLED	If this flag is applied the menu item is disabled and the user cannot select it. The item is grayed out.
		W-SEPARATOR	If this flag is applied the menu item is displayed as a bar separator. The content of the TEXT data item (see below) is not considered.
text	PIC X(n)	Specifies the text of the menu shown on the menu bar. A key letter can be set by adding a "&" character before the desired letter, part of this text. When the menu is displayed, all the key letters are shown underlined, and it is possible to activate that menu item by pressing the [Alt] key in conjunction with the key letter. A "\t" character works as a "tab". If you need to insert a "\t" in the menu description you have to write "\\t".	
ID	any numeric data item or numeric literal	Specifies the menu item. This ID is returned to the program when the user selects this item. In this way it is possible to identify the selected item.	
submenu	USAGE HANDLE	If this item is set to a valid menu handle the newly added item will be a submenu and it will show the menu pointed by submenu when selected by the user. This parameter should be omitted or set to zero for separators and items not containing submenus.	

Return code:

returnCode can be any signed numeric data item and provides additional information:

<=0	Operation failed.
>0	Operation successful.

WMENU-ADD-BITMAP

The WMENU-ADD-BITMAP function adds a bitmap to a menu item.

Syntax:

```
CALL "W$MENU" USING WMENU-ADD-BITMAP
                    menuHandle
                    ID
                    bitmapHandle
                   [bitmapNumber]
                   [bitmapSize]
              GIVING returnCode
```

Parameters:

WMENU-ADD-BITMAP	Constant	
menuHandle	USAGE HANDLE	Specifies the handle of an existing menu.
ID	any numeric data item or numeric literal	Specifies the menu entry you want to add a bitmap.
bitmapHandle	USAGE HANDLE	Specifies the handle of a bitmap or a bitmap strip (see W$BITMAP).
bitmapNumber	any numeric data item or numeric literal	Specifies the ordinal position of the bitmap in the bitmap strip. If the bitmap handle does not reference a bitmap strip this parameter is optional.
bitmapSize	any numeric data item or numeric literal	Specifies the size of each bitmap in the bitmap strip.

Return code:

returnCode can be any signed numeric data item and provides additional information:

<=0	Operation failed.
>0	Operation successful.

WMENU-BLOCK

The WMENU-BLOCK function increases the menu blocking count by 1. The menu is blocked when the blocking count is greater than zero. See also the WMENU-UNBLOCK, WMENU-GET-BLOCK, and WMENU-SET-BLOCK functions.

Syntax:

```
CALL "W$MENU" USING WMENU-BLOCK
              GIVING returnCode
```

Parameters:

WMENU-BLOCK	Constant

Return code:

returnCode can be any signed numeric data item and provides additional information:

<=0	Operation failed.
>0	Operation successful.

WMENU-CHANGE

The WMENU-CHANGE function modifies a menu item. If an item is modified from being a submenu to a normal item than that submenu is destroyed.

Syntax:

```
CALL "W$MENU" USING WMENU-CHANGE
                    menuHandle
                    position
                    flags
                    text
                    ID
                   [submenu]
              GIVING returnCode
```

Parameters:

WMENU-CHANGE	Constant	
menuHandle	USAGE HANDLE	Specifies the handle of an existing menu.
position	any numeric data item or numeric literal	Specifies the ID of the menu entry you want to change. That entry is deleted and the entry described by the current parameters is inserted in the same location. It must be greater than zero.
flags	any numeric data item or numeric literal	It defines the item characteristics. Possible values, that can be added together, are:

		W-CHECKED	If this flag is applied a check mark is put beside the menu item.
		W-DISABLED	If this flag is applied the menu item is disabled and the user cannot select it. The item is grayed out.
		W-SEPARATOR	If this flag is applied the menu item is displayed as a bar separator. The content of the TEXT data item (see below) is not considered.
text	PIC X(n)	Indicates the text of the menu shown on the menu bar. A key letter can be set by adding a "&" character before the desired letter, part of this text. When the menu is displayed, all the key letters are shown underlined, and it is possible to activate that menu item by pressing the [Alt] key in conjunction with the key letter.	
ID	any numeric data item or numeric literal	Specifies the menu item. This ID is returned to the program when the user selects this item. In this way it is possible to identify the selected item.	
submenu	USAGE HANDLE	If this item is set with a menu handle, than this item, when selected, opens up a submenu. For normal operation you cam omit this parameter or set it to "0".	

Return code:

returnCode can be any signed numeric data item and provides additional information:

<=0	Operation failed.
>0	Operation successful.

WMENU-CHANGE-BITMAP

The WMENU-CHANGE-BITMAP function changes the bitmap in a menu item.

Syntax:

```
CALL "W$MENU" USING WMENU-CHANGE-BITMAP
                    menuHandle
                    ID
                    bitmapHandle
                   [bitmapNumber]
                   [bitmapSize]
              GIVING returnCode
```

Parameters:

WMENU-CHANGE-BITMAP	Constant	
menuHandle	USAGE HANDLE	Specifies the handle of an existing menu.

ID	any numeric data item or numeric literal	Specifies the menu entry you want to change a bitmap.
bitmapHandle	USAGE HANDLE	It specifies an handle of a bitmap or a bitmap strip (see W$BITMAP).
bitmapNumber	any numeric data item or numeric literal	Specifies the ordinal position of the bitmap in the bitmap strip. If this parameter is omitted, the first bitmap is used.
bitmapSize	any numeric data item or numeric literal	Specifies the size of each bitmap in the bitmap strip.

Return code:

returnCode can be any signed numeric data item and provides additional information:

<=0	Operation failed.
>0	Operation successful.

WMENU-CHECK

The WMENU-CHECK function adds a check mark to a menu item.

Syntax:

```
CALL "W$MENU" USING WMENU-CHECK
                    menuHandle
                    ID
              GIVING returnCode
```

Parameters:

WMENU-CHECK	Constant	
menuHandle	USAGE HANDLE	Specifies the handle of an existing menu.
ID	any numeric data item or numeric literal	Specifies the menu entry to be checked.

Return code:

returnCode can be any signed numeric data item and provides additional information:

<=0	Operation failed.
>0	Operation successful.

WMENU-DELETE

The WMENU-DELETE function deletes a menu item. If the item to be deleted refers to a submenu, that submenu is destroyed.

Syntax:

```
CALL "W$MENU" USING WMENU-DELETE
                    menuHandle
                    ID
              GIVING returnCode
```

Parameters:

WMENU-DELETE	Constant	
menuHandle	USAGE HANDLE	Specifies the handle of an existing menu.
ID	any numeric data item or numeric literal	Specifies the menu entry to be deleted.

Return code:

returnCode can be any signed numeric data item and provides additional information:

<=0	Operation failed.
>0	Operation successful.

WMENU-DELETE-BITMAP

The WMENU-DELETE-BITMAP removes a bitmap from a menu item.

Syntax:

```
CALL "W$MENU" USING WMENU-DELETE-BITMAP
                    menuHandle
                    ID
              GIVING returnCode
```

Parameters:

WMENU-DELETE-BITMAP	Constant	
menuHandle	USAGE HANDLE	Specifies the handle of an existing menu.
ID	any numeric data item or numeric literal	Specifies the menu entry you want to delete a bitmap.

Return code:

returnCode can be any signed numeric data item and provides additional information:

<=0	Operation failed.
>0	Operation successful.

WMENU-DESTROY

The WMENU-DESTROY function destroys a menu, including optional sub-menus. The destroyed menu is removed from the screen and all the associated memory is released.

Note: Using the DESTROY statement instead of this op-code will not remove the handle from memory.

Syntax:

```
CALL "W$MENU" USING WMENU-DESTROY
                    menuHandle
              GIVING returnCode
```

Parameters:

WMENU-DESTROY	Constant	
menuHandle	USAGE HANDLE	Specifies the handle of an existing menu.

Return code:

returnCode can be any signed numeric data item and provides additional information:

<=0	Operation failed.
>0	Operation successful.

WMENU-DESTROY-DELAYED

The WMENU-DESTROY-DELAYED function destroys a menu, including optional sub-menus. The menu is destroyed only if it is not displayed on the screen. If it is displayed the menu is destroyed only when a WMENU-SHOW function is executed.

Syntax:

```
CALL "W$MENU" USING WMENU-DESTROY-DELAYED
                    menuHandle
              GIVING returnCode
```

Parameters:

WMENU-DESTROY-DELAYED	Constant	
menuHandle	USAGE HANDLE	Specifies the handle of an existing menu.

Return code:

returnCode can be any signed numeric data item and provides additional information:

<=0	Operation failed.
>0	Operation successful.

WMENU-DISABLE

The WMENU-DISABLE function disables a menu item or the entire menu. Users cannot select disabled menu items.

Syntax:

```
CALL "W$MENU" USING WMENU-DISABLE
                    menuHandle
                   [ID]
              GIVING returnCode
```

Parameters:

WMENU-DISABLE	Constant	
menuHandle	USAGE HANDLE	Specifies the handle of an existing menu.
ID	any numeric data item or numeric literal	Specifies the menu entry to be disabled. If this parameter is omitted, the entire menu is disabled.

Return code:

returnCode can be any signed numeric data item and provides additional information:

<=0	Operation failed.
>0	Operation successful.

WMENU-ENABLE

The WMENU-ENABLE function enables a menu item or the entire menu.

Syntax:

```
CALL "W$MENU" USING WMENU-ENABLE
                    menuHandle
                   [ID]
              GIVING returnCode
```

Parameters:

WMENU-ENABLE	Constant	
menuHandle	USAGE HANDLE	Specifies the handle of an existing menu.
ID	any numeric data item or numeric literal	Specifies the menu entry to be enabled. If this parameter is omitted then the entire menu is enabled.

Return code:

returnCode can be any signed numeric data item and provides additional information:

<=0	Operation failed.
>0	Operation successful.

WMENU-ENSURE-VISIBLE

The WMENU-ENSURE-VISIBLE function automatically expands the necessary menus in order to show a given menu item.

Syntax:

```
CALL "W$MENU" USING WMENU-ENSURE-VISIBLE
                    menuHandle
                    ID
              GIVING returnCode
```

Parameters:

WMENU-ENABLE	Constant	
menuHandle	USAGE HANDLE	Specifies the handle of an existing menu.
ID	any numeric data item or numeric literal	Specifies the menu entry to be shown.

Return code:

returnCode can be any signed numeric data item and provides additional information:

<=0	Operation failed.
>0	Operation successful.

WMENU-GET-BLOCK

The WMENU-GET-BLOCK function retrieves the current menu blocking count. See also the WMENU-BLOCK, WMENU-UNBLOCK, and WMENU-SET-BLOCK functions.

Syntax:

```
CALL "W$MENU" USING WMENU-GET-BLOCK
              GIVING blockingCount
```

Parameters:

WMENU-GET-BLOCK	Constant

Return code:

blockingCount can be any signed numeric data item and provides additional information:

<0	Operation failed.
>=0	This is the current blocking count.

WMENU-GET-MENU

The WMENU-GET-MENU function retrieves the handle of the current menu bar.

Syntax:

```
CALL "W$MENU" USING WMENU-GET-MENU
              GIVING menuHandle
```

Parameters:

WMENU-GET-MENU	Constant

Return code:

menuHandle must be an USAGE HANDLE data item and provides additional information:

<=0	Operation failed.
>0	This is the handle to the currently displayed menu bar.

WMENU-NEW

The WMENU-NEW function creates a new empty menu bar.

The optional parameters allow to implement a scroll feature, that is useful for long menus in order to avoid them getting over the screen. The number of items to fill the vertical space is calculated according to the text of menu items so, if bitmaps are involved, the filling may not be accurate.

Syntax:

```
CALL "W$MENU" USING WMENU-NEW
                    [scrollItems]
                    [fixedTopItems]
                    [fixedBottomItems]
                    [scrollingInterval]
              GIVING menuHandle
```

Parameters:

WMENU-NEW	Constant	
scrollItems	PIC 9(n)	Specifies the number of items in the middle with up/down arrows to scroll. If omitted, the scroll feature will not be available.
fixedTopItems	PIC 9(n)	Specifies the number of items at the top that must be always visible. If omitted, 0 is assumed.
fixedBottomItems	PIC 9(n)	Specifies the number of items at the bottom that must be always visible. If omitted, 0 is assumed.
scrollingInterval	PIC 9(n)	Specifies the number of milliseconds used to scroll items. If omitted, the system default is used.

Return code:

menuHandle must be an USAGE HANDLE data item and provides additional information:

<=0	Operation failed.
>0	Receives the handle to the newly created menu.

WMENU-NEW-POPUP

The WMENU-NEW-POPUP function creates a new empty pop-up menu.

The optional parameters allow to implement a scroll feature, that is useful for long menus in order to avoid them getting over the screen. The number of items to fill the vertical space is calculated according to the text of menu items so, if bitmaps are involved, the filling may not be accurate.

Syntax:

```
CALL "W$MENU" USING WMENU-NEW-POPUP
                    [scrollItems]
                    [fixedTopItems]
                    [fixedBottomItems]
                    [scrollingInterval]
              GIVING menuHandle
```

Parameters:

WMENU-NEW-POPUP	Constant	
scrollItems	PIC 9(n)	Specifies the number of items in the middle with up/down arrows to scroll. If omitted, the scroll feature will not be available.
fixedTopItems	PIC 9(n)	Specifies the number of items at the top that must be always visible. If omitted, 0 is assumed.
fixedBottomItems	PIC 9(n)	Specifies the number of items at the bottom that must be always visible. If omitted, 0 is assumed.
scrollingInterval	PIC 9(n)	Specifies the number of milliseconds used to scroll items. If omitted, the system default is used.

Return code:

menuHandle must be an USAGE HANDLE data item and provides additional information:

<=0	Operation failed.
>0	Receives the handle to the newly created menu.

WMENU-NEW-TRAY

The WMENU-NEW-TRAY function creates a new empty menu for they system tray.

The menu is shown when you right click on the tray icon. Clicking with the left mouse button, instead, produces an exception that can be intercepted by the program in the CRT STATUS, same as clicking on the menu items.

Syntax:

```
CALL "W$MENU" USING WMENU-NEW-TRAY
                    text
                    ID
                    ID2
                   [bitmapHandle]
                   [bitmapNumber]
                   [bitmapWidth]
              GIVING menuHandle
```

Parameters:

WMENU-NEW-TRAY	Constant	
text	PIC X(n)	It specifies the content of the tool-tip shown when the user leaves the mouse pointer over the tray icon.
ID	PIC 9(n)	It specifies the exception value returned upon a single click on the tray icon. If zero, no exception is returned.
ID2	PIC 9(n)	It specifies the exception value returned upon a double click on the tray icon. If zero, no exception is returned.
bitmapHandle	PIC S9(9) COMP-4	It specifies the bitmap image shown as tray icon. If this parameter is omitted, then the isCOBOL logo is shown as tray icon.
bitmapNumber	PIC 9(n)	It specifies which bitmap (among the ones in the bitmap strip referenced by *bitmap-Handle*) is to be displayed in the item identified by the Item property.
bitmapWidth	PIC 9(n)	This property identifies the width in pixels of the image displayed as tray icon. The bitmap strip identified by the *bitmapHandle* property is divided into several smaller images. The width of each image is the value assigned to this property.

Return code:

menuHandle must be an USAGE HANDLE data item and provides additional information:

<=0	Operation failed.
>0	Receives the handle to the newly created menu.

WMENU-POPUP

The WMENU-POPUP function displays a pop-up menu. The menu is automatically removed after the user has selected an item from the menu. The user response is treated as if the user had chosen an item from the standard menu bar.

WMENU-POPUP is asynchronous and therefore it should be followed by an ACCEPT statement that waits for

user input in order to intercept user choice.

Syntax:

```
CALL "W$MENU" USING WMENU-POPUP
                    menuHandle
                   [row]
                   [column]
              GIVING returnCode
```

Parameters:

WMENU-POPUP		
menuHandle	USAGE HANDLE	This is a handle of an existing pop-up menu, created with the WMENU-NEW-POPUP function.
row	any numeric data item or numeric literal	It represents, in pixels, the vertical position where the pop-up menu is displayed. If this parameter is omitted, the pop-up menu will appear at the mouse cursor's position.
column	any numeric data item or numeric literal	It represents, in pixels, the horizontal position where the pop-up menu is displayed.

Return code:

returnCode can be any signed numeric data item and provides additional information:

<=0	Operation failed.
>0	Operation successful.

WMENU-RELEASE

The WMENU-RELEASE function logically removes a menu from the screen, but it is still visible. This is useful, for example, when you want to clear an entire window containing a menu bar. In this way the menu bar and the window content disappears in just one operation.

Syntax:

```
CALL "W$MENU" USING WMENU-RELEASE
              GIVING returnCode
```

Parameters:

WMENU-RELEASE	Constant

Return code:

returnCode can be any signed numeric data item and provides additional information:

<=0	Operation failed.
>0	Operation successful.

WMENU-SET-BLOCK

The WMENU-SET-BLOCK function sets the current menu blocking count. When a program changes the menu blocking count, it should retrieve the current value before changing it and restore it before exiting. See also the WMENU-BLOCK, WMENU-UNBLOCK, and WMENU-GET-BLOCK functions.

Syntax:

```
CALL "W$MENU" USING WMENU-SET-BLOCK
                    blockingCount
              GIVING returnCode
```

Parameters:

WMENU-SET-BLOCK	Constant	
blockingCount	any numeric data item or numeric literal	Represents the value to be assigned to the menu blocking count. When it is zero, the menu is available to the user. When it is greater than zero, the menu is blocked.

Return code:

returnCode can be any signed numeric data item and provides additional information:

<=0	Operation failed.
>0	Operation successful.

WMENU-SHOW

The WMENU-SHOW function displays a menu.

If the menu was created by WMENU-NEW, it is displayed on the screen. If the window has already a menu, it is

replaced. The previous menu is not destroyed, its memory is not released and it can be displayed again with the WMENU-SHOW function.

If the menu was created by WMENU-NEW-TRAY, it is displayed as a tray icon in the system tray.

Syntax:

```
CALL "W$MENU" USING WMENU-SHOW
                    menuHandle
                   [windowHandle]
              GIVING returnCode
```

Parameters:

WMENU-SHOW	Constant	
menuHandle	USAGE HANDLE	Specifies the handle of an existing menu. Setting this parameter to 0 removes the menu.
windowHandle	USAGE HANDLE	Specifies the window where the menu has to be displayed. If this parameter is omitted then the menu is displayed on the current window. This parameter is ignored when the menu is shown on the system tray.

Return code:

returnCode can be any signed numeric data item and provides additional information:

<=0	Operation failed.
>0	Operation successful.

WMENU-UNBLOCK

The WMENU-UNBLOCK function decreases the current menu blocking count by 1. The menu is blocked when the blocking count is greater than zero. Blocking count cannot be less than zero. See also the WMENU-BLOCK, WMENU-GET-BLOCK, and WMENU-SET-BLOCK functions.

Syntax:

```
CALL "W$MENU" USING WMENU-UNBLOCK
              GIVING returnCode
```

Parameters:

WMENU-UNBLOCK	Constant

Return code:

returnCode can be any signed numeric data item and provides additional information:

<=0	Operation failed.
>0	Operation successful.

WMENU-UNCHECK

The WMENU-UNCHECK function removes the check mark from a menu item.

Syntax:

```
CALL "W$MENU" USING WMENU-UNCHECK
                    menuHandle
                    ID
              GIVING returnCode
```

Parameters:

WMENU-UNCHECK	Constant	
menuHandle	USAGE HANDLE	Specifies the handle of an existing menu.
ID	any numeric data item or numeric literal	Specifies the menu entry to be unchecked.

Return code:

returnCode can be any signed numeric data item and provides additional information:

<=0	Operation failed.
>0	Operation successful.

W$MOUSE

The W$MOUSE library routine provides a number of functions to manage the mouse.

Syntax:

```
CALL "W$MOUSE" USING opCode
                     parameters
```

Parameters:

opCode	Function to be executed. Valid values, defined in isgui.def, are:	
	SET-MOUSE-SHAPE	Modify the mouse pointer shape.
	GET-MOUSE-STATUS	Returns coordinates of the mouse pointer.
	TEST-MOUSE-PRESENCE	Detects the presence of a mouse.
parameters	Parameters depend on the opcode.	

Examples:

Example - Get mouse position

```
*> copy "isgui.def" on working-storage
call "w$mouse" using get-mouse-status
                     mouse-info
display "row: "   at 0270
display mouse-row at 0275
display "col: "   at 0470
display mouse-col at 0475.
```

Example - Change mouse shape to wait pointer

```
*> copy "isgui.def" on working-storage
call "w$mouse" using set-mouse-shape, wait-pointer.
```

SET-MOUSE-SHAPE

The SET-MOUSE-SHAPE modifies the shape of the mouse pointer.

Syntax:

```
CALL "W$MOUSE" USING SET-MOUSE-SHAPE
                      mouseShape
                     [bitmapHandle]
                     [column]
                     [line]
               GIVING returnCode
```

Parameters:

SET-MOUSE-SHAPE	Constant	
mouseShape	any numeric data item or numeric literal	Specifies the new shape of the mouse pointer. Valid values, defined in isgui.def, are:

		ARROW-POINTER	It represents an arrow
		BAR-POINTER	It represents a vertical bar
		CROSS-POINTER	It represents a cross
		CUSTOM-POINTER	It shows the custom image identified by *bitmapHandle*. The image is automatically scaled to fit the standard cursors dimensions in the current operating system.
		HELP-POINTER	It represents a hand shape
		WAIT-POINTER	It represents an work in progress animation
bitmapHandle	PIC S9(9) COMP-4	Specifies the handle of a bitmap (see W$BITMAP). This parameter should be passed only along with CUSTOM-POINTER.	
column	PIC 9(n)	It specifies the X coordinate of the cursor hotspot, starting from the top-left corner. This parameter should be passed only along with CUSTOM-POINTER. If omitted, 0 is assumed.	
line	PIC 9(n)	It specifies the Y coordinate of the cursor hotspot, starting from the top-left corner. This parameter should be passed only along with CUSTOM-POINTER. If omitted, 0 is assumed.	

GET-MOUSE-STATUS

The GET-MOUSE-STATUS returns information about the mouse's location and the state of each of its buttons.

The various row and column fields are set to the location of the mouse within the current window. If the mouse is outside of the current window, then these values are set to zero.

After an ACCEPT statement is executed, all CALLs to GET-MOUSE-STATUS relate to that ACCEPT statement, until another ACCEPT is executed.

Syntax:

```
CALL "W$MOUSE" USING GET-MOUSE-STATUS
                     MOUSE-INFO
               GIVING returnCode
```

Parameters:

GET-MOUSE-STATUS	Constant	
MOUSE-INFO	Group item	Group item defined in isgui.def as follows:

```
01 mouse-info.
   03  mouse-row               pic xx comp-x.
       88 mouse-off-screen      value zero.
   03  mouse-col               pic xx comp-x.
   03  lbutton-status          pic 9.
       88 lbutton-down         value 1.
   03  mbutton-status          pic 9.
       88 mbutton-down         value 1.
   03  rbutton-status          pic 9.
       88 rbutton-down         value 1.
   03  mouse-row-ex            pic 9(6)v99 comp-
4 sync.
   03  mouse-col-ex            pic 9(6)v99 comp-4.
   03  mouse-row-pixel         pic 9(8) comp-4.
   03  mouse-col-pixel         pic 9(8) comp-4.
```

TEST-MOUSE-PRESENCE

The TEST-MOUSE-PRESENCE tests if mouse is available.

Syntax:

```
CALL "W$MOUSE" USING TEST-MOUSE-PRESENCE
               GIVING returnCode
```

Parameters:

TEST-MOUSE-PRESENCE	Constant

Return Code:

returnCode can be any signed numeric data item and provides additional information:

1	Mouse is available
0	Mouse is not available.

W$PALETTE

The W$PALETTE library routine provides a number of functions to manage RGB colors and COBOL attributes.

NOTE - isCOBOL allows to work with RGB colors. A maximum of 16 million concurrent colors can be displayed on the screen at the same time. However, some old languages display only 16 colors at a time. For compatibility reasons isCOBOL supports this library routine, that allows definition of a palette of usable colors. In any case it is important to pinpoint that the use of this routine is deprecated.

Syntax:

```
CALL "W$PALETTE" USING opCode
                       parameters
                 GIVING paletteResult
```

Parameters:

opCode	Function to be executed. Valid values, defined in ispalette.def, are:	
	WPALETTE-CHOOSE-COLOR	Show a dialog window to choose a color.
	WPALETTE-GET-COLOR	Retrieve the RGB color associated to a COBOL attribute.
	WPALETTE-NUM-COLORS	Retrieve the maximum number of colors that the system can render simultaneously.
	WPALETTE-SET-COLOR	Set the RGB color associated to a COBOL attribute.
	WPALETTE-SUPPORTED	Check if the host system supports the W$PALETTE library routine.
parameters	Parameters depend on the opcode.	

Return code:

returnCode definition and meaning depend on the opcode.

Examples:

Example - Select a color from palette and change a numbered color with the selection

```
working-storage section.
copy "isgui.def"
78  78color   value 3.
77  result    pic 9(2).

procedure division.
...
change-color.
   initialize wpalette-data
   call "w$palette" using wpalette-choose-color,
        wpalette-data,
        giving result
   move 78color to wpal-color-id
   call "w$palette" using wpalette-set-color,
        wpalette-data,
        giving result.
   ...
```

Example - Select a color from palette and change a numbered color with the selection

```
working-storage section.
copy "isgui.def"
78  78color   value 3.
77  result    pic 9(2).

procedure division.
...
change-color.
   initialize wpalette-data
   call "w$palette" using wpalette-choose-color,
        wpalette-data,
        giving result
   move 78color to wpal-color-id
   call "w$palette" using wpalette-set-color,
        wpalette-data,
        giving result.
   ...
```

WPALETTE-CHOOSE-COLOR

The WPALETTE-CHOOSE-COLOR function opens a dialog box that allows the user to pick a color.

Syntax:

```
CALL "W$PALETTE" USING WPALETTE-CHOOSE-COLOR
                       WPALETTE-DATA
                 GIVING returnCode
```

Parameters:

WPALETTE-CHOOSE-COLOR	Constant	

WPALETTE-DATA	Group Item	Group item that receives the user's selection. This group item, defined in ispalette.def, has the following structure:
		
		`01 wpalette-data.`
		`  03 wpal-color-id  pic x comp-x.`
		`  03 wpal-flags`
		`  redefines wpal-color-id  pic x comp-x.`
		`  03 wpal-red  pic x comp-x.`
		`  03 wpal-user-color-id`
		`  redefines wpal-red  pic x comp-x.`
		`  03 wpal-green  pic x comp-x.`
		`  03 wpal-blue  pic x comp-x.`
		
		wpal-color-id
		Not used.
		
		wpal-flags
		Flags affecting the behavior of the dialog box. Possible values are:
		
		wpchoose-use-default — The dialog box is initialized according to the RGB values set in wpaletteData.
		
		wpal-red
		Receives the red component of the color chosen by the user. It may range from 0 to 255.
		
		wpal-user-color-id
		Not used.
		
		wpal-green
		Receives the green component of the color chosen by the user. It may range from 0 to 255.
		
		wpal-blue
		Receives the blue component of the color chosen by the user. It may range from 0 to 255.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful. wpaletteData receives the user's selection.

WPERR-UNSUPPORTED	The WPALETTE-CHOOSE-COLOR function is not supported. wpaletteData is not updated.
WPERR-BAD-ARG	The function has received incorrect parameters. wpaletteData is not updated.
WPERR-CANCELLED	The user has pressed the "Cancel" button. wpaletteData is not updated.

WPALETTE-GET-COLOR

The WPALETTE-GET-COLOR function retrieves the RGB color associated to a COBOL color attribute.

Syntax:

```
CALL "W$PALETTE" USING WPALETTE-GET-COLOR
                       WPALETTE-DATA
                 GIVING returnCode
```

Parameters:

WPALETTE-GET-COLOR	Constant

WPALETTE-DATA	Group Item	Group item that specifies the COBOL color attribute the function refers to and receives the corresponding RGB color. This group item, defined in ispalette.def, has the following structure:
		
		`01 wpalette-data.`
		`  03 wpal-color-id  pic x comp-x.`
		`  03 wpal-flags`
		`  redefines wpal-color-id  pic x comp-x.`
		`  03 wpal-red  pic x comp-x.`
		`  03 wpal-user-color-id`
		`  redefines wpal-red  pic x comp-x.`
		`  03 wpal-green  pic x comp-x.`
		`  03 wpal-blue  pic x comp-x.`
		
		wpal-color-id
		It must be set to the COBOL color attribute before calling the function. It must range from 1 to 16.
		
		wpal-flags
		Not used.
		
		wpal-red
		Receives the red component of the color associated to the COBOL color attribute. It may range from 0 to 255.
		
		wpal-user-color-id
		Not used.
		
		wpal-green
		Receives the green component of the color associated to the COBOL color attribute. It may range from 0 to 255.
		
		wpal-blue
		Receives the blue component of the color associated to the COBOL color attribute. It may range from 0 to 255.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPERR-UNSUPPORTED	The W$PALETTE library routine is not supported. wpaletteData is not updated.
WPERR-BAD-ARG	The function has received incorrect parameters. wpaletteData is not updated.

WPALETTE-NUM-COLORS

The WPALETTE-NUM-COLORS function retrieves the maximum number of colors the system can render simultaneously.

Syntax:

```
CALL "W$PALETTE" USING WPALETTE-NUM-COLORS
                 GIVING returnCode
```

Parameters:

WPALETTE-NUM-COLORS	Constant

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Operation successful. It receives the number of colors the system can render simultaneously.
WPERR-UNSUPPORTED	The W$PALETTE library routine is not supported.

WPALETTE-SET-COLOR

The WPALETTE-SET-COLOR function associates an RGB color to a COBOL color attribute.

Syntax:

```
CALL "W$PALETTE" USING WPALETTE-SET-COLOR
                       WPALETTE-DATA
                 GIVING returnCode
```

Parameters:

WPALETTE-SET-COLOR	Constant

WPALETTE-DATA	Group Item	Group item that specifies the COBOL color attribute the function refers and the RGB color that it will be associated to. This group item, defined in ispalette.def, has the following structure:
		
		`01 wpalette-data.`
		`  03 wpal-color-id  pic x comp-x.`
		`  03 wpal-flags`
		`  redefines wpal-color-id  pic x comp-x.`
		`  03 wpal-red  pic x comp-x.`
		`  03 wpal-user-color-id`
		`  redefines wpal-red  pic x comp-x.`
		`  03 wpal-green  pic x comp-x.`
		`  03 wpal-blue  pic x comp-x.`
		
		wpal-color-id
		It must be set to the COBOL color attribute before calling the function. It must range from 1 to 16.
		
		wpal-flags
		Not used.
		
		wpal-red
		Specifies the red component of the color to be associated to the COBOL color attribute. It may range from 0 to 255.
		
		wpal-user-color-id
		Not used.
		
		wpal-green
		Specifies the green component of the color to be associated to the COBOL color attribute. It may range from 0 to 255.
		
		wpal-blue
		Specifies the blue component of the color to be associated to the COBOL color attribute. It may range from 0 to 255.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPERR-UNSUPPORTED	The W$PALETTE library routine is not supported.
WPERR-BAD-ARG	The function has received incorrect parameters.

WPALETTE-SUPPORTED

The WPALETTE-SUPPORTED function checks if the W$PALETTE library routine is supported by the host system.

Syntax:

```
CALL "W$PALETTE" USING WPALETTE-SUPPORTED
                 GIVING returnCode
```

Parameters:

WPALETTE-SUPPORTED	Constant

Return code:

returnCode can be any signed numeric data item and provides additional information:

WPAL-NO-SUPPORT	The W$PALETTE library routine is not supported.
WPAL-PALETTE-SUPPORTED	The W$PALETTE library routine is supported, except for the WPALETTE-CHOOSE-COLOR function.
WPAL-FULL-SUPPORT	The W$PALETTE library routine is fully supported.

NOTE - Currently, the WPALETTE-SUPPORTED function always returns WPAL-FULL-SUPPORT because the W$PALETTE library routine is fully supported on all platforms.

W$PROGRESSDIALOG

The W$PROGRESSDIALOG library routine provides a general way to show a user how an operation is progressing.

Syntax

```
CALL "W$PROGRESSDIALOG" USING opCode,
                              parameters
                        GIVING returnCode
```

Parameters:

opCode	It is the function to be executed. Valid values, defined in isgui.def, are: WPROGRESSDIALOG-CREATE Create and start the progress dialog. WPROGRESSDIALOG-DESTROY Closes the progress dialog box. WPROGRESSDIALOG-SET-PROGRESS Updates the progress dialog box. WPROGRESSDIALOG-QUERY-CANCEL Check if the user has pressed the Cancel button. WPROGRESSDIALOG-SET-LINE Set the text lines displayed in the progress dialog box. WPROGRESSDIALOG-RESET-TIMER Reset the progress dialog box timer to zero.
parameters	Parameters depend on the opcode.

Return code:

The definition and meaning of the *returnCode* depend on the opcode.

Examples:

Example - Show a work in progress dialog during i/o operations.

```
call "w$progressdialog" using wprogressdialog-create
                              "Elaborating..."
                              "Operation cancelled"
                              wprogressdialog-noprogressbar
                              omitted
                         giving pDialogHandle. |defined as usage handle
                                               |in working-storage
call "w$progressdialog" using wprogressdialog-set-line
                              pdialoghandle
                              "exporting data from files"
                               1, 0.

perform export-data-from-files.
call "w$progressdialog" using wprogressdialog-destroy
                              pDialogHandle.
```

WPROGRESSDIALOG-CREATE

The WPROGRESSDIALOG-CREATE function creates and starts the progress dialog.

Syntax:

```
CALL "W$PROGRESSDIALOG" USING WPROGRESSDIALOG-CREATE
                              title
                              cancelMessage
                              flags
                             [bitmapHandle]
                        GIVING pDialogHandle
```

Parameters:

WPROGRESSDIALOG-CREATE	Constant	
title	PIC X(n)	It specifies the text that will appear in the title bar of the progress dialog.
cancelMessage	PIC X(n)	It specifies the text that is shown on line 3 (below the progress bar) when the user clicks the Cancel button.
flags	PIC 9(n)	It specifies flags that determine the operation of the progress dialog. This can be a combination of the following values, defined in isgui.def: WPROGRESSDIALOG-NORMAL (value 0): Normal progress dialog behavior. WPROGRESSDIALOG-MODAL (value 1): The progress dialog box will be modal to the current window. By default, a progress dialog box is modeless. WPROGRESSDIALOG-AUTOTIME (value 2): Automatically estimate the remaining time and display the estimate on line 3. If this flag is set, WPROGRESSDIALOG-SET-LINE can be used only to display text on lines 1 and 2. WPROGRESSDIALOG-NOTIME (value 4): Do not show the "time remaining" text. WPROGRESSDIALOG-NOPROGRESSBAR (value 16): Display a progress bar in indeterminate state.
bitmapHandle	PIC S9(9) COMP-4	Optional parameter. It specifies an icon to be shown top-right of the dialog.

Return code:

pDialogHandle must be a USAGE HANDLE data item. It receives the handle of the list of users and will be used with the other W$PROGRESSDIALOG functions.

WPROGRESSDIALOG-DESTROY

The WPROGRESSDIALOG-DESTROY function destroys the progress dialog box.

Syntax:

```
CALL "W$PROGRESSDIALOG" USING WPROGRESSDIALOG-DESTROY
                              pDialogHandle
```

Parameters:

WPROGRESSDIALOG-DESTROY	Constant	
pDialogHandle	HANDLE	Handle returned by WPROGRESSDIALOG-CREATE.

WPROGRESSDIALOG-SET-PROGRESS

The WPROGRESSDIALOG-SET-PROGRESS function updates the progress dialog box with the current state of the work being monitored.

Syntax:

```
CALL "W$PROGRESSDIALOG" USING WPROGRESSDIALOG-SET-PROGRESS
                              pDialogHandle
                              completed
                              total
```

Parameters:

WPROGRESSDIALOG-SET-PROGRESS	Constant	
pDialogHandle	HANDLE	Handle returned by WPROGRESSDIALOG-CREATE.
completed	PIC 9(n)	It specifies what proportion of the work has been completed so far.
total	PIC 9(n)	It specifies what value the *completed* parameter will have when the work is complete.

WPROGRESSDIALOG-QUERY-CANCEL

The WPROGRESSDIALOG-QUERY-CANCEL function checks whether the user has pressed the Cancel button. You should periodically use this function to poll the progress dialog box object to determine whether the operation has been canceled.

Syntax:

```
CALL "W$PROGRESSDIALOG" USING WPROGRESSDIALOG-QUERY-CANCEL
                              pDialogHandle
                        GIVING returnCode
```

Parameters:

WPROGRESSDIALOG-QUERY-CANCEL	Constant	
pDialogHandle	HANDLE	Handle returned by WPROGRESSDIALOG-CREATE.

Return code:

returnCode can be any numeric data item and provides additional information:

0	The Cancel button was not clicked.
1	The Cancel button was clicked.

WPROGRESSDIALOG-SET-LINE

The WPROGRESSDIALOG-SET-LINE function sets the text lines that are displayed in the progress dialog.

Syntax:

```
CALL "W$PROGRESSDIALOG" USING WPROGRESSDIALOG-SET-LINE
                              pDialogHandle
                              string
                              lineNum
                              compactPath
```

Parameters:

WPROGRESSDIALOG-SET-LINE	Constant	
pDialogHandle	HANDLE	Handle returned by WPROGRESSDIALOG-CREATE.
string	PIC X(n)	It specifies the text to display.
lineNum	PIC 9(1)	It specifies the line number on which the text is to be displayed. This can be either 1, 2, or 3. If WPROGRESSDIALOG-AUTOTIME was specified in the flags parameter when the progress dialog was created, then only lines 1 and 2 can be used. The estimated time will be displayed on line 3.
compactPath	PIC 9(1)	It activates (if set to "1") or deactivates (if set to "0") path string compaction.

WPROGRESSDIALOG-RESET-TIMER

The WPROGRESSDIALOG-RESET-TIMER function resets the progress dialog box timer to zero.

Syntax:

```
CALL "W$PROGRESSDIALOG" USING WPROGRESSDIALOG-RESET-TIMER
                              pDialogHandle
```

Parameters:

WPROGRESSDIALOG-RESET-TIMER	Constant	
pDialogHandle	HANDLE	Handle returned by WPROGRESSDIALOG-CREATE.

W$ROTATE

The W$ROTATE library routine rotates an image.

The routine creates a new image resource and returns its handle. The newly created resource is not released automatically. It must be destroyed by the programmer with the WBITMAP-DESTROY function when it is no longer needed.

Syntax:

```
CALL "W$ROTATE" USING bitmapHandle
                      rotationRadians
                GIVING rotatedBitmapHandle
```

Parameters:

bitmapHandle	USAGE HANDLE	It specifies an handle of an existing image.
rotationRadians	any numeric data item or numeric literal	Specifies the rotation angle, expressed in radians. Decimal values are allowed. A circle contains 2? radians. Thus, a radian measures 360/2? degrees, i.e. 57.295779513082320876798154814105 degrees. On the other hand, one degree measures 2?/360 radians, i.e. 0.017453292519943295769236907684886 radians. The most convenient way to handle angles is to define the constant PI as 3.14159265358979323 and use it in calculations. You can also define the constant DEG as 0.017453292519943295 and calculate RotationRadians = (Degrees) * DEG.

Return code:

rotatedBitmapHandle can be any data item declared PIC S9(9) and receives the handle to the newly created image, the result of the rotation.

Examples:

Example - Rotate an image 45 degrees

```
*> 77 angle           pic 9v9(6).
*> 77 h-bitmap        handle.
*> 77 h-bitmap-rotate handle.

call "w$bitmap" using wbitmap-load, "c:\tmp\images\img1.png"
     giving h-bitmap
compute angle = 45 * 0.01745
call "w$rotate" using h-bitmap angle
     giving h-bitmap-rotate
```

W$SAVE_IMAGE

The W$SAVE_IMAGE routine saves the image pointed by a bitmap handle into a regular disk file. The save process allows to convert the image to a different format and set some attributes like quality and transparency.

Syntax:

```
CALL "W$SAVE_IMAGE" USING bitmapHandle
                          fileName
                         [WSAVE-OPTIONS]
                    GIVING returnCode
```

Parameters:

bitmapHandle	USAGE HANDLE	It specifies an handle of an existing image.
fileName	PIC X(n)	Specifies the name of a regular disk file that will store the saved image.

WSAVE-OPTIONS	Group Item	Structure that allows to customize the image format, quality and attributes. It's defined in iscobol.def as follows:
		
		`01 wsave-options.`
		`  03 wsave-format pic x(1).`
		`  88 wsave-png values "P", " ".`
		`  88 wsave-bmp value  "B".`
		`  88 wsave-gif value  "G".`
		`  88 wsave-jpg value  "J".`
		`  03 wsave-other.`
		`  05 wsave-quality pic 9(10) comp-x.`
		`  05 wsave-transparency redefines wsave-quality.`
		`  07 filler pic 9(1) comp-x.`
		`  88 no-transparency value 0 false 1.`
		`  07 wsave-transparent-color pic 9(9) comp-x.`
		`  03 wsave-client-server pic x.`
		`  88 wsave-server values "S", " ".`
		`  88 wsave-client value  "C".`
		
		wsave-format
		Image format between BMP, GIF, JPG and PNG. By default PNG is used.
		
		wsave-quality
		Image quality value. The valid range is from 0 (best compression, lower quality) to 100 (no compression, best quality). It's evaluated only for the JPG format.
		
		wsave-transparency
		Activates the transparency and identifies the RGB of the transparent color. *wsave-transparent-color* can be calculated as follows: (RED * 65536) + (GREEN * 256) + BLUE. It's evaluated only for the PNG and GIF formats.
		
		wsave-client-server
		Specifies if the image must be saved server side or client side in a thin client environment. By default the image is saved on the same machine where the runtime system is running, so server side in thin client.

Return code:

returnCode can be any numeric data item and provides additional information:

0	Operation failed.
1	Operation successful.

Example:

Example - Load a BMP file and save it as JPG

```
working-storage section.
copy "isgui.def".
copy "iscobol.def".
77 hbmp handle.

procedure division.
main.
     call "w$bitmap" using wbitmap-load, "src.bmp", giving hbmp.
     initialize wsave-options.
     set wsave-jpg to true.
     move 75 to wsave-quality.
     call "w$save_image" using hbmp, "dest.jpg", wsave-options.
     goback.
```

W$SCALE

The W$SCALE library routine resizes an image.

The routine creates a new image resource and returns its handle. The newly created resource is not released automatically. It must be destroyed by the programmer with the WBITMAP-DESTROY function when it is no longer needed.

The quality of the scaled image is controlled by the iscobol.bitmap_scale.best_quality (boolean) configuration property.

Syntax:

```
CALL "W$SCALE" USING bitmapHandle
                     newWidth
                     newHeight
                     windowHandle
                     scaleMode
                     scaleAlign
               GIVING scaledBitmapHandle
```

Parameters:

bitmapHandle	USAGE HANDLE	It specifies an handle of an existing image.
newWidth	any numeric data item or numeric literal	Specifies the width, in window's cells, of the target area that will contain the new image.
newHeight	any numeric data item or numeric literal	Specifies the height, in window's cells, of the target area that will contain the new image.
windowHandle	USAGE HANDLE	Specifies the handle of an existing window. It is used to calculate the cell size. If this parameter is omitted or zero, then *newWidth* and *newHeight* are considered pixels.

scaleMode	any numeric data item or numeric literal	Specifies the scaling method. Valid values, defined in isgui.def, are:	
		WSCALE-STRETCH	The image is stretched to fit the target area
		WSCALE-RESIZE-XY	Aspect ratio is maintained. The unused part of the target area, if any, will be transparent.
		WSCALE-RESIZE-X	The image is resized so that its width matches the width of the target area, maintaining the aspect ratio. If the new image does not fit vertically the target area, that part of the image that exceeds the space available is truncated. Otherwise, the unused part of the target area will be transparent.
		WSCALE-RESIZE-Y	The image is resized so that its height matches the height of the target area, maintaining the aspect ratio. If the new image does not fit horizontally the target area, that part of the image that exceeds the space available is truncated. Otherwise, the unused part of the target area will be transparent.
scaleAlign	any numeric data item or numeric literal	Specifies the placement of the new image in the target area. Valid values, defined in isgui.def, are:	
		WSCALE-AL-BOTTOM-LEFT	The new image is bottom/left aligned.
		WSCALE-AL-BOTTOM-CENTER	The new image is bottom/center aligned.
		WSCALE-AL-BOTTOM-RIGHT	The new image is bottom/right aligned.
		WSCALE-AL-MIDDLE-LEFT	The new image is middle/left aligned.
		WSCALE-AL-MIDDLE-CENTER	The new image is middle/center aligned.
		WSCALE-AL-MIDDLE-RIGHT	The new image is middle/right aligned.
		WSCALE-AL-TOP-LEFT	The new image is top/left aligned.
		WSCALE-AL-TOP-CENTER	The new image is top/center aligned.
		WSCALE-AL-TOP-RIGHT	The new image is top/right aligned.

Return code:

scaledBitmapHandle can be any data item declared PIC S9(9) and receives the handle to the newly created image, the result of the scaling.

Examples:

Example - Scale an image to width=22 and height=10

```
*> 77 h-bitmap        handle.
*> 77 h-bitmap-resize handle.
*> 77 hWin            handle.

display independent graphical window
                  color  65793
                  with   system menu
                  title  "W$SCALE Routine"
                  handle hWin.
...
call "w$scale" using h-bitmap 22 10 hWin
     wscale-resize-xy
     wscale-al-top-left
     giving h-bitmap-resize.
```

W$TEXTSIZE

The W$TEXTSIZE library routine measures the height and the width of a text.

Syntax:

```
 CALL "W$TEXTSIZE" USING textString
                         TEXTSIZE-DATA
```

Parameters:

textString	PIC X(n)	Specifies the text to be measured.

TEXTSIZE-DATA group item

This group item, defined in isgui.def, has the following structure:

```
01  textsize-data.
    03 textsize-font             handle of font value null.
    03 textsize-window           handle of window
                                 value null.
    03 textsize-size-x           pic 9(7)v99 comp-4.
    03 textsize-cells-x          pic 9(7)v99 comp-4.
    03 textsize-base-x           pic 9(9) comp-4.
    03 textsize-size-y           pic 99v99 comp-4.
    03 textsize-cells-y          pic 99v99 comp-4.
    03 textsize-base-y           pic 9(4) comp-4.
    03 textsize-flags            pic x comp-x value zero.
       88 textsize-strip-spaces  value 1 false zero.
```

textsize-font

Specifies the handle of the font used to measure the text specified in textString. **Note:** if the font has been loaded with wfont-angle different than zero, results are unpredictable. Rotated fonts should not be used for cell measurement.

textsize-window

Specifies the handle of the window used to measure the text specified in textString. This is necessary to return the size of the text in cells. If this member is zero, the current window is used.

textsize-size-x

Receives the width of textString, expressed in columns. A column is the width of the character "0" (zero) when drawn with the font specified in textsize-font.

textsize-cells-x

Receives the width of textString, expressed in window's cells. Cells refer to the window pointed by textsize-window.

textsize-base-x

Receives the width of textString, expressed in pixels.

textsize-size-y

Receives the height of textString, expressed in lines. A line is the height of the character "0" (zero) when drawn with the font specified in textsize-font.

textsize-cells-y

Receives the height of textString, expressed in window's cells. Cells refer to the window pointed by textsize-window.

textsize-base-y

Receives the height of textString, expressed in pixels.

textsize-flags

It specifies if trailing spaces found in textString are relevant to the measurement.

When the textsize-strip-spaces condition is set to true, trailing spaces not measured. When it is set to false, trailing spaces are measured.

Examples:

Example - retrieve cell dimensions on the current window with default font

```
working-storage section.
77  hWin          handle of window.
77  text-string   pic x(20).
77  h-font        handle of font.

procedure division.
main.
   display independent graphical window
           color  65793
           with   system menu
           title  "W$TEXTSIZE Routine"
           handle hWin.
   accept h-font from standard object "default-font"
   ...

   move hWin   to textsize-window
   move h-font to textsize-font
   set textsize-strip-spaces to true
   inquire event-control-handle value in text-string
   call "W$TEXTSIZE" using text-string, textsize-data
   display message textsize-size-x  "  "
                   textsize-cells-x   "  "
                   textsize-base-x  "  "
                   textsize-size-y  "  "
                   textsize-cells-y  "  "
                   textsize-base-y.
```

WD2$CLIENT_INFO

The WD2$CLIENT_INFO library routine returns information about the web browser. It can be used only in webDirect environment.

Syntax:

```
CALL "WD2$CLIENT_INFO" USING opCode
                             parameter
                       GIVING returnCode
```

Parameters:

opCode	PIC 9	Function to be executed. Valid values, defined in iscobol.def, are:
	WD2-GET-USER-AGENT	Returns the user agent http header
	WD2-GET-BROWSER-NAME	Returns the browser name
parameter	PIC X(n)	Parameter depends on the opcode.

Return code:

returnCode definition and meaning depend on the opcode.

Examples:

Example - Get Browser and Agent from client (this works only on webDirect)

```
*> On working-storage copy "isgui.def"

call "WD2$CLIENT_INFO" using wd2-get-browser-name
                             browser-info
call "WD2$CLIENT_INFO" using wd2-get-user-agent
                             user-agent
display message user-agent X"0D0A" browser-info.
```

WD2-GET-USER-AGENT

The WD2-GET-USER-AGENT function returns the user agent http header.

Syntax:

```
CALL "WD2$CLIENT_INFO" USING WD2-GET-USER-AGENT
                             userAgent
                       GIVING returnCode
```

Parameters:

WD2-GET-USER-AGENT	CONSTANT	
userAgent	PIC X(n)	Receives the user agent. For example, using Mozilla Firefox, the result is something like "*mozilla/4.0 (compatible; msie 7.0; windows nt 6.0; slcc1; .net clr 2.0.50727; media center pc 5.0; .net clr 3.5.30729; .net clr 3.0.30618)*"

Return code:

0	operation successful
-1	running outside webDirect environment

WD2-GET-BROWSER-NAME

The WD2-BROWSER-NAME function returns the name of the web browser.

Syntax:

```
CALL "WD2$CLIENT_INFO" USING WD2-GET-BROWSER-NAME
                             browserName
                       GIVING returnCode
```

Parameters:

WD2-GET-BROWSER-NAME	CONSTANT	
browserName	PIC X(n)	Receives the browser name. For example, using Mozilla Firefox, the result is "*firefox*".

Return code:

0	operation successful
-1	running outside webDirect environment

WD2$EXECJS

The WD2$EXECJS library routine executes JavaScript code. It can be used only in webDirect environment.

Note - if you use this routine to redirect the browser to a file created by the COBOL program (e.g. a PDF printed by the program) and you wish to specify a relative path to the file in your JavaScript, be aware that the relative path for the JavaScript doesn't always match with the relative path used by the COBOL program. When you run your programs from the IDE the relative paths match between COBOL program and JavaScript. Instead, when you deploy your programs in a servlet container (e.g. Tomcat), JavaScript appends the relative path to the webapp main folder while the COBOL program appends them to the servlet container working directory.

Syntax:

```
CALL "WD2$EXECJS" USING jsCode
                  GIVING returnCode
```

Parameters:

jsCode	PIC X(n)	Specifies the JavaScript code to execute. It must be pure JavaScript code without any decoration.

Return code:

0	Operation successful.
-1	Operation failed. Possible causes: • Invalid or missing parameter • Running outside webDirect environment

Examples:

Example - Show an alert message via JavaScript (this works only on webDirect)

```
call "wd2$execjs" using "function display_alert(message) { alert (message);  }
display_alert('Hello world');"
```

WD2$REDIRECT

The WD2$REDIRECT library routine redirects the browser to a new page or resource. It can be used only in webDirect environment.

Note - if you use this routine to redirect the browser to a file created by the COBOL program (e.g. a PDF printed by the program) and you wish to specify a relative path to the file in your JavaScript, be aware that the relative path for the JavaScript doesn't always match with the relative path used by the COBOL program. When you run your programs from the IDE the relative paths match between COBOL program and JavaScript. Instead, when you deploy your programs in a servlet container (e.g. Tomcat), JavaScript appends the relative path to the webapp main folder while the COBOL program appends them to the servlet container working directory.

Syntax:

```
CALL "WD2$REDIRECT" USING newURL
                          [target]
                    GIVING returnCode
```

Parameters:

newURL	PIC X(n)	Specifies the link to the resource to be opened. It can be a link to a website, such as "http://www.veryant.com" or a link to a file, such as "resources/pdf_file.pdf".
target	PIC X(n)	Optional parameter. It specifies the target where to open the document, and it can be any value accepted for an <a> html tag. Valid values are: • "_blank" • "_parent" • "_self" • "_top" • *framename* (the name o a frame in a frameset). If omitted, then "_blank" is assumed as default.

Return code:

0	Operation successful.
-1	The redirect function is not available (program not running in a webDirect environment)
-2	Invalid or missing parameter.

Examples:

Example - Redirect to the Veryant's web site

```
call "wd2$redirect" using "http://www.veryant.com" "_self"
```

WD2$RUN_JS

The WD2$RUN_JS library routine executes JavaScript code. It can be used only in webDirect environment.

The routine sends a html snippet that includes a <script> tag to the browser that will interpret it. Calling this routine to execute JavaScript code is suggested only if you need to include js files in the code, otherwise it is prefereable to call WD2$EXECJS.

Note - if you use this routine to redirect the browser to a file created by the COBOL program (e.g. a PDF printed by the program) and you wish to specify a relative path to the file in your JavaScript, be aware that the relative path for the JavaScript doesn't always match with the relative path used by the COBOL program. When you run your programs from the IDE the relative paths match between COBOL program and JavaScript.

Instead, when you deploy your programs in a servlet container (e.g. Tomcat), JavaScript appends the relative path to the webapp main folder while the COBOL program appends them to the servlet container working directory.

Syntax:

```
CALL "WD2$RUN_JS" USING jsCode
                  GIVING returnCode
```

Parameters:

jsCode	PIC X(n)	Specifies the JavaScript code to execute. The code must be included in a <script> html tag. If you include a js file in the Java Script code, the file name in the src attribute must being with / and the js file must be placed in the webapp root folder. Example: `CALL "WD2$RUN_JS" USING "<script type=""text/javascript"" src=""/functions.js"">function1()</script>"`

Return code:

0	Operation successful.
-1	Operation failed. Possible causes: • Invalid or missing parameter • Running outside webDirect environment

Examples:

Example - Run some Java Script code (this works only on webDirect)

```
call "wd2$run_js" using
     "<script type=""text/javascript"" src=""/functions.js"">function1()</script>"
```

WD2$SESSION

The WD2$SESSION library routine manages session fields allowing to share information between JSP and webDirect. It can be used only in webDirect environment.

Syntax:

```
CALL "WD2$SESSION" USING opCode
                         parameters
                  GIVING returnCode
```

Parameters:

opCode	PIC 9	It is the function to be executed. Valid values, defined in iscobol.def, are: WD2-GET-SESSION-VALUE retrieves a session value WD2-PUT-SESSION-VALUE sets a session value
parameters	PIC X(n)	Parameters depends on the opcode.

Return code:

returnCode definition and meaning depend on the opcode.

Examples:

Example - Get the screen width (this works on webDirect only)

```
*> on working-storage copy "isgui.def"
*> g-field and g-value are pic x(n)

move "iscobol.wd2.on_client_info.screen.width" to g-field
call "wd2$session" using wd2-get-session-value
     g-field, g-value
*> display message "Screen width is " g-value
```

Example - Set the session username (this works on webDirect only)

```
*> on working-storage copy "isgui.def"
*> ws-user-name, s-field and s-value are pic x(n)

move "wd2.username" to s-field
move ws-user-name   to s-value
call "wd2$session" using wd2-put-session-value
     s-field, s-value.
```

WD2-GET-SESSION-VALUE

The WD2-GET-SESSION-VALUE function retrieves the value of a session field.

The function can be used also to retrieve screen dimensions since at the display of the initial window the following session values are set by webDirect:

- iscobol.wd2.on_client_info.desktop.width
- iscobol.wd2.on_client_info.desktop.height
- iscobol.wd2.on_client_info.screen.width
- iscobol.wd2.on_client_info.screen.height

Dimensions are expressed in pixel.

Note - Screen dimensions are available after the display of the first window. They could be not available soon after the display as they depend by an event that the ZK Framework generates in response to the creation of a window. It's good practice to wait few seconds before inquiring screen dimensions.

The function can be used also to retrieve information about the HTTP session and the servlet context behind the webDirect application. The following session values are set by webDirect:

- iscobol.wd2.servletcontext.name
- iscobol.wd2.servletcontext.realpath
- iscobol.wd2.servletcontext.path
- iscobol.wd2.servletcontext.serverinfo
- iscobol.wd2.servletcontext.majorversion
- iscobol.wd2.servletcontext.minorversion
- iscobol.wd2.httpsession.id
- iscobol.wd2.httpsession.creationtime

Syntax:

```
CALL "WD2$SESSION" USING WD2-GET-SESSION-VALUE
                         fieldName
                         fieldValue
                 GIVING returnCode
```

Parameters:

opCode	Constant	
fieldName	PIC X(n)	Specifies the name of the session field whose value must be retrieved.
fieldValue	PIC X(n)	Receives the field value.

Return code:

0	Operation successful.
-1	Running outside webDirect environment.
-2	Invalid parameters.
-3	Invalid opCode.
-4	Internal error.
-5	Internal error.
-6	Field not found.

WD2-PUT-SESSION-VALUE

The WD2-PUT-SESSION-VALUE function sets the value of a session field.

Syntax:

```
CALL "WD2$SESSION" USING WD2-PUT-SESSION-VALUE
                         fieldName
                         fieldValue
                 GIVING returnCode
```

Parameters:

opCode	Constant	
fieldName	PIC X(n)	Specifies the name of the session field whose value must be set.
fieldValue	PIC X(n)	Specifies the field value.

Return code:

0	Operation successful.
-1	Running outside webDirect environment.
-2	Invalid parameters.
-3	Invalid opCode.
-4	Internal error.
-5	Internal error.
-6	Field not found.

WIN$PLAYSOUND

The WIN$PLAYSOUND library routine plays a ".wav" or a ".aiff" file.

Syntax:

```
CALL "WIN$PLAYSOUND" USING soundName
                           soundFlags
                     GIVING returnCode
```

Parameters:

soundName	PIC X(n)	Specifies the name of the file to be played. It can be any .WAV or .AIFF file. If soundName is spaces, any sound currently playing is stopped.	
soundFlags	any numeric data item or numeric literal	It affects the way the sound is played. Valid values, defined in isgui.def, are:	
		SND-SYNC	The execution of the program is suspended for the whole duration of the sound.
		SND-ASYNC	The sound is played asynchronously.

Return code:

returnCode can be any signed numeric data item and provides additional information:

-1	The WIN$PLAYSOUND library routine is not supported.
0	Operation failed.
1	Operation successful.

Examples:

Example - Play a sound file synchronously

```
*> on working-storage copy "isgui.def"
*> snd-filename is pic x(n)

move "c:\tmp\sounds\startsound.wav" to snd-filename
call "win$playsound" using snd-filename, snd-sync
```

WIN$PRINTER

The WIN$PRINTER library routine provides a number of functions to manage printers.

The WIN$PRINTER functions can be divided in two groups.

1. Printer management functions:
   - WINPRINT-CANCEL-JOB
   - WINPRINT-GET-CURRENT-INFO
   - WINPRINT-GET-CURRENT-INFO-EX
   - WINPRINT-GET-NO-PRINTERS

- WINPRINT-GET-PRINTER-INFO
- WINPRINT-GET-PRINTER-INFO-EX
- WINPRINT-GET-PRINTER-MEDIA
- WINPRINT-SET-ATTRIBUTE
- WINPRINT-SET-PRINTER
- WINPRINT-SET-PRINTER-AS
- WINPRINT-SET-PRINTER-EX
- WINPRINT-UPDATE-PRINTERS
- WINPRINT-SETUP

2. Print job management functions:

   - WINPRINT-CLEAR-DATA-COLUMNS
   - WINPRINT-CLEAR-PAGE-COLUMNS
   - WINPRINT-GET-PAGE-LAYOUT
   - WINPRINT-GRAPH-BRUSH
   - WINPRINT-GRAPH-DRAW
   - WINPRINT-GRAPH-PEN
   - WINPRINT-PRINT-BITMAP
   - WINPRINT-SET-BACKGROUND-COLOR
   - WINPRINT-SET-CURSOR
   - WINPRINT-SET-CUSTOM-PAPER
   - WINPRINT-SET-DATA-COLUMNS
   - WINPRINT-SET-FONT
   - WINPRINT-SET-HEADER-FOOTER
   - WINPRINT-SET-JOB
   - WINPRINT-SET-LINES-PER-PAGE
   - WINPRINT-SET-MARGINS
   - WINPRINT-SET-PAGE-COLUMN
   - WINPRINT-SET-STD-FONT
   - WINPRINT-SET-TEXT-COLOR

The second group of functions requires an X11 display.

Syntax:

```
CALL "WIN$PRINTER" USING opCode
                         parameters
                   GIVING returnCode
```

Parameters:

opCode	Function to be executed. Valid values, defined in isprint.def, are:

WINPRINT-CANCEL-JOB	Cancel the print job.
WINPRINT-CLEAR-DATA-COLUMNS	Erase any column setting previously defined by the WINPRINT-SET-DATA-COLUMNS function.
WINPRINT-CLEAR-PAGE-COLUMNS	Erase any page setting previously defined by the WINPRINT-SET-PAGE-COLUMN function.
WINPRINT-GET-CAPABILITIES	Not supported, included for compatibility reasons only. This function always returns WPRTERR-UNSUPPORTED.
WINPRINT-GET-CURRENT-INFO	Retrieve information concerning the currently selected printer. A more complete set of information can be retrieved with the WINPRINT-GET-CURRENT-INFO-EX function.
WINPRINT-GET-CURRENT-INFO-EX	Retrieve complete information concerning the currently selected printer.
WINPRINT-GET-JOB-STATUS	Not supported, included for compatibility reasons only. This function always returns WPRTERR-UNSUPPORTED.
WINPRINT-GET-NO-PRINTERS	Retrieve the number of printers installed on the host system.
WINPRINT-GET-PAGE-COLUMN	Not supported, included for compatibility reasons only. This function always returns WPRTERR-UNSUPPORTED.
WINPRINT-GET-PAGE-LAYOUT	Retrieve the number of lines and columns currently available.
WINPRINT-GET-PRINTER-INFO	Retrieve information concerning a specific printer.
WINPRINT-GET-PRINTER-INFO-EX	Retrieve complete information concerning a specific printer.
WINPRINT-GET-PRINTER-MEDIA	Returns the supported trays and paper sizes.
WINPRINT-GET-PRINTER-STATUS	Not supported, included for compatibility reasons only. This function always returns WPRTERR-UNSUPPORTED.
WINPRINT-GRAPH-BRUSH	Set the fill-area of the shape drawn by WINPRINT-GRAPH-DRAW.
WINPRINT-GRAPH-DRAW	Draw a line or a box.
WINPRINT-GRAPH-PEN	Set the border of the shape drawn by WINPRINT-GRAPH-DRAW.
WINPRINT-PRINT-BITMAP	Print an image.
WINPRINT-SET-ATTRIBUTE	Set PDF file attributes.
WINPRINT-SET-BACKGROUND-COLOR	Set the text background color.
WINPRINT-SET-CURSOR	Change the cursor position.
WINPRINT-SET-CUSTOM-PAPER	Sets the paper size.
WINPRINT-SET-DATA-COLUMNS	Set the starting position of information in the print buffer for each column, starting at 1.
WINPRINT-SET-FONT	Set the font to be used by the subsequent WRITE statements.
WINPRINT-SET-HEADER-FOOTER	Print information on each page header and footer.
WINPRINT-SET-LINES-PER-PAGE	Changes the number of printable lines per page.

parameters	Parameters depend on the opcode.

Return code:

returnCode can be any signed numeric data item. The meaning depends on the opcode.

Examples:

Example - Calling win$printer with different op-codes to generate a graphical document

```
       program-id. winprinter.

       input-output section.
       file-control.
       select print-job assign to printer spooler-name
           organization line sequential.

       file section.
       FD  print-job.
       01  print-record            pic x(80).

       working-storage section.
       copy "isgui.def".
       copy "iscrt.def".
       copy "isopensave.def".
       copy "isprint.def".
       77  crt-status              special-names crt status pic 9(5).
       77  hWin                    handle of window.
       77  close-win               pic 9 value 0.

       77  hBmp                    pic s9(9) comp-4.
       77  spooler-name            pic x(128).
       77  printer-num             pic 9(3).
       77  winprint-status         pic s99.


       screen section.
       01  Mask.
           03 label
              line                  2
              col                   2
              size                  10 cells
              title                 "Printers:"
              .
           03 cb-printers
              combo-box
              line                  4
              col                   2
              size                  30 cells
              .
           03 push-button
              line                  7
              col                   2
              title                 "&Print"
              exception-value       101
              .
```

```
           03 push-button
              line                  7
              col                   + 2
              title                 "Pre&view"
              exception-value       102
              .
           03 push-button
              line                 7
              col                  + 2
              title                "PD&F"
              exception-value      103
              .

       procedure division.
       main.
           call "w$bitmap" using wbitmap-load, "files/img.png"
                           giving hBmp

           display independent graphical window
                   color  65793
                   with   system menu
                   title  "win$printer routine"
                   handle hWin
                   event  win-evt

           display Mask

           perform get-printers-names

           perform until crt-status = 27 or close-win = 1
             accept Mask
                    on exception
                       continue
             end-accept
             evaluate crt-status
                      when 101
                           perform normal-print
                      when 102
                           perform print-preview
                      when 103
                           perform print-pdf
             end-evaluate
           end-perform

           destroy Mask
           destroy hWin
           call "w$bitmap" using wbitmap-destroy, hBmp
           goback
           .


       print-preview.
```

```
           move "-p preview" to spooler-name
           perform print-procedure
           .


       print-pdf.
           initialize opensave-data, spooler-name.
           accept opnsav-default-dir from environment "user-path"
           move "PDF Files (*.pdf)|*.pdf" to opnsav-filters
           move "pdf"                     to opnsav-default-ext
           call "c$opensavebox" using opensave-save-box
                                      opensave-data
           if return-code < 0
              exit paragraph
           end-if

           string "-p pdf "       delimited by size
                  opnsav-filename delimited by trailing spaces
                                  into spooler-name

           perform PRINT-PROCEDURE
           .

       normal-print.
           move "-p spooler" to spooler-name

           initialize winprint-selection
           inquire cb-printers value winprint-name
           call "win$printer" using winprint-set-printer
                                    winprint-selection

           perform print-procedure
           .



       get-printers-names.
           modify cb-printers reset-list 1
           perform varying printer-num from 1 by 1 until 1 = 2
              initialize winprint-selection
              move printer-num to winprint-no-of-printers
              call "win$printer" using winprint-get-printer-info
                                       winprint-selection
                                giving winprint-status
              if winprint-status < 1
                 exit perform
              end-if
              modify cb-printers item-to-add winprint-name
```

```
            if wprt-is-default
                modify cb-printers value winprint-name
            end-if
        end-perform
        .


 print-procedure.
        open output print-job
      *print of bitmap pictures
        initialize wprtdata-print-bitmap
        move hBmp to wprtdata-bitmap
        move 3    to wprtdata-bitmap-row
        move 3    to wprtdata-bitmap-col
        move 5    to wprtdata-bitmap-height
        move 6    to wprtdata-bitmap-width
        move wprtbitmap-scale-centimeters to wprtdata-bitmap-flags
        add  wprtbitmap-units-centimeters to wprtdata-bitmap-flags
        call "win$printer" using winprint-print-bitmap
                                 winprint-data
      *print of colored strings (RGB = 96,106,232)
        initialize wprtdata-text-color.
        compute wprtdata-text-color = (232 * 65536) +
                                      (106 * 256)  +
                                       96
        call "win$printer" using winprint-set-text-color
                                 wprtdata-text-color
        write print-record from "colored string"
      *print of graphical shapes (how to create a table)
        initialize wprtdata-draw
        move 2  to wprtdata-draw-start-x
        move 10  to wprtdata-draw-start-y
        move 18 to wprtdata-draw-stop-x
        move 15 to wprtdata-draw-stop-y
        move wprtunits-centimeters to wprtdata-draw-units
        move wprt-draw-rectangle   to wprtdata-draw-shape
        call "win$printer" using winprint-graph-draw
                                 winprint-data
        initialize wprtdata-draw
        move 5  to wprtdata-draw-start-x
        move 10 to wprtdata-draw-start-y
        move 5  to wprtdata-draw-stop-x
        move 15 to wprtdata-draw-stop-y
        move wprtunits-centimeters to wprtdata-draw-units
        move wprt-draw-line        to wprtdata-draw-shape
        call "win$printer" using winprint-graph-draw
                                 winprint-data
        initialize wprtdata-draw
```

```
           move 2  to wprtdata-draw-start-x
           move 12 to wprtdata-draw-start-y
           move 18 to wprtdata-draw-stop-x
           move 12 to wprtdata-draw-stop-y
           move wprtunits-centimeters to wprtdata-draw-units
           move wprt-draw-line        to wprtdata-draw-shape
           call "win$printer" using winprint-graph-draw
                                    winprint-data

           close print-job
           .


       win-evt.
           if event-type = cmd-close
              move 1 to close-win
           end-if.
```

WINPRINT-CANCEL-JOB

The WINPRINT-CANCEL-JOB function cancels the current print job. The subsequent CLOSE statement on the print file will release all the job resources without printing out anything. This function affects standard print jobs as well as print preview and PDF generation. It has no effect if the printing device is SPOOLER DIRECT or any other physical device.

Note: This function should always be called between the OPEN OUTPUT and the CLOSE of the print file. If the function is called after the CLOSE of the print file, the next print job will be cancelled.

Syntax:

```
CALL "WIN$PRINTER" USING WINPRINT-CANCEL-JOB
                   GIVING returnCode
```

Parameters:

WINPRINT-CANCEL-JOB	Constant

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.
WPRTERR-BAD-ARG	The WIN$PRINTER library routine has been called with bad parameters.

WINPRINT-CLEAR-DATA-COLUMNS

The WINPRINT-CLEAR-DATA-COLUMNS function erases any column setting previously defined with the WINPRINT-SET-DATA-COLUMNS function.

Syntax:

```
CALL "WIN$PRINTER" USING WINPRINT-CLEAR-DATA-COLUMNS
                   GIVING returnCode
```

Parameters:

WINPRINT-CLEAR-DATA-COLUMNS	Constant

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.
WPRTERR-BAD-ARG	The WIN$PRINTER library routine has been called with bad parameters.

WINPRINT-CLEAR-PAGE-COLUMNS

The WINPRINT-CLEAR-PAGE-COLUMNS function erases any column setting previously defined through the WINPRINT-SET-PAGE-COLUMN function.

Syntax:

```
CALL "WIN$PRINTER" USING WINPRINT-CLEAR-PAGE-COLUMNS
                   GIVING returnCode
```

Parameters:

WINPRINT-CLEAR-PAGE-COLUMNS	Constant

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.
WPRTERR-BAD-ARG	The WIN$PRINTER library routine has been called with bad parameters.

WINPRINT-GET-CURRENT-INFO

The WINPRINT-GET-CURRENT-INFO function retrieves information concerning the currently selected printer. A more complete set of information can be retrieved through the WINPRINT-GET-CURRENT-INFO-EX function.

Syntax:

```
CALL "WIN$PRINTER" USING WINPRINT-GET-CURRENT-INFO
                         WINPRINT-SELECTION
                   GIVING returnCode
```

Parameters:

WINPRINT-GET-CURRENT-INFO	Constant	

WINPRINT-SELECTION	Group Item	Structure that receives the information. This group item, defined in isprint.def, has the following structure:
		
		`01 winprint-selection.`
		`  03 winprint-name pic x(80).`
		`  03 winprint-port pic x(80).`
		`  03 winprint-driver pic x(80).`
		`  03 winprint-drv-version signed-int.`
		`  03 winprint-no-of-printers signed-short.`
		`  88 wprterr-no-printers value -1.`
		`  03 winprint-is-default signed-short.`
		`  88 wprt-is-not-default value 0.`
		`  88 wprt-is-default value 1.`
		`  03 winprint-copies signed-short.`
		`  03 winprint-orientation signed-short.`
		`  03 winprint-quality signed-short.`
		`  03 winprint-curr-orientation signed-short.`
		`  03 winprint-curr-copies signed-short.`
		
		Note - members not mentioned below are not used by this function.
		
		winprint-name
		It is the name of the printer
		
		winprint-no-of-printers
		It is the number of the currently selected printer in the isCOBOL framework.
		
		winprint-is-default
		It is a flag indicating if the printer is the system's default printer. If it is the default printer, the wprt-is-default condition is set to true, otherwise the wprt-is-not-default is set to true.
		
		winprint-curr-copies
		Is the number of copies the printer will print with the next job.
		
		winprint-curr-orientation
		Is the current orientation: portrait or landscape.

Note - If no printer is available in the system, set *wprterr-no-printers* to true before calling this function in order to avoid errors. *winprint-curr-copies* and *winprint-curr-orientation* are returned anyway.

Note - If this function is called before WINPRINT-SETUP, it returns only the following information: winprint-name, winprint-no-of-printers, winprint-is-default, and winprint-job-title. All other fields are set to default values that may not match with the current printer settings.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.
WPRTERR-BAD-ARG	The WIN$PRINTER library routine has been called with bad parameters.

WINPRINT-GET-CURRENT-INFO-EX

The WINPRINT-GET-CURRENT-INFO-EX function retrieves information concerning the currently selected printer.

Syntax:

```
CALL "WIN$PRINTER" USING WINPRINT-GET-CURRENT-INFO-EX
                         WINPRINT-SELECTION
                   GIVING returnCode
```

Parameters:

WINPRINT-GET-CURRENT-INFO-EX	Constant

WINPRINT-SELECTION	Group Item	Structure that receives the information. This group item, defined in isprint.def, has the following structure:
		
		`01 winprint-selection.`
		`03 winprint-name pic x(80).`
		`03 winprint-port pic x(80).`
		`03 winprint-driver pic x(80).`
		`03 winprint-drv-version signed-int.`
		`03 winprint-no-of-printers signed-short.`
		`88 wprterr-no-printers value -1.`
		`03 winprint-is-default signed-short.`
		`88 wprt-is-not-default value 0.`
		`88 wprt-is-default value 1.`
		`03 winprint-copies signed-short.`
		`03 winprint-orientation signed-short.`
		`03 winprint-quality signed-short.`
		`03 winprint-curr-orientation signed-short.`
		`03 winprint-curr-copies signed-short.`
		`03 winprint-duplex signed-short.`
		`03 winprint-collate signed-short.`
		`03 winprint-color signed-short.`
		`88 wprt-has-no-color value 0.`
		`88 wprt-has-color value 1.`
		`03 winprint-curr-duplex signed-short.`
		`03 winprint-curr-collate signed-short.`
		`03 winprint-curr-papersize signed-short.`
		`03 winprint-curr-tray signed-short.`
		`03 winprint-curr-color signed-short.`
		`88 wprt-monochrome value 0.`
		`88 wprt-color value 1.`
		`03 winprint-job-title pic x(80).`

Note - members not mentioned below are not used by this function.

winprint-name
It is the name of the printer

winprint-no-of-printers
It is the number of the currently selected printer in the isCOBOL framework.

winprint-is-default
It is a flag indicating if the printer is the system's default printer. If it is the default printer, the wprt-is-default condition is set to true, otherwise the wprt-is-not-default is set to true.

winprint-color
It is a flag indicating if the printer supports colors. If color is supported, the wprt-has-color condition is set to true, otherwise the wprt-has-no-color is set to true.

winprint-curr-copies
Is the number of copies the printer will print with the next job.

winprint-curr-orientation
Is the current orientation: portrait or landscape.

winprint-curr-duplex
Is the current duplex: simplex, vertical or horizontal.

winprint-curr-collate
Is the current collate state.

winprint-curr-papersize
Is the current paper-size. Due to Java implementation, the value 9 is always returned.

winprint-curr-tray
Is the current tray. Possible values are defined in isprint.def.

winprint-curr-color
It is a flag indicating if the printer is properly set to print colors. If it is properly set, the wprt-color condition is set to true, otherwise the wprt-monochrome is set to true.

winprint-job-title
It is the name of the current job.

Note - If no printer is available in the system, set *wprterr-no-printers* to true before calling this function in order to avoid errors. *winprint-curr-copies* and *winprint-curr-orientation* are returned anyway.

Note - If this function is called before WINPRINT-SETUP, it returns only the following information: winprint-name, winprint-no-of-printers, winprint-is-default, and winprint-job-title. All other fields are set to default values that may not match with the current printer settings.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.
WPRTERR-BAD-ARG	The WIN$PRINTER library routine has been called with bad parameters.

WINPRINT-GET-NO-PRINTERS

The WINPRINT-GET-NO-PRINTERS function retrieves the number of printers installed on the host system.

Syntax:

```
CALL "WIN$PRINTER" USING WINPRINT-GET-NO-PRINTERS
                         WINPRINT-SELECTION
                   GIVING returnCode
```

Parameters:

WINPRINT-GET-NO-PRINTERS	Constant	

WINPRINT-SELECTION	Group Item	Structure that receives the information. This group item, defined in isprint.def, has the following structure:
		
		`01 winprint-selection.`
		`  03 winprint-name pic x(80).`
		`  03 winprint-port pic x(80).`
		`  03 winprint-driver pic x(80).`
		`  03 winprint-drv-version signed-int.`
		`  03 winprint-no-of-printers signed-short.`
		`  88 wprterr-no-printers value -1.`
		`  03 winprint-is-default signed-short.`
		`  88 wprt-is-not-default value 0.`
		`  88 wprt-is-default value 1.`
		`  03 winprint-copies signed-short.`
		`  03 winprint-orientation signed-short.`
		`  03 winprint-quality signed-short.`
		`  03 winprint-curr-orientation signed-short.`
		`  03 winprint-curr-copies signed-short.`
		
		Note - members not mentioned below are not used by this function.
		
		winprint-no-of-printers
		It is the number of available printers.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.
WPRTERR-BAD-ARG	The WIN$PRINTER library routine has been called with bad parameters.

WINPRINT-GET-PAGE-LAYOUT

The WINPRINT-GET-PAGE-LAYOUT function retrieves the number of printable lines and columns. The result depends on the chosen font, the page size, margins and orientation.

Syntax:

```
CALL "WIN$PRINTER" USING WINPRINT-GET-PAGE-LAYOUT
                         WINPRINT-DATA
                   GIVING returnCode
```

Parameters:

WINPRINT-GET-PAGE-LAYOUT	Constant	

WINPRINT-DATA	Group Item	Structure that receives the information. This group item, defined in isprint.def, has the following structure:
		
		`01 winprint-data.`
		`  03 wprtdata-page-layout.`
		`  05 wprtdata-lines-per-page unsigned-short.`
		`  05 wprtdata-columns-per-page unsigned-short.`
		
		wprtdata-lines-per-page
		It is the number of printable lines in a page.
		
		wprtdata-columns-per-page
		It is the number of printable columns in a page.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.
WPRTERR-BAD-ARG	The WIN$PRINTER library routine has been called with bad parameters.

WINPRINT-GET-PRINTER-INFO

The WINPRINT-GET-PRINTER-INFO function retrieves information concerning a specific printer. A more complete set of information can be retrieved through the WINPRINT-GET-PRINTER-INFO-EX function.

Syntax:

```
CALL "WIN$PRINTER" USING WINPRINT-GET-PRINTER-INFO
                         WINPRINT-SELECTION
                   GIVING returnCode
```

Parameters:

WINPRINT-GET-PRINTER-INFO	Constant

WINPRINT-SELECTION	Group Item	Structure that receives the information. This group item, defined in isprint.def, has the following structure:
		
		`01 winprint-selection.`
		`  03 winprint-name  pic x(80).`
		`  03 winprint-port  pic x(80).`
		`  03 winprint-driver  pic x(80).`
		`  03 winprint-drv-version  signed-int.`
		`  03 winprint-no-of-printers  signed-short.`
		`  88 wprterr-no-printers  value -1.`
		`  03 winprint-is-default  signed-short.`
		`  88 wprt-is-not-default  value 0.`
		`  88 wprt-is-default  value 1.`
		`  03 winprint-copies  signed-short.`
		`  03 winprint-orientation  signed-short.`
		`  03 winprint-quality  signed-short.`
		`  03 winprint-curr-orientation  signed-short.`
		`  03 winprint-curr-copies  signed-short.`
		
		Note - members not mentioned below are not used by this function.
		
		winprint-name
		It is the name of the printer
		
		When the function is called and winprint-no-of-printers is zero, the function searches for a printer with the name specified. Otherwise, it receives the name of the printer identified by winprint-no-of-printers.
		
		winprint-no-of-printers
		It is the number of the currently selected printer in the isCOBOL framework.
		
		winprint-is-default
		It is a flag indicating if the printer is the system's default printer. If it is the default printer, the wprt-is-default condition is set to true, otherwise the wprt-is-not-default is set to true.
		
		winprint-curr-copies
		Is the number of copies the printer will print with the next job.
		
		winprint-curr-orientation
		Is the current orientation: portrait or landscape.

Note - If no printer is available in the system, set *wprterr-no-printers* to true before calling this function in order to avoid errors. *winprint-curr-copies* and *winprint-curr-orientation* are returned anyway.

Note - If this function is called before WINPRINT-SETUP, it returns only the following information: winprint-name, winprint-no-of-printers, winprint-is-default, and winprint-job-title. All other fields are set to default values that may not match with the current printer settings.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.

WPRTERR-BAD-ARG	The WIN$PRINTER library routine has been called with bad parameters.

Example:

The following code snippet retrieves information for all installed printers.

```
call "WIN$PRINTER" using WINPRINT-GET-NO-PRINTERS, winprint-selection
move winprint-no-of-printers to printerCount
perform varying printerIdx from 1 by 1 until printerIdx > printerCount
   initialize winprint-selection
   move printerIdx to winprint-no-of-printers
   call "WIN$PRINTER" using WINPRINT-GET-PRINTER-INFO, winprint-selection
   display "Printer name: " winprint-name
end-perform
```

WINPRINT-GET-PRINTER-INFO-EX

The WINPRINT-GET-PRINTER-INFO-EX function retrieves information concerning a specific printer.

Syntax:

```
 CALL "WIN$PRINTER" USING WINPRINT-GET-PRINTER-INFO-EX
                          WINPRINT-SELECTION
                  GIVING returnCode
```

Parameters:

WINPRINT-GET-PRINTER-INFO-EX	Constant	
WINPRINT-SELECTION	Group Item	Structure that receives the information. This group item, defined in isprint.def, has the following structure:

```
01  winprint-selection.
    03 winprint-name              pic x(80).
    03 winprint-port              pic x(80).
    03 winprint-driver            pic x(80).
    03 winprint-drv-version       signed-int.
    03 winprint-no-of-printers    signed-short.
       88 wprterr-no-printers     value -1.
    03 winprint-is-default        signed-short.
       88 wprt-is-not-default     value 0.
       88 wprt-is-default         value 1.
    03 winprint-copies            signed-short.
    03 winprint-orientation       signed-short.
    03 winprint-quality           signed-short.
    03 winprint-curr-orientation signed-short.
    03 winprint-curr-copies       signed-short.
    03 winprint-duplex            signed-short.
    03 winprint-collate           signed-short.
    03 winprint-color             signed-short.
       88 wprt-has-no-color       value 0.
       88 wprt-has-color          value 1.
    03 winprint-curr-duplex       signed-short.
    03 winprint-curr-collate      signed-short.
    03 winprint-curr-papersize    signed-short.
    03 winprint-curr-tray         signed-short.
    03 winprint-curr-color        signed-short.
       88 wprt-monochrome         value 0.
       88 wprt-color              value 1.
    03 winprint-job-title         pic x(80).
```

Note - members not mentioned below are not used by this function.

winprint-name
It is the name of the printer.

When the function is called and winprint-no-of-printers is zero, the function searches for a printer with the name specified. Otherwise, it receives the name of the printer identified by winprint-no-of-printers.

winprint-no-of-printers
It is the number of the printer information is retrieved for. When this member is zero, the function searches for a printer with the name specified in winprint-name.

winprint-is-default
It is a flag indicating if the printer is the system's default printer. If it is the default printer, the wprt-is-default condition is set to true, otherwise the wprt-is-not-default is set to true.

winprint-curr-copies
Is the number of copies the printer will print with the next job.

	winprint-color It is a flag indicating if the printer supports colors. If color is supported, the wprt-has-color condition is set to true, otherwise the wprt-has-no-color is set to true. *winprint-curr-orientation* Is the current orientation: portrait or landscape. *winprint-curr-collate* Is the current collate state. *winprint-curr-duplex* Is the current duplex setting: simplex, vertical or horizontal. *winprint-curr-papersize* Is the current paper-size. See List of supported papersizes for a list of possible values. *winprint-curr-tray* Is the current tray. Possible values are defined in isprint.def. *winprint-curr-color* It is a flag indicating if the printer is properly set to print colors. If it is properly set, the wprt-color condition is set to true, otherwise the wprt-monochrome is set to true. *winprint-job-title* It is the name of the printer job.

Note - If no printer is available in the system, set *wprterr-no-printers* to true before calling this function in order to avoid errors. *winprint-curr-copies* and *winprint-curr-orientation* are returned anyway.

Note - If this function is called before WINPRINT-SETUP, it returns only the following information: winprint-name, winprint-no-of-printers, winprint-is-default, and winprint-job-title. All other fields are set to default values that may not match with the current printer settings.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.
WPRTERR-BAD-ARG	The WIN$PRINTER library routine has been called with bad parameters.

Example:

The following code snippet retrieves information for all installed printers.

```
call "WIN$PRINTER" using WINPRINT-GET-NO-PRINTERS, winprint-selection
move winprint-no-of-printers to printerCount
perform varying printerIdx from 1 by 1 until printerIdx > printerCount
   initialize winprint-selection
   move printerIdx to winprint-no-of-printers
   call "WIN$PRINTER" using WINPRINT-GET-PRINTER-INFO-EX, winprint-selection
   display "Printer name: " winprint-name
end-perform
```

WINPRINT-GET-PRINTER-MEDIA

The WINPRINT-GET-PRINTER-MEDIA function allows you to access the paper sizes and paper trays supported by the printer.

Syntax

```
CALL "WIN$PRINTER" USING WINPRINT-GET-PRINTER-MEDIA
                         WINPRINT-MEDIA
                   GIVING returnCode
```

Parameters

WINPRINT-GET-PRINTER-MEDIA	Constant	

WINPRINT-MEDIA	Group Item	Structure defined in isprint.def as follows:
		
		`01 winprint-media.`
		`  03 winprint-media-printer pic x(80).`
		`  03 winprint-media-port pic x(80).`
		`  03 winprint-media-papercount signed-short.`
		`  03 winprint-media-traycount signed-short.`
		`  03 winprint-media-paper signed-short occurs max-paper-sizes.`
		`  03 winprint-media-trays signed-short occurs max-paper-trays.`
		
		winprint-media-printer
		It must be set to the value of WINPRINT-NAME as obtained through a call to WINPRINT-GET-PRINTER-INFO(-EX) or WINPRINT-GET-CURRENT-INFO(-EX).
		
		winprint-media-port
		It must be set to the value of WINPRINT-PORT as obtained through a call to WINPRINT-GET-PRINTER-INFO(-EX) or WINPRINT-GET-CURRENT-INFO(-EX).
		
		winprint-media-papercount
		Returns the total number of paper sizes supported by the selected printer driver. This number varies from printer to printer.
		
		winprint-media-traycount
		Returns the total number of paper trays supported by the printer driver. This number varies from printer to printer.
		
		winprint-media-paper
		Returns an array of supported paper sizes. The array is limited to a maximum of *max-paper-sizes* possible sizes. Each number in the array corresponds to a paper size defined by WINPRINT-CURR-PAPERSIZE in *isprint.def*. The numbers in the array may not appear in sequential order .
		
		winprint-media-trays
		Returns an array of supported paper trays. The array is limited to a maximum of *max-paper-trays* possible trays. Each number in the array corresponds to a paper tray defined by WINPRINT-CURR-TRAY in *isprint.def*. The numbers in the array may not appear in sequential order .

Note: The information returned by WINPRINT-GET-PRINTER-MEDIA may be incomplete because Java implements only some of the possible media paper and media tray, in particular the ones defined by the IPP (Internet Printing Protocol). However in the dialog box for the printer selection you could see more formats than the ones isCOBOL is able to handle, because, depending on the OS, special classes are supplied in order

to cope with all the formats handled by the OS. So it could happen that, for example, there are four trays available but two of them are identified by the number 0 (unknown), meaning that those two trays are handled only on that operating system.

Return code

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.
WPRTERR-BAD-ARG	The WIN$PRINTER library routine has been called with bad parameters.

WINPRINT-GRAPH-BRUSH

The WINPRINT-GRAPH-BRUSH function sets a pattern and a color to fill an area before calling the WINPRINT-GRAPH-DRAW function.

Syntax:

```
CALL "WIN$PRINTER" USING WINPRINT-GRAPH-BRUSH
                         WINPRINT-DATA
                   GIVING returnCode
```

Parameters:

WINPRINT-GRAPH-BRUSH	Constant	
WINPRINT-DATA	Group Item	Structure that contains the information to be passed to the function. This group item, defined in isprint.def, has the following structure:

```
01  winprint-data.
    03 wprtdata-brush.
       05 wprtdata-brush-style      unsigned-short.
       05 wprtdata-brush-color      pic 9(9) comp-5.
```

wprtdata-brush-style
Specifies the pattern type used to fill an area. Valid values are:

WPRT-BRUSH-SOLID	It defines a solid pattern. As a result, the area will be filled with the color defines in wprtdata-brush-color.
WPRT-BRUSH-NULL	No patter is defined and the area will appear transparent.
WPRT-BRUSH-BDIAGONAL	It defines a pattern of 45 degrees angled lines (//////).
WPRT-BRUSH-CROSS	It defines a pattern of crosses (++++++).
WPRT-BRUSH-DIAGCROSS	It defines a pattern of diagonal crosses (xxxxxxx).
WPRT-BRUSH-FDIAGONAL	It defines a pattern of 315 degrees angled lines (\\\\\\\\).
WPRT-BRUSH-HORIZONTAL	It defines a pattern of horizontal dashes (--------).
WPRT-BRUSH-VERTICAL	It defines a pattern of vertical bars (|||||||||).
WPRT-BRUSH-DKGRAY	Colors the area with dark gray.
WPRT-BRUSH-GRAY	Colors the area with gray.
WPRT-BRUSH-LTGRAY	Colors the area with light gray.

wprtdata-brush-color
Specifies the RGB color used to fill the area in the following hexadecimal form: 0x00BBGGRR.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.
WPRTERR-BAD-ARG	The WIN$PRINTER library routine has been called with bad parameters.

WINPRINT-GRAPH-DRAW

The WINPRINT-GRAPH-DRAW function draws a line or a box. Shapes are drawn using the current pen and brush settings. Use the WINPRINT-GRAPH-BRUSH and WINPRINT-GRAPH-PEN functions to change them.

Syntax:

```
CALL "WIN$PRINTER" USING WINPRINT-GRAPH-DRAW
                         WINPRINT-DATA
                   GIVING returnCode
```

Parameters:

WINPRINT-GRAPH-DRAW	Constant	

WINPRINT-DATA	Group Item	Structure that contains the information to be passed to the function. This group item, defined in isprint.def, has the following structure:
		
		`01 winprint-data.`
		`  03 wprtdata-draw.`
		`  05 wprtdata-draw-start-x  pic 9(7)v99 comp-5.`
		`  05 wprtdata-draw-start-y  pic 9(7)v99 comp-5.`
		`  05 wprtdata-draw-stop-x  pic 9(7)v99 comp-5.`
		`  05 wprtdata-draw-stop-y  pic 9(7)v99 comp-5.`
		`  05 wprtdata-draw-units  unsigned-short.`
		`  05 wprtdata-draw-shape  unsigned-short.`
		
		wprtdata-draw-start-x
		Specifies the horizontal coordinate of the top-left corner of the shape. The position refers to the physical left margin.
		
		wprtdata-draw-start-y
		Specifies the vertical coordinate of the top-left corner of the shape. The position refers to the physical top margin.
		
		wprtdata-draw-stop-x
		Specifies the horizontal coordinate of the lower-right corner of the shape.
		
		wprtdata-draw-stop-y
		Specifies the vertical coordinate of the lower-right corner of the shape.
		
		wprtdata-draw-units
		Specifies the unit of measure for the drawing coordinates. Valid values are:
		
		WPRTUNITS-CELLS — Values are expressed in cells.
		
		WPRTUNITS-INCHES — Values are expressed in inches.
		
		WPRTUNITS-CENTIMETERS — Values are expressed in centimeters.
		
		WPRTUNITS-PIXELS — Values are expressed in pixels.
		
		wprtdata-draw-shape
		Specifies the shape to be drawn. Valid values are:
		
		WPRT-DRAW-RECTANGLE — Draws a rectangle with square corners.
		
		WPRT-DRAW-ROUND-RECTANGLE — Draws a rectangle with rounded corners.
		
		WPRT-DRAW-LINE — Draws a line.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.
WPRTERR-BAD-ARG	The WIN$PRINTER library routine has been called with bad parameters.

WINPRINT-GRAPH-PEN

The WINPRINT-GRAPH-PEN function sets the appearance of the line used to draw the border of a shape before calling the WINPRINT-GRAPH-DRAW function.

Syntax:

```
CALL "WIN$PRINTER" USING WINPRINT-GRAPH-PEN
                         WINPRINT-DATA
                   GIVING returnCode
```

Parameters:

WINPRINT-GRAPH-PEN	Constant	
WINPRINT-DATA	Group Item	Structure that contains the information to be passed to the function. This group item, defined in isprint.def, has the following structure:

```
01  winprint-data.
    03 wprtdata-pen.
       05 wprtdata-pen-style        unsigned-short.
       05 wprtdata-pen-width        unsigned-short.
       05 wprtdata-pen-color        pic 9(9) comp-5.
```

wprtdata-pen-style
Specifies the line style. Valid values are:

WPRT-PEN-SOLID	A solid line is drawn.
WPRT-PEN-DASH	A dashed line is drawn (-------).
WPRT-PEN-DOT	A dotted line is drawn (.....).
WPRT-PEN-DASHDOT	A line composed of dashes and dots is drawn (-.-.-.-.)
WPRT-PEN-DASHDOTDOT	A line composed of one dash and two dots is drawn (-..-..-..)
WPRT-PEN-NULL	The line is not visible. Background stays visible.
WPRT-PEN-INSIDEFRAME	A solid line is drawn inside the area of the shape.

wprtdata-pen-width
It specifies, in pixels, the width of the line. Note that actual pixel size depends on the printer resolution.

wprtdata-pen-color
It specifies RGB color of the line in the following hexadecimal form: 0x00BBGGRR.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.
WPRTERR-BAD-ARG	The WIN$PRINTER library routine has been called with bad parameters.

WINPRINT-PRINT-BITMAP

The WINPRINT-PRINT-BITMAP function prints an image.

If you need to print text over the bitmap, do it after this call. If you do it before, the bitmap covers the text.

Syntax:

```
CALL "WIN$PRINTER" USING WINPRINT-PRINT-BITMAP
                         WINPRINT-DATA
                   GIVING returnCode
```

Parameters:

WINPRINT-PRINT-BITMAP	Constant	
WINPRINT-DATA	Group Item	Structure that contains the information to be passed to the function. This group item, defined in isprint.def, has the following structure:

```
01  winprint-data.
    03 wprtdata-print-bitmap.
       05 wprtdata-bitmap           pic x(4) comp-n.
       05 wprtdata-bitmap-row       pic 9(7)v99 comp-5.
       05 wprtdata-bitmap-col       pic 9(7)v99 comp-5.
       05 wprtdata-bitmap-height    pic 9(7)v99 comp-5.
       05 wprtdata-bitmap-width     pic 9(7)v99 comp-5.
       05 wprtdata-bitmap-flags     unsigned-short.
```

wprtdata-bitmap
Specifies the handle of the bitmap to be printed.
Do not destroy this handle before closing the print file.

wprtdata-bitmap-row
Specifies the vertical coordinate of top-left corner of the bitmap, in cells. The position refers to the physical left margin.

wprtdata-bitmap-col
Specifies the horizontal coordinate of top-left corner of the bitmap, in cells. The position refers to the physical left margin.

wprtdata-bitmap-height
Specifies the height of the printed image. If this member is zero, the height is automatically calculated so that the aspect ratio is kept.

wprtdata-bitmap-width
Specifies the width of the printed image. If this member is zero, the width is automatically calculated so that the aspect ratio is kept.

wprtdata-bitmap-flags
Specifies the units of measure for location (wprtdata-bitmap-row and wprtdata-bitmap-col) and dimensions (wprtdata-bitmap-width and wprtdata-bitmap-height) of the bitmap. The value is the sum of location unit and scale unit.

Scale unit can be one of the following:

0	Values are expressed in cells.
WPRTBITMAP-SCALE-CELLS	Values are expressed in cells.
WPRTBITMAP-SCALE-INCHES	Values are expressed in inches.
WPRTBITMAP-SCALE-CENTIMETERS	Values are expressed in centimeters.
WPRTBITMAP-SCALE-PIXELS	Values are expressed in pixels.

Location unit can be one of the following:

0	Values are expressed in cells.
WPRTBITMAP-UNITS-INCHES	Values are expressed in inches.
WPRTBITMAP-UNITS-CENTIMETERS	Values are expressed in centimeters.
WPRTBITMAP-UNITS-PIXELS	Values are expressed in pixels.

To inform the runtime that the bitmap was designed directly for printing on the current printer and should not be scaled, set WPRTDATA-BITMAP-FLAGS to WPRTBITMAP-PRINTER-BITMAP.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.
WPRTERR-BAD-ARG	The WIN$PRINTER library routine has been called with bad parameters.

WINPRINT-SET-ATTRIBUTE

The WINPRINT-SET-ATTRIBUTE function sets attributes for the next PDF created by writing on a file assigned to "-P PDF" or by using the *Save As* function in the print preview dialog.

This function overrides the default settings set in the configuration (see Print Configuration).

Note: The attributes are set only for the first PDF print performed after the call to this op-code. If you need to set the attributes also for other PDF print jobs, then you need to call the op-code before each one of them.

Syntax:

```
CALL "WIN$PRINTER" USING WINPRINT-SET-ATTRIBUTE
                         attributeName
                         attributeValue
                   GIVING returnCode
```

Parameters:

WINPRINT-SET-ATTRIBUTE	Constant	
attributeName	PIC X(n)	Identifies the attribute to set. Valid values are listed below.
attributeValue	PIC X(n)	Value for the attribute

List of supported attributes:

Attribute	**Description and possible values**
AUTHOR	The author of the PDF document. It can be any text.

Attribute	Description and possible values

ENCRYPTION	Numeric bitwise value where each bit sets a specific feature.
	
	You can rely on the following data-items, defined in isprint.def, to activate the desired feature:
	`77 pdfcrypt-type pic 9(9) value 0.`
	`78 pdfcrypt-no value 0.`
	`78 pdfcrypt-std-40 value 1.`
	`78 pdfcrypt-std-128 value 2.`
	`78 pdfcrypt-aes-128 value 3.`
	`78 pdfcrypt-no-metadata value x#08.`
	`78 pdfcrypt-embedded-files-only value x#10.`
	
	`78 pdfcrypt-allow-printing value x#0100.`
	`78 pdfcrypt-allow-modify-content value x#0200.`
	`78 pdfcrypt-allow-copy value x#0400.`
	`78 pdfcrypt-allow-modify-annotations value x#0800.`
	`78 pdfcrypt-allow-fill-in value x#1000.`
	`78 pdfcrypt-allow-screenreaders value x#2000.`
	`78 pdfcrypt-allow-assembly value x#4000.`
	`78 pdfcrypt-allow-degraded-printing value x#8000.`
	`78 pdfcrypt-all-permissions value x#FF00.`
	
	If this value is set to 0 then no encryption takes place.
	Permissions are applied only if combined with a valid encryption, otherwise *all-permissions* is assumed.
	
	Usage example:
	
	`add pdfcrypt-std-128`
	`pdfcrypt-allow-printing`
	`giving`
	`pdfcrypt-type`
	`call "win$printer" using winprint-set-attribute`
	`"ENCRYPTION" pdfcrypt-type`
	
	The resulting PDF will be printable, but it will not be possible to add annotations or copy the text to clipboard.

EXPIRES	The custom property "Expires". It can be any text.
FONT_FOLDER	The folders where the fonts used in the PDF document are installed. You can specify multiple folders separated by pipe, e.g. "C:\myCustomFonts|C:\WINDOWS\Fonts". The fonts loaded from these folders are not marked as "embedded". The following rule applies to fonts loaded via W$CREATEFONT: if the font referenced in the print job is not installed in the system, only the specific TTF file loaded by the routine is included in the PDF. If the font is installed in the system, instead, then also alternative versions of the font may be included in the PDF in order to render bold and italic styles.

Attribute	Description and possible values
FONT_FOLDER_EMBED	The folders where the fonts used in the PDF document are installed. You can specify multiple folders separated by pipe, e.g. "C:\myCustomFonts|C:\WINDOWS\Fonts". The fonts loaded from these folders are marked as "embedded". The following rule applies to fonts loaded via W$CREATEFONT: if the font referenced in the print job is not installed in the system, only the specific TTF file loaded by the routine is included in the PDF. If the font is installed in the system, instead, then also alternative versions of the font may be included in the PDF in order to render bold and italic styles.
JPEG	The compression applied to images in the PDF document. It can be "0" if you want to keep images unchanged (default) or it can range from "1" to "100" to indicate the image quality, where "1" is the lowest quality and "100" is the highest quality. When this attribute is set, all images are internally translated to jpeg; this will remove transparency, if any.
KEYWORDS	The keywords of the PDF document. It can be any text.
OWNER_PASSWORD	The password of the owner of the document. If this value is not set, then a random password is created. It works only along with ENCRYPTION.
PDFA	Creates a PDF/A document following a specific standard. Possible values are "PDF/A-1A" and "PDF/A-1B", case insensitive. This attribute must be set in conjunction with either FONT_FOLDER or FONT_FOLDER_EMBED as all the fonts must be available.
SUBJECT	The subject of the PDF document, it can be any text.
TITLE	The title of the PDF document, it can be any text.
USER_PASSWORD	The password of the user of the document. If this value is not set, then a default password is used as specified in the PDF specifics. It works only along with ENCRYPTION.

Note - if neither FONT_FOLDER nor FONT_FOLDER_EMBED are set, the PDF file will not use the fonts you specified through WINPRINT-SET-FONT.

Return Code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.
WPRTERR-BAD-ARG	The WIN$PRINTER library routine has been called with bad parameters.

WINPRINT-SET-BACKGROUND-COLOR

The WINPRINT-SET-BACKGROUND-COLOR function sets the text background color.

Syntax:

```
CALL "WIN$PRINTER" USING WINPRINT-SET-BACKGROUND-COLOR
                         WINPRINT-DATA
                   GIVING returnCode
```

Parameters:

WINPRINT-SET-BACKGROUND-COLOR	Constant	

WINPRINT-DATA	Group Item	Structure that contains the information to be passed to the function. This group item, defined in isprint.def, has the following structure:
		
		`01  winprint-data.`
		`    03 wprtdata-text-color          pic 9(9) comp-5.`
		
		wprtdata-text-color
		Specifies the color to be used, in RGB format.
		
		Color value is computed as follows: *Red* + *Green* * 256 + *Blue* * 65536. *Red*, *Green* and *Blue* range from 0 to 255.
		
		A more convenient way to define the color is to use the hexadecimal notation X#RRGGBB.
		
		To print a light gray text you may use one of the following, they are equivalent:
		
		`compute wprtdata-text-color = 192 * 65536 + 192 * 256 + 192`
		
		or
		
		`move X#C0C0C0 to wprtdata-text-color`
		
		For the most common colors, the following constants can be used:
		`78 wprt-color-black value x#000000.`
		`78 wprt-color-red value x#0000FF.`
		`78 wprt-color-green value x#00FF00.`
		`78 wprt-color-blue value x#FF0000.`
		`78 wprt-color-yellow value x#00FFFF.`
		`78 wprt-color-magenta value x#FF00FF.`
		`78 wprt-color-cyan value x#FFFF00.`
		`78 wprt-color-white value x#FFFFFF.`
		
		Once the background color is set, it remains in use until the next background setting; it can be reset by setting the white color as background (x#FFFFFF or 16777215).
		
		Trailing spaces will not have a background color except when a WRITE WITH NO CONVERSION is used.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.
WPRTERR-BAD-ARG	The WIN$PRINTER library routine has been called with bad parameters.

WINPRINT-SET-CURSOR

The WINPRINT-SET-CURSOR function changes the cursor position in the current report page.

The subsequent WRITE statement must specify the WITH NO CONTROL clause.

Syntax:

```
CALL "WIN$PRINTER" USING WINPRINT-SET-CURSOR
                         WINPRINT-DATA
                   GIVING returnCode
```

Parameters:

WINPRINT-SET-CURSOR	Constant	
WINPRINT-DATA	Group Item	Structure that contains the cursor position to be set and returns the previous cursor position. This group item, defined in isprint.def, has the following structure:

```
01  winprint-data.
    03 wprtdata-draw.
       05 wprtdata-draw-start-x     pic 9(7)v99 comp-5.
       05 wprtdata-draw-start-y     pic 9(7)v99 comp-5.
       05 wprtdata-draw-stop-x      pic 9(7)v99 comp-5.
       05 wprtdata-draw-stop-y      pic 9(7)v99 comp-5.
       05 wprtdata-draw-units       unsigned-short.
       05 wprtdata-draw-shape       unsigned-short.
```

wprtdata-draw-start-x
Specifies the horizontal coordinate of the cursor location. The position refers to the physical left margin.

wprtdata-draw-start-y
Specifies the vertical coordinate of the cursor location. The position refers to the physical top margin.

wprtdata-draw-stop-x
It returns the horizontal coordinate of the current cursor location. The position refers to the physical left margin.

wprtdata-draw-stop-y
It returns the vertical coordinate of the current cursor location. The position refers to the physical top margin.

wprtdata-draw-units
Specifies the unit of measure for the coordinates. Valid values are:

WPRTUNITS-CELLS	Values are expressed in cells.
WPRTUNITS-INCHES	Values are expressed in inches.
WPRTUNITS-CENTIMETERS	Values are expressed in centimeters.
WPRTUNITS-PIXELS	Values are expressed in pixels.

wprtdata-draw-shape
It is a flag that specifies if the cursor position must be changed or not.

When it is set to zero, the function will change the cursor position.

When it is set to a non-zero value, the function will not change the cursor position. Use this setting to retrieve the current cursor location.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.
WPRTERR-BAD-ARG	The WIN$PRINTER library routine has been called with bad parameters.

WINPRINT-SET-CUSTOM-PAPER

The WINPRINT-SET-CUSTOM-PAPER function defines the paper dimensions.

The custom paper format must be set on the printer as the default format.

Syntax:

```
CALL "WIN$PRINTER" USING WINPRINT-SET-CUSTOM-PAPER
                         paperWidth
                         paperHeight
                         units
                   GIVING returnCode
```

Parameters:

WINPRINT-SET-CUSTOM-PAPER	Constant	
paperWidth	any numeric data item or numeric literal	Specifies the paper width.
paperHeight	any numeric data item or numeric literal	Specifies the paper height.
units	any numeric data item or numeric literal	Specifies the measurement unit for the *paperWidth* and *paperHeight* values. Valid values are 1 (inches) and 2 (centimeters). The constants WPRTUNITS-INCHES and WPRTUNITS-CENTIMETERS defined in isprint.def can be used.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.
WPRTERR-BAD-ARG	The WIN$PRINTER library routine has been called with bad parameters.

WINPRINT-SET-DATA-COLUMNS

The WINPRINT-SET-DATA-COLUMNS function defines the way the buffer is split into columns. When a page is divided into columns (see the WINPRINT-SET-PAGE-COLUMN function) the program can use a single WRITE statement to print the content of all columns at a time. You can call this function specifying all character positions or call it repeatedly specifying one character position at a time. Each time this function is called, it adds the values passed to the list that will be used to split information when the WRITE statement is executed. To clear that list, use the WINPRINT-CLEAR-DATA-COLUMNS function.

Syntax:

```
CALL "WIN$PRINTER" USING WINPRINT-SET-DATA-COLUMNS
                         { charPosition } ...
                   GIVING returnCode
```

Parameters:

WINPRINT-SET-DATA-COLUMNS	Constant	
charPosition	any numeric data item or numeric literal	Contains the starting position of a character in the print buffer. If the print buffer is a non-national group item, the most convenient way is to use the RECORD-POSITION syntax, instead of computing values manually.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.
WPRTERR-BAD-ARG	The WIN$PRINTER library routine has been called with bad parameters.

WINPRINT-SET-FONT

The WINPRINT-SET-FONT function sets the font to be used by the subsequent WRITE statements.

Syntax:

```
CALL "WIN$PRINTER" USING WINPRINT-SET-FONT
                         WINPRINT-DATA
                   GIVING returnCode
```

Parameters:

WINPRINT-SET-FONT	Constant	

WINPRINT-DATA	Group Item	Structure that contains the information to be passed to the function. This group item, defined in isprint.def, has the following structure:
		
		`01 winprint-data.`
		`  03 wprtdata-set-font.`
		`  05 wprtdata-font handle of font.`
		
		wprtdata-font
		Specifies the font handle to be used by the subsequent WRITE statements. It should be previously loaded with the WFONT-GET-FONT or the WFONT-GET-CLOSEST-FONT function.
		Do not destroy this handle before closing the print file.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.

WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.
WPRTERR-BAD-ARG	The WIN$PRINTER library routine has been called with bad parameters.

Warning:

For fonts to work correctly, the -P PDF and -P PREVIEW "Save" feature require setting either the FONT_FOLDER or FONT_FOLDER_EMBED printer attributes using WINPRINT-SET-ATTRIBUTE opcode. For example:

```
call "win$printer" using winprint-set-attribute
                   "FONT_FOLDER_EMBED", "C:\WINDOWS\fonts"
```

WINPRINT-SET-HEADER-FOOTER

The WINPRINT-SET-HEADER-FOOTER function specifies the information to be printed on every page header and footer.

Syntax:

```
 CALL "WIN$PRINTER" USING WINPRINT-SET-HEADER-FOOTER
                          headerContent
                          footerContent
                         [font]
                    GIVING returnCode
```

Parameters:

WINPRINT-SET-HEADER-FOOTER	Constant	
headerContent	PIC X(n)	Specifies the text to be printed on the page header. The following escape characters can be used inside this text: &p = page number; &P = total number of pages; &b = the following information will be printed on the right side of the sheet; &d = current date in short format according to the locale, e.g. 4/16/15; &D = current date in long format according to the locale, e.g. April 16, 2015; &u = name of the report; &w = not handled; && = the character '&'
footerContent	PIC X(n)	Specifies the text to be printed on the page footer. The following escape characters can be used inside this text: &p = page number; &P = total number of pages; &b = the following information will be printed on the right side of the sheet; &d = current date in short format according to the locale, e.g. 4/16/15; &D = current date in long format according to the locale, e.g. April 16, 2015; &u = name of the report; &w = not handled; && = the character '&'

font	USAGE HANDLE OF FONT	Optional parameter. Specifies the font handle to be used for the header and footer text. It should be previously loaded with the WFONT-GET-FONT or the WFONT-GET-CLOSEST-FONT function. Do not destroy this handle before closing the print file.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.
WPRTERR-BAD-ARG	The WIN$PRINTER library routine has been called with bad parameters.

Example:

The following code will print page count information on the bottom right of each page.

```
move "&bPage &p of &P" to footer-data.
call "win$printer"  using winprint-set-header-footer
                          header-data, footer-data.
```

WINPRINT-SET-JOB

The WINPRINT-SET-JOB function sets or returns the identifier of the job that is currently spooling into the printer.

Syntax:

```
CALL "WIN$PRINTER" USING WINPRINT-SET-JOB
                         jobID
                   GIVING returnCode
```

Parameters:

WINPRINT-SET-JOB	Constant	
jobID	SIGNED-INT	If set to 0, the function returns the job ID of the current print job. If set to a value greater than zero, the function causes the next calls to WIN$PRINTER to affect the print job identified by *jobID*.

Return code:

returnCode can be any signed numeric data item and provides additional information:

>0	Operation successful. It contains the current jobID if the *jobID* parameter was zero.
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.

WPRTERR-BAD-ARG	The WIN$PRINTER library routine has been called with bad parameters.

Example:

Produce two print jobs simoultaneously using different fonts and margins for each job.

```
      input-output section.
       file-control.
           select file1 assign to print "-p pdf print1.pdf"
                 organization is sequential.
           select file2 assign to print "-p pdf print2.pdf"
                 organization is sequential.
       data division.
       file section.
       fd  file1.
       01  rec-file1 pic x(90).
       fd  file2.
       01  rec-file2 pic x(90).
       working-storage section.
           copy "isprint.def".
           copy "fonts.def".
       77  ind  pic 99.
       77  job1-id usage signed-int.
       77  job2-id usage signed-int.
       77  h-font1 usage handle of font.
       77  h-font2 usage handle of font.
       procedure division.
       main-logic.
           initialize wfont-data.
           set wfdevice-win-printer to true.
           move "Courier New" to wfont-name.
           move 10 to wfont-size.
           call   "w$font" using wfont-get-font,
                                 h-font1, wfont-data
           initialize wfont-data.
           set wfdevice-win-printer to true.
           move "times new roman" to wfont-name.
           move 10 to wfont-size.
           call   "w$font" using wfont-get-font,
                                 h-font2, wfont-data

           open output file1
           call "win$printer" using winprint-set-job 0
                            giving job1-id.

           open output file2
           call "win$printer" using winprint-set-job 0
                            giving job2-id.

           call "win$printer" using winprint-set-job job1-id.

           initialize wprtdata-set-font.
           move h-font1 to wprtdata-font.
           call "win$printer" using winprint-set-font,
                                    winprint-data
```

```
          initialize wprtdata-margins
           move 5  to wprtdata-top-margin
                      wprtdata-bottom-margin
                      wprtdata-left-margin
                      wprtdata-right-margin
           move wprtmargin-centimeters to wprtdata-margin-units
           call "win$printer" using winprint-set-margins
                                    winprint-data

           call "win$printer" using winprint-set-job job2-id.

           initialize wprtdata-set-font.
           move h-font2 to wprtdata-font.
           call "win$printer" using winprint-set-font,
                                     winprint-data
           initialize wprtdata-margins
           move 10  to wprtdata-top-margin
                       wprtdata-bottom-margin
                       wprtdata-left-margin
                       wprtdata-right-margin
           move wprtmargin-centimeters to wprtdata-margin-units
           call "win$printer" using winprint-set-margins
                                    winprintdata
           perform varying ind from 1 by 1 until ind > 10
              initialize rec-file1
              string "out1 row" ind delimited by size into rec-file1
              write rec-file1
              initialize rec-file2
              string "out2 row" ind delimited by size into rec-file2
              write rec-file2
           end-perform
           close file1
           close file2
           goback
           .
```

WINPRINT-SET-LINES-PER-PAGE

The WINPRINT-SET-LINES-PER-PAGE function changes the number of printable lines per page. The lines per page currently available can be retrieved with the WINPRINT-GET-PAGE-LAYOUT function. Note that character size is not changed, this function only changes the line spacing.

Syntax:

```
CALL "WIN$PRINTER" USING WINPRINT-SET-LINES-PER-PAGE
                         WINPRINT-DATA
                   GIVING returnCode
```

Parameters:

WINPRINT-SET-LINES-PER-PAGE	Constant

WINPRINT-DATA	Group Item	Structure that contains the information to be passed to the function. This group item, defined in isprint.def, has the following structure:
		
		`01 winprint-data.`
		`  05 wprtdata-lines-per-page unsigned-short.`
		`  05 wprtdata-columns-per-page unsigned-short.`
		
		Note - members not mentioned below are not used by this function.
		
		wprtdata-lines-per-page
		Specifies the number of lines per page.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.
WPRTERR-BAD-ARG	The WIN$PRINTER library routine has been called with bad parameters.

WINPRINT-SET-MARGINS

The WINPRINT-SET-MARGINS function changes the page margins. Margins are applied at the close of the print file and affect all the pages of the print job. If this function is called multiple times within the same print job, only margins set by the last call are considered.

Syntax:

```
CALL "WIN$PRINTER" USING WINPRINT-SET-MARGINS
                         WINPRINT-DATA
                 GIVING returnCode
```

Parameters:

WINPRINT-SET-MARGINS	Constant	
WINPRINT-DATA	Group Item	Structure that contains the information to be passed to the function. This group item, defined in isprint.def, has the following structure:

```
01  winprint-data.
    03 wprtdata-margins.
       05 wprtdata-top-margin       pic 9(7)v99 comp-5.
       05 wprtdata-bottom-margin    pic 9(7)v99 comp-5.
       05 wprtdata-left-margin      pic 9(7)v99 comp-5.
       05 wprtdata-right-margin     pic 9(7)v99 comp-5.
       05 wprtdata-margin-units     unsigned-short.
```

wprtdata-top-margin
Specifies the top margin.

wprtdata-bottom-margin
Specifies the bottom margin.

wprtdata-left-margin
Specifies the left margin.

wprtdata-right-margin
Specifies the right margin.

wprtdata-margin-units
Specifies the unit of measure of the margins. Valid values are:

WPRTMARGIN-DEFAULT-MARGINS	The function resets the default printer margins. Other members are ignored.
WPRTMARGIN-INCHES	Margins are expressed in inches.
WPRTMARGIN-CENTIMETERS	Margins are expressed in centimeters.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.
WPRTERR-BAD-ARG	The WIN$PRINTER library routine has been called with bad parameters.

WINPRINT-SET-PAGE-COLUMN

The WINPRINT-SET-PAGE-COLUMN function defines a column in a page. When a page is divided into columns the program can use a single WRITE statement to print the content of all columns at a time. The content of the print buffer is placed into columns according to the settings defined with the WINPRINT-SET-DATA-COLUMNS function. This function must be called repeatedly specifying one column configuration at a time. Each time this function is called, it adds the column configuration passed to a list. To clear that list, use the WINPRINT-CLEAR-PAGE-COLUMNS function.

Syntax:

```
CALL "WIN$PRINTER" USING WINPRINT-SET-PAGE-COLUMN
                         winprintColumn
                   GIVING returnCode
```

Parameters:

WINPRINT-SET-PAGE-COLUMN	Constant	
WINPRINT-DATA	Group Item	Structure that contains the information to be passed to the function. This group item, defined in isprint.def, has the following structure:

```
01  winprint-column.
    03 winprint-col-start      pic 9(7)v99 comp-5.
    03 winprint-col-indent     pic 9(7)v99 comp-5.
    03 winprint-col-separation pic 9(7)v99 comp-5.
    03 winprint-col-font       handle of font.
    03 winprint-col-units      pic 99 comp-x.
    03 winprint-col-alignment  pic x.
    03 winprint-transparency   pic 99 comp-x.
       88 winprint-transparent value 1, false 0.
```

winprint-col-start
It defines the starting position of the column. The starting position of a column determines the ending position of the previous one. The last column extends to the right page margin. Text not fitting the column space will be truncated.

winprint-col-indent
It defines the amount of blank space to be left between the starting position of the column and the printed text.

winprint-col-separation
It defines the amount of blank space to be left between printed text and the ending position of the column.

winprint-col-font
Specifies the font handle to be used to print the column content.

winprint-col-units
Specifies the unit of measure of winprint-col-start, winprint-col-indent and winprint-col-separation. Valid values are:

WPRTUNITS-CELLS	Values are expressed in cells.
WPRTUNITS-INCHES	Values are expressed in inches.
WPRTUNITS-CENTIMETERS	Values are expressed in centimeters.
WPRTUNITS-PIXELS	Values are expressed in pixels.

winprint-col-alignment
Specifies the alignment of the text in the column being defined. Valid values are:

WPRTALIGN-NONE	No alignment. Leading spaces are kept.
WPRTALIGN-LEFT	Left alignment. Leading spaces are removed.
WPRTALIGN-RIGHT	Right alignment. Trailing spaces are removed.
WPRTALIGN-CENTER	Centered. Leading and trailing spaces are removed.
WPRTALIGN-DECIMAL	Right alignment on the decimal point. Trailing spaces are removed.
WPRTALIGN-DECIMAL-SUPPRESS	Right alignment on the decimal point. The decimal point character is not printed, a space is printed, instead. Trailing spaces are removed.
WPRTALIGN-RIGHT-SIGN	Right alignment. A certain amount of space is left on the right side to accommodate the sign symbol. Trailing spaces are removed.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.
WPRTERR-BAD-ARG	The WIN$PRINTER library routine has been called with bad parameters.

WINPRINT-SET-PRINTER

The WINPRINT-SET-PRINTER function selects a printer and set some of its features. More features can be set with the WINPRINT-SET-PRINTER-EX routine.

The printer settings are evaluated only at the OPEN OUTPUT of the print-file. Calling this op-code in the middle of the print job has no effect.

When printing to PDF or Print Preview, the active printer is used to calculate margins. If you wish to avoid this, call the WINPRINT-SET-PRINTER function with *wprterr-no-printers* set to true. You will be allowed to set only *winprint-curr-copies* and *winprint-curr-orientation*.

Syntax:

```
CALL "WIN$PRINTER" USING WINPRINT-SET-PRINTER
                         WINPRINT-SELECTION
                   GIVING returnCode
```

Parameters:

WINPRINT-SET-PRINTER	Constant	
WINPRINT-SELECTION	Group Item	Structure that contains the information to be passed to the function. This group item, defined in isprint.def, has the following structure:

```
01  winprint-selection.
    03 winprint-name              pic x(80).
    03 winprint-port              pic x(80).
    03 winprint-driver            pic x(80).
    03 winprint-drv-version       signed-int.
    03 winprint-no-of-printers    signed-short.
       88 wprterr-no-printers     value -1.
    03 winprint-is-default        signed-short.
       88 wprt-is-not-default     value 0.
       88 wprt-is-default         value 1.
    03 winprint-copies            signed-short.
    03 winprint-orientation       signed-short.
    03 winprint-quality           signed-short.
    03 winprint-curr-orientation signed-short.
    03 winprint-curr-copies       signed-short.
```

Note - members not mentioned below are not used by this function.

winprint-name
It is the name of the printer.

When the function is called and winprint-no-of-printers is zero, the function searches for a printer with the name specified and selects it.

winprint-no-of-printers
It is the number of the printer to be selected. When this member is zero, the function selects the printer with the name specified in winprint-name.

winprint-quality
It is the print quality. Not that most laser printers do not support quality, while most dot-matrix printers do. Valid values are:

WPRTSEL-QUALITY-DEFAULT	Default quality.
WPRTSEL-QUALITY-HIGH	High quality.
WPRTSEL-QUALITY-MEDIUM	Medium quality.
WPRTSEL-QUALITY-LOW	Low quality.
WPRTSEL-QUALITY-DRAFT	Draft quality.

winprint-curr-orientation
It is the paper orientation. Valid values are:

WPRTSEL-ORIENT-DEFAULT	Default paper orientation.
WPRTSEL-ORIENT-PORTRAIT	Portrait.
WPRTSEL-ORIENT-LANDSCAPE	Landscape.

winprint-curr-copies
Is the number of copies the printer will print with the next job.

Note - If no printer is available in the system, set *wprterr-no-printers* to true before calling this function in order to avoid errors. You will be allowed to set only *winprint-curr-copies* and *winprint-curr-orientation*.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.
WPRTERR-BAD-ARG	The WIN$PRINTER library routine has been called with bad parameters.

WINPRINT-SET-PRINTER-AS

In an Application Server environment the WINPRINT-SET-PRINTER-AS function redirects the print job on the server machine or the client machine. By default print jobs are redirected on the client machine.

All the print jobs created after the call to this function are affected.

Syntax:

```
CALL "WIN$PRINTER" USING WINPRINT-SET-PRINTER-AS
                         winprint-as
                   GIVING returnCode
```

Parameters:

WINPRINT-SET-PRINTER-AS	Constant		
winprint-as	Numeric Item	Valid values are:	
		WPRT-AS-CLIENT	print job is redirect to the client machine.
		WPRT-AS-SERVER	print job is performed on the server machine.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.
WPRTERR-BAD-ARG	The WIN$PRINTER library routine has been called with bad parameters.

WINPRINT-SET-PRINTER-EX

This functions selects a printer and set more features of that printer than the WINPRINT-SET-PRINTER allows.

The printer settings are evaluated only at the OPEN OUTPUT of the print-file. Calling this op-code in the middle of the print job has no effect.

When printing to PDF or Print Preview, the active printer is used to calculate margins. If you wish to avoid this, call the WINPRINT-SET-PRINTER-EX function with *wprterr-no-printers* set to true. You will be allowed to set only *winprint-curr-copies* and *winprint-curr-orientation*.

Syntax:

```
CALL "WIN$PRINTER" USING WINPRINT-SET-PRINTER-EX
                         WINPRINT-SELECTION
                   GIVING returnCode
```

Parameters:

WINPRINT-SET-PRINTER-EX	Constant	

WINPRINT-SELECTION	Group Item	Structure that contains the information to be passed to the function. This group item, defined in isprint.def, has the following structure:
		
		`01 winprint-selection.`
		`  03 winprint-name  pic x(80).`
		`  03 winprint-port  pic x(80).`
		`  03 winprint-driver  pic x(80).`
		`  03 winprint-drv-version  signed-int.`
		`  03 winprint-no-of-printers  signed-short.`
		`  88 wprterr-no-printers  value -1.`
		`  03 winprint-is-default  signed-short.`
		`  88 wprt-is-not-default  value 0.`
		`  88 wprt-is-default  value 1.`
		`  03 winprint-copies  signed-short.`
		`  03 winprint-orientation  signed-short.`
		`  03 winprint-quality  signed-short.`
		`  03 winprint-curr-orientation  signed-short.`
		`  03 winprint-curr-copies  signed-short.`
		`  03 winprint-duplex  signed-short.`
		`  03 winprint-collate  signed-short.`
		`  03 winprint-color  signed-short.`
		`  03 winprint-curr-duplex  signed-short.`
		`  03 winprint-curr-collate  signed-short.`
		`  03 winprint-curr-papersize  signed-short.`
		`  88 wprt-{format}  value ...`
		`  03 winprint-curr-tray  signed-short.`
		`  03 winprint-curr-color  signed-short.`
		`  03 winprint-job-title  pic x(80).`
		
		Note - members not mentioned below are not used by this function.
		
		winprint-name
		It is the name of the printer.
		
		When the function is called and winprint-no-of-printers is zero, the function searches for a printer with the name specified and selects it.
		
		winprint-no-of-printers
		It is the number of the printer to be selected. When this member is zero, the function selects the printer with the name specified in winprint-name.

winprint-quality
It is the print quality. Not that most laser printers do not support quality, while most dot-matrix printers do. Valid values, defined in isprint.def , are:

WPRTSEL-QUALITY-DEFAULT	Default quality.
WPRTSEL-QUALITY-HIGH	High quality.
WPRTSEL-QUALITY-MEDIUM	Medium quality.
WPRTSEL-QUALITY-LOW	Low quality.
WPRTSEL-QUALITY-DRAFT	Draft quality.

winprint-curr-orientation
It is the paper orientation. Valid values are:

WPRTSEL-ORIENT-DEFAULT	Default paper orientation.
WPRTSEL-ORIENT-PORTRAIT	Portrait.
WPRTSEL-ORIENT-LANDSCAPE	Landscape.

winprint-curr-copies
Is the number of copies the printer will print with the next job.

winprint-curr-duplex
Is the duplex setting.

WPRT-SIMPLEX	No duplex.
WPRT-DUPLEX-VERTICAL	Duplex vertical.
WPRT-DUPLEX-HORIZONTAL	Duplex horizontal.

winprint-curr-collate
Is the current collate state. Valid values are:

WPRT-COLLATE-OFF	Collate disabled
WPRT-COLLATE-ON	Collate enabled

winprint-curr-papersize
It is the paper format. Set one of the available conditions to true. For example, setting the wprt-a4 condition to true would set the paper format to A4, setting the wprt-letter condition to true would set the paper format to Letter.

Available papersizes are listed below.

winprint-curr-tray
It is the printer tray. Valid values, defined in isprint.def , are:

WPRT-UPPER-TRAY

WPRT-LOWER-TRAY

WPRT-MIDDLE-TRAY

	WPRT-MANUAL-TRAY WPRT-ENVELOPE-TRAY WPRT-ENVMANUAL-TRAY WPRT-AUTO-TRAY WPRT-TRACTOR-TRAY WPRT-SMALLFMT-TRAY WPRT-LARGEFMT-TRAY WPRT-LARGECAPACITY-TRAY WPRT-CASSETTE-TRAY WPRT-FORMSOURCE-TRAY *winprint-curr-color* It turns colors on or off. Set the variable to 1 in order to use colors or set it to 0 if you want a monochrome print. Note - Setting winprint-curr-color to 0 doesn't guarantee that the printer output will be black and white. The runtime doesn't perform any color conversion, it just passes the monochrome setting to the printer, then it's the printer duty to honor it. Most of the physical printers are able to convert the graphics to black and white, but the isCOBOL Print Preview window and the internal PDF printer are not. *winprint-job-title* It is the title of the job in the print spooler.

Note - If no printer is available in the system, set *wprterr-no-printers* to true before calling this function in order to avoid errors. You will be allowed to set only *winprint-curr-copies* and *winprint-curr-orientation*.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.
WPRTERR-BAD-ARG	The WIN$PRINTER library routine has been called with bad parameters.

List of supported papersizes

Data-item (available in isprint.def)	Value	Description
wprt-letter	1	na-letter, 215900x279400 um
wprt-lettersmall	2	na-letter, 215900x279400 um
wprt-tabloid	3	tabloid, null
wprt-ledger	4	ledger, 279400x431800 um
wprt-legal	5	na-legal, 215900x355600 um
wprt-statement	6	Statement 5 1/2 x 8 1/2 in, 139700x215900 um
wprt-executive	7	executive, 184150x266700 um
wprt-a3	8	iso-a3, 297000x420000 um
wprt-a4	9	iso-a4, 210000x297000 um
wprt-a4small	10	iso-a4, 210000x297000 um
wprt-a5	11	iso-a5, 148000x210000 um
wprt-b4	12	jis-b4, 257000x364000 um
wprt-b5	13	jis-b5, 182000x257000 um
wprt-folio	14	folio, 215900x330200 um
wprt-quarto	15	quarto, 215900x275082 um
wprt-10x14	16	na-10x14-envelope, 254000x355600 um
wprt-11x17	17	11x17 in, 279400x431800 um
wprt-note	18	Note 8 1/2 x 11 in, 215900x279400 um
wprt-env_9	19	na-number-9-envelope, 98425x225425 um
wprt-env_10	20	na-number-10-envelope, 104775x241300 um
wprt-env_11	21	na-number-11-envelope, 114300x263525 um
wprt-env_12	22	na-number-12-envelope, 120650x279400 um
wprt-env_14	23	na-number-14-envelope, 127000x292100 um
wprt-csheet	24	c, 431800x558800 um
wprt-dsheet	25	d, 558800x863600 um
wprt-esheet	26	e, 863600x1117600 um
wprt-env_dl	27	Envelope DL 110 x 220mm, 110000x220000 um
wprt-env_c5	28	iso-c5, 162000x229000 um
wprt-env_c3	29	iso-c3, 324000x458000 um
wprt-env_c4	30	iso-c4, 229000x324000 um
wprt-env_c6	31	iso-c6, 114000x162000 um
wprt-env_c65	32	Envelope C65 114 x 229 mm, 114000x229000 um
wprt-env_b4	33	iso-b4, 250000x353000 um
wprt-env_b5	34	iso-b5, 176000x250000 um
wprt-env_b6	35	iso-b6, 125000x176000 um
wprt-env_italy	36	italian-envelope, 110000x230000 um
wprt-env_monarch	37	monarch-envelope, 98298x190500 um
wprt-env_personal	38	personal-envelope, 92075x165100 um
wprt-fanfold_us	39	US Std Fanfold 14 7/8 x 11 in, 198120x279400 um
wprt-fanfold_std_german	40	German Std Fanfold 8 1/2 x 12 in, 215900x304800 um
wprt-fanfold_lgl_german	41	German Legal Fanfold 8 1/2 x 13 in, 215900x330200 um
wprt-japanese-postcard	43	japanese-postcard, 100000x148000 um
wprt-env-9x11	44	na-9x11-envelope, 228600x279400 um
wprt-10x11	45	10 x 11 in, 254000x279400 um
wprt-15x11	46	15 x 11 in, 279400x381000 um
wprt-env-invite	47	Envelope Invite 220 x 220 mm, 220000x220000 um
wprt-us-letter-extra	50	US Letter Extra 9 1/2 x 12 in, 241300x304800 um
wprt-us-legal-extra	51	US Legal Extra 9 1/2 x 15 in, 241300x381000 um
wprt-tabloid-extra	52	US Tabloid Extra 11.69 x 18 in, 296926x457200 um
wprt-a4-extra	53	A4 Extra 9.27 x 12.69 in , 235458x322326 um
wprt-letter-transverse	54	Letter Transverse 8 1/2 x 11 in, 215900x279400 um
wprt-a4-transverse	55	A4 Transverse 210 x 297 mm, 210000x297000 um
wprt-letter-extra-transverse	56	Letter Extra Transverse 9 1/2 x 12 in, 241300x304800 um
wprt-supera-a4	57	SuperA/SuperA/A4 227 x 356 mm, 227000x356000 um
wprt-superb-a3	58	SuperB/SuperB/A3 305 x 487 mm, 305000x487000 um
wprt-us-letter-plus	59	US Letter Plus 8.5 x 12.69 in, 215900x322326 um
wprt-a4-plus	60	A4 Plus 210 x 330 mm, 210000x330000 um

Data-item (available in isprint.def)	Value	Description
wprt-a5-transverse	61	A5 Transverse 148 x 210 mm, 148000x210000 um
wprt-b5-transverse	62	B5 (JIS) Transverse 182 x 257 mm, 182000x257000 um
wprt-a3-extra	63	A3 Extra 322 x 445 mm, 322000x445000 um
wprt-a5-extra	64	A5 Extra 174 x 235 mm, 174000x235000 um
wprt-b5-extra	65	B5 (ISO) Extra 201 x 276 mm, 201000x276000 um
wprt-a2	66	iso-a2, 420000x594000 um
wprt-a3-transverse	67	A3 Transverse 297 x 420 mm, 297000x420000 um
wprt-a3-extra-transverse	68	A3 Extra Transverse 322 x 445 mm, 322000x445000 um
wprt-oufuko-postcard	69	oufuko-postcard, 148000x200000 um
wprt-a6	70	iso-a6, 105000x148000 um

WINPRINT-SET-STD-FONT

The WINPRINT-SET-STD-FONT selects one of the pre-defined fonts to be used by the subsequent WRITE statements.

A printer is required for this function to work correctly.

Syntax:

```
CALL "WIN$PRINTER" USING WINPRINT-SET-STD-FONT
                         WINPRINT-DATA
                   GIVING returnCode
```

Parameters:

WINPRINT-SET-STD-FONT	Constant	
WINPRINT-DATA	Group Item	Structure that contains the information to be passed to the function. This group item, defined in isprint.def, has the following structure:

```
01  winprint-data.
    03 wprtdata-set-std-font.
       05 wprtdata-std-font          pic x comp-x.
       05 filler                     pic x(21).
```

wprtdata-std-font
Specifies the font to be used by the subsequent WRITE statements. Valid values are:

WPRTFONT-DEFAULT	It is the default printer font.
WPRTFONT-COURIER-12	It uses a 12-point TrueType Courier font.
WPRTFONT-COURIER-12-COMP	It uses a 12-point TrueType Courier font and rescales it so that at least 132 columns of print will fit on a page.
WPRTFONT-COURIER-10	It uses a 10-point TrueType Courier font.
WPRTFONT-COURIER-10-COMP	It uses a 10-point TrueType Courier font and rescales it so that at least 132 columns of print will fit on a page.

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.
WPRTERR-BAD-ARG	The WIN$PRINTER library routine has been called with bad parameters.

WINPRINT-SET-TEXT-COLOR

The WINPRINT-SET-TEXT-COLOR function sets the text foreground color.

Syntax:

```
CALL "WIN$PRINTER" USING WINPRINT-SET-TEXT-COLOR
                         WINPRINT-DATA
                   GIVING returnCode
```

Parameters:

WINPRINT-SET-TEXT-COLOR	Constant

WINPRINT-DATA	Group Item	Structure that contains the information to be passed to the function. This group item, defined in isprint.def, has the following structure:
		
		`01 winprint-data.`
		`  03 wprtdata-text-color  pic 9(9) comp-5.`
		
		wprtdata-text-color
		Specifies the color to be used, in RGB format.
		
		Color value is computed as follows: *Red* + *Green* * 256 + *Blue* * 65536. *Red*, *Green* and *Blue* range from 0 to 255.
		
		A more convenient way to define the color is to use the hexadecimal notation X#RRGGBB.
		
		To print a light gray text you may use one of the following, they are equivalent:
		
		`compute wprtdata-text-color = 192 * 65536 + 192 * 256 + 192`
		
		or
		
		`move X#C0C0C0 to wprtdata-text-color`
		
		For the most common colors, the following constants can be used:
		`78 wprt-color-black value x#000000.`
		`78 wprt-color-red value x#0000FF.`
		`78 wprt-color-green value x#00FF00.`
		`78 wprt-color-blue value x#FF0000.`
		`78 wprt-color-yellow value x#00FFFF.`
		`78 wprt-color-magenta value x#FF00FF.`
		`78 wprt-color-cyan value x#FFFF00.`
		`78 wprt-color-white value x#FFFFFF.`

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.
WPRTERR-BAD-ARG	The WIN$PRINTER library routine has been called with bad parameters.

WINPRINT-SETUP

The WINPRINT-SETUP function shows a dialog window to choose the current printer.

Syntax:

```
CALL "WIN$PRINTER" USING WINPRINT-SETUP
                   GIVING returnCode
```

Parameters:

WINPRINT-SETUP	Constant

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.
WPRTERR-BAD-ARG	The WIN$PRINTER library routine has been called with bad parameters.

WINPRINT-SUPPORTED

The WINPRINT-SUPPORTED function checks if the host system supports the WIN$PRINTER library routine.

Syntax:

```
CALL "WIN$PRINTER" USING WINPRINT-SUPPORTED
                   GIVING returnCode
```

Parameters:

WINPRINT-SUPPORTED	Constant

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.

Note: currently, the WINPRINT-SUPPORTED function always returns 1 because the WIN$PRINTER library routine is supported on all platforms.

WINPRINT-UPDATE-PRINTERS

The WINPRINT-UPDATE-PRINTERS function reloads the list of available printers so that any change to that list (e.g. the addition of a new printer) is detected by the COBOL program.

Syntax:

```
CALL "WIN$PRINTER" USING WINPRINT-UPDATE-PRINTERS
                   GIVING returnCode
```

Parameters:

WINPRINT-UPDATE-PRINTERS	Constant

Return code:

returnCode can be any signed numeric data item and provides additional information:

1	Operation successful.
WPRTERR-UNSUPPORTED	The WIN$PRINTER library routine is not supported.

WIN$VERSION

The WIN$VERSION library routine retrieves extended information about the Windows operating system where the isCOBOL framework is running. See the ACCEPT FROM SYSTEM INFO statement for additional information about the host operating system.

Note - If your current JVM was released before the current operating system, then values returned by this routine may not be accurate.

Syntax:

```
CALL "WIN$VERSION" USING WINVERSION-DATA
```

Parameters:

WINVERSION-DATA	Group Item	This group data item should have the following structure:

```
01  winversion-data.
    03 win-major-version         pic x comp-x.
    03 win-minor-version         pic x comp-x.
    03 win-platform              pic x comp-x.
       88 platform-win-31        value 1.
       88 platform-win-95        value 2.
       88 platform-win-9x        value 2.
       88 platform-win-nt        value 3.
    03 win-wordsize              pic x comp-x.
       88 win-wordsize-16        value 1.
       88 win-wordsize-32        value 2.
       88 win-wordsize-64        value 3.
```

win-major-version
Receives the operating system's major version.

win-minor-version
Receives the operating system's major version.

win-major-version and win-minor-version should be evaluated together, referring to the following table:

Operating system	*Major version*	*Minor version*
Windows 10	10	0
Windows 8.1	6	3
Windows 8	6	2
Windows Server 2012	6	2
Windows 7	6	1
Windows Server 2008 R2	6	1
Windows Server 2008	6	0
Windows Vista	6	0
Windows Server 2003 R2	5	2
Windows Server 2003	5	2
Windows XP 64-bit	5	2
Windows XP	5	1
Windows 2000	5	0

win-platform
Receives the operating system's major platform.

win-wordsize
Receives the runtime's word size.

Examples:

Example - Get Windows version information

```
*> on working-storage copy "iswinvers.def"

call "win$version" using winversion-data
display message "Win major version : " win-major-version x"0d0a"
                "Win minor version : " win-minor-version x"0d0a"
                "Win platform      : " win-platform.
```

Interpreting the return code as a file status code

The below information is applicable to the following routines:

- CBL_CHANGE_DIR
- CBL_CHECK_FILE_EXIST
- CBL_COPY_FILE
- CBL_CREATE_DIR
- CBL_CREATE_FILE
- CBL_DELETE_DIR
- CBL_DELETE_FILE
- CBL_FLUSH_FILE
- CBL_OPEN_FILE
- CBL_READ_FILE
- CBL_RENAME_FILE
- CBL_WRITE_FILE

If any of these routines fails, the RETURN-CODE register contains a file status value indicating the failure. This file status is always the standard ANSI'74 file status value. If no ANSI'74 file status is defined for the error, an extended file status is returned (9/*nnn* where *nnn* is the run-time system error number).

You should, therefore, use RETURN-CODE and not a RETURNING clause. If RETURN-CODE is non-zero after calling one of these routines, you must process it as a file status, for example:

```
working-storage section.
...
01 file-status      pic xx comp-x.
01 redefines file-status.
   03 fs-byte-1  pic x.
   03 fs-byte-2  pic x comp-x.
...
procedure division.
    ...
    call "CBL_xxx_xxx" using parameters
    if return-code not = 0
       move return-code to file-status
    ...
```

At this point *fs-byte-1* contains "9" and *fs-byte-2* contains the run-time system error number.

Appendix C

Graphical Control List

The following four tables represent supported Graphical Controls. Please refer to the isCOBOL Graphic Controls Reference manual for more details.

1. Table one contains the list of all Control and for each, the related properties, styles and events.
2. Table two contains the list of all properties and for each the controls that support that property.
3. Table three contains the list of all styles and for each the controls that support that style.
4. Table four contains the list of all events and for each the controls that support that event.
5. Table five contains the list of all properties and for each the statements allowed on that property.

Table 1

This table shows the list of all properties, styles and events for each graphical control.

Name	Properties	Styles	Events
BAR	Col, Color, Colors, Column, Css-Style-Name, Custom-Data, Enabled, Event-List, Exclude-Event-List, Font, Foreground-Color, Help-Id, Id, Layout-data,Leading-Shift, Line, Lines, Max-Height, Max-Width, Min-Height, Min-Width, Pop-Up Menu, Pos, Position, Position-Shift, Shading, Size, Trailing-Shift, Visible, Width.	Bold, Dashed, Dot-Dash, Dotted, Height-In-Cells, High, Highlight, Low, Lowlight, Notify-Mouse, Permanent, Standard, Temporary, Width-In-Cells.	MSG-END-MENU, MSG-INIT-MENU, MSG-MENU-INPUT, MSG-MOUSE-CLICKED, MSG-MOUSE-DBLCLICK, MSG-MOUSE-ENTER, MSG-MOUSE-EXIT.

BITMAP	Background-Color, Bitmap-End, Bitmap-Handle, Bitmap-Number, Bitmap-Start, Bitmap-Timer, Col, Column, Css-Style-Name, Custom-Data, Enabled, Event-List, Exclude-Event-List, Font, Foreground-Color, Help-Id, Hint, Id, Layout-data, Line, Lines, Max-Height, Max-Width, Min-Height, Min-Width, Pop-Up Menu, Pos, Position, Size, Transparent-Color, Visible	Background-High, Background-Low, Background-Standard, Bold, Height-In-Cells, High, Highlight, Low, Lowlight, Notify-Mouse, Permanent, Standard, Temporary, Width-In-Cells.	MSG-END-MENU, MSG-INIT-MENU, MSG-MENU-INPUT, MSG-MOUSE-CLICKED, MSG-MOUSE-DBLCLICK, MSG-MOUSE-ENTER, MSG-MOUSE-EXIT.
CHECK-BOX	Background-Color, Bitmap-Disabled, Bitmap-Disabled-Selected, Bitmap-Handle, Bitmap-Number, Bitmap-Pressed, Bitmap-Rollover, Bitmap-Rollover-Selected, Bitmap-Width, Col, Color, Column, Css-Style-Name, Custom-Data, Enabled, Event-List, Exception-Value, Exclude-Event-List, Font, Foreground-Color, Help-Id, Hint, Id, Layout-data, Line, Lines, Max-Height, Max-Width, Min-Height, Min-Width, Pop-Up Menu, Pos, Position, Size, Termination-Value, Title, Title-Position, Value, Visible.	Background-High, Background-Low, Background-Standard, Bitmap, Bold, Flat, Framed, Height-In-Cells, High, Highlight, Left-Text, Low, Lowlight, Multiline, No-Tab, Notify, Notify-Mouse, Permanent, Self-Act, Square, Standard, Temporary, Transparent, Unframed, Vtop, Width-In-Cells.	CMD-CLICKED, CMD-GOTO, CMD-HELP, MSG-END-MENU, MSG-INIT-MENU, MSG-MENU-INPUT, MSG-MOUSE-ENTER, MSG-MOUSE-EXIT, MSG-VALIDATE.

COMBO-BOX	Background-Color, Bitmap-Handle, Bitmap-Number, Bitmap-Width, Col, Color, Column, Css-Style-Name, Cursor, Custom-Data, Enabled, Event-List, Exception-Value, Exclude-Event-List, Font, Foreground-Color, Help-Id, Hidden-Data, Hint, Id, Insertion-Index, Item, Item-Background-Color, Item-Color, Item-Foreground-Color, Item-Height, Item-Text, Item-To-Add, Item-To-Delete, Layout-data, Line, Lines, Mass-Update, Max-Height, Max-Text, Max-Width, Min-Height, Min-Width, Placeholder, Pop-Up Menu, Pos, Position, Query-Index, Reset-List, Selection-Background-Color, Selection-Color, Selection-Foreground-Color, Size, Termination-Value, Value, Visible.	Background-High, Background-Low, Background-Standard, Bold, Drop-Down, Drop-List, Height-In-Cells, High, Highlight, Low, Lower, Lowlight, No-Tab, Notify-Dblclick, Notify-Mouse, Notify-Selchange, Permanent, Standard, Static-List, Temporary, Unsorted, Upper, Width-In-Cells.	CMD-DBLCLICK, CMD-GOTO, CMD-HELP, MSG-END-MENU, MSG-INIT-MENU, MSG-MENU-INPUT, MSG-MOUSE-ENTER, MSG-MOUSE-EXIT, MSG-VALIDATE, NTF-SELCHANGE.
DATE-ENTRY	Background-Color, Bitmap-Handle, Bitmap-Number, Bitmap-Width, Col, Color, Column, Css-Style-Name, Custom-Data, Decoration-Background, Display-Format, Enabled, Event-List, Exclude-Event-List, Font, Foreground-Color, Help-Id, Hint, Id, Layout-data, Line, Lines, Max-Height, Max-Val, Max-Width, Maxday-Characters, Min-Height, Min-Val, Min-Width, Pop-Up Menu, Pos, Position, Size, Sunday-Foreground, Value, Value-Format, Visible, Weekday-Foreground	Allow-Empty, Background-High, Background-Low, Background-Standard, Bold, Century-Date, Decoration-Background-Visible, Decoration-Borders-Visible, Height-In-Cells, High, Highlight, Long-Date, Low, Lowlight, No-F4, No-Tab, No-Updown, Notify-Change, Notify-Mouse, Numeric, Permanent, Read-Only, Right-Align, Short-Date, Spinner, Standard, Temporary, Time, Week-Of-Year-Visible, Width-In-Cells.	CMD-GOTO, CMD-HELP, MSG-END-MENU, MSG-INIT-MENU, MSG-MENU-INPUT, MSG-MOUSE-ENTER, MSG-MOUSE-EXIT, MSG-VALIDATE, NTF-CHANGED

ENTRY-FIELD	Action, Auto-Decimal, Background-Color, Bitmap-Disabled, Bitmap-Handle, Bitmap-Hint, Bitmap-Number, Bitmap-Rollover, Bitmap-Trailing-Disabled, Bitmap-Trailing-Hint, Bitmap-Trailing-Number, Bitmap-Trailing-Rollover, Bitmap-Width, Border-Color, Border-Width, Col, Color, Column, Css-Style-Name, Cursor, Cursor-Col, Cursor-Row, Custom-Data, Enabled, Event-List, Exclude-Event-List, Fill-Char, Font, Foreground-Color, Format-String, Help-Id, Hint, Id, Layout-data, Line, Lines, Max-Height, Max-Lines, Max-Text, Max-Val, Max-Width, Min-Height, Min-Val, Min-Width, Notify-Change-Delay, Placeholder, Pop-Up Menu, Pos, Position, Proposal, Proposal-Delay, Proposal-Index, Proposal-Min-Text, Proposal-To-Delete, Reset-Proposals, Selection-Text, Size, Spell-Checking, Text-Orientation, Validation-Errmsg, Validation-Opts, Validation-Regexp Value, Visible, Visible-Proposal-Count.	3-D, Auto, Auto-Spin, Background-High, Background-Low, Background-Standard, Bold, Boxed, Center, Centered, Height-In-Cells, High, Highlight, Left, Low, Lower, Lowlight, Multiline, No-Autosel, No-Box, No-Tab, No-Wrap, Notify-Change, Notify-Mouse, Numeric, Permanent, Proposals-Unsorted, Read-Only, Right, Secure, Spinner, Standard, Temporary, Upper, Use-Return, Use-Tab, Vscroll, Vscroll-Bar, Width-In-Cells.	CMD-GOTO, CMD-HELP, MSG-BITMAP-CLICKED, MSG-BITMAP-DBLCLICK, MSG-END-MENU, MSG-INIT-MENU, MSG-MENU-INPUT, MSG-MOUSE-ENTER, MSG-MOUSE-EXIT, MSG-SPIN-DOWN, MSG-SPIN-UP, MSG-VALIDATE, NTF-CHANGED.
FRAME	Background-Color, Col, Color, Column, Css-Style-Name, Custom-Data, Event-List, Exclude-Event-List, Fill-Color, Fill-Color2, Fill-Percent, Font, Foreground-Color, Help-Id, High-Color, Hint, Id, Layout-data, Line, Lines, Low-Color, Max-Height, Max-Width, Min-Height, Min-Width, Pop-Up Menu, Pos, Position, Size, Title, Title-Position, Visible.	Alternate, Background-High, Background-Low, Background-Standard, Bold, Engraved, Full-Height, Heavy, Height-In-Cells, High, Highlight, Low, Lowered, Lowlight, Notify-Mouse, Permanent, Raised, Rimmed, Standard, Temporary, Transparent, Very-Heavy, Width-In-Cells	MSG-END-MENU, MSG-INIT-MENU, MSG-MENU-INPUT, MSG-MOUSE-CLICKED, MSG-MOUSE-DBLCLICK, MSG-MOUSE-ENTER, MSG-MOUSE-EXIT.

GRID	Action, Alignment, Background-Color, Bitmap, Bitmap-Number, Bitmap-Trailing, Bitmap-Width, Border-Color, Border-Width, Cell-Background-Color, Cell-Color, Cell-Columns-Span, Cell-Current-Background-Color, Cell-Current-Color, Cell-Current-Font, Cell-Current-Foreground-Color, Cell-Current-Protection, Cell-Data, Cell-Entry-Background-Color, Cell-Entry-Color, Cell-Entry-Foreground-Color, Cell-Font, Cell-Foreground-Color, Cell-Hint, Cell-Protection, Cell-Rows-Span, Cell-Selected-Background-Color, Cell-Selected-Color, Cell-Selected-Foreground-Color, Cells-Selected, Col, Color, Column, Column-Background-Color, Column-Color, Column-Dividers, Column-Font, Column-Foreground-Color, Column-Headings-Height, Column-Headings-Layout, Column-Hiding, Column-Protection, Column-Selected-Background-Color, Column-Selected-Color, Column-Selected-Foreground-Color, Columns-Selected, Css-Style-Name, Cursor-Background-Color, Cursor-Color, Cursor-Foreground-Color, Cursor-Frame-Width, Cursor-X, Cursor-Y, Custom-Data, Data-Columns, Data-Types, Display-Columns, Divider-Color, Drag-Background-Color, Drag-Color,Drag-Foreground-Color, Editor-Show-Always, Enabled, End-Color, Entry-Reason, Event-List, Exclude-Event-List, Export-File-Format, Export-File-Name, File-Pos, Finish-Reason, Font, Foreground-Color, Heading-Background-Color, Heading-Color, Heading-Cursor-Background-Color, Heading-Cursor-Color, Heading-Cursor-Foreground-Color, Heading-Divider-Color, Heading-Font, Heading-Foreground-Color, Heading-Menu-Popup, Help-Id, Hidden-Data, Hint,	3-D, Adjustable-Columns, Auto, Background-High, Background-Low, Background-Standard, Bold, Boxed, Centered-Headings, Column-Headings, Filterable-Columns, Height-In-Cells, High, Highlight, Hscroll, Low, Lowlight, No-Box, No-Autosel, No-Cell-Drag, No-Search, Notify-Mouse, Paged, Permanent, Reordering-Columns, Row-Headings, Sortable-Columns, Standard, Temporary, Tiled-Headings, Use-Tab, Vscroll, Width-In-Cells	CMD-GOTO, CMD-HELP, MSG-BEGIN-DRAG, MSG-BEGIN-ENTRY, MSG-BEGIN-HEADING-DRAG, MSG-BEGIN-HEADING-MENU-POPUP, MSG-BEGIN-SORT, MSG-BITMAP-CLICKED, MSG-BITMAP-DBLCLICK, MSG-CANCEL-ENTRY, MSG-COL-WIDTH-CHANGED, MSG-END-DRAG, MSG-END-HEADING-DRAG, MSG-END-MENU, MSG-FINISH-ENTRY, MSG-FINISH-SORT, MSG-GD-DBLCLICK, MSG-GOTO-CELL, MSG-GOTO-CELL-DRAG, MSG-GOTO-CELL-MOUSE, MSG-GOTO-CELL-OUT-NEXT, MSG-GOTO-CELL-OUT-PREV, MSG-GRID-RBUTTON-DOWN, MSG-GRID-RBUTTON-UP, MSG-HEADING-DRAGGED, MSG-HEADING-CLICKED, MSG-HEADING-MENU-POPUP, MSG-INIT-MENU, MSG-MENU-INPUT, MSG-MOUSE-ENTER, MSG-MOUSE-EXIT, MSG-PAGED-FIRST, MSG-PAGED-LAST, MSG-PAGED-NEXT, MSG-PAGED-NEXTPAGE, MSG-PAGED-PREV, MSG-PAGED-PREVPAGE, MSG-VALIDATE.

GRID (continued)	Hscroll-Pos, Id, Insert-Rows, Insertion-Index, Last-Row, Layout-data, Line, Lines, Lm-On-Columns, Mass-Update, Max-Height, Max-Width, Min-Height, Min-Width, Model-To-View-Y, Mouse-Wheel-Scroll, Num-Col-Headings, Num-Row-Headings, Num-Rows, Pop-Up Menu, Pos, Position, Protection, Record-Data, Record-To-Add, Record-To-Delete, Region-Background-Color, Region-Color, Region-Foreground-Color, Reordering-Col-Index, Reset-Grid, Row-Background-Color, Row-Background-Color-Pattern, Row-Capacity, Row-Color, Row-Color-Pattern, Row-Cursor-Background-Color, Row-Cursor-Color, Row-Cursor-Foreground-Color, Row-Dividers, Row-Font, Row-Foreground-Color, Row-Foreground-Color-Pattern, Row-Hiding, Row-Protection, Row-Selected-Background-Color, Row-Selected-Color, Row-Selected-Foreground-Color, Rows-Filtered, Rows-Per-Page, Rows-Selected, Search-Options, Search-Panel, Search-Text, Search-Text-In-View, Selection-Mode, Separation, Size, Sort-data, Sort-Types, Start-X, Start-Y, VPadding, View-Cursor-Y, View-To-Model-Y, Virtual-Width, Visible, Vscroll-Pos, X, Y.

JAVA-BEAN	Background-Color, Border-Color, Border-Width, Clsid, Col, Column, Css-Style-Name, Custom-Data, Enabled, Event-List, Exclude-Event-List, Font, Foreground-Color, Help-Id, Hint, Id, Init-Params, Init-Signature, Layout-data, Line, Lines, Max-Height, Max-Width, Min-Height, Min-Width, Object, Pop-Up Menu, Pos, Position, Size, Visible.	3-D, Background-High, Background-Low, Background-Standard, Bold, Boxed, Height-In-Cells, High, Highlight, Low, Lowlight, No-Box, Notify-Mouse, Permanent, Self-Act, Standard, Use-Return, Use-Tab, Temporary, Width-In-Cells.	MSG-END-MENU, MSG-INIT-MENU, MSG-JB-EVENT, MSG-MENU-INPUT, MSG-MOUSE-ENTER, MSG-MOUSE-EXIT.
LABEL	Background-Color, Col, Color, Column, Css-Style-Name, Custom-Data, Enabled, Event-List, Exclude-Event-List, Font, Foreground-Color, Help-Id, Hint, Id, Label-Offset, Layout-data, Line, Lines, Max-Height, Max-Width, Min-Height, Min-Width, Pop-Up Menu, Pos, Position, Size, Title, Visible.	Background-High, Background-Low, Background-Standard, Bold, Bottom, Center, Centered, Height-In-Cells, High, Highlight, Left, Low, Lowlight, No-Key-Letter, Notify-Mouse, Permanent, Right, Standard, Temporary, Top, Transparent, Vertical, Width-In-Cells	MSG-END-MENU, MSG-INIT-MENU, MSG-MENU-INPUT, MSG-MOUSE-CLICKED, MSG-MOUSE-DBLCLICK, MSG-MOUSE-ENTER, MSG-MOUSE-EXIT.

LIST-BOX	Action, Alignment, Background-Color, Border-Color, Border-Width, Col, Color, Column, Data-Columns, Css-Style-Name, Custom-Data, Display-Columns, Dividers, Enabled, Event-List, Exception-Value, Exclude-Event-List, Export-File-Format, Export-File-Name, Font, Foreground-Color, Help-Id, Hint, Id, Hidden-Data, Insertion-Index, Item-Background-Color, Item-Color, Item-Foreground-Color, Item-To-Add, Item-To-Delete, Item-Value, Layout-data, Line, Lines, Lm-On-Columns, Mass-Update, Max-Height, Max-Width, Min-Height, Min-Width, Mouse-Wheel-Scroll, Pop-Up Menu, Pos, Position, Query-Index, Reset-List, Row-Background-Color-Pattern, Row-Color-Pattern, Row-Foreground-Color-Pattern, Rows-Selected, Selection-Background-Color, Selection-Color, Selection-Foreground-Color, Search-Text, Selection-Mode, Selection-Index, Separation, Size, Sort-Order, Termination-Value, Thumb-Position, Value, Visible.	3-D, Background-High, Background-Low, Background-Standard, Bold, Boxed, Check-List, Height-In-Cells, High, Highlight, Low, Lower, Lowlight, No-Box, No-Search, Notify-Dblclick, Notify-Selchange, Notify-Mouse, Paged, Permanent, Standard, Temporary, Unsorted, Upper, Width-In-Cells.	CMD-DBLCLICK, CMD-GOTO, CMD-HELP, MSG-END-MENU, MSG-INIT-MENU, MSG-MENU-INPUT, MSG-MOUSE-ENTER, MSG-MOUSE-EXIT, MSG-VALIDATE, NTF-PL-FIRST, NTF-PL-LAST, NTF-PL-NEXT, NTF-PL-NEXTPAGE, NTF-PL-PREV, NTF-PL-PREVPAGE, NTF-PL-SEARCH, NTF-SELCHANGE.
PUSH-BUTTON	Background-Color, Bitmap-Disabled, Bitmap-Handle, Bitmap-Number, Bitmap-Pressed, Bitmap-Rollover, Bitmap-Width, Col, Color, Column, Css-Style-Name, Custom-Data, Enabled, Event-List, Exception-Value, Exclude-Event-List, Font, Foreground-Color, Help-Id, Hint, Id, Layout-data, Line, Lines, Max-Height, Max-Width, Min-Height, Min-Width, Pop-Up Menu, Pos, Position, Size, Termination-Value, Title, Title-Position, Visible.	Background-High, Background-Low, Background-Standard, Bitmap, Bold, Bottom, Cancel-Button, Center, Default-Button, Escape-Button, Flat, Framed, Height-In-Cells, High, Highlight, Left, Low, Lowlight, Multiline, No-Auto-Default, No-Tab, Notify-Mouse, Ok-Button, On-Header, Permanent, Right, Self-Act, Square, Standard, Temporary, Top, Transparent, Unframed, Width-In-Cells.	CMD-CLICKED, CMD-GOTO, CMD-HELP, MSG-END-MENU, MSG-INIT-MENU, MSG-MENU-INPUT, MSG-MOUSE-ENTER, MSG-MOUSE-EXIT, MSG-VALIDATE.

RADIO-BUTTON	Background-Color, Bitmap-Disabled, Bitmap-Disabled-Selected, Bitmap-Handle, Bitmap-Number, Bitmap-Pressed, Bitmap-Rollover, Bitmap-Rollover-Selected, Bitmap-Width, Col, Color, Column, Css-Style-Name, Custom-Data, Enabled, Event-List, Exception-Value, Exclude-Event-List, Font, Foreground-Color, Group, Group-Value, Help-Id, Hint, Id, Layout-data, Line, Lines, Max-Height, Max-Width, Min-Height, Min-Width, Pop-Up Menu, Pos, Position, Size, Termination-Value, Title, Title-Position, Value, Visible.	Background-High, Background-Low, Background-Standard, Bitmap, Bold, Flat, Framed, Height-In-Cells, High, Highlight, Left-Text, Low, Lowlight, Multiline, No-Tab, Notify, Notify-Mouse, Permanent, Self-Act, Square, Standard, Temporary, Transparent, Unframed, Vtop, Width-In-Cells.	CMD-CLICKED, CMD-GOTO, CMD-HELP, MSG-END-MENU, MSG-INIT-MENU, MSG-MENU-INPUT, MSG-MOUSE-ENTER, MSG-MOUSE-EXIT, MSG-VALIDATE.
RIBBON	Background-Color, Bitmap-Handle, Bitmap-Number, Bitmap-Width, Collapse, Color, Css-Style-Name, Enabled, Event-List, Exclude-Event-List, Font, Foreground-Color, Header-Align, Hint, Id, Insertion-Index, Layout-Manager, Lines, Pop-Up Menu, Reset-Tabs, Tab-Enabled, Tab-Index, Tab-Text, Tab-To-Add, Tab-To-Delete, Value, Visible.	Background-High, Background-Low, Background-Standard, Bold, Height-In-Cells, High, Highlight, Low, Lowlight, Notify-Mouse, Permanent, Relative-Offset, Standard, Temporary, Width-In-Cells.	CMD-TABCHANGED, MSG-END-MENU, MSG-INIT-MENU, MSG-MENU-INPUT, MSG-MOUSE-CLICKED, MSG-MOUSE-DBLCLICK, MSG-MOUSE-ENTER, MSG-MOUSE-EXIT.
SCROLL-BAR	Background-Color, Col, Color, Column, Css-Style-Name, Custom-Data, Enabled, Event-List, Exclude-Event-List, Font, Foreground-Color, Help-Id, Hint, Id, Layout-data, Line, Lines, Max-Height, Max-Val, Max-Width, Min-Height, Min-Val, Min-Width, Page-Size, Pop-Up Menu, Pos, Position, Size, Visible.	Background-High, Background-Low, Background-Standard, Bold, Height-In-Cells, High, Highlight, Horizontal, Low, Lowlight, Notify-Mouse, Permanent, Standard, Temporary, Track-Thumb, Width-In-Cells.	CMD-GOTO, CMD-HELP, MSG-END-MENU, MSG-INIT-MENU, MSG-MENU-INPUT, MSG-MOUSE-ENTER, MSG-MOUSE-EXIT, MSG-SB-THUMB, MSG-VALIDATE.

SCROLL-PANE	Background-Color, Border-Color, Col, Color, Column, Css-Base-Style-Name, Css-Style-Name, Custom-Data, Enabled, Event-List, Exclude-Event-List, Font, Foreground-Color, Hint, Id, Line, Lines, Pos, Position, Size, Visible.	Background-High, Background-Low, Background-Standard, Bold, Boxed, Height-In-Cells, High, Highlight, Low, Lowlight, No-Box, Notify-Mouse, Permanent, Standard, Temporary, Transparent, Width-In-Cells.	MSG-MOUSE-CLICKED, MSG-MOUSE-DBLCLICK, MSG-MOUSE-ENTER, MSG-MOUSE-EXIT.
SLIDER	Background-Color, Col, Color, Column, Css-Style-Name, Custom-Data, Enabled, Event-List, Exclude-Event-List, Font, Foreground-Color, Help-Id, Hint, Id, Layout-data, Line, Lines, Major-Tick-Spacing, Max-Height, Max-Val, Max-Width, Min-Height, Min-Val, Min-Width, Minor-Tick-Spacing, Pop-Up Menu, Pos, Position, Size, Value, Visible.	Background-High, Background-Low, Background-Standard, Bold, Height-In-Cells, High, Highlight, Horizontal, Inverted, Low, Lowlight, Notify-Mouse, Permanent, Show-Labels, Show-Ticks, Standard, Temporary, Width-In-Cells.	CMD-GOTO, CMD-HELP, MSG-END-MENU, MSG-INIT-MENU, MSG-MENU-INPUT, MSG-MOUSE-ENTER, MSG-MOUSE-EXIT, MSG-SL-THUMB, MSG-VALIDATE.
STATUS-BAR	Background-Color, Col, Color, Column, Css-Style-Name, Custom-Data, Enabled, Event-List, Exclude-Event-List, Font, Foreground-Color, Help-Id, Hint, Layout-data, Line, Lines, Max-Height, Max-Width, Min-Height, Min-Width, Panel-Bitmap, Panel-Bitmap-Alignment, Panel-Background-Color, Panel-Bitmap-Number, Panel-Bitmap-Width, Panel-Color, Panel-Foreground-Color, Panel-Hint, Panel-Index, Panel-Style, Panel-Text, Panel-Widths, Pop-Up Menu, Pos, Position, Visible.	Background-High, Background-Low, Background-Standard, Bold, Grip, High, Highlight, Low, Lowlight, Notify-Mouse, Permanent, Standard, Temporary.	CMD-GOTO, CMD-HELP, MSG-END-MENU, MSG-INIT-MENU, MSG-MENU-INPUT, MSG-MOUSE-CLICKED, MSG-MOUSE-DBLCLICK, MSG-MOUSE-ENTER, MSG-MOUSE-EXIT, MSG-ST-DBLCLICK, MSG-VALIDATE.

TAB-CONTROL	Background-Color, Bitmap-Handle, Bitmap-Number, Bitmap-Width, Col, Color, Column, Css-Style-Name, Custom-Data, Enabled, Event-List, Exclude-Event-List, Font, Foreground-Color, Help-Id, Hint, Id, Insertion-Index, Line, Layout-data, Lines, Max-Height, Max-Width, Min-Height, Min-Width, Pop-Up Menu, Pos, Position, Reset-Tabs, Size, Tab-Alignment, Tab-Background-color, Tab-Color, Tab-Delay, Tab-Enabled, Tab-Foreground-Color, Tab-Index, Tab-Rollover-Color, Tab-Text, Tab-To-Add, Tab-To-Delete, Value, Visible.	Accordion, Allow-Container, Background-High, Background-Low, Background-Standard, Bold, Bottom, Height-In-Cells, High, Highlight, Low, Lowlight, Multiline, Notify-Mouse, Permanent, Relative-Offset, Standard, Tab-Flat, Temporary, Vertical, Width-In-Cells.	CMD-HELP, CMD-TABCHANGED, MSG-END-MENU, MSG-INIT-MENU, MSG-MENU-INPUT, MSG-MOUSE-CLICKED, MSG-MOUSE-DBLCLICK, MSG-MOUSE-ENTER, MSG-MOUSE-EXIT, MSG-VALIDATE.
TOOL-BAR	Background-Color, Cell Height, Cell Size, Cell Width, Color, Control Font, Custom-Data, Foreground-Color, Help-Id, Hint, Id, Layout-Mana, Lines, Pop-Up Menu.	Background-High, Background-Low, Background-Standard, Bold, High, Highlight, Low, Lowlight, Moveable, Multiline, Standard.	MSG-END-MENU, MSG-INIT-MENU, MSG-MENU-INPUT.

TREE-VIEW	Action, Alignment, Background-Color, Bitmap-Handle, Bitmap-Number, Bitmap-Width, Border-Color, Border-Width, Col, Color, Column, Column-Hiding, Css-Style-Name, Custom-Data, Data-Columns, Display-Columns, Enabled, End-Color, Ensure-Visible, Event-List, Exclude-Event-List, Expand, Font, Foreground-Color, Has-Children, Heading-Background-Color, Heading-Color, Heading-Font, Heading-Foreground-Color, Heading-Menu-Popup, Help-Id, Hidden-Data, Hint, Id, Item, Item-Background-Color, Item-Color, Item-Foreground-Color, Item-Hint, Item-Text, Item-To-Add, Item-To-Delete, Item-To-Empty, Layout-data, Line, Lines, Lm-On-Columns, Mass-Update, Max-Height, Max-Width, Min-Height, Min-Width, Next-Item, Parent, Placement, Pop-Up Menu, Pos, Position, Reset-List, Selection-Background-Color, Selection-Color, Selection-Foreground-Color, Size, Sort-Types, Value, Virtual-Width, Visible, X.	3-D, Adjustable-Columns, Background-High, Background-Low, Background-Standard, Bold, Boxed, Buttons, Centered-Headings, Column-Headings, Height-In-Cells, High, Highlight, Lines-At-Root, Low, Lowlight, No-Box, Notify-Mouse, Permanent, Reordering-Columns, Show-Lines, Show-Sel-Always, Sortable-Columns, Standard, Temporary, Tiled-Headings, Width-In-Cells.	CMD-GOTO, CMD-HELP, MSG-BEGIN-ENTRY, MSG-CANCEL-ENTRY, MSG-END-MENU, MSG-FINISH-ENTRY, MSG-INIT-MENU, MSG-MENU-INPUT, MSG-MOUSE-ENTER, MSG-MOUSE-EXIT, MSG-TV-DBLCLICK, MSG-TV-EXPANDED, MSG-TV-EXPANDING, MSG-TV-SELCHANGE, MSG-TV-SELCHANGE-OUT-NEXT, MSG-TV-SELCHANGE-OUT-PREV, MSG-TV-SELCHANGING, MSG-VALIDATE.
WEB-BROWSER	Busy, Col, Column, Css-Style-Name, Custom-Data, Enabled, Event-List, Exclude-Event-List, Font, Go-Back, Go-Forward, Go-Home, Go-Search, Help-Id, Hint, Id, Layout-data, Line, Lines, Max-Height, Max-Progress, Max-Width, Min-Height, Min-Width, Pos, Position, Progress, Refresh, Size, Status-Text, Stop-Browser, Title, Value, Visible.	Background-High, Background-Low, Background-Standard, Bold, Height-In-Cells, High, Highlight, Low, Lowlight, No-Msg-Before-Navigate, Notify-Mouse, Permanent, Standard, Temporary, Use-Alt, Use-Return, Width-In-Cells.	MSG-MOUSE-ENTER, MSG-MOUSE-EXIT, MSG-WB-BEFORE-NAVIGATE, MSG-WB-DOWNLOAD-BEGIN, MSG-WB-DOWNLOAD-COMPLETE, MSG-WB-NAVIGATE-COMPLETE, MSG-WB-PROGRESS-CHANGE, MSG-WB-STATUS-TEXT-CHANGE, MSG-WB-TITLE-CHANGE.

WINDOW	Action, Background-Color, Cell Height, Cell Size, Cell Width, Col, Color, Column, Control Font, Custom-Data, Enabled, Font, Foreground-Color, Gradient-Color-1, Gradient-Color-2, Gradient-Orientation, Help-Id, Hint, Icon, Layout-manager, Line, Lines, Max-Lines, Max-Size, Min-Lines, Min-Size, Pop-Up Menu, Pos, Position, Screen-Index, Screen Col, Screen Column, Screen Line, Screen Pos, Screen Position, Size, Title, Visible.	Auto-Resize, Background-High, Background-Low, Background-Standard, Bind To Thread, Blank, Bold, Boxed, Controls-Uncropped, High, Highlight, Link To Thread, Low, Lowlight, Modal, Modeless, No Scroll, No Wrap, No-Close, Permanent, Resizable, Reverse, Shadow, Standard, System Menu, Temporary, Title-Bar, User-Colors, User-Gray, User-White.	CMD-ACTIVATE, CMD-CLOSE, MSG-CLOSE, MSG-DEICONIFIED, MSG-END-MENU, MSG-ICONIFIED, MSG-INIT-MENU, MSG-MENU-INPUT, NTF-RESIZED.

Table 2

This table shows the list of all graphical controls for each property.

Action	ENTRY-FIELD, GRID, LIST-BOX, TREE-VIEW, WINDOW
Alignment	GRID, LIST-BOX, TREE-VIEW
Auto-Decimal	ENTRY-FIELD
Background-Color	BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, RIBBON, SCROLL-BAR, SCROLL-PANE, SLIDER, STATUS-BAR, TAB-CONTROL, TOOL-BAR, TREE-VIEW, WINDOW
Bitmap	GRID
Bitmap-Disabled	CHECK-BOX, ENTRY-FIELD, PUSH-BUTTON, RADIO-BUTTON
Bitmap-Disabled-Selected	CHECK-BOX, PUSH-BUTTON, RADIO-BUTTON
Bitmap-End	BITMAP
Bitmap-Handle	BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, PUSH-BUTTON, RADIO-BUTTON, TAB-CONTROL, TREE-VIEW
Bitmap-Hint	ENTRY-FIELD
Bitmap-Number	BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, GRID, PUSH-BUTTON, RADIO-BUTTON, TAB-CONTROL, TREE-VIEW
Bitmap-Pressed	CHECK-BOX, PUSH-BUTTON, RADIO-BUTTON
Bitmap-Rollover	CHECK-BOX, ENTRY-FIELD, PUSH-BUTTON, RADIO-BUTTON
Bitmap-Rollover-Selected	CHECK-BOX, RADIO-BUTTON

Bitmap-Start	BITMAP
Bitmap-Timer	BITMAP
Bitmap-Trailing	GRID
Bitmap-Trailing-Disabled	ENTRY-FIELD
Bitmap-Trailing-Hint	ENTRY-FIELD
Bitmap-Trailing-Number	ENTRY-FIELD
Bitmap-Trailing-Rollover	ENTRY-FIELD
Bitmap-Width	CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, GRID, PUSH-BUTTON, RADIO-BUTTON, TAB-CONTROL, TREE-VIEW
Border-Color	ENTRY-FIELD, GRID, JAVA-BEAN, LIST-BOX, TREE-VIEW
Border-Width	ENTRY-FIELD, GRID, JAVA-BEAN, LIST-BOX, TREE-VIEW
Busy	WEB-BROWSER
Cell Height	TOOL-BAR, WINDOW
Cell Size	TOOL-BAR, WINDOW
Cell Width	TOOL-BAR, WINDOW
Cell-Background-Color	GRID
Cell-Color	GRID
Cell-Columns-Span	GRID
Cell-Current-Background-Color	GRID
Cell-Current-Color	GRID
Cell-Current-Font	GRID
Cell-Current-Foreground-Color	GRID
Cell-Current-Protection	GRID
Cell-Data	GRID
Cell-Entry-Background-Color	GRID
Cell-Entry-Color	GRID
Cell-Entry-Foreground-Color	GRID
Cell-Font	GRID
Cell-Foreground-Color	GRID

Cell-Hint	GRID
Cell-Protection	GRID
Cell-Rows-Span	GRID
Cell-Selected-Background-Color	GRID
Cell-Selected-Color	GRID
Cell-Selected-Foreground-Color	GRID
Cells-Selected	GRID
Clsid	JAVA-BEAN
Col	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, SCROLL-BAR, SCROLL-PANE, SLIDER, STATUS-BAR, TAB-CONTROL, TREE-VIEW, WEB-BROWSER, WINDOW
Color	BAR, CHECK-BOX. COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, SCROLL-BAR, SCROLL-PANE, SLIDER, STATUS-BAR, TAB-CONTROL, TOOL-BAR, TREE-VIEW, WINDOW
Colors	BAR
Column	BAR, BITMAP, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, SCROLL-BAR, SCROLL-PANE, SLIDER, STATUS-BAR, TAB-CONTROL, TREE-VIEW, WEB-BROWSER, WINDOW, CHECK-BOX
Column-Background-Color	GRID
Column-Color	GRID
Column-Dividers	GRID
Column-Font	GRID
Column-Foreground-Color	GRID
Column-Headings-Height	GRID
Column-Headings-Layout	GRID
Column-Hiding	GRID, TREE-VIEW
Column-Protection	GRID
Column-Selected-Background-Color	GRID
Column-Selected-Color	GRID
Column-Selected-Foreground-Color	GRID

Columns-Selected	GRID
Control Font	TOOL-BAR, WINDOW
Css-Base-Style-Name	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, SCROLL-BAR, SCROLL-PANE, SLIDER, STATUS-BAR, TAB-CONTROL, TREE-VIEW, WEB-BROWSER
Css-Style-Name	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, SCROLL-BAR, SCROLL-PANE, SLIDER, STATUS-BAR, TAB-CONTROL, TREE-VIEW, WEB-BROWSER
Cursor	ENTRY-FIELD, COMBO-BOX
Cursor-Background-Color	GRID
Cursor-Col	ENTRY-FIELD
Cursor-Color	GRID
Cursor-Foreground-Color	GRID
Cursor-Frame-Width	GRID
Cursor-Row	ENTRY-FIELD
Cursor-X	GRID
Cursor-Y	GRID
Custom-Data	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, SCROLL-BAR, SCROLL-PANE, SLIDER, STATUS-BAR, TAB-CONTROL, TOOL-BAR, TREE-VIEW, WEB-BROWSER, WINDOW
Data-Columns	GRID, LIST-BOX, TREE-VIEW
Data-Types	GRID
Decoration-Background	DATE-ENTRY
Display-Columns	GRID, LIST-BOX, TREE-VIEW
Display-Format	DATE-ENTRY
Divider-Color	GRID
Dividers	LIST-BOX
Drag-Background-Color	GRID
Drag-Color	GRID
Drag-Foreground-Color	GRID

Editor-Show-Always	GRID
Enabled	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, SCROLL-BAR, SCROLL-PANE, SLIDER, STATUS-BAR, TAB-CONTROL, TREE-VIEW, WEB-BROWSER, WINDOW
End-Color	GRID, TREE-VIEW
Ensure-Visible	TREE-VIEW
Entry-Reason	GRID
Event-List	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, RIBBON, SCROLL-BAR, SCROLL-PANE, SLIDER, STATUS-BAR, TAB-CONTROL, TREE-VIEW, WEB-BROWSER
Exception-Value	CHECK-BOX, COMBO-BOX, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON
Exclude-Event-List	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, RIBBON, SCROLL-BAR, SCROLL-PANE, SLIDER, STATUS-BAR, TAB-CONTROL, TREE-VIEW, WEB-BROWSER
Expand	TREE-VIEW
Export-File-Format	GRID, LIST-BOX
Export-File-Name	GRID, LIST-BOX
File-Pos	GRID
Fill-Char	ENTRY-FIELD
Fill-Color	FRAME
Fill-Color2	FRAME
Fill-Percent	FRAME
Finish-Reason	GRID
Font	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, RIBBON, SCROLL-BAR, SCROLL-PANE, SLIDER, STATUS-BAR, TAB-CONTROL, TREE-VIEW, WEB-BROWSER, WINDOW
Foreground-Color	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, RIBBON, SCROLL-BAR, SCROLL-PANE, SLIDER, STATUS-BAR, TAB-CONTROL, TOOL-BAR, TREE-VIEW, WINDOW
Format-String	ENTRY-FIELD
Go-Back	WEB-BROWSER

Go-Forward	WEB-BROWSER
Go-Home	WEB-BROWSER
Go-Search	WEB-BROWSER
Gradient-Color-1	WINDOW
Gradient-Color-2	WINDOW
Gradient-Orientation	WINDOW
Group	RADIO-BUTTON
Group-Value	RADIO-BUTTON
Has-Children	TREE-VIEW
Heading-Background-Color	GRID, TREE-VIEW
Heading-Color	GRID, TREE-VIEW
Heading-Cursor-Background-Color	GRID
Heading-Cursor-Color	GRID
Heading-Cursor-Foreground-Color	GRID
Heading-Divider-Color	GRID
Heading-Font	GRID, TREE-VIEW
Heading-Foreground-Color	GRID, TREE-VIEW
Heading-Menu-Popup	GRID, TREE-VIEW
Help-Id	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, SCROLL-BAR, SLIDER, STATUS-BAR, TAB-CONTROL, TOOL-BAR, TREE-VIEW, WEB-BROWSER
Hidden-Data	COMBO-BOX, LIST-BOX, GRID, TREE-VIEW
High-Color	FRAME
Hint	BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, SCROLL-BAR, SCROLL-PANE, SLIDER, STATUS-BAR, TAB-CONTROL, TOOL-BAR, TREE-VIEW, WEB-BROWSER
Hscroll-Pos	GRID
Icon	WINDOW

Id	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, RIBBON, SCROLL-BAR, SCROLL-PANE, SLIDER, TAB-CONTROL, TOOL-BAR, TREE-VIEW, WEB-BROWSER
Init-Params	JAVA-BEAN
Init-Signature	JAVA-BEAN
Insertion-Index	COMBO-BOX, GRID, LIST-BOX, RIBBON, TAB-CONTROL
Insert-Rows	GRID
Item	COMBO-BOX, TREE-VIEW
Item-Background-Color	COMBO-BOX, LIST-BOX, TREE-VIEW
Item-Color	COMBO-BOX, LIST-BOX, TREE-VIEW
Item-Foreground-Color	COMBO-BOX, LIST-BOX, TREE-VIEW
Item-Height	COMBO-BOX
Item-Hint	TREE-VIEW
Item-Text	COMBO-BOX, TREE-VIEW
Item-To-Add	COMBO-BOX, LIST-BOX, TREE-VIEW
Item-To-Delete	COMBO-BOX, LIST-BOX, TREE-VIEW
Item-To-Empty	TREE-VIEW
Item-Value	LIST-BOX
Label-Offset	LABEL
Last-Row	GRID
Layout-data	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, SCROLL-BAR, SLIDER, STATUS-BAR, TAB-CONTROL, TREE-VIEW, WEB-BROWSER
Layout-manager	RIBBON, TOOL-BAR, WINDOW
Leading-Shift	BAR
Line	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, SCROLL-BAR, SCROLL-PANE, SLIDER, STATUS-BAR, TAB-CONTROL, TREE-VIEW, WEB-BROWSER

Lines	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, SCROLL-BAR, SCROLL-PANE, SLIDER, STATUS-BAR, TAB-CONTROL, TOOL-BAR, TREE-VIEW, WEB-BROWSER, WINDOW
Lm-On-Columns	GRID, LIST-BOX, TREE-VIEW
Low-Color	FRAME
Major-Tick-Spacing	SLIDER
Mass-Update	COMBO-BOX, GRID, LIST-BOX, TREE-VIEW
Maxday-Characters	DATE-ENTRY
Max-Height	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, SCROLL-BAR, SLIDER, STATUS-BAR, TAB-CONTROL, TREE-VIEW, WEB-BROWSER
Max-Lines	ENTRY-FIELD, WINDOW
Max-Progress	WEB-BROWSER
Max-Size	WINDOW
Max-Text	COMBO-BOX, ENTRY-FIELD
Max-Val	DATE-ENTRY, ENTRY-FIELD, SCROLL-BAR, SLIDER
Max-Width	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, SCROLL-BAR, SLIDER, STATUS-BAR, TAB-CONTROL, TREE-VIEW, WEB-BROWSER
Min-Height	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, SCROLL-BAR, SLIDER, STATUS-BAR, TAB-CONTROL, TREE-VIEW, WEB-BROWSER
Min-Lines	WINDOW
Minor-Tick-Spacing	SLIDER
Min-Size	WINDOW
Min-Val	DATE-ENTRY, ENTRY-FIELD, SCROLL-BAR, SLIDER
Min-Width	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, SCROLL-BAR, SLIDER, STATUS-BAR, TAB-CONTROL, TREE-VIEW, WEB-BROWSER
Model-To-View-Y	GRID
Mouse-Wheel-Scroll	GRID, LIST-BOX

Next-Item	TREE-VIEW
Notify-Change-Delay	ENTRY-FIELD
Num-Col-Headings	GRID
Num-Row-Headings	GRID
Num-Rows	GRID
Object	JAVA-BEAN
Page-Size	SCROLL-BAR
Panel-Background-Color	STATUS-BAR
Panel-Bitmap	STATUS-BAR
Panel-Bitmap-Alignment	STATUS-BAR
Panel-Bitmap-Number	STATUS-BAR
Panel-Bitmap-Width	STATUS-BAR
Panel-Color	STATUS-BAR
Panel-Foreground-Color	STATUS-BAR
Panel-Hint	STATUS-BAR
Panel-Index	STATUS-BAR
Panel-Style	STATUS-BAR
Panel-Text	STATUS-BAR
Panel-Widths	STATUS-BAR
Parent	TREE-VIEW
Placeholder	COMBO-BOX, ENTRY-FIELD
Placement	TREE-VIEW
Pop-Up Menu	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, RIBBON, SCROLL-BAR, SLIDER, STATUS-BAR, TAB-CONTROL, TOOL-BAR, WINDOW
Pos	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, RIBBON, SCROLL-BAR, SCROLL-PANE, SLIDER, STATUS-BAR, TAB-CONTROL, TREE-VIEW, WEB-BROWSER, WINDOW

Position	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, RIBBON, SCROLL-BAR, SCROLL-PANE, SLIDER, STATUS-BAR, TAB-CONTROL, TREE-VIEW, WEB-BROWSER, WINDOW
Position-Shift	BAR
Progress	WEB-BROWSER
Proposal	ENTRY-FIELD
Proposal-Delay	ENTRY-FIELD
Proposal-Index	ENTRY-FIELD
Proposal-Min-Text	ENTRY-FIELD
Proposal-To-Delete	ENTRY-FIELD
Protection	GRID
Query-Index	COMBO-BOX, LIST-BOX
Record-Data	GRID
Record-To-Add	GRID
Record-To-Delete	GRID
Refresh	WEB-BROWSER
Region-Background-Color	GRID
Region-Color	GRID
Region-Foreground-Color	GRID
Reordering-Col-Index	GRID
Reset-Grid	GRID
Reset-List	COMBO-BOX, LIST-BOX, TREE-VIEW
Reset-Proposals	ENTRY-FIELD
Reset-Tabs	TAB-CONTROL
Row-Background-Color	GRID
Row-Background-Color-Pattern	GRID, LIST-BOX
Row-Capacity	GRID
Row-Color	GRID
Row-Color-Pattern	GRID, LIST-BOX

Row-Cursor-Background-Color	GRID
Row-Cursor-Color	GRID
Row-Cursor-Foreground-Color	GRID
Row-Dividers	GRID
Row-Font	GRID
Row-Foreground-Color	GRID
Row-Foreground-Color-Pattern	GRID, LIST-BOX
Row-Hiding	GRID
Row-Protection	GRID
Row-Selected-Background-Color	GRID
Row-Selected-Color	GRID
Row-Selected-Foreground-Color	GRID
Rows-Per-Page	GRID
Rows-Selected	GRID, LIST-BOX
Screen Col	WINDOW
Screen Column	WINDOW
Screen Line	WINDOW
Screen Pos	WINDOW
Screen Position	WINDOW
Screen-Index	WINDOW
Search-Options	GRID
Search-Panel	GRID
Search-Text	GRID, LIST-BOX
Search-Text-In-View	GRID
Selection-Background-Color	COMBO-BOX, LIST-BOX, TREE-VIEW
Selection-Color	COMBO-BOX, LIST-BOX, TREE-VIEW
Selection-Foreground-Color	COMBO-BOX, LIST-BOX, TREE-VIEW
Selection-Index	LIST-BOX
Selection-Mode	GRID, LIST-BOX

Selection-Text	ENTRY-FIELD
Separation	GRID, LIST-BOX
Shading	BAR
Size	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, SCROLL-BAR, SCROLL-PANE, SLIDER, TAB-CONTROL, TREE-VIEW, WEB-BROWSER, WINDOW
Sort-data	GRID
Sort-types	GRID, TREE-VIEW
Sort-Order	LIST-BOX
Spell-Checking	ENTRY-FIELD
Start-X	GRID
Start-Y	GRID
Status-Text	WEB-BROWSER
Stop-Browser	WEB-BROWSER
Sunday-Foreground	DATE-ENTRY
Tab-Alignment	TAB-CONTROL
Tab-Background-Color	TAB-CONTROL
Tab-Color	TAB-CONTROL
Tab-Delay	TAB-CONTROL
Tab-Enabled	TAB-CONTROL
Tab-Foreground-Color	TAB-CONTROL
Tab-Index	RIBBON, TAB-CONTROL
Tab-Text	RIBBON, TAB-CONTROL
Tab-To-Add	RIBBON, TAB-CONTROL
Tab-To-Delete	RIBBON, TAB-CONTROL
Tab-Rollover-Color	TAB-CONTROL
Termination-Value	CHECK-BOX, COMBO-BOX, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON
Text-Orientation	ENTRY-FIELD
Thumb-Position	LIST-BOX

Title	CHECK-BOX, FRAME, LABEL, PUSH-BUTTON, RADIO-BUTTON, WEB-BROWSER, WINDOW
Title-Position	CHECK-BOX, FRAME, PUSH-BUTTON, RADIO-BUTTON
Trailing-Shift	BAR
Transparent-Color	BITMAP
Validation-Errmsg	ENTRY-FIELD
Validation-Opts	ENTRY-FIELD
Validation-Regexp	ENTRY-FIELD
Value	CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, LIST-BOX, RADIO-BUTTON, RIBBON, SLIDER, TAB-CONTROL, TREE-VIEW, WEB-BROWSER
Value-Format	DATE-ENTRY
View-Cursor-Y	GRID
View-To-Model-Y	GRID
Virtual-Width	GRID
Visible	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, RIBBON, SCROLL-BAR, SCROLL-PANE, SLIDER, STATUS-BAR, TAB-CONTROL, TREE-VIEW, WEB-BROWSER, WINDOW
Visible-Proposal-Count	ENTRY-FIELD
VPadding	GRID
Vscroll-Pos	GRID
Weekday-Foreground	DATE-ENTRY
Width	BAR
X	GRID, TREE-VIEW
Y	GRID

Table 3

This table shows the list of all graphical controls for each style.

3-D	ENTRY-FIELD, GRID, JAVA-BEAN, LIST-BOX, SCROLL-PANE, TREE-VIEW
Adjustable-Columns	GRID, TREE-VIEW

Accordion	TAB-CONTROL
Allow-Container	TAB-CONTROL
Allow-Empty	DATE-ENTRY
Alternate	FRAME
Auto	ENTRY-FIELD
Auto-Resize	WINDOW
Auto-Spin	ENTRY-FIELD
Background-High	BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, RIBBON, SCROLL-BAR, SLIDER, STATUS-BAR, SCROLL-PANE, TAB-CONTROL, TOOL-BAR, TREE-VIEW, WEB-BROWSER, WINDOW
Background-Low	BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, RIBBON, SCROLL-BAR, SLIDER, STATUS-BAR, SCROLL-PANE, TAB-CONTROL, TOOL-BAR, TREE-VIEW, WEB-BROWSER, WINDOW
Background-Standard	BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, RIBBON, SCROLL-BAR, SLIDER, STATUS-BAR, SCROLL-PANE, TAB-CONTROL, TOOL-BAR, TREE-VIEW, WEB-BROWSER, WINDOW
Bind To Thread	WINDOW
Bitmap	CHECK-BOX, PUSH-BUTTON, RADIO-BUTTON
Blank	WINDOW
Bold	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, RIBBON, SCROLL-BAR, SLIDER, STATUS-BAR, SCROLL-PANE, TAB-CONTROL, TOOL-BAR, TREE-VIEW, WEB-BROWSER, WINDOW
Bottom	LABEL, PUSH-BUTTON, TAB-CONTROL
Boxed	ENTRY-FIELD, GRID, JAVA-BEAN, LIST-BOX, SCROLL-PANE, TREE-VIEW, WINDOW
Buttons	TREE-VIEW
Cancel-Button	PUSH-BUTTON
Center	ENTRY-FIELD, LABEL, PUSH-BUTTON
Centered	ENTRY-FIELD, LABEL
Centered-Headings	GRID, TREE-VIEW
Century-Date	DATE-ENTRY

Check-List	LIST-BOX
Column-Headings	GRID, TREE-VIEW
Controls-Uncropped	WINDOW
Dashed	BAR
Decoration-Background-Visible	DATE-ENTRY
Decoration-Borders-Visible	DATE-ENTRY
Default-Button	PUSH-BUTTON
Dot-Dash	BAR
Dotted	BAR
Drop-Down	COMBO-BOX
Drop-List	COMBO-BOX
Engraved	FRAME
Escape-Button	PUSH-BUTTON
Filterable-Columns	GRID
Flat	CHECK-BOX, PUSH-BUTTON, RADIO-BUTTON
Framed	CHECK-BOX, PUSH-BUTTON, RADIO-BUTTON
Full-Height	FRAME
Grip	STATUS-BAR
Heavy	FRAME
Height-In-Cells	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, RIBBON, SCROLL-BAR, SCROLL-PANE, SLIDER, TAB-CONTROL, TREE-VIEW, WEB-BROWSER
High	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, RIBBON SCROLL-BAR, SCROLL-PANE, SLIDER, STATUS-BAR, TAB-CONTROL, TOOL-BAR, TREE-VIEW, WEB-BROWSER, WINDOW
Highlight	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, RIBBON, SCROLL-BAR, SCROLL-PANE, SLIDER, STATUS-BAR, TAB-CONTROL, TOOL-BAR, TREE-VIEW, WEB-BROWSER, WINDOW
Horizontal	SCROLL-BAR, SLIDER
Hscroll	GRID

Inverted	SLIDER
Left	ENTRY-FIELD, LABEL, PUSH-BUTTON
Left-Text	CHECK-BOX, RADIO-BUTTON
Lines-At-Root	TREE-VIEW
Link To Thread	WINDOW
Long-Date	DATE-ENTRY
Low	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, RIBBON, SCROLL-BAR, SCROLL-PANE, SLIDER, STATUS-BAR, TAB-CONTROL, TOOL-BAR, TREE-VIEW, WEB-BROWSER, WINDOW
Lower	COMBO-BOX, ENTRY-FIELD, LIST-BOX
Lowered	FRAME
Lowlight	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, RIBBON, SCROLL-BAR, SCROLL-PANE, SLIDER, STATUS-BAR, TAB-CONTROL, TOOL-BAR, TREE-VIEW, WEB-BROWSER, WINDOW
Modal	WINDOW
Modeless	WINDOW
Moveable	TOOL-BAR
Multiline	CHECK-BOX, ENTRY-FIELD, PUSH-BUTTON, RADIO-BUTTON, TAB-CONTROL, TOOL-BAR
No Scroll	WINDOW
No Wrap	WINDOW
No-Auto-Default	PUSH-BUTTON
No-Autosel	ENTRY-FIELD, GRID
No-Box	ENTRY-FIELD, GRID, JAVA-BEAN, LIST-BOX, TREE-VIEW
No-Cell-Drag	GRID
No-Close	WINDOW
No-F4	DATE-ENTRY
No-Key-Letter	LABEL
No-Msg-Before-Navigate	WEB-BROWSER
No-Search	GRID, LIST-BOX

No-Tab	CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, PUSH-BUTTON, RADIO-BUTTON
No-Wrap	ENTRY-FIELD
Notify	CHECK-BOX, RADIO-BUTTON
Notify-Change	DATE-ENTRY, ENTRY-FIELD
Notify-Dblclick	COMBO-BOX, LIST-BOX
Notify-Mouse	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, RIBBON, SCROLL-BAR, SCROLL-PANE, SLIDER, TAB-CONTROL, TREE-VIEW, WEB-BROWSER
Notify-Selchange	COMBO-BOX, LIST-BOX
No-Updown	DATE-ENTRY
Numeric	ENTRY-FIELD, DATE-ENTRY
Ok-Button	PUSH-BUTTON
On-Header	PUSH-BUTTON
Paged	GRID, LIST-BOX
Permanent	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, RIBBON, SCROLL-BAR, SCROLL-PANE, SLIDER, STATUS-BAR, TAB-CONTROL, TREE-VIEW, WEB-BROWSER, WINDOW
Raised	FRAME
Read-Only	ENTRY-FIELD, DATE-ENTRY
Relative-Offset	TAB-CONTROL, RIBBON
Reordering-Columns	GRID, TREE-VIEW
Resizable	WINDOW
Reverse	WINDOW
Right	ENTRY-FIELD, LABEL, PUSH-BUTTON
Right-Align	DATE-ENTRY
Rimmed	FRAME
Row-Headings	GRID
Secure	ENTRY-FIELD
Self-Act	CHECK-BOX, PUSH-BUTTON, RADIO-BUTTON, JAVA-BEAN

Shadow	WINDOW
Short-Date	DATE-ENTRY
Show-Labels	SLIDER
Show-Lines	TREE-VIEW
Show-Sel-Always	TREE-VIEW
Show-Ticks	SLIDER
Sortable-Columns	GRID, TREE-VIEW
Spinner	DATE-ENTRY, ENTRY-FIELD
Square	CHECK-BOX, PUSH-BUTTON, RADIO-BUTTON
Standard	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, RIBBON, SCROLL-BAR, SCROLL-PANE, SLIDER, STATUS-BAR, TAB-CONTROL, TOOL-BAR, TREE-VIEW, WEB-BROWSER, WINDOW
Static-List	COMBO-BOX
System Menu	WINDOW
Temporary	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, RIBBON, SCROLL-BAR, SCROLL-PANE, SLIDER, STATUS-BAR, TAB-CONTROL, TREE-VIEW, WEB-BROWSER, WINDOW
Tiled-Headings	GRID, TREE-VIEW
Time	DATE-ENTRY
Title-Bar	WINDOW
Top	LABEL, PUSH-BUTTON
Track-Thumb	SCROLL-BAR
Transparent	CHECK-BOX, FRAME, LABEL, PUSH-BUTTON, RADIO-BUTTON, SCROLL-PANE
Unframed	CHECK-BOX, PUSH-BUTTON, RADIO-BUTTON
Unsorted	COMBO-BOX, LIST-BOX
Upper	COMBO-BOX, ENTRY-FIELD, LIST-BOX
User-Colors	WINDOW
Use-Alt	WEB-BROWSER
Use-Return	ENTRY-FIELD, JAVA-BEAN, WEB-BROWSER

User-Gray	WINDOW
User-White	WINDOW
Use-Tab	ENTRY-FIELD, GRID, JAVA-BEAN, WEB-BROWSER
Vertical	LABEL, TAB-CONTROL
Very-Heavy	FRAME
Vscroll	ENTRY-FIELD, GRID
Vscroll-Bar	ENTRY-FIELD
Vtop	CHECK-BOX, RADIO-BUTTON
Week-Of-Year-Visible	DATE-ENTRY
Width-In-Cells	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, RIBBON, SCROLL-BAR, SCROLL-PANE, SLIDER, TAB-CONTROL, TREE-VIEW, WEB-BROWSER

Table 4

This table shows the list of all graphical controls for each event.

CMD-ACTIVATE	WINDOW
CMD-CLICKED	CHECK-BOX, PUSH-BUTTON, RADIO-BUTTON
CMD-CLOSE	WINDOW
CMD-DBLCLICK	COMBO-BOX, LIST-BOX
CMD-GOTO	CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, GRID, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, SCROLL-BAR, SLIDER, STATUS-BAR, TAB-CONTROL, TREE-VIEW
CMD-HELP	CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, GRID, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, SCROLL-BAR, SLIDER, STATUS-BAR, TAB-CONTROL, TREE-VIEW
CMD-TABCHANGED	TAB-CONTROL, RIBBON
MSG-BEGIN-DRAG	GRID
MSG-BEGIN-ENTRY	GRID, TREE-VIEW
MSG-BEGIN-HEADING-DRAG	GRID
MSG-BEGIN-HEADING-MENU-POPUP	GRID
MSG-BEGIN-SORT	GRID

MSG-BITMAP-CLICKED	ENTRY-FIELD, GRID
MSG-BITMAP-DBLCLICK	ENTRY-FIELD, GRID
MSG-CANCEL-ENTRY	GRID, TREE-VIEW
MSG-CLOSE	WINDOW
MSG-COL-WIDTH-CHANGED	GRID
MSG-DEICONIFIED	WINDOW
MSG-END-DRAG	GRID
MSG-END-HEADING-DRAG	GRID
MSG-END-MENU	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, RIBBON, SCROLL-BAR, SLIDER, STATUS-BAR, TAB-CONTROL, TOOL-BAR, TREE-VIEW, WINDOW
MSG-FINISH-ENTRY	GRID, TREE-VIEW
MSG-FINISH-SORT	GRID
MSG-GOTO-CELL	GRID
MSG-GD-DBLCLICK	GRID
MSG-GOTO-CELL-DRAG	GRID
MSG-GOTO-CELL-MOUSE	GRID
MSG-GOTO-CELL-OUT-NEXT	GRID
MSG-GOTO-CELL-OUT-PREV	GRID
MSG-GRID-RBUTTON-DOWN	GRID
MSG-GRID-RBUTTON-UP	GRID
MSG-HEADING-CLICKED	GRID
MSG-HEADING-DBLCLICK	GRID
MSG-HEADING-DRAGGED	GRID
MSG-HEADING-MENU-POPUP	GRID
MSG-ICONIFIED	WINDOW
MSG-INIT-MENU	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, RIBBON, SCROLL-BAR, SLIDER, STATUS-BAR, TAB-CONTROL, TOOL-BAR, TREE-VIEW, WINDOW
MSG-JB-EVENT	JAVA-BEAN

MSG-MENU-INPUT	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, RIBBON, SCROLL-BAR, SLIDER, STATUS-BAR, TAB-CONTROL, TOOL-BAR, TREE-VIEW, WINDOW
MSG-MOUSE-CLICKED	BAR, BITMAP, FRAME, LABEL, RIBBON, SCROLL-PANE, STATUS-BAR, TAB-CONTROL
MSG-MOUSE-DBLCLICK	BAR, BITMAP, FRAME, LABEL, RIBBON, SCROLL-PANE, STATUS-BAR, TAB-CONTROL
MSG-MOUSE-ENTER	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, RIBBON, SCROLL-BAR, SCROLL-PANE, SLIDER, STATUS-BAR, TAB-CONTROL, TREE-VIEW, WEB-BROWSER
MSG-MOUSE-EXIT	BAR, BITMAP, CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, FRAME, GRID, JAVA-BEAN, LABEL, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, RIBBON, SCROLL-BAR, SCROLL-PANE, SLIDER, STATUS-BAR, TAB-CONTROL, TREE-VIEW, WEB-BROWSER
MSG-PAGED-FIRST	GRID
MSG-PAGED-LAST	GRID
MSG-PAGED-NEXT	GRID
MSG-PAGED-NEXTPAGE	GRID
MSG-PAGED-PREV	GRID
MSG-PAGED-PREVPAGE	GRID
MSG-ROW-HEIGHT-CHANGED	GRID
MSG-SB-THUMB	SCROLL-BAR
MSG-SL-THUMB	SLIDER
MSG-SPIN-DOWN	ENTRY-FIELD
MSG-SPIN-UP	ENTRY-FIELD
MSG-ST-DBLCLICK	STATUS-BAR
MSG-TV-DBLCLICK	TREE-VIEW
MSG-TV-EXPANDED	TREE-VIEW
MSG-TV-EXPANDING	TREE-VIEW
MSG-TV-SELCHANGE	TREE-VIEW
MSG-TV-SELCHANGE-OUT-NEXT	TREE-VIEW
MSG-TV-SELCHANGE-OUT-PREV	TREE-VIEW

MSG-TV-SELCHANGING	TREE-VIEW
MSG-VALIDATE	CHECK-BOX, COMBO-BOX, DATE-ENTRY, ENTRY-FIELD, GRID, LIST-BOX, PUSH-BUTTON, RADIO-BUTTON, SCROLL-BAR, SLIDER, STATUS-BAR, TAB-CONTROL, TREE-VIEW
MSG-WB-BEFORE-NAVIGATE	WEB-BROWSER
MSG-WB-DOWNLOAD-BEGIN	WEB-BROWSER
MSG-WB-DOWNLOAD-COMPLETE	WEB-BROWSER
MSG-WB-NAVIGATE-COMPLETE	WEB-BROWSER
MSG-WB-PROGRESS-CHANGE	WEB-BROWSER
MSG-WB-STATUS-TEXT-CHANGE	WEB-BROWSER
MSG-WB-TITLE-CHANGE	WEB-BROWSER
NTF-CHANGED	DATE-ENTRY, ENTRY-FIELD
NTF-PL-FIRST	LIST-BOX
NTF-PL-LAST	LIST-BOX
NTF-PL-NEXT	LIST-BOX
NTF-PL-NEXTPAGE	LIST-BOX
NTF-PL-PREV	LIST-BOX
NTF-PL-PREVPAGE	LIST-BOX
NTF-PL-SEARCH	LIST-BOX
NTF-RESIZED	WINDOW
NTF-SELCHANGE	COMBO-BOX, LIST-BOX

Table 5

This table shows the list of all properties informing about when it's allowed to use them, from the control creation to the modifications and inquires made by the program later.

Property	Display	Modify	Inquire	Notes
Action	x	x		
Alignment	x	x		
Auto-Decimal	x	x	x	

Property	Display	Modify	Inquire	Notes
Background-Color	x	x	x	This property can't be modified or inquired on the WINDOW control.
Bitmap	x	x		
Bitmap-Disabled	x	x	x	
Bitmap-Disabled-Selected	x	x	x	
Bitmap-End	x	x	x	
Bitmap-Handle	x	x	x	
Bitmap-Hint	x	x	x	
Bitmap-Number	x	x	x	
Bitmap-Pressed	x	x	x	
Bitmap-Rollover	x	x	x	
Bitmap-Rollover-Selected	x	x	x	
Bitmap-Start	x	x	x	
Bitmap-Timer	x	x	x	
Bitmap-Trailing	x	x		
Bitmap-Trailing-Disabled	x	x	x	
Bitmap-Trailing-Hint	x	x	x	
Bitmap-Trailing-Number	x	x	x	
Bitmap-Trailing-Rollover	x	x	x	
Bitmap-Width	x	x	x	
Border-Color	x	x	x	
Border-Width	x	x	x	
Busy			x	
Cell Height	x			
Cell Size	x			
Cell Width	x			
Cell-Background-Color	x	x	x	
Cell-Color	x	x	x	

Property	Display	Modify	Inquire	Notes
Cell-Columns-Span	x	x		Preferably use modify instead of display for setting this property
Cell-Current-Background-Color			x	
Cell-Current-Color			x	
Cell-Current-Font			x	
Cell-Current-Foreground-Color			x	
Cell-Current-Protection			x	
Cell-Data	x	x	x	
Cell-Entry-Background-Color	x	x	x	
Cell-Entry-Color	x	x	x	
Cell-Entry-Foreground-Color	x	x	x	
Cell-Font	x	x	x	
Cell-Foreground-Color		x	x	
Cell-Hint		x	x	
Cell-Protection		x	x	
Cell-Rows-Span	x	x		Preferably use modify instead of display for setting this property
Cell-Selected-Background-Color	x	x	x	
Cell-Selected-Color	x	x	x	
Cell-Selected-Foreground-Color	x	x	x	
Cells-Selected			x	
Clsid	x			
Col	x	x	x	
Color	x	x	x	This property can't be modified or inquired on the WINDOW control.
Colors	x	x	x	
Column	x	x	x	
Column-Background-Color		x	x	

Property	Display	Modify	Inquire	Notes
Column-Color		x	x	
Column-Dividers	x	x		
Column-Font		x	x	
Column-Foreground-Color		x	x	
Column-Headings-Height		x	x	
Column-Headings-Layout		x	x	
Column-Hiding		x	x	
Column-Protection		x	x	
Column-Selected-Background-Color	x	x	x	
Column-Selected-Color	x	x	x	
Column-Selected-Foreground-Color	x	x	x	
Columns-Selected	x	x	x	
Control Font	x			
Css-Style-Name	x	x		
Cursor	x	x	x	
Cursor-Background-Color	x	x	x	
Cursor-Col	x	x	x	
Cursor-Color	x	x	x	
Cursor-Foreground-Color	x	x	x	
Cursor-Frame-Width	x	x	x	
Cursor-Row	x	x	x	
Cursor-X	x	x	x	
Cursor-Y	x	x	x	
Custom-Data	x	x	x	
Data-Columns	x	x		
Data-Types	x	x		
Decoration-Background	x	x	x	
Display-Columns	x	x	x	

Property	Display	Modify	Inquire	Notes
Display-Format	x	x	x	
Divider-Color	x	x	x	
Dividers	x	x		
Drag-Background-Color	x	x	x	
Drag-Color	x	x	x	
Drag-Foreground-Color	x	x	x	
Editor-Show-Always	x	x		
Enabled	x	x	x	
End-Color	x	x	x	
Ensure-Visible		x		
Entry-Reason			x	
Event-List	x	x		
Exception-Value	x	x	x	
Exclude-Event-List	x	x		
Expand		x		
Export-File-Format	x	x	x	
Export-File-Name	x	x	x	
File-Pos	x	x	x	
Fill-Char	x	x	x	
Fill-Color	x	x	x	
Fill-Color2	x	x	x	
Fill-Percent	x	x	x	
Finish-Reason			x	
Font	x	x	x	This property can't be modified or inquired on the WINDOW control.
Foreground-Color	x	x	x	This property can't be modified or inquired on the WINDOW control.
Format-String	x	x	x	

Property	Display	Modify	Inquire	Notes
Gradient-Color-1	x	x	x	
Gradient-Color-1	x	x	x	
Gradient-Orientation	x	x	x	
Go-Back		x		
Go-Forward		x		
Go-Home		x		
Go-Search		x		
Group	x	x	x	
Group-Value	x	x	x	
Has-Children		x	x	
Heading-Background-Color	x	x	x	
Heading-Color	x	x	x	
Heading-Cursor-Background-Color	x	x	x	
Heading-Cursor-Color	x	x	x	
Heading-Cursor-Foreground-Color	x	x	x	
Heading-Divider-Color	x	x	x	
Heading-Font	x	x	x	
Heading-Foreground-Color	x	x	x	
Heading-Menu-Popup	x	x	x	
Help-Id	x	x	x	
Hidden-Data		x	x	
High-Color	x	x	x	
Hint	x	x	x	
Hscroll-Pos	x	x	x	
Icon	x			
Id	x	x	x	
Init-Params	x			
Init-Signature	x			

Property	Display	Modify	Inquire	Notes
Insertion-Index		x	x	
Insert-Rows		x		
Item		x	x	
Item-Background-Color		x	x	
Item-Color		x	x	
Item-Foreground-Color		x	x	
Item-Height	x	x		
Item-Hint		x	x	
Item-Text		x	x	
Item-To-Add	x	x		Preferably use modify instead of display for setting this property
Item-To-Delete		x		
Item-To-Empty		x		
Item-Value		x	x	
Label-Offset	x	x	x	
Last-Row			x	
Layout-data	x	x	x	
Layout-manager	x			
Leading-Shift	x	x		
Line	x	x	x	
Lines	x	x	x	
Lm-On-Columns	x	x		
Low-Color	x	x	x	
Major-Tick-Spacing	x	x	x	
Mass-Update		x	x	
Maxday-Characters	x	x	x	
Max-Height	x	x	x	
Max-Lines	x	x	x	

Property	Display	Modify	Inquire	Notes
Max-Progress	x	x	x	
Max-Size	x	x	x	
Max-Text	x	x	x	
Max-Val	x	x	x	
Max-Width	x	x	x	
Min-Height	x	x	x	
Min-Lines	x	x	x	
Minor-Tick-Spacing	x	x	x	
Min-Size	x	x	x	
Min-Val	x	x	x	
Min-Width	x	x	x	
Model-To-View-Y			x	
Mouse-Wheel-Scroll	x	x	x	
Next-Item		x		
Notify-Change-Delay	x	x	x	
Num-Col-Headings	x	x	x	
Num-Row-Headings	x	x	x	
Num-Rows	x	x	x	
Object	x			
Page-Size	x	x	x	
Panel-Background-Color		x	x	
Panel-Bitmap		x	x	
Panel-Bitmap-Alignment		x	x	
Panel-Bitmap-Number		x	x	
Panel-Bitmap-Width		x	x	
Panel-Color		x	x	
Panel-Foreground-Color		x	x	
Panel-Hint		x	x	

Property	Display	Modify	Inquire	Notes
Panel-Index		x	x	
Panel-Style		x	x	
Panel-Text		x	x	
Panel-Widths	x	x		
Parent		x		
Placeholder	x	x	x	
Placement		x		
Pop-Up Menu	x	x		
Pos	x	x	x	
Position	x	x	x	
Position-Shift	x	x	x	
Progress	x	x	x	
Proposal	x	x		
Proposal-Delay	x	x	x	
Proposal-Index		x	x	
Proposal-Min-Text	x	x	x	
Proposal-To-Delete		x		
Protection	x	x	x	
Query-Index		x		
Record-Data	x	x	x	Preferably use modify instead of display for setting this property
Record-To-Add	x	x		Preferably use modify instead of display for setting this property
Record-To-Delete		x		
Refresh	x	x	x	
Region-Background-Color		x	x	
Region-Color		x	x	
Region-Foreground-Color		x	x	

Property	Display	Modify	Inquire	Notes
Reordering-Col-Index	x	x	x	
Reset-Grid		x		
Reset-List		x		
Reset-Proposals		x		
Reset-Tabs		x		
Row-Background-Color		x	x	
Row-Background-Color-Pattern	x	x		
Row-Capacity			x	
Row-Color		x	x	
Row-Color-Pattern	x	x		
Row-Cursor-Background-Color		x	x	
Row-Cursor-Color		x	x	
Row-Cursor-Foreground-Color		x	x	
Row-Dividers	x	x		
Row-Font		x	x	
Row-Foreground-Color		x	x	
Row-Foreground-Color-Pattern	x	x		
Row-Hiding		x	x	
Row-Protection		x	x	
Row-Selected-Background-Color	x	x	x	
Row-Selected-Color	x	x	x	
Row-Selected-Foreground-Color	x	x	x	
Rows-Filtered			x	
Rows-Per-Page	x	x	x	
Rows-Selected	x	x	x	
Screen Col	x			
Screen Column	x			
Screen Line	x			

Property	Display	Modify	Inquire	Notes
Screen Pos	x			
Screen Position	x			
Screen-Index	x	x	x	
Search-Options	x	x	x	
Search-Panel	x	x		
Search-Text		x		
Search-Text-In-View		x		
Selection-Background-Color	x	x	x	
Selection-Color	x	x	x	
Selection-Foreground-Color	x	x	x	
Selection-Index	x	x	x	
Selection-Mode		x	x	
Selection-Text			x	
Separation	x	x		
Shading	x	x		
Size	x	x	x	
Sort-data	x	x	x	
Sort-types	x	x	x	
Sort-Order	x	x	x	
Spell-Checking	x	x	x	
Start-X		x		
Start-Y		x		
Status-Text	x	x	x	
Stop-Browser		x		
Sunday-Foreground	x	x	x	
Tab-Alignment		x	x	
Tab-Background-Color	x	x	x	
Tab-Color	x	x	x	

Property	Display	Modify	Inquire	Notes
Tab-Delay	x	x	x	
Tab-Enabled		x	x	
Tab-Foreground-Color	x	x	x	
Tab-Index	x	x	x	
Tab-Insertion-Index		x		
Tab-Text		x	x	
Tab-To-Add	x	x		Preferably use modify instead of display for setting this property
Tab-To-Delete		x		
Tab-Rollover-Color	x	x	x	
Termination-Value	x	x	x	
Text-Orientation	x	x	x	
Thumb-Position	x	x	x	
Title	x	x	x	
Title-Position	x	x	x	
Trailing-Shift	x	x	x	
Transparent-Color	x	x	x	
Validation-Errmsg	x	x	x	
Validation-Opts	x	x	x	
Validation-Regexp	x	x	x	
Value	x	x	x	
Value-Format	x	x	x	
View-Cursor-Y			x	
View-To-Model-Y			x	
Virtual-Width	x	x	x	
Visible	x	x	x	
Visible-Proposal-Count	x	x	x	
VPadding	x	x	x	

Property	Display	Modify	Inquire	Notes
Vscroll-Pos	x	x	x	
Weekday-Foreground	x	x	x	
Width	x	x	x	
X		x	x	
Y		x	x	

Appendix F

Intrinsic Functions

The table below shows all available intrinsic functions.

The **Function name** column defines the name of the function.

The **Function type** column defines the type of the function, it can be: Integer, Numeric, Alphabetic, Alphanumeric. When "any" word appears it means that the type of the function depends on the argument type.

The **Argument number** column defines the number of the arguments. When "arbitrary" word appears it means that the number or arguments is undefined and it can be arbitrary. When 0 appears, it means that the function doesn't has arguments.

The **Argument type** column defines the type of the arguments, they can be: Integer, Numeric, Alphabetic, Alphanumeric. When "any" appears it means that the type of the argument can be any of theme.

A detailed documentation of each single function follows.

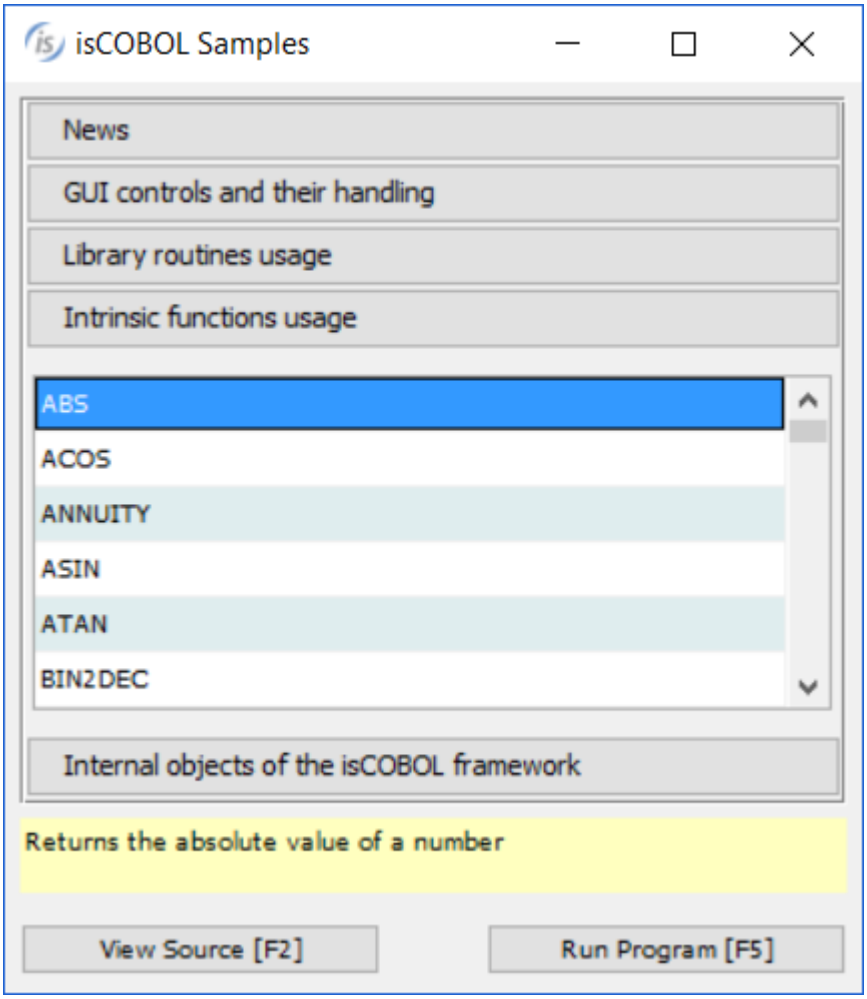
Function name	Function type	Argument number	Argument type	Description
ABS	Numeric	1	Numeric	Absolute value of the argument passed
ACOS	Numeric	1	Numeric	Arccosine of argument
ANNUITY	Numeric	2	Numeric	Annuity of the arguments
ASIN	Numeric	1	Numeric	Arcsine of argument
ATAN	Numeric	1	Numeric	Arctangent of argument
BIN2DEC	Numeric	1	Alphanumeric	Decimal representation of binary data
BYTE-LENGTH	Numeric	1	any	Length of argument in number of bytes
CAPACITY	Numeric	1	Occurs	Current capacity of a dynamic capacity table
CHAR	Alphanumeric	1	Integer	Character in position of argument
COS	Numeric	1	Numeric	Cosine of argument
CURRENT-DATE	Alphanumeric	0		Current date and time and difference from Greenwich Mean Time

DATE-OF-INTEGER	Integer	1	Integer	Standard date equivalent (YYYYMMDD) of integer date
DATE-TO-YYYYMMDD	Integer	2	Integer	Date with four digits year of a date with two digits year. The second optional parameter is the threshold (default 50)
DAY-OF-INTEGER	Integer	1	Integer	Julian date equivalent (YYYYDDD) of integer date
DAY-TO-YYYYMMDD	Integer	2	Integer	Date with four digits year of a date with two digits year. in the format *YYddd* The second optional parameter is the threshold (default 50)
DEC2BIN	Alphanumeric	1	Numeric	Binary representation of decimal data
DEC2HEX	Alphanumeric	1	Numeric	Hexadecimal representation of decimal data
DEC2OCT	Alphanumeric	1	Numeric	Octal representation of decimal data
E	Numeric	0		The value of E, the natural base
EXP	Numeric	1	Numeric	E raised to the power of argument
EXP10	Numeric	1	Numeric	E raised to the power of argument
FACTORIAL	Integer	1	Integer	Factorial of argument
FRACTION-PART	Numeric	1	Numeric	Fraction part of argument
HANDLE-TYPE	Integer	1	Integer	resource identified by the handle
HEX2DEC	Numeric	1	Alphanumeric	Decimal representation of hexadecimal data
INTEGER	Integer	1	Numeric	The greatest integer not greater than argument
INTEGER-OF-DATE	Integer	1	Integer	Integer date equivalent of standard date (YYYYMMDD)
INTEGER-OF-DAY	Integer	1	Integer	Integer date equivalent of Julian date (YYYYDDD)
INTEGER-PART	Integer	1	Numeric	Integer part of argument
LENGTH	Integer	1	any	Length of argument
LOG	Numeric	1	Numeric	Natural logarithm of argument
LOG10	Numeric	1	Numeric	Logarithm to base 10 of argument
LOWER-CASE	Alphanumeric	1	Alphanumeric	All letters in the argument are set to lowercase

MAX	any	arbitrary	any	Value of maximum argument
MEAN	Numeric	arbitrary	Numeric	Arithmetic mean of arguments
MEDIAN	Numeric	arbitrary	Numeric	Median of arguments
MIDRANGE	Numeric	arbitrary	Numeric	Mean of minimum and maximum arguments
MIN	any	arbitrary	any	Value of minimum argument
MOD	Integer	2	Integer	argument 1 modulo argument 2
NUMVAL	Numeric	1	Alphanumeric	Numeric value of simple numeric string
NUMVAL-C	Numeric	2	Alphanumeric	Numeric value of numeric string with optional commas and currency sign
OCT2DEC	Numeric	1	Numeric	Decimal representation of octal data
ORD	Integer	1	Alphanumeric	Ordinal position of the argument in collating sequence
ORD-MAX	Integer	1	any	Ordinal position of maximum argument
ORD-MIN	Integer	1	any	Ordinal position of minimum argument
PI	Numeric	0		Value of pi
PRESENT-VALUE	Numeric	arbitrary	Numeric	Present value of a series of future period-end amounts, argument 2 at a discount rate of argument 1
RANDOM	Numeric	1	Integer	Random number between 0 and 1
RANGE	Numeric	arbitrary	Numeric	Value of maximum argument minus value of minimum argument
REM	Numeric	2	Numeric	Remainder of arg 1 / arg 2
REVERSE	Alphanumeric	1	Alphanumeric	Reverse order of the characters of the argument
SIGN	Numeric	1	Numeric	1 if argument is positive 0 if argument is zero -1 if argument is negative
SIN	Numeric	1	Numeric	Sine of argument
SQRT	Numeric	1	Numeric	Square root of argument
STANDARD-DEVIATION	Numeric	arbitrary	Numeric	Standard deviation of arguments
SUM	Numeric	arbitrary	Numeric	Sum of arguments
TAN	Numeric	1	Numeric	Tangent of argument

TRIM	Alphanumeric	1	Alphanumeric	The argument is trimmed
TRIML	Alphanumeric	1	Alphanumeric	The argument is left trimmed
TRIMR	Alphanumeric	1	Alphanumeric	The argument is right trimmed
UPPER-CASE	Alphanumeric	1	Alphanumeric	All letters in the argument are set to uppercase
VARIANCE	Numeric	arbitrary	Numeric	Variance of argument
WHEN-COMPILED	Alphanumeric	0		Date and time program was compiled
YEAR-TO-YYYY	Numeric	2	Numeric	Four digits year of the first argument, the second optional parameter is the threshold (default 50)

Sample programs for each function are available between isCOBOL Samples.



ABS

The ABS function returns the absolute value of a number.

Syntax 1

```
function abs (arg-1)
```

Syntax 2

```
$abs (arg-1)
```

Arguments

- *arg-1* must be a numeric data item or literal.

Result

The function returns a number that is the absolute value of arg-1.

Examples

Example - Display the absolute value of -1.

```
display function abs(-1).
```

ACOS

The ACOS function returns the arccosine of a value.

Syntax 1

```
function acos (arg-1)
```

Syntax 2

```
$acos (arg-1)
```

Arguments

- *arg-1* must be a numeric data item or literal.

Result

The function returns a number that is the arccosine of arg-1.

Examples

Example - Display the arccosine of a cosine.

```
working-storage section.
77 wrk-cosine pic 9v999.
...
procedure division.
...
display function acos(wrk-cosine).
```

ANNUITY

The ANNUITY function returns ratio of annuity using a given rate.

Syntax 1

```
function annuity (arg-1, arg-2)
```

Syntax 2

```
$annuity (arg-1, arg-2)
```

Arguments

- *arg-1* must be a numeric data item or literal whose value is greater than zero.
- *arg-2* must be a numeric data item or literal whose value is greater than zero.

Result

If the value of arg-1 is zero, the value of the function is the approximation of *1 / arg-2*, else the value of the function is the approximation of *arg-1 / (1 - (1 + arg-1) ** (- arg-2))* .

Examples

Example - Display the ratio of annuity of 2 periods with a interest rate of 0.5.

```
display function annuity(2, 0.5).
```

ASIN

The ASIN function returns the arcsine of a value.

Syntax 1

```
function asin (arg-1)
```

Syntax 2

```
$asin (arg-1)
```

Arguments

- *arg-1* must be a numeric data item or literal.

Result

The function returns a number that is the arcsine of arg-1.

Examples

Example - Display the arcsine of -0.5 .

```
display function asin(-0.5).
```

ATAN

The ATAN function returns the arctangent of a value.

Syntax 1

```
function atan (arg-1)
```

Syntax 2

```
$atan (arg-1)
```

Arguments

- *arg-1* must be a numeric data item or literal.

Result

The function returns a number that is the arctangent of arg-1.

Examples

Example - Display the arctangent of 1 .

```
display function atan(1).
```

BIN2DEC

The BIN2DEC function converts a binary value in its decimal format.

Syntax 1

```
function bin2dec (arg-1)
```

Syntax 2

```
$bin2dec (arg-1)
```

Arguments

- *arg-1* must be a alphanumeric data item or literal.

Result

The function returns a number that is the decimal representation of arg-1.

Examples

Example - Display the decimal representation of "0101" .

```
display function bin2dec("0101").
```

BYTE-LENGTH

The BYTE-LENGTH function returns the size in bytes (not in characters) of a variable.

Syntax 1

```
function byte-length (arg-1)
```

Syntax 2

```
$byte-length (arg-1)
```

Arguments

- *arg-1* must be a data-item.

Result

The function returns the size in bytes of arg-1.

Examples

Example - Display the size in bytes of a national data item.

```
working-storage section.
77 unicode-var pic n(10).
...
procedure division.
...
display function byte-length(unicode-var).
```

CAPACITY

The CAPACITY function returns the current capacity of a dynamic capacity table.

Syntax 1

```
function capacity (arg-1)
```

Syntax 2

```
$capacity (arg-1)
```

Arguments

- *arg-1* must be an item of type OCCURS DYNAMIC.

Result

The function returns capacity of arg-1.

Examples

Example - Display the capacity of a table that stores 2 elements.

```
working-storage section.
77 var pic x(10) occurs dynamic.
...
procedure division.
...
move "xxx" to var(1).
move "yyy" to var(2).
display function capacity(var).
```

CHAR

The CHAR function returns the character in the ASCII character set occupying the ordinal position of the argument.

Syntax 1

```
function char (arg-1)
```

Syntax 2

```
$char (arg-1)
```

Arguments

- *arg-1* must be a numeric data item or literal greater than zero.

Result

The function returns the character in the ASCII character set occupying the ordinal position of arg-1.

The ordinal position corresponds to the ASCII character's decimal value minus one.

Examples

Example - Display the ASCII character at the ordinal position 66, that is the upper case letter "A".

```
display function char(66).
```

COS

The COS function returns the cosine of a value.

Syntax 1

```
function cos (arg-1)
```

Syntax 2

```
$cos (arg-1)
```

Arguments

- *arg-1* must be a numeric data item or literal.

Result

The function returns a number that is the cosine of arg-1.

Examples

Example - Display the cosine of 1.048 radians.

```
display function cos(1.048).
```

CURRENT-DATE

The CURRENT-DATE function returns the current date and time.

Syntax 1

```
function current-date ()
```

Syntax 2

```
$current-date ()
```

Result

The function returns a number that is the current date and time in the format YYYYMMDDHHNNSSCC.

Examples

Example - Set a timestamp field to the current date and time.

```
working-storage section.
01 w-date-time.
   03 w-date.
      05 w-year  pic 9(4).
      05 w-month pic 9(2).
      05 w-day   pic 9(2).
   03 w-time.
      05 w-hour  pic 9(2).
      05 w-min   pic 9(2).
      05 w-sec   pic 9(2).
      05 w-cent  pic 9(2).
...
procedure division.
...
move function current-date() to w-date-time.
```

DATE-OF-INTEGER

The DATE-OF-INTEGER function converts a gregorian number into a date.

Syntax 1

```
function date-of-integer (arg-1)
```

Syntax 2

```
$date-of-integer (arg-1)
```

Arguments

- *arg-1* must be a numeric data item or literal.

Result

The function returns a number that is the date corresponding to the value of arg-1. The date is returned in the format YYYYMMDD.

Examples

Example - Display the date corresponding to the gregorian value 1234.

```
display function date-of-integer(1234).
```

DAY-OF-INTEGER

The DAY-OF-INTEGER function converts a gregorian number into a julian date.

Syntax 1

```
function day-of-integer (arg-1)
```

Syntax 2

```
$day-of-integer (arg-1)
```

Arguments

- *arg-1* must be a numeric data item or literal.

Result

The function returns a number that is the julian date corresponding to the value of arg-1. The date is returned in the format YYYYDDD, where DDD ranges from 1 to 366.

Examples

Example - Display the julian date corresponding to the gregorian value 1234.

```
display function day-of-integer(1234).
```

DATE-TO-YYYYMMDD

The DATE-TO-YYYYMMDD function converts a 6 digits date into 8 digits date.

Syntax 1

```
function date-to-yyyymmdd (arg-1)
```

Syntax 2

```
$date-to-yyyymmdd (arg-1)
```

Arguments

- *arg-1* must be a numeric data item or literal.

Result

The function returns a number that is the 8 digits date corresponding to the value of arg-1. The date is returned in the format YYYYMMDD.

Examples

Example - Display the 8 digits date of a 6 digits date.

```
working-storage section.
01 w-date-time.
   03 w-date.
      05 w-year  pic 9(2).
      05 w-month pic 9(2).
      05 w-day   pic 9(2).
...
procedure division.
...
move 171231 to w-date.
display function date-to-yyyymmdd(w-date)
```

DAY-TO-YYYYMMDD

The DAY-TO-YYYYMMDD function converts a Converts YYDDD date into YYYYDDD date.

Syntax 1

```
function day-to-yyyyddd (arg-1)
```

Syntax 2

```
$day-to-yyyyddd (arg-1)
```

Arguments

- *arg-1* must be a numeric data item or literal.

Result

The function returns a number that is the 7 digits julian date corresponding to the value of arg-1. The date is returned in the format YYYYDDD, where DDD ranges from 1 to 366.

Examples

Example - Display the 7 digits julian date of a 5 digits julian date.

```
working-storage section.
01 w-julian-date.
   03 w-year  pic 9(2).
   03 w-day   pic 9(3).
...
procedure division.
...
move 17001 to w-julian-date.
display function day-to-yyyyddd(w-julian-date)
```

DEC2BIN

The DEC2BIN function converts a decimal value in its binary format.

Syntax 1

```
function dec2bin (arg-1)
```

Syntax 2

```
$dec2bin (arg-1)
```

Arguments

- *arg-1* must be a numeric data item or literal.

Result

The function returns a number that is the binary representation of arg-1.

Examples

Example - Display the binary representation of the number 15 .

```
display function dec2bin(15).
```

DEC2HEX

The DEC2HEX function converts a decimal value in its hexadecimal format.

Syntax 1

```
function dec2hex (arg-1)
```

Syntax 2

```
$dec2hex (arg-1)
```

Arguments

- *arg-1* must be a numeric data item or literal.

Result

The function returns a string that is the hexadecimal representation of arg-1.

Examples

Example - Display the hex representation of the number 15 .

```
display function dec2hex(15).
```

DEC2OCT

The DEC2OCT function converts a decimal value in its octal format.

Syntax 1

```
function dec2oct (arg-1)
```

Syntax 2

```
$dec2oct (arg-1)
```

Arguments

- *arg-1* must be a numeric data item or literal.

Result

The function returns a string that is the octal representation of arg-1.

Examples

Example - Display the octal representation of the number 15 .

```
display function dec2oct(15).
```

E

The E function returns an approximation of e, the base of natural logarithms.

Syntax 1

```
function e ()
```

Syntax 2

```
$e ()
```

Result

The function returns a number that is the approximation of e.

Examples

Example - Display the approximation of e.

```
display function e().
```

EXP

The EXP function returns an approximation of the value of e raised to the power of the argument.

Syntax 1

```
function exp (arg-1)
```

Syntax 2

```
$exp (arg-1)
```

Arguments

- *arg-1* must be a numeric data item or literal.

Result

The function returns a number that is the approximation of the value of e raised to the power of arg-1.

Examples

Example - Display the approximation of e raised to the power of 2 .

```
display function exp(2).
```

EXP10

The EXP10 function returns the value of 10 raised to the power of the argument.

Syntax 1

```
function exp10 (arg-1)
```

Syntax 2

```
function $exp10 (arg-1)
```

Arguments

- *arg-1* must be a numeric data item or literal.

Result

The function returns a number that is the value of 10 raised to the power of arg-1.

Examples

Example - Display the result of 10 raised to the power of 3.

```
display function exp10(3).
```

FACTORIAL

The FACTORIAL function returns the factorial of a number.

Syntax 1

```
function factorial (arg-1)
```

Syntax 2

```
$factorial (arg-1)
```

Arguments

- *arg-1* must be a numeric data item or literal.

Result

The function returns a number that is the factorial of arg-1.

Examples

Example - Display the factorial of 123.

```
display function factorial(123).
```

FRACTION-PART

The FRACTION-PART function returns the fraction portion of the argument.

Syntax 1

```
function fraction-part (arg-1)
```

Syntax 2

```
$fraction-part (arg-1)
```

Arguments

- *arg-1* must be a numeric data item or literal.

Result

The function returns a number that is the fraction portion of arg-1.

Examples

Example - Display the decimal part of a data item with virtual.

```
working-storage section.
77 price pic 9(5)v99.
...
procedure division.
...
move 134.99 to price.
display function fraction-part(price)
```

HANDLE-TYPE

The HANDLE-TYPE function returns the resource type pointed by a handle or zero if the handle is invalid.

Syntax 1

```
function handle-type (arg-1)
```

Syntax 2

```
$handle-type (arg-1)
```

Arguments

- *arg-1* must be a USAGE HANDLE item.

Result

The function returns a number from 0 to 9. The isgui.def copybook includes constants that describe these values:

```
78  handle-is-invalid                        value 0.
78  handle-of-window                         value 1.
78  handle-of-subwindow                      value 2.
78  handle-of-font                           value 3.
78  handle-of-thread                         value 4.
78  handle-of-menu                           value 5.
78  handle-of-control                        value 6.
78  handle-of-bitmap                         value 7.
78  handle-is-unknown                        value 9.
```

Examples

Example - Test if a handle is invalid.

```
working-storage section.
77 my-handle usage handle.
...
procedure division.
...
if function handle-type(my-handle) = 0
   display "my-handle points to nothing!"
end-if.
```

HEX2DEC

The HEX2DEC function converts an hexadecimal value in its decimal format.

Syntax 1

```
function hex2dec (arg-1)
```

Syntax 2

```
$hex2dec (arg-1)
```

Arguments

- *arg-1* must be a alphanumeric data item or literal.

Result

The function returns a number that is the decimal representation of arg-1.

Examples

Example - Display the decimal representation of 0xFF.

```
display function hex2dec("FF").
```

INTEGER

The INTEGER function returns the integer portion of a numeric value.

Syntax 1

```
function integer (arg-1)
```

Syntax 2

```
$integer (arg-1)
```

Arguments

- *arg-1* must be a numeric data item or literal.

Result

The function returns a number that is the integer portion of arg-1.

Examples

Example - Display the integer part of a data item with virtual.

```
working-storage section.
77 price pic 9(5)v99.
...
procedure division.
...
move 134.99 to price.
display function integer(price)
```

INTEGER-OF-DATE

The INTEGER-OF-DATE function converts a date into a gregorian number.

Syntax 1

```
function integer-of-date (arg-1)
```

Syntax 2

```
$integer-of-date (arg-1)
```

Arguments

- *arg-1* must be a numeric data item or literal and must specify a date in the format YYYYMMDD.

Result

The function returns a number that is the gregorian value corresponding to arg-1.

Examples

Example - Display the gregorian value corresponding to the date 1st January 2017.

```
display function integer-of-date(20170101).
```

INTEGER-OF-DAY

The INTEGER-OF-DAY function converts a julian date into a gregorian number.

Syntax 1

```
function integer-of-day (arg-1)
```

Syntax 2

```
$integer-of-day (arg-1)
```

Arguments

- *arg-1* must be a numeric data item or literal and must specify a date in the format YYYYDDD where DDD ranges fro 1 to 366.

Result

The function returns a number that is the gregorian value corresponding to arg-1.

Examples

Example - Display the gregorian value corresponding to the date 1st January 2017.

```
display function integer-of-day(2017001).
```

INTEGER-PART

The INTEGER-PART function returns the integer portion of a numeric value.

Syntax 1

```
function integer-part (arg-1)
```

Syntax 2

```
$integer-part (arg-1)
```

Arguments

- *arg-1* must be a numeric data item or literal.

Result

The function returns a number that is the integer portion of arg-1.

Examples

Example - Display the integer part of a data item with virtual.

```
working-storage section.
77 price pic 9(5)v99.
...
procedure division.
...
move 134.99 to price.
display function integer-part(price)
```

LENGTH

The LENGTH function returns the size of a variable. When used on national items, it returns the number of characters, not the number of bytes; use BYTE-LENGTH in order to know the number of bytes.

Syntax 1

```
function length (arg-1)
```

Syntax 2

```
$length (arg-1)
```

Arguments

- *arg-1* must be a data-item.

Result

The function returns the size of arg-1.

Examples

Example - Display the size in characters of a national data item.

```
working-storage section.
77 unicode-var pic n(10).
...
procedure division.
...
display function length(unicode-var).
```

LOG

The LOG function returns the logarithm of a number.

Syntax 1

```
function log (arg-1)
```

Syntax 2

```
$log (arg-1)
```

Arguments

- *arg-1* must be a data-item.

Result

The function returns the logarithm of arg-1.

Examples

Example - Display the logarithm of the number 100.

```
display function log(100).
```

LOG10

The LOG10 function returns the base-10 logarithm of a number.

Syntax 1

```
function log10 (arg-1)
```

Syntax 2

```
$log10 (arg-1)
```

Arguments

- *arg-1* must be a data-item.

Result

The function returns the base-10 logarithm of arg-1.

Examples

Example - Display the base-10 logarithm of the number 100.

```
display function log10(100).
```

LOWER-CASE

The LOWER-CASE function returns the lower-case version of a text string.

Syntax 1

```
function lower-case (arg-1)
```

Syntax 2

```
$lower-case (arg-1)
```

Arguments

- *arg-1* must be a alphanumeric data item or literal.

Result

The function returns returns the lower-case version of arg-1.

Examples

Example - Test if the provided user name is "admin" in a case-insensitive way.

```
working-storage section.
77 w-user pic x(32).
...
procedure division.
...
if function lower-case(w-user) = "admin"
   |do something
else
   |do something else
end-if.
```

MAX

The MAX function returns the maximum value between the passed arguments.

Syntax 1

```
function max (arg-1[, arg2 ...,arg-n])
```

Syntax 2

```
$max (arg-1[, arg2 ...,arg-n])
```

Arguments

- *arg-1* can be either a numeric or a alphanumeric data item or literal. Other arguments, if any, must be of the same type.

Result

The function returns returns the maximum value between the passed arguments.

Examples

Example - Display the highest number between 1, 2 and 3.

```
display function max(1, 2, 3)
```

MEAN

The MEAN function returns the mean value between the passed arguments.

Syntax 1

```
function mean (arg-1[, arg2 ...,arg-n])
```

Syntax 2

```
$mean (arg-1[, arg2 ...,arg-n])
```

Arguments

- *arg-1* can be either a numeric or a alphanumeric data item or literal. Other arguments, if any, must be of the same type.

Result

The function returns returns the mean value between the passed arguments.

Examples

Example - Display the mean value between 1, 2 and 3.

```
display function mean(1, 2, 3)
```

MEDIAN

The MEDIAN function returns the median value between the passed arguments.

Syntax 1

```
function median (arg-1[, arg2 ...,arg-n])
```

Syntax 2

```
$median (arg-1[, arg2 ...,arg-n])
```

Arguments

- *arg-1* can be either a numeric or a alphanumeric data item or literal. Other arguments, if any, must be of the same type.

Result

The function returns returns the median value between the passed arguments.

Examples

Example - Display the median value between 1, 2 and 3.

```
display function median(1, 2, 3)
```

MIDRANGE

The MIDRANGE function returns the midrange value between the passed arguments.

Syntax 1

```
function midrange (arg-1[, arg2 ...,arg-n])
```

Syntax 2

```
$midrange (arg-1[, arg2 ...,arg-n])
```

Arguments

- *arg-1* can be either a numeric or a alphanumeric data item or literal. Other arguments, if any, must be of the same type.

Result

The function returns returns the midrange value between the passed arguments.

Examples

Example - Display the midrange value between 1, 2 and 3.

```
display function midrange(1, 2, 3)
```

MIN

The MIN function returns the minimum value between the passed arguments.

Syntax 1

```
function min (arg-1[, arg2 ...,arg-n])
```

Syntax 2

```
$min (arg-1[, arg2 ...,arg-n])
```

Arguments

- *arg-1* can be either a numeric or a alphanumeric data item or literal. Other arguments, if any, must be of the same type.

Result

The function returns returns the minimum value between the passed arguments.

Examples

Example - Display the lowest value between 1, 2 and 3.

```
display function min(1, 2, 3)
```

MOD

The MOD function returns the remainder of the division between two arguments.

Syntax 1

```
function mod (arg-1, arg-2)
```

Syntax 2

```
$mod (arg-1, arg-2)
```

Arguments

- *arg-1* must be a numeric data item or literal.
- *arg-2* must be a numeric data item or literal.

Result

The function returns returns the remainder of *arg-1 / arg-2*.

Examples

Example - Check if a number is odd or even.

```
working-storage section.
77 my-num pic 9(3).
...
procedure division.
...
if function mod(my-num, 2) = 0
   |the number is even
else
   |the number is odd
end-if.
```

NUMVAL

The NUMVAL function returns the numbers found in a given alphanumeric value.

Syntax 1

```
function numval (arg-1)
```

Syntax 2

```
$numval (arg-1)
```

Arguments

- *arg-1* must be a alphanumeric data item or literal.

Result

The function returns returns the numbers found in arg-1.

Examples

Example - Remove separators from a date field.

```
working-storage section.
77 w-date-display pic x(10) value "2017/12/31".
77 w-date         pic 9(8).
...
procedure division.
...
move function numval(w-date-display) to w-date.
```

NUMVAL-C

The NUMVAL-C function returns the numbers found in a given alphanumeric value. Any optional currency sign specified by the second argument and any optional commas preceding the decimal point are ignored.

Syntax 1

```
function numval-c (arg-1 [, arg-2])
```

Syntax 2

```
$numval-c (arg-1 [, arg-2])
```

Arguments

- *arg-1* must be a alphanumeric data item or literal.
- *arg-2* must be a alphanumeric data item or literal. It must not contain any of the digits 0 through 9, any leading or trailing spaces, or any of the special characters '+', '-', '.', or ','. If not specified, the character used for currency symbol is the one specified for the program.

Result

The function returns returns the numbers found in arg-1.

Examples

Example - Display the price value.

```
working-storage section.
77 w-price pic x(20) value "$1,234.99".
...
procedure division.
...
display function numval-c(w-price).
```

OCT2DEC

The OCT2DEC function converts a octal value in its decimal format.

Syntax 1

```
function oct2dec (arg-1)
```

Syntax 2

```
$oct2dec (arg-1)
```

Arguments

- *arg-1* must be a numeric data item or literal.

Result

The function returns a number that is the decimal representation of arg-1.

Examples

Example - Display the decimal representation of 123 .

```
display function oct2dec(123).
```

ORD

The ORD function returns the ordinal position in the ASCII character set of an alphanumeric value.

Syntax 1

```
function ord (arg-1)
```

Syntax 2

```
$ord (arg-1)
```

Arguments

- *arg-1* must be a alphanumeric data item or literal.

Result

The function returns a number that is the ordinal position in the ASCII character set of the character identified by arg-1.

The ordinal position corresponds to the ASCII character's decimal value minus one.

Examples

Example - Display the ordinal position of the upper case letter "A" .

```
display function ord("A").
```

ORD-MAX

The ORD-MAX function returns the ordinal position of maximum argument.

Syntax 1

```
function ord-max (arg-1[, arg2 ...,arg-n])
```

Syntax 2

```
$ord-max (arg-1[, arg2 ...,arg-n])
```

Arguments

- *arg-1* can be either a numeric or a alphanumeric data item or literal. Other arguments, if any, must be of the same type.

Result

The function returns the ordinal position of the maximum value between the passed arguments.

Examples

Example - Display the maximum ordinal value between 1, 2 and 3.

```
display function ord-max(1, 2, 3)
```

ORD-MIN

The ORD-MIN function returns the ordinal position of minimum argument.

Syntax 1

```
function ord-min (arg-1[, arg2 ...,arg-n])
```

Syntax 2

```
$ord-min (arg-1[, arg2 ...,arg-n])
```

Arguments

- *arg-1* can be either a numeric or a alphanumeric data item or literal. Other arguments, if any, must be of the same type.

Result

The function returns the ordinal position of the minimum value between the passed arguments.

Examples

Example - Display the minimum ordinal value between 1, 2 and 3.

```
display function ord-min(1, 2, 3)
```

PI

The PI function returns a value that is an approximation of the ratio of the circumference of a circle to its diameter (PI Greek).

Syntax 1

```
function pi ()
```

Syntax 2

```
$pi ()
```

Result

returns a value that is an approximation of the ratio of the circumference of a circle to its diameter (PI Greek).

Examples

Example - Display the PI Greek value.

```
display function pi()
```

PRESENT-VALUE

The PRESENT-VALUE function returns a value that approximates the present value of a series of future period-end amounts specified by the second argument at a discount rate specified by the first argument.

Syntax 1

```
function present-value (arg-1, arg2)
```

Syntax 2

```
$present-value (arg-1, arg2)
```

Arguments

- *arg-1* must be a numeric data item or literal and must not be negative.
- *arg-2* must be a numeric data item or literal.

Result

The function returns a value that approximates the present value of a series of future period-end amounts specified by arg-2 at a discount rate specified by arg-1.

Examples

Example - Display the present value of an amount of 5000 with a discount rate of 0.10.

```
display function present-value(0.10, 5000)
```

RANDOM

The RANDOM function returns a random value between 0 and 1.

Syntax 1

```
function random ([arg-1])
```

Syntax 2

```
$random ([arg-1])
```

Arguments

- *arg-1*, if specified, must be zero or a positive integer. It is used as the seed value to generate a sequence of pseudo-random numbers.

Result

The function returns a random value between 0 and 1.

Examples

Example - Display a random value between 0 and 1.

```
display function random()
```

RANGE

The RANGE function returns the difference between the maximum value and the minimum value in a series of arguments.

Syntax 1

```
function range (arg-1[, arg2 ...,arg-n])
```

Syntax 2

```
$range (arg-1[, arg2 ...,arg-n])
```

Arguments

- *arg-1* can be either a numeric or a alphanumeric data item or literal. Other arguments, if any, must be of the same type.

Result

The function returns the difference between the maximum value and the minimum between the passed arguments.

Examples

Example - Display a range between 1, 2 and 3.

```
display function range(1, 2, 3)
```

REM

The REM function returns the remainder of the division between two arguments.

Syntax 1

```
function rem (arg-1, arg-2)
```

Syntax 2

```
$rem (arg-1, arg-2)
```

Arguments

- *arg-1* must be a numeric data item or literal.
- *arg-2* must be a numeric data item or literal.

Result

The function returns returns the remainder of *arg-1 / arg-2*.

Examples

Example - Check if a number is odd or even.

```
working-storage section.
77 my-num pic 9(3).
...
procedure division.
...
if function rem(my-num, 2) = 0
   |the number is even
else
   |the number is odd
end-if.
```

REVERSE

The REVERSE function returns the content of the argument.

Syntax 1

```
function reverse (arg-1)
```

Syntax 2

```
$reverse (arg-1)
```

Arguments

- *arg-1* must be a alphanumeric data item or literal.

Result

The function returns the content of arg-1 with bytes in the opposite order.

Examples

Example - Discover an hidden message.

```
display function reverse ("looc si LOBOCsi").
```

SIGN

The SIGN function returns +1, 0, or -1 depending on the sign of the argument.

Syntax 1

```
function sign (arg-1)
```

Syntax 2

```
$sign (arg-1)
```

Arguments

- *arg-1* must be a numeric data item or literal.

Result

The function returns returns +1, 0, or -1 depending on the sign of arg-1.

Examples

Example - Check if a value is positive or negative.

```
working-storage section.
77 var pic s9(9).
...
procedure division.
...
evaluate function sign(var)
when -1
   |var contains a negative value
when 0
   |var contains zero
when 1
   |var contains a positive value
end-evaluate.
```

SIN

The SIN function returns the sine of a value.

Syntax 1

```
function sin (arg-1)
```

Syntax 2

```
$sin (arg-1)
```

Arguments

- *arg-1* must be a numeric data item or literal.

Result

The function returns a number that is the sine of arg-1.

Examples

Example - Display the sine of 3.5 .

```
display function sin(3.5).
```

SQRT

The SQRT function returns the square root of a number.

Syntax 1

```
function sqrt (arg-1)
```

Syntax 2

```
$sqrt (arg-1)
```

Arguments

- *arg-1* must be a numeric data item or literal.

Result

The function returns a number that is the square root of arg-1.

Examples

Example - Display the square root of 4 .

```
display function sqrt(4).
```

STANDARD-DEVIATION

The STANDARD-DEVIATION function returns an approximation of the standard deviation of its arguments.

Syntax 1

```
function standard-deviation (arg-1[, arg2 ...,arg-n])
```

Syntax 2

```
$standard-deviation (arg-1[, arg2 ...,arg-n])
```

Arguments

- *arg-1* can be either a numeric data item or literal. Other arguments, if any, must be of the same type.

Result

The function returns a number value that approximates the standard deviation of the passed arguments. If only arg-1 is passed or all the arguments have the same value, then the returned value is zero.

Examples

Example - Display the standard deviation between 1 and 3.

```
display function standard-deviation(1, 3)
```

SUM

The SUM function returns the sum of its arguments.

Syntax 1

```
function sum (arg-1[, arg2 ...,arg-n])
```

Syntax 2

```
$sum (arg-1[, arg2 ...,arg-n])
```

Arguments

- *arg-1* can be either a numeric data item or literal. Other arguments, if any, must be of the same type.

Result

The function returns a number value that is the sum of the passed arguments.

Examples

Example - Display the result of 1+2+3.

```
display function sum(1, 2, 3)
```

TAN

The TAN function returns the tangent of a value.

Syntax 1

```
function tan (arg-1)
```

Syntax 2

```
$tan (arg-1)
```

Arguments

- *arg-1* must be a numeric data item or literal.

Result

The function returns a number that is the tangent of arg-1.

Examples

Example - Display the tangent of 1.

```
display function tan(1).
```

TRIM

The TRIM function trims a string.

Syntax 1

```
function trim (arg-1)
```

Syntax 2

```
$trim (arg-1)
```

Arguments

- *arg-1* must be a alphanumeric data item or literal.

Result

The function returns the value of arg-1 stripped of leading and trailing spaces.

Examples

Example - Display a combined message after removing the spaces from the variable part.

```
working-storage section.
77 w-name pic x(32) value " John ".
...
procedure division.
...
display "Hello " function trim(w-name) ", how are you?".
```

TRIML

The TRIML function removes leading spaces from a string.

Syntax 1

```
function triml (arg-1)
```

Syntax 2

```
$triml (arg-1)
```

Arguments

- *arg-1* must be a alphanumeric data item or literal.

Result

The function returns the value of arg-1 stripped of leading spaces.

Examples

Example - Display a combined message after trimming the spaces from the variable part.

```
working-storage section.
77 w-name pic x(32) value " John".
...
procedure division.
...
display "Hello " function triml(w-name) ", how are you?".
```

TRIMR

The TRIMR function removes trailing spaces from a string.

Syntax 1

```
function trimr (arg-1)
```

Syntax 2

```
$trimr (arg-1)
```

Arguments

- *arg-1* must be a alphanumeric data item or literal.

Result

The function returns the value of arg-1 stripped of trailing spaces.

Examples

Example - Display a combined message after trimming the spaces from the variable part.

```
working-storage section.
77 w-name pic x(32) value "John ".
...
procedure division.
...
display "Hello " function trimr(w-name) ", how are you?".
```

UPPER-CASE

The UPPER-CASE function returns the upper-case version of a text string.

Syntax 1

```
function upper-case (arg-1)
```

Syntax 2

```
$upper-case (arg-1)
```

Arguments

- *arg-1* must be a alphanumeric data item or literal.

Result

The function returns returns the upper-case version of arg-1.

Examples

Example - Test if the provided user name is "admin" in a case-insensitive way.

```
working-storage section.
77 w-user pic x(32).
...
procedure division.
...
if function upper-case(w-user) = "ADMIN"
   |do something
else
   |do something else
end-if.
```

VARIANCE

The VARIANCE function returns an approximation of the variance of its arguments.

Syntax 1

```
function standard-deviation (arg-1[, arg2 ...,arg-n])
```

Syntax 2

```
$standard-deviation (arg-1[, arg2 ...,arg-n])
```

Arguments

- *arg-1* can be either a numeric data item or literal. Other arguments, if any, must be of the same type.

Result

The function returns a number value that approximates the variance of the passed arguments. If only arg-1 is passed or all the arguments have the same value, then the returned value is zero.

Examples

Example - Display the variance between 1 and 3.

```
display function variance(1, 3)
```

WHEN-COMPILED

The WHEN-COMPILED function returns the compilation date and time of the program.

Syntax 1

```
function when-compiled ()
```

Syntax 2

```
$when-compiled ()
```

Result

The function returns a number that is the compilation date and time in the format YYYYMMDDHHNNSSCC.

Examples

Example - Set a timestamp field to the compilation date and time of the program.

```
working-storage section.
01 w-date-time.
   03 w-date.
      05 w-year  pic 9(4).
      05 w-month pic 9(2).
      05 w-day   pic 9(2).
   03 w-time.
      05 w-hour  pic 9(2).
      05 w-min   pic 9(2).
      05 w-sec   pic 9(2).
      05 w-cent  pic 9(2).
...
procedure division.
...
move function when-compiled() to w-date-time.
```

YEAR-TO-YYYY

The YEAR-TO-YYYY function converts 2 digits year into 4 digits year.

Syntax 1

```
function year-to-yyyy (arg-1)
```

Syntax 2

```
$year-to-yyyy (arg-1)
```

Arguments

- *arg-1* must be a numeric data item or literal, two bytes in size.

Result

The function returns the four digits version of the two digits year specified by arg-1.

Examples

Example - Convert a two digits year to a four digits year.

```
working-storage section.
77 w-year-yy   pic 99 value 17.
77 w-year-yyyy pic 9(4).
...
procedure division.
...
move function year-to-yyyy(w-year-yy) to w-year-yyyy.
```

Appendix E

File Status Codes

The table below shows all file status codes. Set the iscobol.file.status * property to specify which file status codes to use.

Regardless of which set of status codes is being used:

- Any code that starts with a "0" is considered successful.
- Any code that starts with a "1" is considered to be an "at end" condition.
- Any code starting with a "2" is considered to be an "invalid key" condition.

2002	85	74	VAX	DG	IBM	MF	MS	Condition
00	00	00	00	00	00	00	00	Operation successful.
02	02	02	00	00	00	02	02	The current key of reference in the record just read is duplicated in the next record (read next) or the operation added a duplicate key to the file where duplicates were allowed (write, rewrite). **Note -** this status is returned only by Vision and DCI file handlers.
04	04	04	04	04	04	04	04	Wrong record length in READ statement.
05	05	00	05	00	10	05	05	Optional file missing and Optional file created.
0D	0D	0D	0D	0D	0D	0D	0D	Command execution failed.
10	10	10	13	10	10	10	10	End/begin of file.
22	22	22	22	22	22	22	22	Duplicate record.
23	23	23	23	23	23	23	23	Record not found.
24	24	24	24	24	24	24	24	Invalid write.
30	30	30	30	30	30	30	95	Permanent I-O error / Invalid path / No write permissions.
35	35	94	35	91	93	35	30	Missing file.

37, 07	37, 07	90, 07	39, 07	91, 07	93	9\u001F	37,07	Invalid permissions.
37, 09	37, 09	95, 09	37, 09	91, 09	93	37, 09	37,09	Invalid open mode.
38	38	93, 03	38	92	93	38	38	File closed with lock.
39,xx	39,xx	94,xx	39,xx	9A,xx	95,xx	39,xx	91,xx	File mismatch. *xx* might be set to one of the following values: 00 - unknown cause of mismatch 01 - generic mismatch reported by the host file system 02 - mismatch on the maximum record size 03 - mismatch on the minimum record size 04 - mismatch on the number of keys or in keys structure **Note -** keys structure is checked only if iscobol.file.index.check_all_keys (boolean) is set to true in the configuration.
41	41	92	41	91	93	41	41	File is already opened.
42	42	91	42	92	92	42	42	file not open during CLOSE.
42	42	91	94	91	92	42	42	file not open during UNLOCK.
43	43	90,02	43	92	23	43	21	invalid delete/rewrite in sequential mode.
44	44	97	44	92	21	44	44	record size changed.
46	46	96	46	92	21	46	46	no current record defined.
47	47	91,02	47, 02	92, 02	13	47	47	file not open during START/READ.
47, 01	47, 01	90, 01	47, 01	92, 01	13	47, 01	47,01	File not open for input or I-O.
48, 01	48, 01	90, 01	48, 01	92, 01	13	48, 01	48,01	File not open for output or I-O.
48, 02	48, 02	91, 02	48, 02	92, 02	13	48, 02	48,02	File not open during WRITE statement.
49, 01	49, 01	90, 01	49, 01	92, 01	13	49, 01	49,01	File not open for I-O.
49, 02	49, 02	91, 02	49, 02	92, 02	13	49, 02	49,02	File not open during REWRITE/DELETE statement.
51	99	99	92	94	23	9D	94	Record locked.
53	9C	9C	9C	9C	23	9\u00D5	53	No more locks available.
61	93	93	91	94	93	9A	94	File locked by another user.
94, 10	94, 10	94, 10	97	97, 10	93	9\u000E	94,10	Too many files opened.
98	98	98	98	98	98	98	98	File corrupt.

9B	9B	9B	9B	9B	23	9B	9B	Operation not supported by the current file handler.
9D	9D	9D	9D	9D	9D	9D	9D	No remappable error. This error is usually followed by a secondary code and and error message that can be retrieved through the C$RERR routine. The same extended information is also stored in the runtime log if iscobol.tracelevel includes the value 8 (trace file activity).
9E	9E	9E	9E	9E	9E	9E	9E	Transaction Error Codes.
9N,xx	9N,xx	9N,xx	9N,xx	9N,xx	9N,xx	9N,xx	9N,xx	Network error under isCOBOL File Server. *xx* might be set to one of the following values: 00 - general error 01 - invalid initialization 02 - no method found 03 - missing parameter 04 - invalid header received 05 - Communication problems 06 - Exception in response 07 - Timeout 08 - Invalid IOR string 09 - Invalid type description 10 - Server side exception 11 - Connection closed
9X	9X	9X	9X	9X	9X	9X	9X	Missing encryption key. This error is returned only by JIsam.
9?	9?	9?	9?	9?	9?	9?	9?	Extended status, check the next list of status for details.

Extended File Status Codes

Instead of one of the above file status codes, an extended status can be returned. The status is composed by the number 9 followed by a letter or symbol. The character after the 9 is the representation of the extended file status using ASCII encoding. For example, 9i is the same as file status 9 with extended status 105 (i.e. EXFS=105) because 'i' is 105 in ASCII.

The following table lists the most common extended codes.

Code	Condition
100	Duplicate record.
101	File not open.

Code	Condition
102	Illegal argument.
103	Illegal key description.
104	Too many files open.
105	Bad isam file format.
106	Non-exclusive access.
107	Record locked.
108	Key already exists.
109	Is primary key.
110	End/begin of file.
111	No record found.
112	No current record.
113	File locked.
114	File name too long.
115	Unknown error.
116	Can't allocate memory.
117	Bad custom collating.
118	Cannot read log file record.
119	Record format of transaction-log file cannot be recognized.
120	Cannot open transaction-log file.
121	Cannot write to transaction-log file.
122	Not in transaction.
123	Unknown error.
124	Beginning of transaction not found.
125	Operation incompatible with open mode
126	Function not supported
127	Disk full
128	Record changed.
129	No more locks available.

Code	Condition
130	Missing file.
131	Invalid permission.
132	File exists.
133	System error
134	Unknown error.
135	Unknown error.
136	Unknown error.
137	Unknown error.
138	Unknown error.
139	Unknown error.
140	Unknown error.
141	Unknown error.
142	Unknown error.
143	Unknown error.
144	Unknown error.
145	Boundary violation.
146	Unknown error.
147	Unknown error.
148	Unknown error.
149	Unknown error.
150	Too many connections.
151	Malformed url.
152	Error on connection: error in iserrio.
153	Error on connection: no permission.

Appendix F

Copybooks

SQLCA	Definitions for the efdParser Class (com.iscobol.lib.efdParser)
fonts.def	Constants and data items to be used in conjunction with the W$FONT Routine and with all the properties related to fonts.
iscobol.def	Most commonly used definitions and group items. Language parts involved are: ACCEPT FROM TERMINAL-INFO Statement ACCEPT FROM SYSTEM-INFO Statement COLOR Phrase and color-related phrases or properties A$GET_DIGEST Routine A$LIST_LOCKS Routine A$LIST_USERS Routine A$USERINFO Routine C$GETRUNENV Routine C$MONITOR Routine C$RERR Routine C$SYSTEM Routine C$LIST_ENVIRONMENT Routine C$LIST_DIRECTORY Routine C$REPLACE_ALL Routine C$XML Routine EDBI_DISCONNECT Routine ESQL$BLOB Routine W$FLUSH Routine W$KEYBUF Routine W$SAVE_IMAGE Routine WD2$CLIENT_INFO Routine
iscontrols.def	This is the complete list of control and property reference, to be used in conjunction with the MODIFY and INQUIRE Statements.
iscrt.def	Group items for event handling and for controlling the ACCEPT Statement behaviors.
isfilesys.def	Constants and data items to be used in conjunction with the I$IO Routine.

isgui.def	Constants and data items for GUI handling. Language parts involved are: DISPLAY MESSAGE BOX Statement C$DESKTOP Routine W$BITMAP Routine W$SCALE Routine W$MENU Routine W$MOUSE Routine W$TEXTSIZE Routine All Event handling Bitmap Transparent-Color Property Date-Entry Value-Format Property Entry-Field Action Property Grid Action Property Grid Cell-Protection Property Grid Column-Protection Property Grid Entry-Reason Property Grid Finish-Reason Property Grid Heading-Menu-Popup Property Grid Protection Property Grid Row-Protection Property Grid Search-Options Property Grid Selection-Mode Propert y Tree-View Placement Property Tree-View Next-Item Property Tree-View Expand Property Status-Bar Panel-Style Property Window Action Property HANDLE-TYPE Function
iskeisen.def	Constants and data items to be used in conjunction with the KEISEN routines: KEISEN, KEISEN1, KEISEN2 and KEISEN_SELECT.
isopensave.def	Constants and data items to be used in conjunction with the C$OPENSAVEBOX Routine.
ispalette.def	Constants and data items to be used in conjunction with the W$PALETTE Routine.
isparseefd.def	Constants and data items to be used in conjunction with the C$PARSEEFD Routine.
isprint.def	Constants and data items to be used in conjunction with the WIN$PRINTER Routine.
isresize.def	Constants and data items to be used in conjunction with the Layout Manager.
iscoblib.def	Container for the issocket.def, iswinhelp.def, iswinvers.def and isreg.def copybooks.
issocket.def	Definitions for the native COBOL library C$SOCKET Routine
iswinhelp.def	Definitions for the native COBOL library $WINHELP Routine

iswinvers.def	Definitions for the native COBOL library WIN$VERSION Routine
isreg.def	Definitions for the native COBOL libraries REG_CLOSE_KEY Routine REG_CREATE_KEY Routine REG_CREATE_KEY_EX Routine REG_DELETE_KEY Routine REG_DELETE_VALUE Routine REG_ENUM_KEY Routine REG_ENUM_VALUE Routine REG_OPEN_KEY Routine REG_OPEN_KEY_EX Routine REG_QUERY_VALUE Routine REG_QUERY_VALUE_EX Routine REG_SET_VALUE Routine REG_SET_VALUE_EX Routine
SQLCA	SQLCA definition for ESQL programs
stdfonts.def	Definition of the internal runtime fonts

efdParser.def

```
*>*****************************************************************<*
*>***** Copyright (c) 2005 - 2020 Veryant. Users of isCOBOL      *****<*
*>***** may freely modify and redistribute this program.         *****<*
*>*****************************************************************<*

       >>SOURCE FORMAT FREE

01  efd-description.
  03  efd-version                          pic x    comp-x.
  03  efd-select-name                      pic x(30).
  03  efd-filename                         pic x(30).
  03  efd-filetype                         pic x    comp-x.
  03  efd-max-record-size                  pic x(4) comp-x.
  03  efd-min-record-size                  pic x(4) comp-x.
  03  efd-number-of-keys                   pic x    comp-x.
  03  efd-number-conditions                pic xx   comp-x.
  03  efd-number-fields                    pic xx   comp-x.
  03  efd-total-number-fields              pic xx   comp-x.
  03  efd-total-number-allfields           pic xx   comp-x.
  03  efd-key-index                        pic xx   comp-x.
  03  efd-field-index                      pic xx   comp-x.
  03  save-efd-field-index                 pic xx   comp-x.
  03  min-efd-field-index                  pic xx   comp-x.
  03  max-efd-field-index                  pic xx   comp-x.
  03  efd-cond-index                       pic xx   comp-x.
  03  efd-max-field-name-len               pic xx   comp-x.
  03  efd-num-key-flds                     pic x    comp-x occurs 120 times.

78  efdmax-segs                           value 16.
```

```
78  efdMaxNumKeyFields                          value 16.

01 efd-key-description-group.
    03  efd-key-description.
        05  efd-number-of-segments             pic 99.
        05  efd-allow-dup-flag                 pic 9.
            88  efd-allow-duplicates           value 1 false 0.
        05  efd-segment-description            occurs efdmax-
segs times indexed by efd-seg-idx.
            07  efd-segment-length             pic x    comp-x.
            07  efd-segment-offset             pic x(4) comp-x.
        05  efd-num-of-key-fields              pic x    comp-x.
        05  efd-key-
fields                      occurs efdMaxNumKeyFields times indexed by efd-key-field-
idx.
            07  efd-key-field-name             pic x(30).
            07  efd-key-field-num              pic xx   comp-x.
    03  efd-key-index-buf                      pic xx   comp-x.

01  efd-condition-description.
    03  efd-condition-type                     pic x comp-x.
        88  efd-equal-condition                value 1.
        88  efd-and-condition                  value 2.
        88  efd-other-condition                value 3.
        88  efd-gt-condition                   value 4.
        88  efd-ge-condition                   value 5.
        88  efd-lt-condition                   value 6.
        88  efd-le-condition                   value 7.
        88  efd-ne-condition                   value 8.
        88  efd-or-condition                   value 9.
        88  efd-comparison-condition           values 1, 4 through 8.
    03  efd-condition-flag                     pic x.
        88  efd-true-condition                 value 'y' false 'n'.
    03  efd-other-conditions.
        05  efd-other-fieldnum                 pic xx comp-x.
        05  efd-other-fieldname                pic x(30).
        05  efd-other-field-val                pic x(50).
        05  efd-other-field-nums               redefines efd-other-field-val.
            07  efd-cond-val-1                 pic s9(18).
            07  efd-cond-val-2                 pic s9(18).
    03  efd-and-conditions                     redefines efd-other-conditions.
        05  efd-condition-1                    pic xx comp-x.
        05  efd-condition-2                    pic xx comp-x.
    03  efd-condition-tablename                pic x(30).


78  efd-NumEdited                              value  0. | Numeric Edited
78  efd-NumUnsigned                            value  1. | Unsigned numeric
78  efd-
NumSignSep                                  value  2. | Signed numeric (trail sep)
78  efd-
NumSigned                                   value  3. | Signed numeric (trail comb)
78  efd-NumSepLead                             value  4. | Signed numeric (lead sep)
78  efd-
NumLeading                                  value  5. | Signed numeric (lead comb)
78  efd-CompSigned                             value  6. | Signed computational
78  efd-CompUnsigned                           value  7. | Unsigned computational
78  efd-PackedPositive                         value  8. | Positive packed-decimal
78  efd-PackedSigned                           value  9. | Signed packed-decimal
```

```
78  efd-PackedUnsigned                  value 10. | Computational-6
78  efd-BinarySigned                    value 11. | Signed binary
78  efd-BinaryUnsigned                  value 12. | Unsigned binary
78  efd-NativeSigned                    value 13. | Signed native-
order binary
78  efd-NativeUnsigned                  value 14. | Unsigned native-
order binary
78  efd-Alphanum                        value 16. | Alphanumeric
78  efd-JustAN                          value 17. | Alphanumeric (justified)
78  efd-Alphbetic                       value 18. | Alphabetic
78  efd-JustAlpha                       value 19. | Alphabetic (justified)
78  efd-AlphaEdited                     value 20. | Alphanumeric Edited
78  efd-Group                           value 22. | Group
78  efd-Flt                             value 23. | Float or Double
78  efd-Nat-type                        value 24. | National
78  efd-JustNat                         value 25. | National (justified)
78  efd-NatEdited                       value 26. | National edited
78  efd-Wide-type                       value 27. | Wide
78  efd-JustWide                        value 28. | Wide (justified)
78  efd-WideEdited                      value 29. | Wide edited
78  efd-NativeVSigned                   value 30. | Signed var-len native-
order binary
78  efd-NativeVUnsigned                 value 31. | Unsigned var-len native-
order binary


78  efd-UserDate                        value 1.
78  efd-UserBinary                      value 2.
78  efd-UserVarLength                   value 3.
78  efd-SecondaryTable                  value 16.
78  efd-UnusedBits                      value 480.

01  efd-field-description.
    03  efd-field-offset                pic x(4) comp-x.
    03  efd-field-length                pic x(4) comp-x.
    03  efd-field-type                  pic x comp-x.
        88  efd-signed-field            values efd-NumSignSep
                                               efd-NumSigned
                                               efd-NumSepLead
                                               efd-NumLeading
                                               efd-CompSigned
                                               efd-PackedSigned
                                               efd-BinarySigned
                                               efd-NativeSigned.
        88  efd-numeric-field           values efd-NumEdited thru efd-
NativeUnsigned.
        88  efd-float-field             value  efd-Flt.
        88  efd-ascii-field             values efd-Alphanum  thru efd-Group.
        88  efd-national-field          values efd-Nat-type  thru efd-
NatEdited.
        88  efd-wide-field              values efd-Wide-type thru efd-
WideEdited.
    03  efd-field-digits                pic x(4) comp-x.
    03  efd-field-scale                 pic s99 comp-4.
    03  efd-field-dbtype                pic x(32).
    03  efd-field-format                pic x(32).
    03  efd-field-user-type             pic xx comp-x.
    03  efd-field-condition             pic xx comp-x.
    03  efd-field-level                 pic x comp-x.
```

```
    03  efd-field-name                          pic x(30).
    03  efd-field-occurs-depth                 pic x comp-x.
    03  efd-field-occurs-table                 occurs efdMaxNumKeyFields times
                                                indexed by efd-field-occurs-level.
        05  efd-field-occ-max-idx              pic xx comp-x.
        05  efd-field-occ-offset               pic xx comp-x.
    03  efd-field-idx                          pic 9(6).
    03  efd-buf-field-type                     pic x(30).

       >>SOURCE FORMAT PREVIOUS
```

fonts.def

```
*>*************************************************************************<*
*>***** Copyright (c) 2005 - 2020 Veryant. Users of isCOBOL      *****<*
*>***** may freely modify and redistribute this program.         *****<*
*>*************************************************************************<*

       >>SOURCE FORMAT FREE

78  wfont-supported                            value 1.
78  wfont-choose-font                          value 2.
78  wfont-get-font                             value 101.
78  wfont-get-closest-font                     value 102.
78  wfont-describe-font                        value 106.

78  wfont-font-support                         value 1.
78  wfont-full-support                         value 2.

78  wfonterr-unsupported                       value 0.
78  wfonterr-cancelled                         value -1.
78  wfonterr-font-not-found                    value -2.
78  wfonterr-invalid-handle                    value -3.

78  wfchoose-fixed-only                        value 1.
78  wfchoose-initialize                        value 2.
78  wfchoose-effects-ok                        value 4.

01  wfont-data.
    03  wfont-face-data.
        05  wfont-device                       handle, value null.
            88  wfdevice-console               value null.
            88  wfdevice-win-printer           value 1.
            88  wfdevice-printer               value 1.
        05  wfont-name                         pic x(33).
        05  wfont-char-set                     pic x comp-x.
            88  wfcharset-dont-care            value 0.
            88  wfcharset-default              value 1.
            88  wfcharset-win-oem              value 2.
            88  wfcharset-win-symbol           value 3.
            88  wfcharset-win-shiftjis         value 4.
            88  wfcharset-win-hangul           value 5.
            88  wfcharset-win-gb2312           value 6.
            88  wfcharset-win-chinesebig5      value 7.
            88  wfcharset-win-johab            value 8.
            88  wfcharset-win-hebrew           value 9.
            88  wfcharset-win-arabic           value 10.
            88  wfcharset-win-greek            value 11.
```

```
           88  wfcharset-win-turkish          value 12.
           88  wfcharset-win-vietnamese       value 13.
           88  wfcharset-win-thai             value 14.
           88  wfcharset-win-easteurope       value 15.
           88  wfcharset-win-russian          value 16.
           88  wfcharset-win-mac              value 17.
           88  wfcharset-win-baltic           value 18.
       05  wfont-size                         pic x comp-x.
       05  wfont-bold-state                   pic x comp-x.
           88  wfont-bold                     value 1, false zero.
       05  wfont-italic-state                 pic x comp-x.
           88  wfont-italic                   value 1, false zero.
       05  wfont-underline-state              pic x comp-x.
           88  wfont-underline                value 1, false zero.
       05  wfont-strikeout-state              pic x comp-x.
           88  wfont-strikeout                value 1, false zero.
       05  wfont-pitch-state                  pic x comp-x.
           88  wfont-fixed-pitch              value 1, false zero.
       05  wfont-family                       pic x comp-x.
           88  wffamily-dont-care             value 0.
           88  wffamily-modern                value 1.
           88  wffamily-roman                 value 2.
           88  wffamily-swiss                 value 3.
           88  wffamily-script                value 4.
           88  wffamily-decorative            value 5.
   03  wfont-choose-data.
       05  wfont-choose-flags                 pic x comp-x.
       05  wfont-choose-min-size              pic x comp-x.
       05  wfont-choose-max-size              pic x comp-x.
       05  wfont-choose-red                   pic x comp-x.
       05  wfont-choose-green                 pic x comp-x.
       05  wfont-choose-blue                  pic x comp-x.
       05  wfont-choose-color-num             pic x comp-x.
   03  wfont-angle                            pic x(2) comp-x.
   03  wfont-scale-x                          float value 0.
   03  wfont-scale-y                          float value 0.

      >>SOURCE FORMAT PREVIOUS
```

iscoblib.def

```
*>*****************************************************************<*
*>***** Copyright (c) 2005 - 2020 Veryant. Users of isCOBOL      *****<*
*>***** may freely modify and redistribute this program.         *****<*
*>*****************************************************************<*

*>***** This copybook is useful for compatibility with
*>***** isCOBOL versions prior 2009.

      >>SOURCE FORMAT FREE

copy "isreg.def".
copy "iswinvers.def".
copy "iswinhelp.def".
copy "issocket.def".

      >>SOURCE FORMAT PREVIOUS
```

iscobol.def

```
*>*****************************************************************<*
*>***** Copyright (c) 2005 - 2020 Veryant. Users of isCOBOL      *****<*
*>***** may freely modify and redistribute this program.         *****<*
*>*****************************************************************<*

       >>SOURCE FORMAT FREE

01  terminal-abilities.
    03  terminal-name                          pic x(10).
    03  filler                                 pic x.
        88  has-reverse                        value "Y".
    03  filler                                 pic x.
        88  has-blink                          value "Y".
    03  filler                                 pic x.
        88  has-underline                      value "Y".
    03  filler                                 pic x.
        88  has-dual-intensity                 value "Y".
    03  filler                                 pic x.
        88  has-132-column-mode                value "Y".
    03  filler                                 pic x.
        88  has-color                          value "Y".
    03  filler                                 pic x.
        88  has-line-drawing                   value "Y".
    03  number-of-screen-lines                 pic 9(3).
    03  number-of-screen-columns               pic 9(3).
    03  filler                                 pic x.
        88  has-local-printer                  value "Y".
    03  filler                                 pic x.
        88  has-visible-attributes             value "Y".
    03  filler                                 pic x.
        88  has-graphical-interface            value "Y".
    03  usable-screen-height                   pic x(2) comp-x.
    03  usable-screen-width                    pic x(2) comp-x.
    03  physical-screen-height                 pic x(2) comp-x.
    03  physical-screen-width                  pic x(2) comp-x.
    03  filler                                 pic x.
        88  is-remote                          value "Y".
    03  client-machine-name                    pic x(64).
    03  filler                                 pic x.
    03  client-user-id                         pic x(20).

01  system-information.
    03  operating-system                       pic x(10).
        88  os-is-msdos                        value "ms-dos".
        88  os-is-os2                          value "OS/2".
        88  os-is-vms                          values "vms", "vax/vms".
        88  os-is-unix                         value "Linux", "AIX", "HP-
UX", "SunOS", "Solaris".
        88  os-is-linux                        value "Linux".
        88  os-is-aos                          value "aos/vs".
        88  os-is-
windows                       values "Windows 95", "Windows 98", "Windows Me".
        88  os-is-win-
nt                          values "Windows NT", "Windows 20", "Windows XP", "Windows V
i", "Windows 7", "WINDOWS", "Windows Se", "Windows 8", "Windows 8.", "Windows 10".
        88  os-is-win-
family                      values "Windows NT", "Windows 95", "Windows 98", "Windows M
```

```
e", "Windows 20", "Windows XP", "Windows Vi", "WINDOWS", "Windows 7", "Windows Se", "W
indows 8", "Windows 8.", "Windows 10".
        88  os-is-amos                          value "amos".
        88  os-is-mpe                           value "MPE/iX".
        88  os-is-mpeix                         value "MPE/iX".
        88  os-is-mac                           value "Mac OS", "Mac OS X".
    03  user-id                                 pic x(12).
    03  station-id                              pic x(12).
    03  filler                                  pic x.
        88  has-indexed-read-previous           value "Y".
    03  filler                                  pic x.
        88  has-relative-read-previous          value "Y".
    03  filler                                  pic x.
        88  can-test-input-status               value "Y".
    03  filler                                  pic x.
        88  is-multi-tasking                    value "Y".
    03  runtime-version.
        05  runtime-major-version               pic 99.
        05  runtime-minor-version               pic 99.
        05  runtime-release                     pic 99.
    03  filler                                  pic x.
        88  is-plugin                           value "Y".
    03  serial-number                           pic x(20).
    03  filler                                  pic x.
        88  has-large-file-support              value "Y".
    03  filler                                  pic x.
    03  filler                                  pic x.
        88  is-64-bit                           value "Y".

78 runenv-standalone                            value 1.
78 runenv-charva                                value 2.
78 runenv-remote-call                           value 3.
78 runenv-thin-client                           value 4.
78 runenv-web-client                            value 5.
78 runenv-wd2                                   value 6.
78 runenv-j2ee                                  value 7.
78 runenv-mobile                                value 8.

78  black                                       value 1.
78  blue                                        value 2.
78  green                                       value 3.
78  cyan                                        value 4.
78  red                                         value 5.
78  magenta                                     value 6.
78  brown                                       value 7.
78  white                                       value 8.
78  dark-gray                                   value 9.
78  bright-blue                                 value 10.
78  bright-green                                value 11.
78  bright-cyan                                 value 12.
78  bright-red                                  value 13.
78  bright-magenta                              value 14.
78  yellow                                      value 15.
78  bright-white                                value 16.
78  bckgrnd-black                               value 32.
78  bckgrnd-blue                                value 64.
78  bckgrnd-green                               value 96.
78  bckgrnd-cyan                                value 128.
78  bckgrnd-red                                 value 160.
```

```
78  bckgrnd-magenta                  value 192.
78  bckgrnd-brown                    value 224.
78  bckgrnd-white                    value 256.
78  bckgrnd-dark-gray                value 288.
78  bckgrnd-bright-blue              value 320.
78  bckgrnd-bright-green             value 352.
78  bckgrnd-bright-cyan              value 384.
78  bckgrnd-bright-red               value 416.
78  bckgrnd-bright-magenta           value 448.
78  bckgrnd-yellow                   value 480.
78  bckgrnd-bright-white             value 512.
78  color-reverse                    value 1024.
78  frgrnd-low                       value 2048.
78  frgrnd-high                      value 4096.
78  color-underline                  value 8192.
78  color-blink                      value 16384.
78  color-protected                  value 32768.
78  bckgrnd-low                      value 65536.
78  bckgrnd-high                     value 131072.
78  window-bright-white              value 131328.

78  csys-async                       value 1.
78  csys-maximized                   value 4.
78  csys-minimized                   value 8.
78  csys-hidden                      value 32.
78  csys-shell                       value 64.
78  csys-desktop                     value 128.

78  listenv-open                     value 1.
78  listenv-next                     value 2.
78  listenv-close                    value 3.

78  listdir-open                     value 1.
78  listdir-next                     value 2.
78  listdir-close                    value 3.

78  listusr-open                     value 1.
78  listusr-next                     value 2.
78  listusr-close                    value 3.

78  listlock-open                    value 1.
78  listlock-next                    value 2.
78  listlock-close                   value 3.

78  get-blob-from-file               value 1.
78  put-blob-into-file               value 2.
78  free-blob-handle                 value 3.

78  edbi-disconnect-connection       value 1.
78  edbi-disconnect-all              value 2.

78  wd2-get-user-agent               value 1.
78  wd2-get-browser-name             value 2.

78  wd2-get-session-value            value 1.
78  wd2-put-session-value            value 2.

78  auserinfo-set                    value 1.
78  auserinfo-get                    value 2.
```

```
78  auserinfo-clear                         value 3.

78  crep-case-insensitive                   value 1.
78  crep-left-trimmed                       value 2.
78  crep-right-trimmed                      value 4.

78  wflush-refresh                          value 1.
78  wflush-inhibit                          value 256.
78  wflush-allow                            value 257.
78  wflush-disable-ui                       value 512.
78  wflush-enable-ui                        value 513.

78  wkbuf-add-to-end                        value 1.
78  wkbuf-add-to-beginning                  value 2.
78  wkbuf-clear-buffer                      value 3.
78  wkbuf-start-recording                   value 4.
78  wkbuf-stop-recording                    value 5.
78  wkbuf-is-recording-active               value 6.
78  wkbuf-start-recording-file              value 7.
78  wkbuf-start-recording-file-append       value 8.
78  wkbuf-load-from-file                    value 9.

78  jget-laf-font                           value 1.
78  jget-laf-color                          value 2.

78  cmonitor-get-no-monitor                 value 0.
78  cmonitor-get-monitor-info               value 1.

78  cprel-error                             value 0.
78  cprel-starting                          value 1.
78  cprel-in-progress                       value 2.
78  cprel-completed                         value 3.

78  cxml-parse-file                         value 1.
78  cxml-release-parser                     value 2.
78  cxml-get-first-child                    value 3.
78  cxml-get-next-sibling                   value 4.
78  cxml-get-parent                         value 5.
78  cxml-get-data                           value 6.
78  cxml-get-attribute-count                value 7.
78  cxml-get-attribute                      value 8.
78  cxml-get-last-error                     value 9.
78  cxml-open-file                          value 10.
78  cxml-parse-string                       value 11.
78  cxml-parse-next-record                  value 12.
78  cxml-get-prev-sibling                   value 13.
78  cxml-new-parser                         value 14.
78  cxml-get-attribute-by-name              value 15.
78  cxml-get-child-by-name                  value 16.
78  cxml-get-child-by-cdata                 value 17.
78  cxml-get-child-by-attr-name             value 18.
78  cxml-get-child-by-attr-value            value 19.
78  cxml-get-sibling-by-name                value 20.
78  cxml-get-sibling-by-cdata               value 21.
78  cxml-get-sibling-by-attr-name           value 22.
78  cxml-get-sibling-by-attr-value          value 23.
78  cxml-get-comment                        value 24.
78  cxml-set-data                           value 25.
78  cxml-modify-cdata                       value 25.
```

```
78  cxml-modify-attribute-value                 value 26.
78  cxml-add-child                              value 27.
78  cxml-add-sibling                            value 28.
78  cxml-add-attribute                          value 29.
78  cxml-add-comment                            value 30.
78  cxml-append-comment                         value 31.
78  cxml-delete-attribute                       value 32.
78  cxml-delete-element                         value 33.
78  cxml-delete-comment                         value 34.
78  cxml-write-file                             value 35.
78  cxml-get-proc-instr-count                   value 36.
78  cxml-get-proc-instr                         value 37.
78  cxml-set-proc-instr                         value 38.
78  cxml-get-version                            value 39.
78  cxml-set-version                            value 40.
78  cxml-get-encoding                           value 41.
78  cxml-set-encoding                           value 42.
78  cxml-get-standalone                         value 43.
78  cxml-set-standalone                         value 44.
78  cxml-write-string                           value 48.

01  listdir-file-information.
    03  listdir-file-type                       pic x.
        88  listdir-file-type-directory         value "D".
        88  listdir-file-type-regular-file      value "F".
        88  listdir-file-type-unknown           value "U".
    03  listdir-file-creation-time.
        05  ldfc-year                           pic xx comp-x.
        05  ldfc-month                          pic x comp-x.
        05  ldfc-day                            pic x comp-x.
        05  ldfc-hour                           pic x comp-x.
        05  ldfc-minute                         pic x comp-x.
        05  ldfc-second                         pic x comp-x.
        05  ldfc-hundreths                      pic x comp-x.
    03  listdir-file-last-access-time.
        05  ldfla-year                          pic xx comp-x.
        05  ldfla-month                         pic x comp-x.
        05  ldfla-day                           pic x comp-x.
        05  ldfla-hour                          pic x comp-x.
        05  ldfla-minute                        pic x comp-x.
        05  ldfla-second                        pic x comp-x.
        05  ldfla-hundreths                     pic x comp-x.
    03  listdir-file-last-modification-time.
        05  ldflm-year                          pic xx comp-x.
        05  ldflm-month                         pic x comp-x.
        05  ldflm-day                           pic x comp-x.
        05  ldflm-hour                          pic x comp-x.
        05  ldflm-minute                        pic x comp-x.
        05  ldflm-second                        pic x comp-x.
        05  ldflm-hundreths                     pic x comp-x.
    03  listdir-file-size                       pic x(8) comp-x.

01 wsave-options.
   03 wsave-format                              pic x(1).
      88 wsave-png                              values "P", " ".
      88 wsave-bmp                              value  "B".
      88 wsave-gif                              value  "G".
      88 wsave-jpg                              value  "J".
   03 wsave-other.
```

```
      05 wsave-quality                          pic 9(10) comp-x.
      05 wsave-transparency                     redefines wsave-quality.
         07 filler                              pic 9(1) comp-x.
            88 no-transparency                  value 0 false 1.
         07 wsave-transparent-color             pic 9(9) comp-x.
   03 wsave-client-server                       pic x.
      88 wsave-server                           values "S", " ".
      88 wsave-client                           value  "C".

 01 cmonitor-data.
    03 cmonitor-usable-screen-height            pic x(2) comp-x.
    03 cmonitor-usable-screen-width             pic x(2) comp-x.
    03 cmonitor-physical-screen-height          pic x(2) comp-x.
    03 cmonitor-physical-screen-width           pic x(2) comp-x.
    03 cmonitor-start-y                         signed-int.
    03 cmonitor-start-x                         signed-int.

>>SOURCE FORMAT PREVIOUS
```

iscontrols.def

```
*>*****************************************************************<*
*>***** Copyright (c) 2005 - 2020 Veryant. Users of isCOBOL      *****<*
*>***** may freely modify and redistribute this program.         *****<*
*>*****************************************************************<*

       >>SOURCE FORMAT FREE

78  ctl-label                                  value 1.
78  ctl-entry-field                           value 2.
78  ctl-push-button                           value 3.
78  ctl-check-box                             value 4.
78  ctl-radio-button                          value 5.
78  ctl-scroll-bar                            value 6.
78  ctl-list-box                              value 7.
78  ctl-combo-box                             value 8.
78  ctl-frame                                 value 9.
78  ctl-tab                                   value 10.
78  ctl-bar                                   value 11.
78  ctl-grid                                  value 12.
78  ctl-bitmap                                value 13.
78  ctl-tree-view                             value 14.
78  ctl-web-browser                           value 15.
78  ctl-status-bar                            value 17.
78  ctl-date-entry                            value 18.
78  ctl-slider                                value 21.
78  ctl-java-bean                             value 22.
78  ctl-ribbon                                value 23.

78  s-permanent                               value 1073741824.
78  s-temporary                               value 536870912.
78  s-notab                                   value 268435456.
78  s-height-in-cells                         value 134217728.
78  s-width-in-cells                          value 67108864.
78  s-3d                                      value 33554432.
78  s-overlap-left                            value 16777216.
78  s-overlap-top                             value 8388608.
78  s-self-act                                value 4194304.
```

```
78  s-notify                            value 2097152.

78  p-termination-value                 value 1.
78  p-exception-value                   value 2.

78  ls-left                             value 1.
78  ls-right                            value 2.
78  ls-center                           value 4.
78  ls-no-key-letter                    value 8.
78  ls-transparent                      value 16.

78  lp-label-offset                     value 1.

78  efs-left                            value 1.
78  efs-right                           value 2.
78  efs-center                          value 4.
78  efs-box                             value 8.
78  efs-no-box                          value 16.
78  efs-multiline                       value 32.
78  efs-vscroll                         value 96.
78  efs-vscroll-bar                     value 224.
78  efs-use-return                      value 256.
78  efs-use-tab                         value 512.
78  efs-upper                           value 1024.
78  efs-lower                           value 2048.
78  efs-no-autosel                      value 4096.
78  efs-read-only                       value 8192.
78  efs-autoterminate                   value 16384.
78  efs-notify-change                   value 32768.
78  efs-secure                          value 65536.
78  efs-numeric                         value 131072.
78  efs-spinner                         value 262144.
78  efs-auto-spin                       value 262208.
78  efs-proposals-unsorted              value 1048576.
78  efs-no-wrap                         value 2097152.

78  efp-max-text                        value 3.
78  efp-max-lines                       value 4.
78  efp-min-val                         value 5.
78  efp-max-val                         value 6.
78  efp-auto-decimal                    value 7.
78  efp-cursor-row                      value 8.
78  efp-cursor                          value 4097.
78  efp-action                          value 4098.
78  efp-selection-text                  value 4099.
78  efp-cursor-col                      value 4100.
78  efp-proposal                        value 4101.
78  efp-proposal-delay                  value 4102.
78  efp-reset-proposals                 value 4103.
78  efp-visible-proposal-count          value 4104.
78  efp-proposal-index                  value 4105.
78  efp-proposal-to-delete              value 4106.
78  efp-proposal-min-text               value 4107.
78  efp-bitmap-number                   value 4108.
78  efp-bitmap-handle                   value 4109.
78  efp-bitmap-width                    value 4110.
78  efp-bitmap-disabled                 value 4111.
78  efp-bitmap-trailing-number          value 4112.
78  efp-bitmap-trailing-disabled        value 4113.
```

```
78  efp-spell-checking                  value 4114.
78  efp-bitmap-hint                     value 4115.
78  efp-bitmap-trailing-hint            value 4116.
78  efp-bitmap-rollover                 value 4117.
78  efp-bitmap-trailing-rollover        value 4118.
78  efp-notify-change-delay             value 4119.

78  pbs-default-button                  value 1.
78  pbs-escape-button                   value 2.
78  pbs-ok-button                       value 4.
78  pbs-cancel-button                   value 8.
78  pbs-no-auto-default                 value 16.
78  pbs-bitmap                          value 32768.
78  pbs-square                          value 16384.
78  pbs-framed                          value 8192.
78  pbs-unframed                        value 4096.
78  pbs-flat                            value 2048.
78  pbs-multiline                       value 1024.
78  pbs-on-header                       value 65536.

78  pbp-bitmap-number                   value 3.
78  pbp-bitmap-handle                   value 4.
78  pbp-bitmap-default                  value pbp-bitmap-number.
78  pbp-bitmap-disabled                 value 21.
78  pbp-bitmap-rollover                 value 22.
78  pbp-bitmap-pressed                  value 23.

78  cbs-transparent                     value 65536.
78  cbs-bitmap                          value 32768.
78  cbs-square                          value 16384.
78  cbs-framed                          value 8192.
78  cbs-unframed                        value 4096.
78  cbs-flat                            value 2048.
78  cbs-multiline                       value 1024.
78  cbs-vtop                            value 512.
78  cbs-left-text                       value 2.

78  cbp-bitmap-number                   value 3.
78  cbp-bitmap-handle                   value 4.
78  cbp-bitmap-default                  value cbp-bitmap-number.
78  cbp-bitmap-disabled                 value 21.
78  cbp-bitmap-rollover                 value 22.
78  cbp-bitmap-pressed                  value 23.
78  cbp-bitmap-rollover-selected        value 24.
78  cbp-bitmap-selected                 value 25.
78  cbp-bitmap-disabled-selected        value 26.

78  rbs-no-group-tab                    value 1.
78  rbs-left-text                       value 2.
78  rbs-bitmap                          value 32768.
78  rbs-square                          value 16384.
78  rbs-framed                          value 8192.
78  rbs-unframed                        value 4096.
78  rbs-flat                            value 2048.
78  rbs-multiline                       value 1024.
78  rbs-vtop                            value 512.
78  rbs-transparent                     value 65536.

78  rbp-bitmap-number                   value 3.
```

```
78  rbp-bitmap-handle                   value 4.
78  rbp-group                           value 5.
78  rbp-group-value                     value 6.
78  rbp-bitmap-default                  value rbp-bitmap-number.
78  rbp-bitmap-disabled                 value 21.
78  rbp-bitmap-rollover                 value 22.
78  rbp-bitmap-pressed                  value 23.
78  rbp-bitmap-rollover-selected        value 24.
78  rbp-bitmap-selected                 value 25.
78  rbp-bitmap-disabled-selected        value 26.

78  sbs-horizontal                      value 1.
78  sbs-track-thumb                     value 2.

78  sbp-min-val                         value 1.
78  sbp-max-val                         value 2.
78  sbp-page-size                       value 3.

78  lbs-unsorted                        value 1.
78  lbs-no-box                          value 2.
78  lbs-box                             value 4.
78  lbs-notify-dblclick                 value 256.
78  lbs-notify-selchange                value 512.
78  lbs-paged                           value 1024.
78  lbs-upper                           value 2048.
78  lbs-lower                           value 4096.
78  lbs-no-search                       value 8192.
78  lbs-check-list                      value 32768.

78  lbp-mass-update                     value 3.
78  lbp-insertion-index                 value 4.
78  lbp-data-columns                    value 5.
78  lbp-display-columns                 value 6.
78  lbp-query-index                     value 7.
78  lbp-alignment                       value 8.
78  lbp-separation                      value 9.
78  lbp-dividers                        value 10.
78  lbp-sort-order                      value 11.
78  lbp-item-to-add                     value 4097.
78  lbp-reset-list                      value 4098.
78  lbp-item-to-delete                  value 4099.
78  lbp-search-text                     value 4100.
78  lbp-selection-index                 value 4103.
78  lbp-item-value                      value 4104.
78  lbp-thumb-position                  value 4105.
78  lbp-selection-background-color      value 4106.
78  lbp-selection-foreground-color      value 4107.
78  lbp-mouse-wheel-scroll              value 4111.
78  lbp-lm-on-columns                   value 4112.
78  lbp-selection-mode                  value 4113.
78  lbp-rows-selected                   value 4114.
78  lbp-action                          value 4115.
78  lbp-export-file-name                value 4116.
78  lbp-export-file-format              value 4117.

78  cms-unsorted                        value 1.
78  cms-drop-down                       value 0.
78  cms-static-list                     value 2.
78  cms-drop-list                       value 4.
```

```
78  cms-box                          value 8.
78  cms-no-box                       value 16.
78  cms-notify-dblclick              value 256.
78  cms-notify-selchange             value 512.
78  cms-upper                        value 2048.
78  cms-lower                        value 4096.

78  cmp-mass-update                  value 3.
78  cmp-max-text                     value 4.
78  cmp-insertion-index              value 5.
78  cmp-item-to-add                  value 4097.
78  cmp-reset-list                   value 4098.
78  cmp-item-to-delete               value 4099.
78  cmp-action                       value 4100.

78  fs-raised                        value 1.
78  fs-lowered                       value 2.
78  fs-engraved                      value 4.
78  fs-rimmed                        value 8.
78  fs-heavy                         value 16.
78  fs-very-heavy                    value 32.
78  fs-alternate                     value 64.
78  fs-full-height                   value 128.
78  fs-vertical                      value 256.
78  fs-horizontal                    value 512.
78  fs-box                           value 1024.
78  fs-keisen                        value 2048.
78  fs-transparent                   value 4096.

78  fp-high-color                    value 1.
78  fp-low-color                     value 2.
78  fp-fill-color                    value 3.
78  fp-fill-percent                  value 4.
78  fp-fill-color2                   value 5.
78  fp-title-position                value 6.

78  ts-multiline                     value 1.
78  ts-buttons                       value 2.
78  ts-fixed-width                   value 4.
78  ts-bottom                        value 8.
78  ts-vertical                      value 17.
78  ts-flat-buttons                  value 32.
78  ts-hot-track                     value 64.
78  ts-no-dividers                   value 128.
78  ts-no-focus                      value 256.
78  ts-text-norotate                 value 512.
78  ts-allow-container               value 1024.
78  ts-relative-offset               value 2048.
78  ts-accordion                     value 4096.
78  ts-tab-flat                      value 8192.

78  tp-bitmap-handle                 value 1.
78  tp-bitmap-width                  value 2.
78  tp-bitmap-number                 value 3.
78  tp-tab-to-add                    value 4097.
78  tp-reset-tabs                    value 4098.
78  tp-tab-to-delete                 value 4099.
78  tp-tab-color                     value 4100.
78  tp-tab-background-color          value 4101.
```

```
78  tp-tab-foreground-color                  value 4102.
78  tp-tab-rollover-color                    value 4103.
78  tp-tab-delay                             value 4104.

78  brs-dotted                               value 1.
78  brs-dashed                               value 2.
78  brs-dotdash                              value 3.

78  brp-width                                value 1.
78  brp-colors                               value 2.
78  brp-shading                              value 3.
78  brp-position-shift                       value 4.
78  brp-leading-shift                        value 5.
78  brp-trailing-shift                       value 6.

78  btp-bitmap-number                        value 1.
78  btp-bitmap-handle                        value 2.
78  btp-bitmap-start                         value 3.
78  btp-bitmap-end                           value 4.
78  btp-bitmap-timer                         value 5.
78  btp-bitmap-transparent-color             value 6.

78  grs-boxed                                value 1.
78  grs-no-box                               value 2.
78  grs-vscroll                              value 4.
78  grs-hscroll                              value 8.
78  grs-column-headings                      value 16.
78  grs-row-headings                         value 32.
78  grs-tiled-headings                       value 64.
78  grs-centered-headings                    value 128.
78  grs-use-tab                              value 256.
78  grs-adjustable-columns                   value 512.
78  grs-paged                                value 1024.
78  grs-reordering-columns                   value 2048.
78  grs-sortable-columns                     value 4096.
78  grs-adjustable-rows                      value 8192.
78  grs-autoterminate                        value 16384.
78  grs-no-search                            value 32768.

78  grp-row-dividers                         value 1.
78  grp-vpadding                             value 2.
78  grp-divider-color                        value 3.
78  grp-insertion-index                      value 4.
78  grp-data-columns                         value 5.
78  grp-display-columns                      value 6.
78  grp-alignment                            value 7.
78  grp-separation                           value 8.
78  grp-column-dividers                      value 9.
78  grp-row-color-pattern                    value 10.
78  grp-y                                    value 11.
78  grp-x                                    value 12.
78  grp-column-color                         value 13.
78  grp-row-color                            value 14.
78  grp-cell-color                           value 15.
78  grp-column-font                          value 16.
78  grp-row-font                             value 17.
78  grp-cell-font                            value 18.
78  grp-bitmap                               value 19.
78  grp-bitmap-number                        value 20.
```

```
78  grp-bitmap-width                    value 21.
78  grp-bitmap-trailing                 value 22.
78  grp-num-rows                        value 23.
78  grp-cursor-y                        value 24.
78  grp-cursor-x                        value 25.
78  grp-cursor-frame-width              value 26.
78  grp-virtual-width                   value 27.
78  grp-data-types                      value 28.
78  grp-cursor-color                    value 29.
78  grp-heading-color                   value 30.
78  grp-heading-font                    value 31.
78  grp-heading-divider-color           value 32.
78  grp-start-x                         value 33.
78  grp-start-y                         value 34.
78  grp-region-color                    value 35.
78  grp-mass-update                     value 36.
78  grp-hidden-data                     value 37.
78  grp-end-color                       value 38.
78  grp-file-pos                        value 39.
78  grp-num-col-headings                value 40.
78  grp-drag-color                      value 41.
78  grp-finish-reason                   value 42.
78  grp-column-protection               value 43.
78  grp-row-protection                  value 44.
78  grp-cell-protection                 value 45.
78  grp-record-to-add                   value 4097.
78  grp-reset-grid                      value 4098.
78  grp-cell-data                       value 4099.
78  grp-record-to-delete                value 4100.
78  grp-record-data                     value 4101.
78  grp-last-row                        value 4102.
78  grp-vscroll-pos                     value 4103.
78  grp-hscroll-pos                     value 4104.
78  grp-action                          value 4105.
78  grp-search-text                     value 4106.
78  grp-search-options                  value 4107.
78  grp-insert-rows                     value 4108.
78  grp-entry-reason                    value 4109.
78  grp-row-heading-line-height         value 4110.
78  grp-mouse-wheel-scroll              value 4111.
78  grp-row-hiding                      value 4112.
78  grp-model-to-view-y                 value 4113.
78  grp-view-cursor-y                   value 4114.
78  grp-heading-menu-popup              value 4115.
78  grp-lm-on-columns                   value 4116.
78  grp-selection-mode                  value 4117.
78  grp-cell-selected-color             value 4118.
78  grp-column-selected-color           value 4119.
78  grp-row-selected-color              value 4120.
78  grp-cells-selected                  value 4121.
78  grp-columns-selected                value 4122.
78  grp-rows-selected                   value 4123.
78  grp-column-headings-layout          value 4124.
78  grp-column-headings-height          value 4125.
78  grp-export-file-name                value 4126.
78  grp-export-file-format              value 4127.
78  grp-row-capacity                    value 4128.

78  tvs-boxed                           value 1.
```

```
78  tvs-no-box                          value 2.
78  tvs-buttons                         value 4.
78  tvs-show-lines                      value 8.
78  tvs-lines-at-root                   value 16.
78  tvs-show-sel-always                 value 32.
78  tvs-table-view                      value 64.
78  tvs-column-headings                 value 128.
78  tvs-tiled-headings                  value 256.
78  tvs-centered-headings               value 512.
78  tvs-adjustable-columns              value 1024.
78  tvs-reordering-columns              value 2048.
78  tvs-sortable-columns                value 4096.

78  tvp-parent                          value 1.
78  tvp-placement                       value 2.
78  tvp-item                            value 3.
78  tvp-bitmap-handle                   value 4.
78  tvp-bitmap-width                    value 5.
78  tvp-item-to-add                     value 4097.
78  tvp-item-text                       value 4098.
78  tvp-next-item                       value 4099.
78  tvp-item-to-delete                  value 4100.
78  tvp-reset-list                      value 4101.
78  tvp-ensure-visible                  value 4102.
78  tvp-expand                          value 4103.
78  tvp-item-to-empty                   value 4104.
78  tvp-bitmap-number                   value 4105.
78  tvp-hidden-data                     value 4106.
78  tvp-has-children                    value 4107.
78  tvp-data-columns                    value 4108.
78  tvp-display-columns                 value 4109.
78  tvp-virtual-width                   value 4110.
78  tvp-heading-color                   value 4111.
78  tvp-heading-background-color        value 4112.
78  tvp-heading-foreground-color        value 4113.
78  tvp-heading-font                    value 4114.
78  tvp-lm-on-columns                   value 4115.
78  tvp-sort-types                      value 4116.
78  tvp-end-color                       value 4117.
78  tvp-heading-menu-popup              value 4118.
78  tvp-column-hiding                   value 4119.
78  tvp-alignment                       value 4120.

78  wbs-notify-change                   value 1.

78  wbp-busy                            value 1.
78  wbp-type                            value 2.
78  wbp-status-text                     value 3.
78  wbp-navigate-url                    value 4.
78  wbp-progress                        value 5.
78  wbp-max-progress                    value 6.
78  wbp-custom-print-template           value 7.
78  wbp-file-name                       value 8.
78  wbp-go-back                         value 4097.
78  wbp-go-forward                      value 4098.
78  wbp-go-home                         value 4099.
78  wbp-go-search                       value 4100.
78  wbp-refresh                         value 4101.
78  wbp-stop                            value 4102.
```

```
78  wbp-print                                   value 4103.
78  wbp-print-no-prompt                         value 4104.
78  wbp-print-preview                           value 4105.
78  wbp-page-setup                              value 4106.
78  wbp-save-as                                 value 4107.
78  wbp-save-as-no-prompt                       value 4108.
78  wbp-properties                              value 4109.
78  wbp-copy-selection                          value 4110.
78  wbp-select-all                              value 4111.
78  wbp-clear-selection                         value 4112.

78  das-short-date                              value 0.
78  das-century-date                            value 1.
78  das-long-date                               value 2.
78  das-time                                    value 3.
78  das-right-align                             value 16.
78  das-notify-change                           value efs-notify-change.
78  das-decoration-background-visible           value 64.
78  das-decoration-borders-visible              value 128.
78  das-week-of-year-visible                    value 256.

78  dap-value-format                            value 1.
78  dap-calendar-font                           value 2.
78  dap-bitmap-handle                           value 3.
78  dap-bitmap-width                            value 4.
78  dap-bitmap-number                           value 5.
78  dap-decoration-background                   value 6.
78  dap-sunday-foreground                       value 7.
78  dap-weekday-foreground                      value 8.
78  dap-max-val                                 value 9.
78  dap-min-val                                 value 10.
78  dap-maxday-characters                       value 11.
78  dap-display-format                          value 4097.

       >>SOURCE FORMAT PREVIOUS
```

iscrt.def

```
*>*****************************************************************<*
*>***** Copyright (c) 2005 - 2020 Veryant. Users of isCOBOL      *****<*
*>***** may freely modify and redistribute this program.         *****<*
*>*****************************************************************<*

       >>SOURCE FORMAT FREE

01  event-status is special-names event status.
    03  event-type                              pic x(4) comp-x.
    03  event-window-handle                     handle of window.
    03  event-control-handle                    handle.
    03  event-control-id                        pic xx comp-x.
    03  event-data-1                            signed-short.
    03  event-data-2                            signed-long.
    03  event-action                            pic x comp-x.

01  screen-control is special-names screen control.
    03  accept-control                          pic 9.
    03  control-value                           pic 999.
    03  control-handle                          handle.
```

```
    03  control-id                              pic xx comp-x.

       >>SOURCE FORMAT PREVIOUS
```

isfilesys.def

```
*>*************************************************************************<*
*>***** Copyright (c) 2005 - 2020 Veryant. Users of isCOBOL      *****<*
*>***** may freely modify and redistribute this program.         *****<*
*>*************************************************************************<*

       >>SOURCE FORMAT FREE

78  max-keys                                value 120.
78  max-segs                                value 16.
78  max-key-size                            value 256.
78  max-record-size                         value 32767.

77  io-function                             pic 99 comp-x.
    88  open-function                       value 1.
    88  close-function                      value 2.
    88  make-function                       value 3.
    88  info-function                       value 4.
    88  read-function                       value 5.
    88  next-function                       value 6.
    88  previous-function                   value 7.
    88  start-function                      value 8.
    88  write-function                      value 9.
    88  rewrite-function                    value 10.
    88  delete-function                     value 11.
    88  unlock-function                     value 12.
    88  remove-function                     value 13.
    88  flush-function                      value 14.
    88  execute-function                    value 15.
    88  start-transaction-function          value 16.
    88  commit-transaction-function         value 17.
    88  rollback-function                   value 18.
    88  recover-function                    value 19.
    88  in-transaction-function             value 21.

77  sio-function                            pic 99 comp-x.
    88  s-open-function                     value 1.
    88  s-close-function                    value 2.
    88  s-make-function                     value 3.
    88  s-read-function                     value 4.
    88  s-write-function                    value 5.
    88  s-rewrite-function                  value 6.

01  seq-type                                signed-short.
    78  s-fixed                             value -1.
    78  s-var-count                         value -2.
    78  s-line                              value -3.

77  rio-function                            pic 99 comp-x.
    88  r-open-function                     value 1.
    88  r-close-function                    value 2.
    88  r-make-function                     value 3.
    88  r-read-function                     value 4.
```

```
    88  r-next-function                  value 5.
    88  r-previous-function              value 6.
    88  r-start-function                 value 7.
    88  r-write-function                 value 8.
    88  r-rewrite-function               value 9.
    88  r-delete-function                value 10.
    88  r-unlock-function                value 11.

77  f-errno                              signed-short external.
    88  f-in-error                      values 1 thru 99.
    88  e-sys-err                       value 1.
    88  e-param-err                     value 2.
    88  e-too-many-files                value 3.
    88  e-mode-clash                    value 4.
    88  e-rec-locked                    value 5.
    88  e-broken                        value 6.
    88  e-duplicate                     value 7.
    88  e-not-found                     value 8.
    88  e-undef-record                  value 9.
    88  e-disk-full                     value 10.
    88  e-file-locked                   value 11.
    88  e-rec-changed                   value 12.
    88  e-mismatch                      value 13.
    88  e-no-memory                     value 14.
    88  e-missing-file                  value 15.
    88  e-permission                    value 16.
    88  e-no-support                    value 17.
    88  e-no-locks                      value 18.
    88  e-interface                     value 19.
    88  e-license-err                   value 20.
    88  e-unknown-err                   value 21.
    88  w-no-support                    value 100.
    88  w-dup-ok                        value 101.

77  f-no-lock                           signed-short external.

77  f-log-errno                         signed-short external.
    88  f-log-in-error                  values 1 thru 99.
    88  e-log-external                  value 1.
    88  e-log-too-many                  value 2.
    88  e-log-missing                   value 3.
    88  e-log-permission                value 4.
    88  e-log-sys-err                   value 5.
    88  e-log-corrupt                   value 6.
    88  e-log-locked                    value 7.
    88  e-log-no-memory                 value 8.
    88  e-log-disk-full                 value 9.
    88  e-no-log                        value 10.
    88  e-rb-log-corrupt                value 11.
    88  e-log-incomplete                value 12.
    88  e-open-not-logged               value 13.
    88  e-log-interface                 value 14.
    88  e-log-remote                    value 15.
    88  e-log-nested-start              value 16.
    88  e-log-temp                      value 17.
    88  w-log-no-support                value 100.

77  f-syserr                            pic x(16)  external.
77  f-errmsg                            pic x(256) external.
```

```
77  record-size-v4                          signed-short.
77  record-size                             unsigned-int.
77  start-key-size                          signed-short.
77  key-num                                 signed-short.
77  flush-all-flag                          signed-short.

01  open-mode                               signed-short.

78  finput                                  value 0.
78  foutput                                 value 1.
78  fio                                     value 2.
78  fextend                                 value 3.
78  fmulti-lock                             value 16.
78  fread-lock                              value 256.
78  fwrite-lock                             value 512.
78  fencrypt                                value 4096.
78  fmass-update                            value 1536.
78  ftrans                                  value 16384.
78  fbulk-addition                          value 34304.

77  start-mode                              signed-short.
    88  f-equals                            value zero.
    88  f-not-less                          value 1.
    88  f-greater                           value 2.
    88  f-less                              value 3.
    88  f-not-greater                       value 4.

77  info-mode                               signed-short.
    88  get-logical-params                  value -1.
    88  get-physical-params                 value -2.
    88  get-comment                         value -3.
    88  get-record-count                    value -4.
    88  get-collating-sequence              value -5.
    88  get-lock-count                      value -6.
    88  get-segment-count                   value -7.
    88  get-segment-info                    value -8.
    88  get-file-size                       value -9.
    88  get-version-number                  value -10.
    88  get-deleted-count                   value -11.
    88  get-os-file-descriptor              value -12.

01  logical-info.
    03  max-rec-size                        pic 9(10).
    03  l-comma-1                           pic x value ",".
    03  min-rec-size                        pic 9(10).
    03  l-comma-2                           pic x value ",".
    03  num-keys                            pic 9(3).
    03  l-end                               pic x value low-values.

01  physical-info.
    03  block-multiple                      pic 99.
    03  p-comma-1                           pic x value ",".
    03  pre-allocation-amount               pic 9(7).
    03  p-comma-2                           pic x value ",".
    03  extension-amount                    pic 9(7).
    03  p-comma-3                           pic x value ",".
    03  compression-factor                  pic 999.
    03  p-comma-4                           pic x value ",".
```

```
    03  encrypted-flag                      pic 9.
    03  p-end                               pic x value low-values.

01  key-info.
    03  key-data.
        05  num-segs                        pic 99.
        05  k-comma-1                       pic x value ",".
        05  dups-allowed                    pic 9.
        05  seg-info
            occurs max-segs times.
            07  k-comma-2                   pic x value ",".
            07  key-size                    pic 9(3).
            07  k-comma-3                   pic x value ",".
            07  key-offset                  pic 9(10).
    03  k-end                               pic x value low-values.

01  record-count-info.
    03  number-of-records                   pic 9(10).
    03  number-of-records-end               pic x.

01  file-segment-count-info.
    03  data-segments                       pic 9(5).
    03  fs-count-comma                      pic x value ",".
    03  index-segments                      pic 9(5).
    03  fs-count-end                        pic x value low-values.

01  file-segment-info.
    03  fs-name                             pic x(127).
    03  fs-name-end                         pic x value low-values.
    03  fs-size                             pic 9(10).
    03  fs-size-end                         pic x value low-values.
    03  fs-type                             signed-short.
        88  fs-data                         value 255.
        88  fs-index                        value 254.
    03  fs-segment-number                   unsigned-short.

01  file-size-info.
    03  whole-file-size                     pic 9(15).
    03  whole-file-size-end                 pic x value low-values.

01  file-version-info.
    03  file-version                        pic 999.
    03  file-version-end                    pic x value low-values.

78  fa-mass-update                          value 1.
78  fa-remote                               value 2.

       >>SOURCE FORMAT PREVIOUS
```

isfonts.def

```
*>*****************************************************************<*
*>***** Copyright (c) 2005 - 2020 Veryant. Users of isCOBOL      *****<*
*>***** may freely modify and redistribute this program.         *****<*
*>*****************************************************************<*

       >>SOURCE FORMAT FREE
```

```
copy "fonts.def".
copy "stdfonts.def".

       >>SOURCE FORMAT PREVIOUS
```

isgui.def

```
*>*****************************************************************<*
*>***** Copyright (c) 2005 - 2020 Veryant. Users of isCOBOL      *****<*
*>***** may freely modify and redistribute this program.         *****<*
*>*****************************************************************<*

       >>SOURCE FORMAT FREE

78  mb-ok                                   value 1.
78  mb-yes-no                              value 2.
78  mb-ok-cancel                           value 3.
78  mb-yes-no-cancel                       value 4.
78  mb-retry-cancel                        value 5.
78  mb-abort-retry-ignore                  value 6.
78  mb-cancel-retry-continue               value 7.

78  mb-yes                                 value 1.
78  mb-no                                  value 2.
78  mb-cancel                              value 3.
78  mb-abort                               value 4.
78  mb-retry                               value 5.
78  mb-ignore                              value 6.
78  mb-continue                            value 7.

78  mb-default-icon                        value 1.
78  mb-warning-icon                        value 2.
78  mb-error-icon                          value 3.

78  wbitmap-display                        value 1.
78  wbitmap-destroy                        value 2.
78  wbitmap-load                           value 3.
78  wbitmap-load-imagelist                 value 5.
78  wbitmap-destroy-imagelist              value 6.
78  wbitmap-load-from-client               value 20.

78  wbitmap-no-fill                        value 1.
78  wbitmap-no-download                    value 2.

78  wscale-stretch                         value 0.
78  wscale-resize-xy                       value 1.
78  wscale-resize-x                        value 2.
78  wscale-resize-y                        value 3.

78  wscale-al-bottom-left                  value 1.
78  wscale-al-bottom-center                value 2.
78  wscale-al-bottom-right                 value 3.
78  wscale-al-middle-left                  value 4.
78  wscale-al-middle-center                value 5.
78  wscale-al-middle-right                 value 6.
78  wscale-al-top-left                     value 7.
78  wscale-al-top-center                   value 8.
78  wscale-al-top-right                    value 9.
```

```
78  wberr-unsupported                    value 0.
78  wberr-file-error                     value -1.
78  wberr-no-memory                      value -2.
78  wberr-not-bitmap                     value -3.
78  wberr-format-unsupported             value -4.
78  wberr-missing-dll                    value -5.

78  wmenu-new                            value 1.
78  wmenu-destroy                        value 2.
78  wmenu-add                            value 3.
78  wmenu-change                         value 4.
78  wmenu-delete                         value 5.
78  wmenu-check                          value 6.
78  wmenu-uncheck                        value 7.
78  wmenu-enable                         value 8.
78  wmenu-disable                        value 9.
78  wmenu-show                           value 10.
78  wmenu-get-menu                       value 11.
78  wmenu-input                          value 12.
78  wmenu-block                          value 13.
78  wmenu-unblock                        value 14.
78  wmenu-get-block                      value 15.
78  wmenu-set-block                      value 16.
78  wmenu-release                        value 17.
78  wmenu-refresh                        value 20.
78  wmenu-destroy-delayed                value 21.
78  wmenu-get-delayed-flag               value 22.
78  wmenu-set-delayed-flag               value 23.
78  wmenu-new-popup                      value 26.
78  wmenu-popup                          value 27.
78  wmenu-add-bitmap                     value 28.
78  wmenu-change-bitmap                  value 29.
78  wmenu-delete-bitmap                  value 30.
78  wmenu-ensure-visible                 value 31.
78  wmenu-new-tray                       value 32.

78  w-unchecked                          value 0.
78  w-checked                            value 1.
78  w-enabled                            value 0.
78  w-disabled                           value 16.
78  w-separator                          value 256.

77  menu-handle                          pic s9(9) comp-4.
77  sub-handle-1                         pic s9(9) comp-4.
77  sub-handle-2                         pic s9(9) comp-4.
77  sub-handle-3                         pic s9(9) comp-4.
77  sub-handle-4                         pic s9(9) comp-4.

78  test-mouse-presence                  value 0.
78  get-mouse-status                     value 1.
78  get-mouse-screen-status              value 2.
78  set-mouse-position                   value 3.
78  set-mouse-screen-position            value 4.
78  set-mouse-shape                      value 5.
78  set-delayed-mouse-shape              value 6.
78  get-mouse-shape                      value 7.
78  capture-mouse                        value 8.
78  release-mouse                        value 9.
```

```
78  enable-mouse                        value 10.
78  set-mouse-help                      value 19.

78  arrow-pointer                       value 1.
78  bar-pointer                         value 2.
78  cross-pointer                       value 3.
78  wait-pointer                        value 4.
78  help-pointer                        value 5.
78  custom-pointer                      value 9.

78  auto-mouse-handling                 value 1.
78  allow-left-down                     value 2.
78  allow-left-up                       value 4.
78  allow-left-double                   value 8.
78  allow-middle-down                   value 16.
78  allow-middle-up                     value 32.
78  allow-middle-double                 value 64.
78  allow-right-down                    value 128.
78  allow-right-up                      value 256.
78  allow-right-double                  value 512.
78  allow-mouse-move                    value 1024.
78  always-arrow-cursor                 value 2048.
78  allow-all-screen-actions            value 16384.

01  textsize-data.
    03  textsize-font                   handle of font value null.
    03  textsize-window                 handle of window value null.
    03  textsize-size-x                 pic 9(7)v99 comp-4.
    03  textsize-cells-x                pic 9(7)v99 comp-4.
    03  textsize-base-x                 pic 9(9) comp-4.
    03  textsize-size-y                 pic 99v99 comp-4.
    03  textsize-cells-y                pic 99v99 comp-4.
    03  textsize-base-y                 pic 9(4) comp-4.
    03  textsize-flags                  pic x comp-x value zero.
        88  textsize-strip-spaces       value 1 false zero.

78  event-action-normal                 value zero.
78  event-action-terminate              value 1.
78  event-action-continue               value 2.
78  event-action-ignore                 value 3.
78  event-action-fail                   value 4.
78  event-action-complete               value 5.
78  event-action-fail-terminate         value 7.

78  w-terminate                         value 91.
78  w-message                           value 95.
78  w-event                             value 96.
78  w-no-fields                         value 97.
78  w-conversion-error                  value 98.
78  w-timeout                           value 99.

78  cmd-close                           value 1.
78  cmd-goto                            value 3.
78  cmd-clicked                         value 4.
78  cmd-dblclick                        value 5.
78  cmd-activate                        value 6.
78  cmd-tabchanged                      value 7.
78  cmd-help                            value 8.
78  cmd-help-mouse                      value 19.
```

```
78  ntf-selchange                        value 4099.
78  ntf-changed                          value 4100.
78  ntf-pl-next                          value 4101.
78  ntf-pl-prev                          value 4102.
78  ntf-pl-nextpage                      value 4103.
78  ntf-pl-prevpage                      value 4104.
78  ntf-pl-first                         value 4105.
78  ntf-pl-last                          value 4106.
78  ntf-pl-search                        value 4107.
78  ntf-resized                          value 4114.

78  msg-sb-thumb                        value 16389.
78  msg-sl-thumb                        value 17001.
78  msg-validate                        value 16391.
78  msg-begin-entry                     value 16392.
78  msg-finish-entry                    value 16393.
78  msg-cancel-entry                    value 16394.
78  msg-goto-cell                       value 16395.
78  msg-goto-cell-mouse                 value 16396.
78  msg-menu-input                      value 16397.
78  msg-init-menu                       value 16398.
78  msg-end-menu                        value 16399.
78  msg-bitmap-clicked                  value 16400.
78  msg-bitmap-dblclick                 value 16401.
78  msg-heading-clicked                 value 16402.
78  msg-heading-dblclick                value 16403.
78  msg-goto-cell-drag                  value 16404.
78  msg-heading-dragged                 value 16405.
78  msg-begin-drag                      value 16406.
78  msg-end-drag                        value 16407.
78  msg-begin-heading-drag              value 16408.
78  msg-end-heading-drag                value 16409.
78  msg-col-width-changed               value 16410.
78  msg-tv-selchanging                  value 16411.
78  msg-tv-selchange                    value 16412.
78  msg-tv-expanding                    value 16413.
78  msg-tv-expanded                     value 16414.
78  msg-close                           value 16415.
78  msg-spin-up                         value 16416.
78  msg-spin-down                       value 16417.
78  msg-paged-next                      value 16419.
78  msg-paged-prev                      value 16420.
78  msg-paged-nextpage                  value 16421.
78  msg-paged-prevpage                  value 16422.
78  msg-paged-first                     value 16423.
78  msg-paged-last                      value 16424.
78  msg-grid-rbutton-down               value 16426.
78  msg-grid-rbutton-up                 value 16427.
78  msg-tv-dblclick                     value 16428.
78  msg-wb-before-navigate              value 16429.
78  msg-wb-navigate-complete            value 16430.
78  msg-wb-download-begin               value 16431.
78  msg-wb-download-complete            value 16432.
78  msg-wb-progress-change              value 16433.
78  msg-wb-status-text-change           value 16434.
78  msg-wb-title-change                 value 16435.
78  msg-ax-event                        value 16436.
78  msg-begin-sort                      value 16437.
```

```
78  msg-row-height-changed                  value 16438.
78  msg-goto-cell-out-prev                  value 16439.
78  msg-goto-cell-out-next                  value 16440.
78  msg-tv-selchange-out-prev               value 16441.
78  msg-tv-selchange-out-next               value 16442.
78  msg-begin-heading-menu-popup            value 16443.
78  msg-heading-menu-popup                  value 16444.
78  msg-jb-event                            value 17011.
78  msg-mouse-clicked                       value 17021.
78  msg-mouse-dblclick                      value 17022.
78  msg-mouse-enter                         value 17023.
78  msg-mouse-exit                          value 17024.
78  msg-st-dblclick                         value 17031.
78  msg-iconified                           value 17032.
78  msg-deiconified                         value 17033.
78  msg-finish-sort                         value 17034.
78  msg-gd-dblclick                         value 17035.

78  action-cut                              value 1.
78  action-copy                             value 2.
78  action-paste                            value 3.
78  action-delete                           value 4.
78  action-undo                             value 5.
78  action-redo                             value 6.
78  action-select-all                       value 7.

78  action-first-page                       value 10.
78  action-last-page                        value 11.
78  action-current-page                     value 12.
78  action-hide-drag                        value 13.
78  action-previous                         value 14.
78  action-previous-page                    value 15.
78  action-next-page                        value 16.
78  action-next                             value 17.
78  action-entry                            value 30.
78  action-sort                             value 32.
78  action-export                           value 33.

78  action-minimize                         value 20.
78  action-maximize                         value 21.
78  action-restore                          value 22.

78  paged-at-start                          value 2147418113.
78  paged-at-end                            value 2147418114.
78  paged-empty                             value 2147418115.

01  grid-search-options                      value all zeros.
    03  grid-search-direction               pic 9.
        88  grid-search-forwards            value zero false 1.
    03  grid-search-wrap-flag               pic 9.
        88  grid-search-wrap                value zero false 1.
    03  grid-search-case-flag               pic 9.
        88  grid-search-ignore-case         value zero false 1.
    03  grid-search-match-flag              pic 9.
        88  grid-search-match-any           value zero.
        88  grid-search-match-leading       value 1.
        88  grid-search-match-all           value 2.
    03  grid-search-location-flag           pic 9.
        88  grid-search-visible             value zero.
```

```
           88  grid-search-hidden                  value 1.
           88  grid-search-all-data                value 2.
       03  grid-search-skip-flag                   pic 9.
           88  grid-search-skip-current           value zero false 1.
       03  grid-search-cursor-flag                 pic 9.
           88  grid-search-moves-cursor           value zero false 1.
       03  grid-search-column                      pic 9(5).
           88  grid-search-all-columns            value zero.

78  grdsrch-not-found                             value 0.
78  grdsrch-found                                 value 1.
78  grdsrch-wrapped                               value 2.

78  grer-enter                                    value x"0D".
78  grer-dblclick                                 value x"00".
78  grer-del                                      value x"01".
78  grer-entry-by-program                         value x"FF".

78  grfr-blank-past-end                           value -1.
78  grfr-terminating                              value -2.
78  grfr-cell-clicked                             value -3.
78  grfr-navigation-key                           value -4.
78  grfr-escape-key                               value -5.
78  grfr-enter-key                                value -6.
78  grfr-tab-key                                  value -7.
78  grfr-autoterminate                            value -8.
78  grfr-page-up-key                              value -9.
78  grfr-page-down-key                            value -10.
78  grfr-shift-tab-key                            value -11.
78  grfr-arrow-up-key                             value -12.
78  grfr-arrow-down-key                           value -13.
78  grfr-arrow-left-key                           value -14.
78  grfr-arrow-right-key                          value -15.

78  grhm-no-menu                                  value 0.
78  grhm-columns-on-right-click                   value 1.
78  grhm-columns-on-button                        value 2.
78  grhm-export-on-right-click                    value 4.
78  grhm-export-on-button                         value 8.
78  grhm-copy-on-right-click                      value 16.
78  grhm-copy-on-button                           value 32.
78  grhm-find-on-right-click                      value 64.
78  grhm-find-on-button                           value 128.

78  grsm-no-selection                             value  0.
78  grsm-single-selection                         value  1.
78  grsm-single-interval-selection                value  2.
78  grsm-multiple-interval-selection              value  4.
78  grsm-row-selection                            value  8.
78  grsm-column-selection                         value 16.
78  grsm-cell-selection                           value 32.

78  lssm-single-selection                         value  1.
78  lssm-single-interval-selection                value  2.
78  lssm-multiple-interval-selection              value  4.

78  grchl-horizontal                              value  0.
78  grchl-vertical-left                           value  1.
78  grchl-vertical-right                          value  2.
```

```
78  grchl-align-center                  value  4.
78  grchl-align-left                    value  8.
78  grchl-align-right                   value 16.
78  grchl-align-top                     value 32.
78  grchl-align-bottom                  value 64.

78  tvplace-first                       value 4294901761.
78  tvplace-last                        value 0.
78  tvplace-sort                        value 4294901763.

78  tvni-child                          value 1.
78  tvni-first-visible                  value 2.
78  tvni-next                           value 3.
78  tvni-next-visible                   value 4.
78  tvni-parent                         value 5.
78  tvni-previous                       value 6.
78  tvni-previous-visible               value 7.
78  tvni-root                           value 8.

78  tvflag-collapse                     value 1.
78  tvflag-expand                       value 2.
78  tvflag-program                      value zero.
78  tvflag-mouse                        value 1.
78  tvflag-keyboard                     value 2.
78  tvflag-normal                       value 1.
78  tvflag-bold                         value 2.

78  tvhm-no-menu                        value 0.
78  tvhm-columns-on-right-click         value 1.
78  tvhm-columns-on-button              value 2.

78  pl-sort-default                     value 0.
78  pl-sort-none                        value 1.
78  pl-sort-native                      value 2.
78  pl-sort-native-ignore-case          value 3.

78  handle-is-invalid                   value 0.
78  handle-of-window                    value 1.
78  handle-of-subwindow                 value 2.
78  handle-of-font                      value 3.
78  handle-of-thread                    value 4.
78  handle-of-menu                      value 5.
78  handle-of-control                   value 6.
78  handle-of-bitmap                    value 7.
78  handle-is-unknown                   value 9.

78  panel-flat                          value 0.
78  panel-lowered                       value 1.
78  panel-raised                        value 2.

78  grd-prt-readonly                    value 1.
78  grd-prt-jump                        value 2.

78  davf-yyyymmdd                       value 0.
78  davf-yymmdd                         value 1.
78  davf-hhmmsshh                       value 2.
78  davf-hhmmss                         value 3.
78  davf-yyyymmddhhmmsshh               value 4.
78  davf-mmddyyyy                       value 5.
```

```
78  davf-mmddyy                                value 6.
78  davf-mmddyyyyhhmmsshh                      value 7.
78  davf-ddmmyyyy                              value 8.
78  davf-ddmmyy                                value 9.
78  davf-ddmmyyyyhhmmsshh                      value 10.
78  davf-hhmm                                  value 11.

78  bm-corner-color                            value x#1000000.

78  snd-sync                                   value 0.
78  snd-async                                  value 1.
78  snd-loop                                   value 8.
78  snd-nostop                                 value 16.

01 mouse-info.
   03  mouse-row                               pic xx comp-x.
       88 mouse-off-screen                     value zero.
   03  mouse-col                               pic xx comp-x.
   03  lbutton-status                          pic 9.
       88 lbutton-down                         value 1.
   03  mbutton-status                          pic 9.
       88 mbutton-down                         value 1.
   03  rbutton-status                          pic 9.
       88 rbutton-down                         value 1.
   03  mouse-row-ex                            pic 9(6)v99 comp-4 sync.
   03  mouse-col-ex                            pic 9(6)v99 comp-4.
   03  mouse-row-pixel                         pic 9(8) comp-4.
   03  mouse-col-pixel                         pic 9(8) comp-4.

78  wprogressdialog-create                     value 1.
78  wprogressdialog-destroy                    value 2.
78  wprogressdialog-set-progress               value 3.
78  wprogressdialog-query-cancel               value 4.
78  wprogressdialog-set-line                   value 5.
78  wprogressdialog-reset-timer                value 6.
78  wprogressdialog-c-copy                     value 7.

78  wprogressdialog-normal                     value 0.
78  wprogressdialog-modal                      value 1.
78  wprogressdialog-autotime                   value 2.
78  wprogressdialog-notime                     value 4.
78  wprogressdialog-nominimize                 value 8.
78  wprogressdialog-noprogressbar              value 16.

78  cdesktop-browse                            value 1.
78  cdesktop-edit                              value 2.
78  cdesktop-mail                              value 3.
78  cdesktop-open                              value 4.
78  cdesktop-print                             value 5.

78  gradient-north-to-
south                       value 0. |default if the property is omitted
78  gradient-northeast-to-southwest            value 1.
78  gradient-east-to-west                      value 2.
78  gradient-southeast-to-northwest            value 3.
78  gradient-south-to-north                    value 4.
78  gradient-southwest-to-northeast            value 5.
78  gradient-west-to-east                      value 6.
78  gradient-northwest-to-southeast            value 7.
```

```
       >>SOURCE FORMAT PREVIOUS
```

iskeisen.def

```
*>*****************************************************************<*
*>***** Copyright (c) 2005 - 2020 Veryant. Users of isCOBOL      *****<*
*>***** may freely modify and redistribute this program.         *****<*
*>*****************************************************************<*

       >>SOURCE FORMAT FREE

01  keisen.
    02  kei-cmd           pic 9(1) comp-x.
    02  kei-line          pic 9(2) comp-x.
    02  kei-col           pic 9(2) comp-x.
    02  kei-lng1          pic 9(2) comp-x.
    02  kei-lng2          pic 9(2) comp-x.
    02  kei-color         pic 9(2) comp-x.
    02  kei-ptn           pic 9(2) comp-x.

01  keisen2.
    02  kei2-cmd          pic 9(1) comp-x.
    02  kei2-start-line   pic 9(2) comp-x.
    02  kei2-start-col    pic 9(2) comp-x.
    02  kei2-end-line     pic 9(2) comp-x.
    02  kei2-end-col      pic 9(2) comp-x.
    02  kei2-prn          pic 9(2) comp-x.
    02  kei2-color        pic 9(2) comp-x.

78  kei-method1 value 1.
78  kei-method2 value 2.
01  kei-param pic 9(1) comp-x value kei-method1.

       >>SOURCE FORMAT PREVIOUS
```

isopensave.def

```
*>*****************************************************************<*
*>***** Copyright (c) 2005 - 2020 Veryant. Users of isCOBOL      *****<*
*>***** may freely modify and redistribute this program.         *****<*
*>*****************************************************************<*

       >>SOURCE FORMAT FREE

78  opensave-supported                  value 1.
78  opensave-open-box                   value 2.
78  opensave-save-box                   value 3.
78  opensave-browse-folder              value 4.
78  opensave-open-box-multi             value 5.
78  opensave-browse-folder-multi        value 6.
78  opensave-next                       value 7.
78  opensave-save-box-checked           value 8.

78  opensave-show-preview-flag          value 1.
78  opensave-overwriteprompt            value 2.
78  opensave-pathmustexist              value 2048.
```

```
78  opensave-filemustexist                  value 4096.
78  opensave-createprompt                   value 8192.
78  opensave-noreadonlyreturn               value 32768.

78  opensave-browse-dontgobelowdomain       value 2.
78  opensave-browse-browseincludefiles      value 16384.

78  opnsaverr-unsupported                   value 0.
78  opnsaverr-cancelled                     value -1.
78  opnsaverr-no-memory                     value -2.
78  opnsaverr-name-too-large                value -3.

01  opensave-data.
    03  opnsav-filename                     pic x(256).
    03  opnsav-flags                        pic 9(4) comp-x value 0.
    03  opnsav-default-ext                  pic x(12).
    03  opnsav-title                        pic x(80).
    03  opnsav-filters                      pic x(512).
    03  opnsav-default-filter               pic 9(4) comp-x value 0.
    03  opnsav-default-dir                  pic x(128).
    03  opnsav-basename                     pic x(128).

       >>SOURCE FORMAT PREVIOUS
```

ispalette.def

```
*>*************************************************************************<*
*>***** Copyright (c) 2005 - 2020 Veryant. Users of isCOBOL      *****<*
*>***** may freely modify and redistribute this program.         *****<*
*>*************************************************************************<*

       >>SOURCE FORMAT FREE

78  wpalette-supported                      value 1.
78  wpalette-num-colors                     value 2.
78  wpalette-get-color                      value 3.
78  wpalette-set-color                      value 4.
78  wpalette-update                         value 5.
78  wpalette-choose-color                   value 6.
78  wpalette-set-user-color                 value 7.

78  wpchoose-use-default                    value 1.

78  wpuser-color-3d                         value 1.
78  wpuser-color-background                 value 2.

78  wpal-no-support                         value 0.
78  wpal-palette-supported                  value 1.
78  wpal-full-support                       value 2.

78  wperr-unsupported                       value 0.
78  wperr-bad-arg                           value -1.
78  wperr-cancelled                         value -2.

01  wpalette-data.
    03  wpal-color-id                       pic x comp-x.
    03  wpal-flags    redefines wpal-color-id   pic x comp-x.
    03  wpal-red                            pic x comp-x.
```

```
    03  wpal-user-color-id   redefines wpal-red  pic x comp-x.
    03  wpal-green                              pic x comp-x.
    03  wpal-blue                               pic x comp-x.

       >>SOURCE FORMAT PREVIOUS
```

isparseefd.def

```
*>*****************************************************************<*
*>***** Copyright (c) 2005 - 2020 Veryant. Users of isCOBOL      *****<*
*>***** may freely modify and redistribute this program.         *****<*
*>*****************************************************************<*

       >>SOURCE FORMAT FREE

copy "isfilesys.def".

78  parseefd-parse                          value 0.
78  parseefd-get-key-info                   value 1.
78  parseefd-get-cond-info                  value 2.
78  parseefd-get-field-info                 value 3.
78  parseefd-test-conditions                value 4.
78  parseefd-release                        value 9.

78  parseefd-flag-include-comments          value 1.
78  parseefd-flag-include-999               value 2.
78  parseefd-flag-exclude-arrays            value 4.
78  parseefd-flag-deep-first                value 8.

78  parseefd-numedited                      value 0.   | Numeric Edited
78  parseefd-numunsigned                    value 1.   | Unsigned numeric
78  parseefd-
numsignsep                          value 2.   | Signed numeric (trail sep)
78  parseefd-
numsigned                           value 3.   | Signed numeric (trail comb)
78  parseefd-numseplead                       value 4.   | Signed numeric (lead sep)
78  parseefd-
numleading                          value 5.   | Signed numeric (lead comb)
78  parseefd-compsigned                       value 6.   | Signed computational
78  parseefd-compunsigned                     value 7.   | Unsigned computational
78  parseefd-packedpositive                   value 8.   | Positive packed-decimal
78  parseefd-packedsigned                     value 9.   | Signed packed-decimal
78  parseefd-packedunsigned                   value 10.  | Computational-6
78  parseefd-binarysigned                     value 11.  | Signed binary
78  parseefd-binaryunsigned                   value 12.  | Unsigned binary
78  parseefd-nativesigned                     value 13.  | Signed native-
order binary
78  parseefd-nativeunsigned                   value 14.  | Unsigned native-
order binary
78  parseefd-alphanum                         value 16.  | Alphanumeric
78  parseefd-justan                           value 17.  | Alphanumeric (justified)
78  parseefd-alphbetic                        value 18.  | Alphabetic
78  parseefd-justalpha                        value 19.  | Alphabetic (justified)
78  parseefd-alphaedited                      value 20.  | Alphanumeric Edited
78  parseefd-group                            value 22.  | Group
78  parseefd-flt                              value 23.  | Float or Double
78  parseefd-nat-type                         value 24.  | National
78  parseefd-justnat                          value 25.  | National (justified)
```

```
78  parseefd-natedited                      value 26. | National edited
78  parseefd-wide-type                      value 27. | Wide
78  parseefd-justwide                       value 28. | Wide (justified)
78  parseefd-wideedited                     value 29. | Wide edited
78  parseefd-nativevsigned                  value 30. | Signed var-len native-
order binary
78  parseefd-nativevunsigned                value 31. | Unsigned var-len native-
order binary

78  parseefd-userdate                       value 1.
78  parseefd-userbinary                     value 2.
78  parseefd-uservarlength                  value 3.
78  parseefd-secondarytable                 value 16.

78  parseefd-maxnumkeyfields                value 16.

01  parseefd-description.
    03  parseefd-header-line.
        05  parseefd-version                pic x comp-n.
        05  parseefd-select-name            pic x(30).
        05  parseefd-filename               pic x(30).
        05  parseefd-filetype               pic x comp-n.
            88  parseefd-sequential-file    value 4.
            88  parseefd-relative-file      value 8.
            88  parseefd-indexed-file       value 12.
    03  parseefd-record-line.
        05  parseefd-max-rec-size           pic x(4) comp-n.
        05  parseefd-min-rec-size           pic x(4) comp-n.
        05  parseefd-num-keys               pic x comp-n.
    03  parseefd-condition-line.
        05  parseefd-number-conditions      pic xx comp-n.
    03  parseefd-fields-line.
        05  parseefd-number-fields          pic x(4) comp-n.
    03  parseefd-v6-information.
        05  parseefd-cobol-trigger          pic x(100).
        05  parseefd-compile-line.
            07  parseefd-sign-flag          pic x(2) comp-n.
                88  parseefd-sign-acu       value 0.
                88  parseefd-sign-ibm       value 4.
                88  parseefd-sign-mf        value 8.
                88  parseefd-sign-ncr       value 20.
                88  parseefd-sign-vax       value 36.
                88  parseefd-sign-mbp       value 72.
                88  parseefd-sign-rea       value 128.
            07  parseefd-max-digits         pic x(2) comp-n.
                88  parseefd-18-digits      value 40.
                88  parseefd-31-digits      value 68.
            07  parseefd-pgm-period         pic x.
            07  parseefd-pgm-comma          pic x.
            07  parseefd-encoding           pic x(2) comp-n.
                88  parseefd_ascii          value 0.
                88  parseefd_wide           value 1.
                88  parseefd_utf-8          value 2.
                88  parseefd_utf-16-le      value 3.
                88  parseefd_utf-16-be      value 5.
                88  parseefd_utf-32-le      value 4.
                88  parseefd_utf-32-be      value 6.

01  parseefd-key-description.
```

```
    03  parseefd-number-segments                pic x comp-n.
    03  parseefd-dup-flag                       pic x comp-n.
        88  parseefd-allow-duplicates           value 1 false 0.
    03  parseefd-segment-description
            occurs max-segs times
            indexed by parseefd-seg-idx.
        05  parseefd-segment-length             pic x comp-n.
        05  parseefd-segment-offset             pic x(4) comp-n.
    03  parseefd-number-key-fields              pic x comp-n.
    03  parseefd-key-fields
            occurs parseefd-maxnumkeyfields times
            indexed by parseefd-key-field-idx.
        05  parseefd-key-field-num              pic xx comp-n.

01  parseefd-condition-description.
    03  parseefd-condition-type                 pic x comp-n.
        88  parseefd-equal-condition            value 1.
        88  parseefd-and-condition              value 2.
        88  parseefd-other-condition            value 3.
        88  parseefd-gt-condition               value 4.
        88  parseefd-ge-condition               value 5.
        88  parseefd-lt-condition               value 6.
        88  parseefd-le-condition               value 7.
        88  parseefd-ne-condition               value 8.
        88  parseefd-or-condition               value 9.
        88  parseefd-comparison-cond            values 1, 4 through 8.
    03  parseefd-condition-flag                 pic x.
        88  parseefd-true-condition             value 'Y' false 'N'.
    03  parseefd-comparison-conditions.
        05  parseefd-comp-fieldnum              pic xx comp-n.
        05  parseefd-comp-fieldname             pic x(30).
        05  parseefd-comp-field-val             pic x(50).
    03  parseefd-other-conditions
                redefines parseefd-comparison-conditions.
        05  parseefd-other-fieldnum             pic xx comp-n.
        05  parseefd-other-fieldname            pic x(30).
    03  parseefd-and-or-conditions
                redefines parseefd-comparison-conditions.
        05  parseefd-condition-1                pic xx comp-n.
        05  parseefd-condition-2                pic xx comp-n.
    03  parseefd-condition-tablename            pic x(30).

01  parseefd-field-description.
    03  parseefd-field-offset                   pic x(4) comp-n.
    03  parseefd-field-length                   pic x(4) comp-n.
    03  parseefd-field-type                     pic x comp-n.
        88  parseefd-signed-field               values parseefd-numsignsep
                                                       parseefd-numsigned
                                                       parseefd-numseplead
                                                       parseefd-numleading
                                                       parseefd-compsigned
                                                       parseefd-packedsigned
                                                       parseefd-binarysigned
                                                       parseefd-nativesigned.
        88  parseefd-num-field                  values parseefd-numedited thru
                                                       parseefd-nativeunsigned.
        88  parseefd-float-field                value  parseefd-flt.
        88  parseefd-ascii-field                values parseefd-
alphanum thru parseefd-group.
```

```
        88  parseefd-nat-field                  values parseefd-nat-
type thru parseefd-natedited.
        88  parseefd-wide-field                values parseefd-wide-type thru
                                                       parseefd-wideedited.
    03  parseefd-field-digits                  pic x comp-n.
    03  parseefd-field-scale                   signed-short.
    03  parseefd-field-user-type               pic xx comp-n.
    03  parseefd-field-condition               pic xx comp-n.
    03  parseefd-field-level                   pic x comp-n.
    03  parseefd-field-name                    pic x(30).
    03  parseefd-field-format                  pic x(30).
    03  parseefd-field-occurs-depth            pic x comp-n.
    03  parseefd-field-occurs-table
            occurs parseefd-maxnumkeyfields times
            indexed by parseefd-field-occurs-level.
        05  parseefd-field-occ-max-idx         pic xx comp-n.
        05  parseefd-field-occ-this-idx        pic xx comp-n.
    03  parseefd-field-in-key-flag             pic x.
        88  parseefd-field-is-in-key           value 'Y' false 'N'.
    03  parseefd-field-secondary-flag          pic x.
        88  parseefd-field-is-secondary        value 'Y' false 'N'.
    03  parseefd-field-hidden-flag             pic x.
        88  parseefd-field-is-hidden           value 'Y' false 'N'.
    03  parseefd-field-read-only-flag          pic x.
        88  parseefd-field-is-read-only        value 'Y' false 'N'.
    03  parseefd-field-dbtype                  pic x(32).
    03  parseefd-field-type-description        pic x(30).

01  parseefd-text-error-messages.
    03  pic x(80).                             | no error 1.
    03  pic x(80)                              value "Error opening EFD file".
    03  pic x(80)                              value "Error reading EFD file".
    03  pic x(80)                              value "EFD file Version error".
    03  pic x(80)                              value "EFD file doesn't match indexed
 file".
    03  pic x(80)                              value "Memory request size error".
    03  pic x(80)                              value "No memory".
    03  pic x(80)                              value "Too many key fields".
    03  pic x(80)                              value "No EFD file has been parsed".
    03  pic x(80)                              value "An EFD file has already been p
arsed".
    03  pic x(80)                              value "Invalid key index value".
    03  pic x(80)                              value "Invalid condition index value"
.
    03  pic x(80)                              value "Invalid field index value".
    03  pic x(80)                              value "This EFD is not for an Indexed
 File".
01  parseefd-text-error-messages-array
    redefines parseefd-text-error-messages.
    03  parseefd-text-error-message
        pic x(80)
        occurs 14 times.

       >>SOURCE FORMAT PREVIOUS
```

isprint.def

```
*>*****************************************************************<*
```

```
*>***** Copyright (c) 2005 - 2020 Veryant. Users of isCOBOL      *****<*
*>***** may freely modify and redistribute this program.         *****<*
*>*****************************************************************<*

       >>SOURCE FORMAT FREE

78  winprint-supported                      value 1.
78  winprint-setup                          value 2.
78  winprint-setup-old                      value 2.
78  winprint-set-std-font                   value 3.
78  winprint-get-page-layout                value 4.
78  winprint-get-settings-size              value 5.
78  winprint-get-settings                   value 6.
78  winprint-set-settings                   value 7.
78  winprint-set-font                       value 8.
78  winprint-set-lines-per-page             value 9.
78  winprint-get-capabilities               value 10.
78  winprint-print-bitmap                   value 11.
78  winprint-set-margins                    value 12.
78  winprint-get-no-printers                value 13.
78  winprint-get-printer-info               value 14.
78  winprint-set-printer                    value 15.
78  winprint-get-current-info               value 16.
78  winprint-set-data-columns               value 18.
78  winprint-clear-data-columns             value 19.
78  winprint-set-page-column                value 20.
78  winprint-clear-page-columns             value 21.
78  winprint-get-page-column                value 22.
78  winprint-graph-brush                    value 23.
78  winprint-graph-pen                      value 24.
78  winprint-graph-draw                     value 25.
78  winprint-set-cursor                     value 26.
78  winprint-get-spool-err                  value 27.
78  winprint-get-printer-info-ex            value 28.
78  winprint-set-printer-ex                 value 29.
78  winprint-get-current-info-ex            value 30.
78  winprint-set-text-color                 value 31.
78  winprint-get-printer-status             value 32.
78  winprint-get-printer-media              value 33.
78  winprint-get-job-status                 value 34.
78  winprint-set-job-status                 value 35.
78  winprint-set-job                        value 36.
78  winprint-update-printers                value 39.
78  winprint-set-background-color           value 94.
78  winprint-set-header-footer              value 95.
78  winprint-set-custom-paper               value 96.
78  winprint-cancel-job                     value 97.
78  winprint-set-attribute                  value 98.
78  winprint-set-printer-as                 value 99.

78  wprterr-unsupported                     value 0.
78  wprterr-bad-arg                         value -1.
78  wprterr-cancelled                       value -2.
78  wprterr-buffer-too-small                value -3.
78  wprterr-no-memory                       value -4.
78  wprterr-spooler-open                    value -5.
78  wprterr-spooler-closed                  value -6.
78  wprterr-device-incapable                value -7.
78  wprterr-enum-fail                       value -8.
```

```
78  wprterr-drv-loadfail                    value -9.
78  wprterr-bad-driver                      value -10.
78  wprterr-spool-err                       value -11.

78  wprtfont-default                        value 1.
78  wprtfont-courier-12                     value 2.
78  wprtfont-courier-12-comp                value 3.
78  wprtfont-courier-10                     value 4.
78  wprtfont-courier-10-comp                value 5.

78  wprtbitmap-printer-bitmap               value 1.
78  wprtbitmap-scale-cells                  value 2.
78  wprtbitmap-scale-inches                 value 4.
78  wprtbitmap-scale-centimeters            value 8.
78  wprtbitmap-scale-pixels                 value 16.
78  wprtbitmap-units-inches                 value 32.
78  wprtbitmap-units-centimeters            value 64.
78  wprtbitmap-units-pixels                 value 128.
78  wprtbitmap-units-cells-abs              value 256.
78  wprtbitmap-units-inches-abs             value 512.
78  wprtbitmap-units-centimeters-abs        value 1024.

78  wprtmargin-default-margins              value 0.
78  wprtmargin-cells                        value 1.
78  wprtmargin-inches                       value 2.
78  wprtmargin-centimeters                  value 3.
78  wprtmargin-pixels                       value 4.

78  wprtsel-orient-default                  value 0.
78  wprtsel-orient-portrait                 value 1.
78  wprtsel-orient-landscape                value 2.
78  wprtsel-quality-default                 value 0.
78  wprtsel-quality-high                    value -1.
78  wprtsel-quality-medium                  value -2.
78  wprtsel-quality-low                     value -3.
78  wprtsel-quality-draft                   value -4.

78  wprtunits-cells                         value 0.
78  wprtunits-inches                        value 1.
78  wprtunits-centimeters                   value 2.
78  wprtunits-pixels                        value 3.
78  wprtunits-cells-abs                     value 4.
78  wprtunits-inches-abs                    value 5.
78  wprtunits-centimeters-abs               value 6.

78  wprtalign-none                          value space.
78  wprtalign-left                          value "L".
78  wprtalign-right                         value "R".
78  wprtalign-center                        value "C".
78  wprtalign-decimal                       value "D".
78  wprtalign-decimal-suppress              value "S".
78  wprtalign-right-sign                    value "-".

78  wprt-draw-rectangle                     value 1.
78  wprt-draw-round-rectangle               value 2.
78  wprt-draw-line                          value 3.

78  wprt-pen-solid                          value 0.
78  wprt-pen-dash                           value 1.
```

```
78  wprt-pen-dot                            value 2.
78  wprt-pen-dashdot                        value 3.
78  wprt-pen-dashdotdot                     value 4.
78  wprt-pen-null                           value 5.
78  wprt-pen-insideframe                    value 6.

78  wprt-brush-solid                        value 0.
78  wprt-brush-null                         value 1.
78  wprt-brush-bdiagonal                    value 2.
78  wprt-brush-cross                        value 3.
78  wprt-brush-diagcross                    value 4.
78  wprt-brush-fdiagonal                    value 5.
78  wprt-brush-horizontal                   value 6.
78  wprt-brush-vertical                     value 7.
78  wprt-brush-dkgray                       value 8.
78  wprt-brush-gray                         value 9.
78  wprt-brush-ltgray                       value 10.

78  wprt-color-black                        value x#000000.
78  wprt-color-red                          value x#0000FF.
78  wprt-color-green                        value x#00FF00.
78  wprt-color-blue                         value x#FF0000.
78  wprt-color-yellow                       value x#00FFFF.
78  wprt-color-magenta                      value x#FF00FF.
78  wprt-color-cyan                         value x#FFFF00.
78  wprt-color-white                        value x#FFFFFF.

01  winprint-data.
    03  wprtdata-set-std-font.
        05  wprtdata-std-font                   pic x comp-x.
        05  filler                              pic x(21).
    03  wprtdata-page-layout  redefines wprtdata-set-std-font.
        05  wprtdata-lines-per-page             unsigned-short.
        05  wprtdata-columns-per-page           unsigned-short.
    03  wprtdata-set-font     redefines wprtdata-set-std-font.
        05  wprtdata-font                       handle of font.
    03  wprtdata-capabilities redefines wprtdata-set-std-font.
        05  wprtdata-bitmaps-ok-flag            pic 9.
            88  wprtdata-bitmaps-ok             value 1, false zero.
    03  wprtdata-print-bitmap redefines wprtdata-set-std-font.
        05  wprtdata-bitmap                     pic x(4) comp-n.
        05  wprtdata-bitmap-row                 pic 9(7)v99 comp-5.
        05  wprtdata-bitmap-col                 pic 9(7)v99 comp-5.
        05  wprtdata-bitmap-height              pic 9(7)v99 comp-5.
        05  wprtdata-bitmap-width               pic 9(7)v99 comp-5.
        05  wprtdata-bitmap-flags               unsigned-short.
    03  wprtdata-margins      redefines wprtdata-set-std-font.
        05  wprtdata-top-margin                 pic 9(7)v99 comp-5.
        05  wprtdata-bottom-margin              pic 9(7)v99 comp-5.
        05  wprtdata-left-margin                pic 9(7)v99 comp-5.
        05  wprtdata-right-margin               pic 9(7)v99 comp-5.
        05  wprtdata-margin-units               unsigned-short.
    03  wprtdata-draw         redefines wprtdata-set-std-font.
        05  wprtdata-draw-start-x               pic 9(7)v99 comp-5.
        05  wprtdata-draw-start-y               pic 9(7)v99 comp-5.
        05  wprtdata-draw-stop-x                pic 9(7)v99 comp-5.
        05  wprtdata-draw-stop-y                pic 9(7)v99 comp-5.
        05  wprtdata-draw-units                 unsigned-short.
        05  wprtdata-draw-shape                 unsigned-short.
```

```
    03  wprtdata-pen          redefines wprtdata-set-std-font.
        05  wprtdata-pen-style                  unsigned-short.
        05  wprtdata-pen-width                  unsigned-short.
        05  wprtdata-pen-color                  pic 9(9) comp-5.
    03  wprtdata-brush        redefines wprtdata-set-std-font.
        05  wprtdata-brush-style                unsigned-short.
        05  wprtdata-brush-color                pic 9(9) comp-5.
    03  wprtdata-text-color   redefines wprtdata-set-std-font
                                                pic 9(9) comp-5.

78  max-paper-sizes                            value 67.
78  max-paper-trays                            value 13.
78  max-printer-status                         value 26.
78  max-job-status                             value 12.

01  winprint-selection.
    03  winprint-name                          pic x(80).
    03  winprint-port                          pic x(80).
    03  winprint-driver                        pic x(80).
    03  winprint-drv-version                   signed-int.
    03  winprint-no-of-printers                signed-short.
        88 wprterr-no-printers                 value -1.
    03  winprint-is-default                    signed-short.
        88 wprt-is-not-default                 value 0.
        88 wprt-is-default                     value 1.
    03  winprint-copies                        signed-short.
        88 wprt-has-no-copy                    value 1.
    03  winprint-orientation                   signed-short.
        88 wprt-has-no-landscape               value 0.
        88 wprt-has-landscape                  value 1.
    03  winprint-quality                       signed-short.
    03  winprint-curr-orientation              signed-short.
        88 wprt-curr-default                   value 0.
        88 wprt-curr-portrait                  value 1.
        88 wprt-curr-landscape                 value 2.
    03  winprint-curr-copies                   signed-short.
    03  winprint-duplex                        signed-short.
        88 wprt-has-no-duplex                  value 0.
        88 wprt-has-duplex                     value 1.
    03  winprint-collate                       signed-short.
        88 wprt-has-no-collate                 value 0.
        88 wprt-has-collate                    value 1.
    03  winprint-color                         signed-short.
        88 wprt-has-no-color                   value 0.
        88 wprt-has-color                      value 1.
    03  winprint-curr-duplex                   signed-short.
        88 wprt-simplex                        value 1.
        88 wprt-duplex-vertical                value 2.
        88 wprt-duplex-horizontal              value 3.
    03  winprint-curr-collate                  signed-short.
        88 wprt-collate-off                    value 0.
        88 wprt-collate-on                     value 1.
    03  winprint-curr-papersize                signed-short.
        88 wprt-letter                         value  1.
        88 wprt-lettersmall                    value  2.
        88 wprt-tabloid                        value  3.
        88 wprt-ledger                         value  4.
        88 wprt-legal                          value  5.
        88 wprt-statement                      value  6.
```

```
88 wprt-executive                    value  7.
88 wprt-a3                           value  8.
88 wprt-a4                           value  9.
88 wprt-a4small                      value 10.
88 wprt-a5                           value 11.
88 wprt-b4                           value 12.
88 wprt-b5                           value 13.
88 wprt-folio                        value 14.
88 wprt-quarto                       value 15.
88 wprt-10x14                        value 16.
88 wprt-11x17                        value 17.
88 wprt-note                         value 18.
88 wprt-env_9                        value 19.
88 wprt-env_10                       value 20.
88 wprt-env_11                       value 21.
88 wprt-env_12                       value 22.
88 wprt-env_14                       value 23.
88 wprt-csheet                       value 24.
88 wprt-dsheet                       value 25.
88 wprt-esheet                       value 26.
88 wprt-env_dl                       value 27.
88 wprt-env_c5                       value 28.
88 wprt-env_c3                       value 29.
88 wprt-env_c4                       value 30.
88 wprt-env_c6                       value 31.
88 wprt-env_c65                      value 32.
88 wprt-env_b4                       value 33.
88 wprt-env_b5                       value 34.
88 wprt-env_b6                       value 35.
88 wprt-env_italy                    value 36.
88 wprt-env_monarch                  value 37.
88 wprt-env_personal                 value 38.
88 wprt-fanfold_us                   value 39.
88 wprt-fanfold_std_german           value 40.
88 wprt-fanfold_lgl_german           value 41.
88 wprt-japanese-postcard            value 43.
88 wprt-env-9x11                     value 44.
88 wprt-10x11                        value 45.
88 wprt-15x11                        value 46.
88 wprt-env-invite                   value 47.
88 wprt-us-letter-extra              value 50.
88 wprt-us-legal-extra               value 51.
88 wprt-tabloid-extra                value 52.
88 wprt-a4-extra                     value 53.
88 wprt-letter-transverse            value 54.
88 wprt-a4-transverse                value 55.
88 wprt-letter-extra-transverse      value 56.
88 wprt-supera-a4                    value 57.
88 wprt-superb-a3                    value 58.
88 wprt-us-letter-plus               value 59.
88 wprt-a4-plus                      value 60.
88 wprt-a5-transverse                value 61.
88 wprt-b5-transverse                value 62.
88 wprt-a3-extra                     value 63.
88 wprt-a5-extra                     value 64.
88 wprt-b5-extra                     value 65.
88 wprt-a2                           value 66.
88 wprt-a3-transverse                value 67.
88 wprt-a3-extra-transverse          value 68.
```

```
           88 wprt-oufuko-postcard                value 69.
           88 wprt-a6                             value 70.

       03  winprint-curr-tray                     signed-short.
           88 wprt-upper-tray                     value 1.
           88 wprt-lower-tray                     value 2.
           88 wprt-middle-tray                    value 3.
           88 wprt-manual-tray                    value 4.
           88 wprt-envelope-tray                  value 5.
           88 wprt-envmanual-tray                 value 6.
           88 wprt-auto-tray                      value 7.
           88 wprt-tractor-tray                   value 8.
           88 wprt-smallfmt-tray                  value 9.
           88 wprt-largefmt-tray                  value 10.
           88 wprt-largecapacity-tray             value 11.
           88 wprt-cassette-tray                  value 14.
           88 wprt-formsource-tray                value 15.
       03  winprint-curr-color                    signed-short.
           88 wprt-monochrome                     value 0.
           88 wprt-color                          value 1.
       03  winprint-job-title                     pic x(80).

   01  winprint-media.
       03  winprint-media-printer                 pic x(80).
       03  winprint-media-port                    pic x(80).
       03  winprint-media-papercount              signed-short.
       03  winprint-media-traycount               signed-short.
       03  winprint-media-paper                   signed-short occurs max-paper-sizes.
       03  winprint-media-trays                   signed-short occurs max-paper-trays.

   01  winprint-job-status.
       03  winprint-job-printer                   pic x(80).
       03  winprint-job-id                        signed-int.
       03  winprint-job-status-no                 pic 9(9) comp-5.
           88 wprt-job-pause                      value 1.
           88 wprt-job-resume                     value 2.
           88 wprt-job-cancel                     value 3.
           88 wprt-job-restart                    value 4.
       03  winprint-job-position                  signed-int.
       03  winprint-job-page-total                signed-int.
       03  winprint-job-page-printed              signed-int.
       03  winprint-job-status-text               pic x(80).

   01  winprint-column.
       03  winprint-col-start                     pic 9(7)v99 comp-5.
       03  winprint-col-indent                    pic 9(7)v99 comp-5.
       03  winprint-col-separation                pic 9(7)v99 comp-5.
       03  winprint-col-font                      handle of font.
       03  winprint-col-units                     pic 99 comp-x.
       03  winprint-col-alignment                 pic x.
       03  winprint-transparency                  pic 99 comp-x.
           88  winprint-transparent               value 1, false 0.

   01  misc-winprint-tables.
       03  paper-sizes.
           05 filler                  pic x(34) value "letter 8 1/2 x 11 in".
           05 filler                  pic x(34) value "letter small 8 1/2 x 11 in".
           05 filler                  pic x(34) value "tabloid 11 x 17 in".
           05 filler                  pic x(34) value "ledger 17 x 11 in".
```

```
        05 filler                   pic x(34) value "legal 8 1/2 x 14 in".
        05 filler                   pic x(34) value "statement 5 1/2 x 8 1/2 in".
        05 filler                   pic x(34) value "executive 7 1/4 x 10 1/2 in".
        05 filler                   pic x(34) value "a3 297 x 420 mm".
        05 filler                   pic x(34) value "a4 210 x 297 mm".
        05 filler                   pic x(34) value "a4 small 210 x 297 mm".
        05 filler                   pic x(34) value "a5 148 x 210 mm".
        05 filler                   pic x(34) value "b4 (jis) 250 x 354 mm".
        05 filler                   pic x(34) value "b5 (jis) 182 x 257 mm".
        05 filler                   pic x(34) value "folio 8 1/2 x 13 in".
        05 filler                   pic x(34) value "quarto 215 x 275 mm".
        05 filler                   pic x(34) value "10x14 in".
        05 filler                   pic x(34) value "11x17 in".
        05 filler                   pic x(34) value "note 8 1/2 x 11 in".
        05 filler                   pic x(34) value "envelope #9 3 7/8 x 8 7/8 in".
        05 filler                   pic x(34) value "envelope #10 4 1/8 x 9 1/
2 in".
        05 filler                   pic x(34) value "envelope #11 4 1/2 x 10 3/
8 in".
        05 filler                   pic x(34) value "envelope #12 4 \276 x 11 in".
        05 filler                   pic x(34) value "envelope #14 5 x 11 1/2 in".
        05 filler                   pic x(34) value "c size sheet".
        05 filler                   pic x(34) value "d size sheet".
        05 filler                   pic x(34) value "e size sheet".
        05 filler                   pic x(34) value "envelope dl 110 x 220 mm".
        05 filler                   pic x(34) value "envelope c5 162 x 229 mm".
        05 filler                   pic x(34) value "envelope c3  324 x 458 mm".
        05 filler                   pic x(34) value "envelope c4  229 x 324 mm".
        05 filler                   pic x(34) value "envelope c6  114 x 162 mm".
        05 filler                   pic x(34) value "envelope c65 114 x 229 mm".
        05 filler                   pic x(34) value "envelope b4  250 x 353 mm".
        05 filler                   pic x(34) value "envelope b5  176 x 250 mm".
        05 filler                   pic x(34) value "envelope b6  176 x 125 mm".
        05 filler                   pic x(34) value "envelope 110 x 230 mm".
        05 filler                   pic x(34) value "envelope monarch 3.875 x 7.5 i
n".
        05 filler                   pic x(34) value "6 3/4 envelope 3 5/8 x 6 1/
2 in".
        05 filler                   pic x(34) value "us std fanfold 14 7/
8 x 11 in".
        05 filler                   pic x(34) value "german std fanfold 8 1/
2 x 12 in".
        05 filler                   pic x(34) value "german legal fanfold 8 1/
2 x 13 in".
        05 filler                   pic x(34) value "japanese-postcard".
        05 filler                   pic x(34) value "na-9x11-envelope".
        05 filler                   pic x(34) value "10 x 11 in".
        05 filler                   pic x(34) value "15 x 11 in".
        05 filler                   pic x(34) value "Envelope Invite 220 x 220 mm".
        05 filler                   pic x(34) value "US Letter Extra 9 1/
2 x 12 in".
        05 filler                   pic x(34) value "US Legal Extra 9 1/2 x 15 in".
        05 filler                   pic x(34) value "US Tabloid Extra 11.69 x 18 in
".
        05 filler                   pic x(34) value "A4 Extra 9.27 x 12.69 in".
        05 filler                   pic x(34) value "Letter Transverse 8 1/
2 x 11 in".
        05 filler                   pic x(34) value "A4 Transverse 210 x 297 mm".
        05 filler                   pic x(34) value "Letter Extra Transverse".
```

```
        05 filler                      pic x(34) value "SuperA/SuperA/
A4 227 x 356 mm".
        05 filler                      pic x(34) value "SuperB/SuperB/
A3 305 x 487 mm".
        05 filler                      pic x(34) value "US Letter Plus 8.5 x 12.69 in"
.
        05 filler                      pic x(34) value "A4 Plus 210 x 330 mm".
        05 filler                      pic x(34) value "A5 Transverse 148 x 210 mm".
        05 filler                      pic x(34) value "B5 (JIS) Transverse 182 x 257
mm".
        05 filler                      pic x(34) value "A3 Extra 322 x 445 mm".
        05 filler                      pic x(34) value "A5 Extra 174 x 235 mm".
        05 filler                      pic x(34) value "B5 (ISO) Extra 201 x 276 mm".
        05 filler                      pic x(34) value "iso-a2".
        05 filler                      pic x(34) value "A3 Transverse 297 x 420 mm".
        05 filler                      pic x(34) value "A3 Extra Transverse 322 x 445
mm".
        05 filler                      pic x(34) value "oufuko-postcard".
        05 filler                      pic x(34) value "iso-a6".
    03  paper-size-table redefines paper-sizes   pic x(34) occurs max-paper-sizes.
    03  paper-trays.
        05 filler                      pic x(34) value "upper tray".
        05 filler                      pic x(34) value "lower tray".
        05 filler                      pic x(34) value "middle tray".
        05 filler                      pic x(34) value "manual".
        05 filler                      pic x(34) value "envelope tray".
        05 filler                      pic x(34) value "manual envelope".
        05 filler                      pic x(34) value "auto".
        05 filler                      pic x(34) value "tractor feeder".
        05 filler                      pic x(34) value "small format tray".
        05 filler                      pic x(34) value "large format tray".
        05 filler                      pic x(34) value "large capacity tray".
        05 filler                      pic x(34) value "cassette tray".
        05 filler                      pic x(34) value "form source tray".
    03  paper-tray-table redefines paper-trays   pic x(34) occurs max-paper-trays.

01  winprint-printer-status                      signed-int.

01  printer-conditions-r.
    03  printer-status-idle                      pic 9(9) comp-5 value 0.
    03  printer-status-paused                    pic 9(9) comp-5 value 1.
    03  printer-status-error                     pic 9(9) comp-5 value 2.
    03  printer-status-pending-deletion          pic 9(9) comp-5 value 4.
    03  printer-status-paper-jam                 pic 9(9) comp-5 value 8.
    03  printer-status-paper-out                 pic 9(9) comp-5 value 16.
    03  printer-status-manual-feed               pic 9(9) comp-5 value 32.
    03  printer-status-paper-problem             pic 9(9) comp-5 value 64.
    03  printer-status-offline                   pic 9(9) comp-5 value 128.
    03  printer-status-io-active                 pic 9(9) comp-5 value 256.
    03  printer-status-busy                      pic 9(9) comp-5 value 512.
    03  printer-status-printing                  pic 9(9) comp-5 value 1024.
    03  printer-status-output-bin-full           pic 9(9) comp-5 value 2048.
    03  printer-status-not-available             pic 9(9) comp-5 value 4096.
    03  printer-status-waiting                   pic 9(9) comp-5 value 8192.
    03  printer-status-processing                pic 9(9) comp-5 value 16384.
    03  printer-status-initializing              pic 9(9) comp-5 value 32768.
    03  printer-status-warming-up                pic 9(9) comp-5 value 65536.
    03  printer-status-toner-low                 pic 9(9) comp-5 value 131072.
    03  printer-status-no-toner                  pic 9(9) comp-5 value 262144.
```

```
    03  printer-status-page-punt                pic 9(9) comp-5 value 524288.
    03  printer-status-user-intervention        pic 9(9) comp-5 value 1048576.
    03  printer-status-out-of-memory            pic 9(9) comp-5 value 2097152.
    03  printer-status-door-open                pic 9(9) comp-5 value 4194304.
    03  printer-status-server-unknown           pic 9(9) comp-5 value 8388608.
    03  printer-status-power-save               pic 9(9) comp-5 value 16777216.
01  printer-conditions redefines printer-conditions-r
                                                pic 9(9) comp-5 occurs max-printer-
status.

01  job-conditions-r.
    03  job-status-paused                       pic 9(9) comp-5 value 1.
    03  job-status-error                        pic 9(9) comp-5 value 2.
    03  job-status-deleting                     pic 9(9) comp-5 value 4.
    03  job-status-spooling                     pic 9(9) comp-5 value 8.
    03  job-status-printing                     pic 9(9) comp-5 value 16.
    03  job-status-offline                      pic 9(9) comp-5 value 32.
    03  job-status-paperout                     pic 9(9) comp-5 value 64.
    03  job-status-printed                      pic 9(9) comp-5 value 128.
    03  job-status-deleted                      pic 9(9) comp-5 value 256.
    03  job-status-stat-blocked-devq            pic 9(9) comp-5 value 512.
    03  job-status-user-intervention            pic 9(9) comp-5 value 1024.
    03  job-status-restart                      pic 9(9) comp-5 value 2048.
01  job-conditions redefines job-conditions-r   pic 9(9) comp-5 occurs max-job-
status.

01  winprint-as                                 pic 9 value 0.
    88 wprt-as-client                          value 0.
    88 wprt-as-server                          value 1.

77  pdfcrypt-type                    pic 9(9) value 0.
78  pdfcrypt-no                      value 0.
78  pdfcrypt-std-40                  value 1.
78  pdfcrypt-std-128                 value 2.
78  pdfcrypt-aes-128                 value 3.
78  pdfcrypt-no-metadata             value x#08.
78  pdfcrypt-embedded-files-only     value x#10.

78  pdfcrypt-allow-printing          value x#0100.
78  pdfcrypt-allow-modify-content    value x#0200.
78  pdfcrypt-allow-copy              value x#0400.
78  pdfcrypt-allow-modify-annotations value x#0800.
78  pdfcrypt-allow-fill-in           value x#1000.
78  pdfcrypt-allow-screenreaders     value x#2000.
78  pdfcrypt-allow-assembly          value x#4000.
78  pdfcrypt-allow-degraded-printing  value x#8000.
78  pdfcrypt-all-permissions         value x#FF00.

       >>SOURCE FORMAT PREVIOUS
```

isresize.def

```
*>*****************************************************************<*
*>***** Copyright (c) 2005 - 2020 Veryant. Users of isCOBOL      *****<*
*>***** may freely modify and redistribute this program.         *****<*
*>*****************************************************************<*

       >>SOURCE FORMAT FREE
```

```
78  rlm-no-action                           value 0.
78  rlm-resize-x                            value 1.
78  rlm-move-x                              value 2.
78  rlm-no-min-x                            value 4.
78  rlm-resize-x-any                        value rlm-resize-x + rlm-no-min-x.
78  rlm-move-x-any                          value rlm-move-x + rlm-no-min-x.

78  rlm-resize-y                            value 16.
78  rlm-move-y                              value 32.
78  rlm-no-min-y                            value 64.
78  rlm-resize-y-any                        value rlm-resize-y + rlm-no-min-y.
78  rlm-move-y-any                          value rlm-move-y + rlm-no-min-y.

78  rlm-resize-both                         value rlm-resize-x + rlm-resize-y.
78  rlm-resize-both-any                     value rlm-resize-x-any + rlm-resize-
y-any.

78  rlm-move-both                           value rlm-move-x + rlm-move-y.
78  rlm-move-both-any                       value rlm-move-x-any + rlm-move-y-
any.

77  lm-resize                               handle of layout-manager, lm-resize.
77  lm-scale                                handle of layout-manager, lm-scale.
77  lm-wrap                                 handle of layout-manager, lm-wrap.
77  lm-responsive                           handle of layout-manager, lm-
responsive.
77  lm-zoom                                 handle of layout-manager, lm-zoom.

77  custom-lm-scale                         handle of layout-manager, lm-
scale "bar=103 label=119".
77  custom-lm-zoom                          handle of layout-manager, lm-
zoom "bar=103 label=119".

       >>SOURCE FORMAT PREVIOUS
```

isreg.def

```
*>*****************************************************************<*
*>***** Copyright (c) 2005 - 2020 Veryant. Users of isCOBOL      *****<*
*>***** may freely modify and redistribute this program.         *****<*
*>*****************************************************************<*

       >>SOURCE FORMAT FREE

78  hkey_classes_root_31                    value 1.

78  hkey_classes_root                       value 2147483648.
78  hkey_current_user                       value 2147483649.
78  hkey_local_machine                      value 2147483650.
78  hkey_users                              value 2147483651.
78  hkey_performance_data                   value 2147483652.
78  hkey_current_config                     value 2147483653.
78  hkey_dyn_data                           value 2147483654.

78  reg_none                                value zero.
78  reg_sz                                  value 1.
78  reg_expand_sz                           value 2.
```

```
78  reg_binary                                  value 3.
78  reg_dword                                   value 4.
78  reg_dword_little_endian                     value 4.
78  reg_dword_big_endian                        value 5.
78  reg_link                                    value 6.
78  reg_multi_sz                                value 7.
78  reg_resource_list                           value 8.
78  reg_full_resource_descriptor                value 9.
78  reg_resource_requirements_list              value 10.

01  win32-registry-value-type                  usage is unsigned-long.
    88  type-is-reg_none                       value zero.
    88  type-is-reg_sz                         value 1.
    88  type-is-reg_expand_sz                  value 2.
    88  type-is-reg_binary                     value 3.
    88  type-is-reg_dword                      value 4.
    88  type-is-reg_dword_little_endian        value 4.
    88  type-is-reg_dword_big_endian           value 5.
    88  type-is-reg_link                       value 6.
    88  type-is-reg_multi_sz                   value 7.
    88  type-is-reg_resource_list              value 8.
    88  type-is-reg_full_resource_descriptor   value 9.
    88  type-is-reg_resource_requirements_list value 10.

78  reg_option_non_volatile                    value 0.
78  reg_option_volatile                        value 1.

78  key_query_value                            value 1.
78  key_set_value                              value 2.
78  key_create_sub_key                         value 4.
78  key_enumerate_sub_keys                     value 8.
78  key_notify                                 value 16.
78  key_create_link                            value 32.
78  key_write                                  value 131078.
78  key_read                                   value 131097.
78  key_execute                                value 131097.
78  key_all_access                             value 983103.

78  reg_created_new_key                        value 1.
78  reg_opened_existing_key                    value 2.

       >>SOURCE FORMAT PREVIOUS
```

issocket.def

```
*>*****************************************************************<*
*>***** Copyright (c) 2005 - 2020 Veryant. Users of isCOBOL      *****<*
*>***** may freely modify and redistribute this program.         *****<*
*>*****************************************************************<*

       >>SOURCE FORMAT FREE

78  ags-create-server                          value 1.
78  ags-accept                                 value 2.
78  ags-create-client                          value 3.
78  ags-close                                  value 4.
78  ags-write                                  value 5.
78  ags-read                                   value 6.
```

```
78  ags-flush                                value 7.
78  ags-empty                                value 8.
78  ags-gethostname                          value 9.
78  ags-last-error                           value 10.
78  ags-next-read                            value 11.
78  ags-remote-name                          value 12.
78  ags-remote-addr                          value 13.
78  ags-read-line                            value 14.

78  csocket-create-server                    value ags-create-server.
78  csocket-accept                           value ags-accept.
78  csocket-create-client                    value ags-create-client.
78  csocket-close                            value ags-close.
78  csocket-write                            value ags-write.
78  csocket-read                             value ags-read.
78  csocket-flush                            value ags-flush.
78  csocket-empty                            value ags-empty.
78  csocket-gethostname                      value ags-gethostname.
78  csocket-last-error                       value ags-last-error.
78  csocket-next-read                        value ags-next-read.
78  csocket-remote-name                      value ags-remote-name.
78  csocket-remote-addr                      value ags-remote-addr.
78  csocket-read-line                        value ags-read-line.
78  csocket-getremoteaddress                 value 31.

01  csocket-remote-address.
    03 csocket-remote-hostname               pic x(64).
    03 csocket-remote-ip-address             pic x(15).
    03 csocket-remote-port                   pic 9(5).

       >>SOURCE FORMAT PREVIOUS
```

iswinhelp.def

```
*>*************************************************************************<*
*>***** Copyright (c) 2005 - 2020 Veryant. Users of isCOBOL      *****<*
*>***** may freely modify and redistribute this program.         *****<*
*>*************************************************************************<*

       >>SOURCE FORMAT FREE

78  help-context                            value 1.
78  help-quit                               value 2.
78  help-contents                           value 3.
78  help-helponhelp                         value 4.
78  help-setcontents                        value 5.
78  help-contextpopup                       value 8.
78  help-forcefile                          value 9.
78  help-key                                value 257.
78  help-command                            value 258.
78  help-partialkey                         value 261.
78  help-multikey                           value 513.
78  help-setwinpos                          value 515.

78  help-contextmenu                        value 10.
78  help-finder                             value 11.
78  help-wm-help                            value 12.
78  help-setpopup-pos                       value 13.
```

```
       >>SOURCE FORMAT PREVIOUS
```

iswinvers.def

```
*>*****************************************************************<*
*>***** Copyright (c) 2005 - 2020 Veryant. Users of isCOBOL      *****<*
*>***** may freely modify and redistribute this program.         *****<*
*>*****************************************************************<*

       >>SOURCE FORMAT FREE

01  winversion-data.
    03  win-major-version                    pic x comp-x.
    03  win-minor-version                    pic x comp-x.
    03  win-platform                         pic x comp-x.
        88  platform-win-31                  value 1.
        88  platform-win-95                  value 2.
        88  platform-win-9x                  value 2.
        88  platform-win-nt                  value 3.
            03  win-wordsize                 pic x comp-x.
                88  win-wordsize-16          value 1.
                88  win-wordsize-32          value 2.
                88  win-wordsize-64          value 3.

       >>SOURCE FORMAT PREVIOUS
```

SQLCA

```
*>*****************************************************************<*
*>***** Copyright (c) 2005 - 2020 Veryant. Users of isCOBOL      *****<*
*>***** may freely modify and redistribute this program.         *****<*
*>*****************************************************************<*

       >>SOURCE FORMAT FREE

       01  SQLCA.
           05  SQLCAID                PIC X(8).
           05  SQLCABC                PIC S9(9) COMP-5.
           05  SQLCODE                PIC S9(9) COMP-5.
           05  SQLERRML               PIC S9(4) COMP-5.
           05  SQLERRMC               PIC X(254).
           05  SQLERRP                PIC X(8).
           05  SQLERRD OCCURS 6 TIMES PIC S9(9) COMP-5.
           05  SQLWARN.
               10 SQLWARN0            PIC X(1).
               10 SQLWARN1            PIC X(1).
               10 SQLWARN2            PIC X(1).
               10 SQLWARN3            PIC X(1).
               10 SQLWARN4            PIC X(1).
               10 SQLWARN5            PIC X(1).
               10 SQLWARN6            PIC X(1).
               10 SQLWARN7            PIC X(1).
           05  SQLSTATE               PIC X(5).
           05  SQLEXT                 PIC S9(5) COMP-3 VALUE 1.

       >>SOURCE FORMAT PREVIOUS
```

stdfonts.def

```
*>*****************************************************************<*
*>***** Copyright (c) 2005 - 2019 Veryant. Users of isCOBOL      *****<*
*>***** may freely modify and redistribute this program.         *****<*
*>*****************************************************************<*

       >>SOURCE FORMAT FREE

77  fixed-font                                 handle of font fixed-font.
77  traditional-font                           handle of font traditional-font.
77  default-font                               handle of font default-font.
77  small-font                                 handle of font small-font.
77  medium-font                                handle of font medium-font.
77  large-font                                 handle of font large-font.

       >>SOURCE FORMAT PREVIOUS
```

Appendix G

Compiler Errors

The isCOBOL Compiler produces a wide range of informative messages, including different kind of Errors and Warnings.

Each message is preceded by its type.

S	Severe Error
E	Error
I	Informational
W	Warning

Severe Errors inhibit the production of the final object. Errors, Informationals and Warnings, instead, let the compilation complete.

Each message is followed by information about file name, row and column where the error has been found.

The following list contains the most common messages produced by the Compiler during the parsing of the cobol code.

Message	Meaning
'void' type not allowed here	A method that does not return any result has been used inside a cobol statement (Object Oriented Programming).
a USE phrase cannot be here	A USE phrase has been found outside of DECLARATIVES.
Ambiguous identifier	A data-item found in the statement is defined more than once in DATA DIVISION or a paragraph name is duplicated in the same SECTION.
ANY LENGTH variable expected	An INITIALIZE WITH SIZE statement has been used on an item that is not ANY LENGTH.
Behavior differs from ACU's	Although the Acucobol feature is supported by isCOBOL, its behavior is different. A typical example is the SET THREAD PRIORITY statement.
Cannot invoke an instance method from a factory method SELF	The SELF keyword can be used only in OBJECTs Procedure Division. If you need to invoke a self method in a FACTORY Procedure Division use the logical name of the class, instead.

Message	Meaning
Cannot open file	The required copybook cannot be open. Copybooks are searched in the current directory and in all of the paths listed in -sp compiler option. If -ce compiler option is used, the desired extension is appended to the copybook name before opening it. When working on platforms different than Windows, the case of the copybook name in the cobol source code must match the case of the disk file name and the Compiler must have read permission on that file. This error is returned also when the Compiler is not able to write the intermediate java file due to permission issues or wrong output directory during the compilation process.
Class not found	One of the classes defined in the REPOSITORY paragraph cannot be found. Please check the CLASSPATH.
Clause clash	There is a conflict between two clauses. Consider having ORGANIZATION SEQUENTIAL and ACCESS DYNAMIC in the same file definition, for example.
Constant already defined	The same constant has been defined more than once in the program. Maybe the same constant appears in more copybooks or a copybooks has been included more times in the same program. Duplicated constants are allowed if -cm compiler option is used.
Directive ignored	This warning is returned for statements that are recognized by the Compiler but are treated as commentary and will not have effect, for example: COPY RESOURCE
Directive not closed	A compiler directive has not been closed properly with the corresponding END-*Directive*

Duplicate key definition: <keyname>	A key has the same definition (same segments in the same order) as another key. For example:
	`ALTERNATE KEY KEY1 = ARC-D1, ARC-K1`
	`ALTERNATE KEY KEY2 = ARC-D1, ARC-K1`
	It is allowed by some file handlers (e.g. c-tree) but not by others (e.g. JIsam).

Duplicate primary key	The record key clause of an indexed file has been specified more than once.
Dynamic-capacity tables and standards tables cannot be mixed	This error is returned when OCCURS DYNAMIC item is found in the same group variable of OCCURS n TIMES item.
End statement required	A cobol statement has not been properly closed by dot or END-*statement*.
ESQL cursor already declared	A cursor has been declared more than once in a ESQL program. Cursors cannot be declared more than once in the same program, not even they are bind to different queries.

Message	Meaning

Exception already caught	The same exception has been caught two times, for example:
	`try`
	`  object:>method()`
	`catch IOException`
	`  |handle error here`
	`catch SecurityException`
	`  |handle error here`
	`catch IOException`
	`  |handle error here`
	`end-try.`
	(Object Oriented Programming)

Exception block required	A method has been invoked without catching its exceptions. If INVOKE statement has been used, add ON EXCEPTION clause to it. If the direct syntax has been used, instead, place your statement into a TRY/CATCH block, for example:
	`try`
	`  object:>method()`
	`catch exception`
	`  |handle the error here`
	`end-try.`
	(Object Oriented Programming)

Exception not raised	The specific exception is not raised by the method, check method documentation for details and correct the CATCH clause of the TRY/CATCH block. (Object Oriented Programming)
EXIT PERFORM outside of PERFORM	An EXIT PERFORM statement has been found outside of a PERFORM block.
Expected/found token mismatch	An unrecognized token has been found inside a statement in place of the expected keyword.
EFC field occurs greater then 2000 are not supported	You can't link a c-tree file that includes a OCCURS field with more than 2000 occurrences in the c-tree SQL Engine.
EFD directive in wrong format	The value of an EFD directive is missing or incorrect. Consult Using EFD directives chapter for details.
$EFD directive required	This warning is returned when the physical name of a file is a variable and -efd or -efc options are used. In this case the Compiler doesn't know how to name the efd file and uses the name of the variable. In order to tell the Compiler how to name the efd file, $EFD directive must be used. See Using EFD directives for more information about this. This warning is printed at the end of the Compiler output and is not included in the count of warnings. This is because the generation of EFD and ISS dictionaries is a separate task performed by the Compiler after the whole program has been compiled.

Message	Meaning
File not found	The source file cannot be found by the Compiler. If you're compiling on platforms different than Windows, be aware that the case of the source name passed to the Compiler must match the case of the disk file.
Function not found	The specific function is not available. See Intrinsic Functions for the list of all supported internal functions.
Identifier expected, found	An unexpected token has been found in place of an expected identifier.
Illegal condition	The specific condition cannot be resolved. A typical situation in which this error appears is when the program contains the following statement: `IF numeric-item = NULL` and -ca compiler option is not used.
Illegal picture	The picture of an elementary data item is invalid.
Illegal receiver from source type	A MOVE statement is invalid for the Compiler because the source-item and the dest-item are incompatible.
Incompatible options	Two or more of the specified Compiler options cannot be used together, for example: -dci -dcm.
Incompatible options: -pt0/-pt2 used for OO programs	The options -pt0 and -pt2 cannot be used to compile object oriented programs.
Incomplete copy statement	This error is returned when the COPY statement and the name of the copybook are not in the same row.
Incomplete replace statement	This error is returned when the REPLACE statement and its parameters are not in the same row.
Incorrect string literal	A text string has not been correctly enclosed between quotes.
Internal error EIS: Missing License	A valid isCOBOL EIS license is required in order to compile • a program that uses the HTTPHandler class, • a program that uses the HTTPClient class, • a program that includes CGI syntax (e.g. EXEC HTML), • a program compiled with -wd2 option
Invalid ALL phrase, ALL removed	This error is returned when the Compiler finds the ALL keyword followed by a numeric literal. In this case the ALL keyword is ignored. For example, `IF item-1 = ALL 1` is considered as `IF item-1 = 1`.
Invalid arguments	This error is returned when a method is invoked with the wrong number or the wrong type of parameters. (Object Oriented Programming)

Message	Meaning
Invalid concatenation	This error is returned when the & operator is used to concatenate resource strings (e.g. r"string"), numeric literals or data items. The concatenation is allowed only between string literals. Hex notation (e.g. x"41") is allowed. When a numeric literal or a data item is included in the concatenation, this error is severe and prevents the compilation to complete. When a resource string is included in the concatenation, this error is not severe; the compilation proceeds treating the resource string as a standard string (E.g. "A" & r"abc" will produce "Aabc").
Invalid file name	The name of the source file does not follow Java rules. For example, this error is returned when the source file name begins with a number or if it contains spaces.

Invalid level number	The level number of a group item is not valid. A potential situation in which this error appears is the following SCREEN SECTION entry:
	`01 screen1.`
	`  03 entry-field, line 2, col 2.`
	`  05 push-button, line 5, col 3.`
	A control cannot be child of another control.

Invalid open mode	The open mode has not been specified or is not supported by the specific file type.
Invalid Program/Class Id	The PROGRAM-ID or CLASS-ID paragraphs are missing or contain typos.
Invalid SQL directive: <type> type is invalid	The SQL type specified in the *iscobol.compiler.esql.procedure* property or in the HOSTVAR directive doesn't match any known SQL type. Refer to the java.sqlTypes javadoc for the list of supported types.
Invalid SQL directive: error parameters < 2	The number of parameters in the *iscobol.compiler.esql.procedure* property or in the HOSTVAR directive is not sufficient. Ensure to specify all the mandatory parameters.
Invalid value	The VALUE clause of a data-item contains a value that is incompatible with the item picture.
Line truncated	A statement exceeds the AREA B limit in the current source format (e.g. it exceeds column 72 in a FIXED source) therefore it's truncated and may not work at runtime.
Maybe a directive	The word includes syntax usually adopted for compiler directives, but it doesn't match with any known compiler directive.
Method not found	The specific method is not found in the invoked class. Consider that the method name is case sensitive (Object Oriented Programming).
Missing dot	A paragraph is not correctly closed. This error may be caused by other errors encountered inside the paragraph code.
Missing EFD/EFC key name	A field of a key referenced in FILE-CONTROL is hidden in the FD due to the EFD directive that were used. The most common case is using EFD USE GROUP on the parent item of the key segment.

Message	Meaning
Missing FD for file	A file has been defined in FILE-CONTROL but its description is not available in FILE SECTION.
Missing license: The license key is missing, invalid or has expired! Check your properties files or contact Veryant or your distributor for technical support.	The license for the Compiler is either expired or missing.
Missing picture clause	The picture of an elementary data item is missing.
Missing primary key	The RECORD KEY clause is missing in the Indexed file definition in FILE-CONTROL.
Missing Procedure Division	The program lacks of the PROCEDURE DIVISION.
Missing receiver	This error is caused by an incomplete MOVE statement.,
Missing record for file	FILE SECTION contains an FD entry without any record definition.
Missing relative key	The RELATIVE KEY clause is missing in the Relative file definition in FILE-CONTROL.
Missing SELECT for file	The FD of a file has been defined in FILE SECTION but the file description is not available in FILE-CONTROL.
Duplicated $EFD directives not allowed, only the last is considered.	The same EFD directive have been used more than one time on the same data item. Example: $EFD NAME=datep $EFD NAME=datePurchased 05 DATE-PURCHASED PIC 9(8). The above field will be named 'datePurchased'.
Must be one-dimensional table	When the FROM, USING or VALUE phrases of a Screen Section item are associated to an Occurs data-item, the Occurs data-item must be one-dimensional.
National with -b option may not work!	Programs containing national items (items defined as PIC N(n)) should not be compiled with -b option.
Not a sort file	A SORT statement has been issued on a file that is not a sort file.
OCCURS DEPENDING must be last in group	If placed inside a group item, an OCCURS DEPENDING item must be the last in the group.

Message	Meaning

Only a host variable or a literal is allowed here	Returned for a query like this:
	
	`exec sql`
	`  update tbl1 set c1 = c1 + 1 where current of cur2`
	`end-exec.`
	
	When you have CURRENT OF in the WHERE clause, then you can only use host variables or literals in the UPDATE statement. The following UPDATE would compile with no errors:
	
	`exec sql`
	`  update tbl1 set c1 = 3 where current of cur2`
	`end-exec.`

Only one-dimensional table allowed here	Only one dimensional OCCURS items can be used in GUI controls properties with the clauses TABLE and MULTIPLE. Only one dimensional dynamic length OCCURS (OCCURS DYNAMIC) can be sorted by the SORT verb.
OPEN INPUT with LOCK may not work!	On some platforms, the lock is not acquired on files open for input.
Option has no effect	This message is returned when the change of a reserved word issued through -rc and -rm options cannot be made on the source code.
Paragraph name found in Area B	This warning is returned when the name of a paragraph or a section is written in the Area B of the current source format. It's never returned with the Free source format, as there are no areas in it.
Program id differs from source name	The name specified in PROGRAM-ID paragraph does not match with the name of the disk source file.
RECORD KEY not in FD	The field associated to a indexed file record key is not found the record definition, it's found somewhere else in DATA DIVISION instead. This invalidates the file record key.
RECORD KEY outside smallest record	The offset of one or more of the record key segments is greater than the minimum record size of the file.
Record len is not equal than declared size	The record length specified in RECORD clause of a file FD does not match with the length of the level 01 item in that FD.
REDEFINES too long	The redefining item is larger in size than the redefined item.
Reference modifier out of range	This error is returned when a data-item is referenced over its size by constant offset and length. For example consider having 77 WRK-ITEM1 PIC X(10) in DATA DIVISION and performing the following check in PROCEDURE DIVISION: IF WRK-ITEM1(12:1) = SPACES...
Screen name not allowed in this context	Screen names can be used only in DISPLAY, ACCEPT, MODIFY and INQUIRE statements. They cannot be used in other statements. For example, they cannot be tested using IF or EVALUATE, they cannot be copied using MOVE, etcetera.

Message	Meaning
Subscript required	A data item that is part of an OCCURS has been referenced without specifying the index between parenthesis or, vice versa, a data item that is not part of an OCCURS has been referenced using an index.
Symbol not in linkage	One of the items listed in the USING clause of PROCEDURE DIVISION is not defined in LINKAGE SECTION.
Syntax error	This error is returned whenever an unrecognized token is found and there isn't a specific error message for it.
The method signature might be ambiguous	This warning is returned when you invoke a method by passing parameters that are compatible with more than one of the method signatures. It's also returned if you invoke a method of a generic class.
Too many parameters the bean code can not be generated! <ServiceBridge bean suffix><nameProgram>;	This warning is returned when the program's Linkage Section includes more than 255 data items and iscobol.compiler.servicebridge.bean is set in the configuration. The bean can't be generated because Java doesn't allow more than 255 parameters in a method.
Unbalanced parenthesis	The number of "(" into a statement does not match the number of ")".
Undeclared cursor	An ESQL statement is trying to operate on a cursor that has not been defined. Involved ESQL statements are: OPEN, FETCH and CLOSE.
Undefined constant	The program tests a Compiler constant that has not been defined. See Compiler Directives for details about Compiler constants definition and testing.
Undefined data item	The specific data-item is not defined in DATA DIVISION.
Undefined data item SQLCA	The PROCEDURE DIVISION contains ESQL statements, but the copybook SQLCA is missing in the WORKING-STORAGE. This copybook is mandatory for programs that take advantage of ESQL statements.
Unexpected end of program	The last statement in PROCEDURE DIVISION has not been properly closed by dot or END-*statement* .
Unexpected token	The specific token was not expected by the Compiler in that part of the source.
Unknown token	The specific token is not recognized by the Compiler.
Unmatched END-*statement*	An END-*statement* has been found, but there is no matching statement above of it.
Unsupported compiler directive	The specific Compiler directive is not supported by isCOBOL. See Compiler Directives for the list of all supported Compiler directives.
Unsupported feature	The specific syntax is recognized as unsupported feature. A typical example is the STATIC-LIST style for COMBO-BOX control.
Unsupported option	The specific option is not recognized by the Compiler. Use -help option to make the Compiler print a list of all supported options.

Message	Meaning
Usage must be DISPLAY	A data-item that is not USAGE DISPLAY has been used in a statement that requires only USAGE DISPLAY parameters. For example, you cannot use a USAGE HANDLE item in STRING and UNSTRING statements.
User defined error	Error traced due to >>ERROR directive. See Compiler Directives for details about the >>ERROR directive.
VALUE size error	The length of the value specified in the VALUE clause of a data-item is greater than the size of the picture.

Variable has zero length	This error is usually returned for variables without pictures that are parent of 88 level items, for example:
	`01 flag.`
	`88 flag-true value "T" false "F".`

Variable not used	This warning message is returned for each useless data-item found in the source if -wu compiler option is used.
WHEN EFD/EFC name not found #	An invalid field name has been used in a WHEN condition. The compiler is not able to find the named field among the fields of the FD where the condition was used.
WHEN OTHER not last	The WHEN OTHER condition should be the last one into an EVALUATE statement.
With decimal point comma, comma can not be separator between two digits: #,	This warning is returned when a comma is used as separator of multiple values and the DECIMAL-POINT IS COMMA clause is specified in the Special Names paragraph. For example the following item would produce the warning: `88 CONDITION-1 VALUE IS 1,2 .`
Wrong compiler directive	A Compiler directive has not been used with the proper syntax. See Compiler Directives for details about Compiler directives.
Wrong copy statement	A COPY statement is either incomplete or not correctly terminated by dot.
Wrong replace statement	A REPLACE statement is either incomplete or not correctly terminated by dot.
Malformed option: compiler.regexp=#	The regular expression specified by iscobol.compiler.regexp * property is not valid. The number of strings in the value can't be odd.

If the parsing of the cobol code completes correctly, the isCOBOL Compiler invokes the Java Compiler in order to generate the final class file.

During this second phase of the compilation process, the following error messages may show up.

java.lang.NoClassDefFoundError: com/sun/tools/javac/Main	This error means that the Java Compiler cannot be found in the CLASSPATH. Ensure that the library $JDK_HOME/lib/tools.jar appears in the CLASSPATH.
java.lang.OutOfMemoryError: Java heap space	This error means that the JVM ran out of memory while compiling the source. To avoid it, you must increase the memory limit by adding -Xmx256m Java option. 256 means 256 MB and it specifies the maximum amount of RAM that the JVM can allocate. If it's not enough, try with higher values.
code too large	This error means that, due to huge paragraphs in the source code, a java method over 64KB of bytecode has been generated and cannot be compiled. To avoid this error try using -sns=200 Compiler option. This setting will break huge paragraphs in smaller paragraphs of 200 lines in size. If the problem still exists, try with lower values. If -sns does not resolve the problem, a manual intervention on the source may be necessary. Before proceeding with the manual intervention, try using `-big` in conjunction with -sns.
Error writing file: too many constants	This error means that too many Java constants were created and the program cannot be compiled. This error is usually returned when compiling very huge source files. To avoid it, add -big option to the Compiler options. Use -big only for programs that returns this error. Other programs may experience performance slowdown if compiled with -big option.
The system is out of resources. Consult the following stack trace for details. java.lang.StackOverflowError	This error means that the thread stack area in the Java compiler was overloaded. To avoid it, increase the thread stack size using the Xss Java option. e.g. `iscc -J-Xss1m myProg.cbl`

Error numbers list

The below tables list the error numbers followed by their description.

Severe Errors

1	Incorrect string literal
2	Unknown token
3	Cannot open file
4	Unexpected compiler directive
5	Unsupported compiler directive
6	Incomplete copy statement
7	Incomplete replace statement
8	Wrong copy statement:
9	Wrong replace statement
10	File not found
11	Syntax error
12	Unexpected end of program
13	Missing dot
14	Missing SECTION keyword
15	Unexpected token
16	Invalid currency sign
17	Identifier expected + found
18	Missing keyword
19	Unsupported feature
20	Malformed variable name
21	String value expected found
22	Integer value expected found
23	Numeric value expected found
24	Data name expected found
25	Missing clause
26	Undefined file
27	Invalid level number
28	Missing picture clause

29	Invalid clause
30	Invalid value:
31	Expected/found token mismatch:
32	Illegal picture:
33	Missing
34	Picture too big
35	Undefined data item
36	Ambiguous identifier
37	Must be a GROUP item:
38	Invalid file name:
39	Cannot open file:
40	Cannot write file:
41	Subscript required:
42	Program id differs from source name:
43	Illegal condition
44	Unbalanced parenthesis
45	Unmatched
46	Integer variable expected:
47	Procedure name required
48	Clause clash
49	Numeric expression expected:
50	Invalid THRU clause
51	Missing FD for file:
52	Picture should be XX:
53	Missing SELECT for file:
54	Invalid open mode:
55	Subscript not allowed here:
56	Invalid key:
57	Missing relative key

58	Missing primary key
59	Illegal receiver from source type:
61	Internal error
62	Object wrong type for subject
63	Class already specified:
64	a USE phrase cannot be here
65	duplicate USE phrase
66	String variable expected:
67	Duplicate procedure name:
68	Usage must be DISPLAY:
69	Must be size 1 in this context:
70	Illegal size:
71	Unknown OBJECT value
72	Positive integer required
73	Missing Procedure Division
74	Class not found
75	Type clash
76	Invalid constructor
77	Method not found
78	Exception block required
79	Invalid return type
80	Invalid argument(s)
81	Object reference variable expected
82	Invalid class name
83	Invalid method name
84	Cannot invoke an instance method from a factory method
85	EXIT PERFORM outside of PERFORM
86	Invalid Program/Class Id
87	REDEFINES too long

88	SUPER not allowed here
89	Invoke with SUPER must be the first statement
90	'S' ignored:
91	Duplicate statement:
92	Repeated or conflicting option:
93	Undeclared cursor:
94	Undeclared prepare:
95	Unsupported fetch:
96	Handle variable expected:
97	String literal expected + found
98	Statement not allowed here
99	Record len is not equal to declared size:
100	Wrong compiler directive
101	Missing record for file:
102	Wrong SELECT for sort file
103	Invalid operation on sort file
104	Not a sort file
105	Procedure name not found:
109	Numeric variable required+ found:
110	Statement not allowed on pointer:
111	Undefined constant
112	User defined error:
113	Function not found
114	VALUE in REDEFINES ignored:
115	Clause not allowed here:
116	Incompatible options:
117	Missing linage clause
118	WHEN OTHER not last
119	$EFD directive required:

120	$EFD directive in wrong format:
121	Only levels 01 & 77 allowed in this context:
122	Illegal expression
123	Conflicting phrases:
124	Invalid resource name
125	Illegal DEPENDING ON
127	Only 2 level of OCCURS allowed.
128	USAGE conflict:
129	Different number of SYMBOLIC names and values
130	Missing method name
131	Exception already caught:
132	Exception not raised:
133	EXCEPTION not last
135	Procedure name not unique
136	ESQL statement not allowed here:
137	ESQL cursor already declared:
138	Dynamic-capacity tables and standards tables cannot be mixed:
139	Symbol not in linkage:
140	Invalid XML structure:
141	Numeric literal too large:
142	Operation not permitted on:
143	Native character specified twice:
144	Invalid name:
145	Duplicate method signature:
146	Unsupported option:
147	Option(s) requires WORKING-STORAGE SECTION. on a single line
148	Behavior differs from ACU's:
149	'void' type not allowed here:
153	File in multiple areas:

155	Duplicate
156	Missing receiver:
157	Constant already defined:
158	Invalid concatenation:
159	Option has no effect:
160	SIZE or LINES phrase required
161	Attempting to override a method that doesn't exist
162	Attempting to use incompatible return type:
163	Only one-dimensional table allowed here:
164	ESQL invalid STRLITERAL:
165	Screen name not allowed in this context:
166	Stack overflow:
168	RECORD KEY not in FD:
170	VALUE size error
174	Incomplete statement:
175	Duplicate primary key
176	Duplicate attribute
177	Invalid OCCURS KEY
178	RECORD KEY outside smallest record:
180	Variable not allowed here:
181	Operand not declared:
182	Not a detail group:
183	Invalid line:
184	Invalid column:
185	Not a CONTROL:
186	Not with CONTROL FOOTING group:
187	Missing clause in RD:
188	Period missing. Period Assumed.
189	Wrong subscript(s):

190	Undeclared database:
191	Constant already defined with different value:
195	The ALTER statement is only supported with the -aa compiler option.
196	Invalid host variable:
197	ESQL invalid end statement:
198	Source literal not numeric:
199	Illegal redefines.
201	Reference modification not allowed here:
202	Data item too long:
203	Malformed option:
204	Too many options:
205	Not allowed in a nested program:
206	Not allowed in an unnamed method :
208	Invalid regular-expression pattern:
209	Not allowed in class:
212	Assumed to be a paragraph without final dot:
213	Condition name not allowed here:
214	Not an interface:
215	Must be public:
216	Interface method missing:
218	Numeric literal treated as alphanumeric:
219	Invalid ALL phrase+ ALL removed:
225	Duplicate definition:
226	$ELK directive in wrong format:
228	Only a host variable or a literal is allowed here:
230	$ELK directive has wrong value:
235	SYMBOLIC value must be between 1 and 256
243	Invalid SQL directive:
256	ANY LENGTH variable expected:

261	DEPENDING ON subsidiary to OCCURS only allowed under -cod1

Errors, Informationals and Warnings

The following error conditions don't make the compilation fail. They're just warnings. You can increase their severity or avoid them to be returned by setting iscobol.compiler.messagelevel.(error-number)=(action)in the Compiler configuration.

106	Directive ignored:
107	Parameters differs from ACU's:
108	Variable not used:
126	OCCURS DEPENDING must be last in group:
134	OPEN INPUT with LOCK may not work!
150	Static context: expression evaluated only once!
151	National with -b option may not work!
152	Variable has zero length:
154	End statement required
158	Invalid concatenation:
167	Redefines not allowed as key:
169	LOCK not supported:
171	variable record len not supported for relative file:
173	Reference modifier out of range
179	WD2: Unsupported
192	Directive not closed:
193	With decimal point comma+ comma can not be separator between two digits:
194	The ALTER statement encourages the use of unstructured programming practices.
200	Maybe a directive:
207	Duplicate EFD/EFC name
210	Duplicate key definition:
211	Variable(s) declared in LINKAGE isn't in the USING clause:
217	WHEN EFD/EFC name not found
220	Possible divide by ZERO without ON SIZE ERROR

221	duplicated $EFD directives not allowed+ only the last is considered.
222	The method signature might be ambiguous:
223	Nested COPY ... REPLACING may cause unexpected results
224	This variable contains a KEY that will be lost in iss file+ because in REDEFINE:
227	ELK Directive ignored:
229	Missing EFD/EFC key name
232	EIS/Mobile: Unsupported
233	Illegal MOVE CONVERT size:
234	MOVE from alphanum to numeric:
236	VALUE has already been specified
250	Too many parameters the bean code can not be generated!
251	Line truncated
253	Paragraph name found in Area B
254	Procedure name same as data name
257	Dynamic items will be ignored
258	Gradient will not be used with ERASE/BLANK clause
263	OCCURS DYNAMIC data item without INITIALIZED involved in MOVE POSITIONAL DELIMITED DEFAULT
264	Subscript out of bounds:
265	Sql indicator not declared:

Appendix H

Runtime Errors

Except for few runtime errors (e.g. "missing license" and "unsatisfied link error") runtime errors are identified by exception messages produced by the Java virtual machine. They can be printed on the console or returned as a graphical message box depending on the error type and on iscobol.exception.message setting.

The exception message is usually followed by a stack that lists all programs and paragraphs that are involved, from the one that produced the error to the first one that were executed. Compile with -g option to see references to the COBOL source into the stack, otherwise you'll see references to the intermediate java source that were generated by the isCOBOL Compiler to produce the final class object. When "unknown source" appears in place of the source reference, it means that the corresponding Java program didn't include debug information (it usually happens with isCOBOL internal objects, since the isCOBOL Framework is compiled without debug information).

This article examines the most common runtime errors, providing suggestions about how to address them.

Standard runtime errors

The errors listed below are produced by the runtime system. They can be reproduced running in stand-alone mode as well as running in thin client mode.

```
Missing license: The license key is missing, invalid or has expired!Check your
properties files or contact Veryant or your distributor for technical support.
```

This error is returned at startup if the license is not found or is expired. Contact your Veryant's representative to obtain a valid license. Consult the chapter Getting Started to know how to install the license correctly.

```
java.lang.NoClassDefFoundError
or
java.lang.ClassNotFoundException
```

These errors mean that a java class was not found by the JVM. It can be returned at startup if the main program name has not been passed correctly in the command line (consider that Java is case-sensitive and the .class extension must be omitted) or if a typo caused Java to treat an option in the command line as if it was the program to launch.

```
java.lang.OutOfMemoryError: Java heap space
```

This error means that the JVM ran out of memory. To avoid it, you must increase the memory limit by adding -Xmx256m Java option to the command line. 256 means 256 MB and it specifies the maximum amount of RAM that the JVM can allocate. If it's not enough, try with higher values.

```
java.lang.OutOfMemoryError: Metaspace
```

This error may occur with Java version 8 or higher and it means that the non-heap memory limit of the JVM has been reached. The non-heap memory is used to store class descriptions, so this error may appear when running huge applications. To avoid it, you must increase the non-heap memory limit by adding -XX:MaxMetaspaceSize=1g Java option to the command line. "1g" means 1 GB and it specifies the maximum amount of non-heap memory that the JVM can allocate. If it's not enough, try with higher values.

```
java.lang.OutOfMemoryError: PermGen space
```

This error may occur with Java version 7 or lower and it means that the non-heap memory limit of the JVM has been reached. The non-heap memory is used to store class descriptions, so this error may appear when running huge applications. To avoid it, you must increase the non-heap memory limit by adding -XX:MaxPermSize=256m Java option to the command line. "256m" means 256 MB and it specifies the maximum amount of non-heap memory that the JVM can allocate. If it's not enough, try with higher values.

```
java.lang.OutOfMemoryError: unable to create new native thread
```

This type of OutOfMemoryError is generated when an application isn't able to create new threads. This error can occur if there is no room in the memory to accommodate new threads or if the number of threads exceeds the operating system limit. In order to know what's the maximum number of thread allowed by the host operating system, you can run the ISCHECK program installed with isCOBOL in the folder "sample/limit-tests".

```
java.lang.UnsatisfiedLinkError: no ### in java.library.path
```

This error means that the JVM failed to load a native library. The library name is traced in the error message. A typical case in which this error can appear is when iscobol.file.index is set to work with c-tree (e.g. *iscobol.file.index=ctreej*) and isCOBOL is not able to find the c-tree client library (ctree.dll on Windows and libctree.so on Unix). On Windows machines, native libraries are loaded from the paths listed in %PATH% environment variable. On Unix, they're loaded from the paths listed in $LD_LIBRARY_PATH environment variable (be aware that the name could be different depending on the o.s.). To specify the library path in a cross-platform way, you can take advantage of the -Djava.library.path Java property (i.e. java -Djava.library.path=/home/isCOBOL2010/native/lib MAIN).

```
java.lang.UnsatisfiedLinkError: Native Library ### already loaded in
another classloader
```

This error is typical in servlet containers like Tomcat. It occurs if two web applications try to load the same dynamic link library. It can happen with COBOL applications using file handlers that have native dependences, like DCI for example. In order to avoid it, consider a file connector solution if available. For example, with DCI you can use The DCI File Connector.

```
java.lang.NullPointerException
```

This error is returned when the JVM tries to use an object that is not available. In COBOL programs it usually refers to a data item. A typical case in which this error appears is when a called program tries to use a linkage item that was not passed by the caller.

```
java.lang.ArrayIndexOutOfBoundsException
```

This error means that a Java array has been referenced out of its bounds. In COBOL programs it usually refers to an OCCURS data item that it referenced outside of its occurrences (i.e. consider having 77 data-item1 PIC X(10) OCCURS 10. and performing MOVE SPACES TO data-item1(11).).

```
java.lang.reflect.InvocationTargetException
```

This error is returned when a problem occurs during the invocation of a Java object. In isCOBOL it's usually caused by missing libraries. Check the exception stack to retrieve more information on the cause.

```
error_description logical_filename
(physical_filename)FS=[file_status],EXFS=[extended_status]: error_description
```

This kind of error is returned when an I/O error occurs while working with a file and the program doesn't contain DECLARATIVES for the file and `iscobol.file.errors_ok` is set to 0.

```
Could not load the main class
```

This error is usually returned by javaw.exe on Windows. It means that the class has been found but there were problems while running it, for example if the license is missing. Use java.exe to see a more complex error message that will help in diagnosing the problem.

```
Invalid isCOBOL rts version build #xxx or later required
```

This error is returned by Database Bridge subroutines when they're not compatible with the current runtime version.

```
Native call not found
```

The error message "Native call not found" means that the COBOL program has called a C language (native) function that has the same name as the native library (.dll or .so), but the function does not exist. If you get the error "Native call not found" and you do not mean to call a native library (i.e. you are trying to call a COBOL program) then there may be a DLL or shared object library with the same name as the COBOL program in the library search path. .

```
Internal error: IIOP: Communication problem(s) [java.io.IOException: Cannot run program
"program-name"
```

This error is returned when `iscobol.file.index` is set either to "dcic", "fscsc", "mfc" or "vfc" and the isCOBOL Framework cannot find the proper executable file. Ensure that the executable is in the system Path or set *iscobol.file.connector.program.** properties to indicate where it is.

```
Active window is modal : cannot change the active window
```

This error is returned when the program tries to activate another window but the current window is modal (floating windows are modal by default) and the focus cannot be moved until the current window is closed.

```
Active window is modal : can't create a modeless child window
```

This error is returned when the program tries to create and activate a modeless window (e.g. an independent window) but the current window is modal (floating windows are modal by default) and the focus cannot be moved until the current window is closed.

```
Accept executed on an invisible window: "window's title"
```

This error is returned when the program tries to accept the user input on a window whose Visible property is set to zero

```
Internal error: java.lang.IllegalArgumentException: ct_init ERROR 19:133:0
```

This error is returned when isCOBOL is not able to connect to the c-tree server. The two common causes are

- the c-tree server is down or unreachable, or
- the c-tree client library version doesn't match with the c-tree server version.

```
CALL not found: C function system()
```

This error is returned when you call either C$SYSTEM or SYSTEM routines having *iscobol.system.exec=c* in the configuration and the dyncall library (installed with isCOBOL) can't be found in the Java library path. Only on Windows, the error can be returned also if msvcrt.dll can't be found in the Java library path.

```
CALL not found: <program_name>
```

This error is returned when a called subroutine cannot be found and the CALL statement doesn't have a ON EXCEPTION|OVERFLOW clause. The same error is returned when a remote subroutine crashes and doesn't return to the calling program.

```
Action in event procedure
```

This error is returned when the ACTION property of a Grid or Tree-View is modified within MSG events handling (e.g. *MODIFY screen1-gr-1 ACTION ACTION-ENTRY* within the handling of the MSG-GOTO-CELL event). Such operation is not allowed by isCOBOL.

```
java.lang.StackOverflowError
```

This error means that the thread stack area in the Java virtual machine was overloaded. To avoid it, increase the thread stack size using the Xss Java option. The error may also depend by an error in the COBOL code, like a program that calls itself recursively a infinite number of times.

Application Server (Thin Client) errors

The errors listed below can occur only in a thin client environment.

```
java.net.BindException: Address already in use
```

This error is returned starting the ApplicationServer if the port on which it must listen for connections is already used by another process.

```
com.iscobol.rts.IscobolRuntimeException: License exceeded #
```

This error is returned when a client connects to the Application Server but there are no more slots available because they have been used by other clients. You must wait for another client to disconnect before being able to connect to the Application Server, or you can upgrade the license in order to have more concurrent connections to the Application Server. See Users count for information about concurrent client connections to the Application Server.

```
Max number of connections reached: (n)
```

This error is returned when the isCOBOL Client connects to an Application Server in thin client environment but there are no more connection slots available. The number at the end of the message is the current connection limit. See Connections count for information about user count in Application Server environment

```
Unable to connect to <ip>, port <port>
```

This error is returned when the server can't be reached by the client. The most common causes are:

- wrong IP or port specified
- different hostname values used on client and server command-lines. E.g. the server was started with *-hostname 127.0.0.1* (server's localhost address) and the client is launched with *-hostname 192.168.1.1* (server's IP address)
- the server is down
- the server port is blocked by a firewall.

```
Disconnected from <ip>, port <port>
```

This error is returned when the connection to the server terminates unexpectedly, e.g. if a severe exception occurs server side. If you don't want to be notified, you can add the *-nodisconnecterr* option to the isCOBOL Client command line.

```
Session terminated by the Server
```

This error is returned when the connection to the server is terminated by the administrator.

```
ERROR: Client release (n1) is incompatible with Application Server (n2)
```

This error is returned when the isCOBOL Client connects to an Application Server in thin client environment, but the versions of the two products are not compatible. Update the isCOBOL installation in order to make these versions match.

```
com.iscobol.rmi.RemoteInvocationHandler$1: Call timed out. Current timeout=0
```

This error is returned when the program invokes a method of a Java-Bean passing an object reference among parameters and the client JVM is not compatible with the server JVM for the transmission of the object reference. In order to ensure that the cause of this error is actually the invocation of a Java-Bean method, check the exception stack and look for occurrences of "com.iscobol.gui.server.CobolGUIJavaBean.callMethod". Use the same JVM version on both client and server side to get rid of this error.

Appendix I

Table of ASCII values

Dec	Hex	Character
0	00	NUL (Null char.)
1	01	SOH (Start of Header)
2	02	STX (Start of Text)
3	03	ETX (End of Text)
4	04	EOT (End of Transmission)
5	05	ENQ (Enquiry)
6	06	ACK (Acknowledgment)
7	07	BEL (Bell)
8	08	BS (Backspace)
9	09	HT (Horizontal Tab)
10	0A	LF (Line Feed)
11	0B	VT (Vertical Tab)
12	0C	FF (Form Feed)
13	0D	CR (Carriage Return)
14	0E	SO (Shift Out)
15	0F	SI (Shift In)
16	10	DLE (Data Link Escape)
17	11	DC1 (XON)(Device Control 1)
18	12	DC2 (Device Control 2)

Dec	Hex	Character
19	13	DC3 (XOFF)(Device Control 3)
20	14	DC4 (Device Control 4)
21	15	NAK (Negative Acknowledgement)
22	16	SYN (Synchronous Idle)
23	17	ETB (End of Trans. Block)
24	18	CAN (Cancel)
25	19	EM (End of Medium)
26	1A	SUB (Substitute)
27	1B	ESC (Escape)
28	1C	FS (File Separator)
29	1D	GS (Group Separator)
30	1E	RS (Request to Send)(Record Separator)
31	1F	US (Unit Separator)
32	20	SP (Space)
33	21	!
34	22	“
35	23	#
36	24	$
37	25	%
38	26	&
39	27	‘
40	28	(
41	29	)
42	2A	*
43	2B	+
44	2C	,
45	2D	-
46	2E	.

Dec	Hex	Character
47	2F	/
48	30	0
49	31	1
50	32	2
51	33	3
52	34	4
53	35	5
54	36	6
55	37	7
56	38	8
57	39	9
58	3A	:
59	3B	;
60	3C	<
61	3D	=
62	3E	>
63	3F	?
64	40	@
65	41	A
66	42	B
67	43	C
68	44	D
69	45	E
70	46	F
71	47	G
72	48	H
73	49	I
74	4A	J

Dec	Hex	Character
75	4B	K
76	4C	L
77	4D	M
78	4E	N
79	4F	O
80	50	P
81	51	Q
82	52	R
83	53	S
84	54	T
85	55	U
86	56	V
87	57	W
88	58	X
89	59	Y
90	5A	Z
91	5B	[
92	5C	\
93	5D	]
94	5E	^
95	5F	_
96	60	`
97	61	a
98	62	b
99	63	c
100	64	d
101	65	e
102	66	f

Dec	Hex	Character
103	67	g
104	68	h
105	69	i
106	6A	j
107	6B	k
108	6C	l
109	6D	m
110	6E	n
111	6F	o
112	70	p
113	71	q
114	72	r
115	73	s
116	74	t
117	75	u
118	76	v
119	77	w
120	78	x
121	79	y
122	7A	z
123	7B	{
124	7C	|
125	7D	}
126	7E	~
127	7F	DEL

Appendix J

External Links

Java Download Page

http://java.sun.com/javase/downloads/index.jsp

Here you can download the latest setups for the official JDK (Java Development Kit) and JRE (Java Runtime Environment) released by Sun Microsystems.

Previous Java versions are also available.

Veryant Download Page

http://www.veryant.com/support/signedin/software.php

Here you can download the latest setups of Veryant's products as well as products documentation.

You must log in with your credentials in order to see the content of this page.

Veryant Knowledge Base

http://www.veryant.com/support/phpkb/

Here you can find additional information and suggestions on Veryant's products, extending the knowledge acquired from the isCOBOL documentation.

Java Docs

http://java.sun.com/javase/6/docs/api/

Here you can find the reference manual of objects provided by Java.

Java objects are available in the JVM and can be invoked by the COBOL program through OOP syntax in order to extend its functionalities.

c-tree Server Administrator's Guide

http://www.faircom.com/doc/ctserver/

Here you can find all the information about the c-tree configuration and administration.

It provides additional information that extends the knowledge acquired from the isCOBOL documentation.

ZK Direct RIA

http://www.zkoss.org/

Here you can find information about the ZK Framework used by Web Direct 2.0.

Visit this site to see what ZK can do and to be up to date to the latest news.

Eclipse Official Site

http://www.eclipse.org/

Here you can find information about the Eclipse environment where the isCOBOL IDE works.

CHARVA Official Site

http://www.pitman.co.za/projects/charva/

Here you can find information about the CHARVA Toolkit.

CHARVA is internally used by isCOBOL to handle character-based screens without using graphical components.

DBMaker Official Site

http://www.dbmaker.com/

Here you can find setups and documentation of DBMaker, the RDBMS connected by the DCI file handler.

Appendix K

Handling Different Character Sets

Introduction

As data management tools COBOL applications need to work with special characters sometimes. Not all text strings are made of standard characters whose ASCII value is less than 128. It may happen that the COBOL application needs to handle special characters such as grave letters or foreign language characters (for example chinese characters).

This chapter provides some information on the ways isCOBOL work with these characters.

National Items

National data items are the most pliant way to manage special characters.They don't need any particular configuration and work the same way on every platform whatever is the current encoding set in the environment.

These items are defined by the N character in the picture. For example, in order to define a 10 digits national string, you would write:

```
77 national-item PIC N(10).
```

If you take advantage of national data items, you don't need to care about the current encoding set in the environment. The isCOBOL Framework handles national data items by encoding data in UTF-16 Big Endian. This is a fixed length encoding where each digit needs two bytes to be stored. The data item shown above, for example, uses 20 bytes of memory.

When national items are used within a group, the group should be defined as GROUP-USAGE NATIONAL:

```
01 national-group GROUP-USAGE NATIONAL.
   03 national-item-1 PIC N(10).
   03 national-item-2 PIC N(20).
```

Note: a GROUP-USAGE NATIONAL field should contain only items whose picture is N(n) or numeric-edited.

National items can be used in the COBOL program as standard alphanumeric items. Every operation you can do on an alphanumeric item (for example: UNSTRING, INSPECT or reference an area) can be done also on national items in the same way.

This is the easiest and most reliable solution to handle special characters.

Note: national items cannot represent data written with a specific encoding (such as UTF-8 or BIG5) but just data written through national items.

The isCOBOL Encoding

If national items cannot be used, and standard alphanumeric items are used instead, special characters are handled by the Framework according to the current encoding set in the environment. To force the Framework to use a particular encoding overriding the operating system settings, you must set the `iscobol.encoding *` configuration property. For example, in order to force UTF-8 as current encoding for the COBOL application, you add the following entry to the configuration:

```
iscobol.encoding=UTF-8
```

The above setting affects standard i-o operations. The isCOBOL Framework will use the specified encoding to represent data on the video after reading and to convert data before writing.

When managing special characters using standard alphanumeric items, you must pay attention to offsets. For example, if you reference a variable by specifying offset and length between parenthesis, like:

```
MOVE var(2:) TO dest.
```

consider that 2 means the second byte and not the second digit, so, if the first character of *var* needs two or more bytes to be represented with the current encoding, the result of the COBOL operation will be a truncated string.

The Java Encoding

For i-o operations other than standard i-o statements (for example reading and writing an xml file using the XmlStream internal class) and for line sequential files managed by programs compiled with `-flsu`, the Java file.encoding setting is considered. This property is usually set in the command line. For example, to force UTF-8 as Java encoding for your program, you launch:

```
iscrun -J-Dfile.encoding=UTF-8 PROG
```

The Java file.encoding is also useful at compile time. If you have special characters in your source code and their ASCII value doesn't match with the current operating system encoding, you should tell the compiler about it. For example, in order to compile a source written on Linux using UTF-8 on a Windows system where the system encoding is not UTF-8, you launch:

```
iscc -J-Dfile.encoding=UTF-8 prog.cbl
```

Appendix L

Performance Tuning

In this book you will find suggestions to speed up the development of your COBOL application as well as its performance at run time.

Guidelines for faster compilation

Guidelines for better runtime performance

Profiling COBOL programs

Guidelines for faster compilation

The time spent by the Compiler to generate objects, dictionaries and bridge programs may affect your productivity. In this chapter we provide some guidelines to optimize the time spent in compiling programs.

General advice

Writing a file on disk has a cost, so it's important to reduce the number of files that the Compiler will generate. Review your Compiler options and configuration looking for the below items, and check if you really need them or if you can discard them:

Compiler options	
-efa -efd	Generate EFD dictionaries required by several tools and file handlers
-efc	Generate ISS dictionaries required by c-tree in order to link files to the SQL Engine
-ef	Generate error files that include the list of errors and warnings returned by the Compiler
-jj -jc	When used together, the intermediate Java source is left on disk
-ld -lf	Generate list files that include the source code and all the copybooks
Compiler configurations / directives	
iscobol.compiler.easydb=1 $SET "easydb" "1"	Generate bridge programs that allow to use COBOL I/O statements on a RDBMS

iscobol.compiler.easylinkage=1 $SET “easylinkage” “1”	Generate a bridge program that allows your COBOL program to be called from Java programs
iscobol.compiler.servicebridge=1 $SET “servicebridge” “1”	Generate bridge classes that allow your COBOL program to be called via WebService technology

EFD and ISS dictionaries as well as Database Bridge routines are generated for each FD found in the COBOL source. The same FD could be included in multiple programs, so the Compiler may regenerate the same dictionary and Database Bridge routine multiple times without reason. In order to avoid the regeneration of the same dictionary and Database Bridge routine it’s good practice to gather all your FDs in a dummy program and compile only this dummy program with the options and configurations that make the Compiler generate dictionaries and Database Bridge routines.

Suggestions for faster compilation from command-line

The isCOBOL Compiler is a Java program. Every time you launch the Compiler, a JVM must be initialized. This operation may take a couple of seconds. If this couple of seconds is spent for each program, then the compilation of a series of programs may take several seconds. For this reason it’s good practice to compile multiple programs at once, using wildcard characters on the command line. For example a command like:

```
iscc src/prog*.cbl
```

Takes less time than:

```
iscc src/prog01.cbl
iscc src/prog02.cbl
iscc src/prog03.cbl
iscc src/prog04.cbl
iscc src/prog05.cbl
iscc src/prog06.cbl
iscc src/prog07.cbl
iscc src/prog08.cbl
iscc src/prog09.cbl
iscc src/prog10.cbl
```

The display on console of the Compiler output has a little cost. If you wish to save more seconds in the compilation of multiple programs, you should consider to redirect the Compiler output to a file that you will review when the Compiler terminates, e.g.

```
iscc src/prog*.cbl 2>/tmp/iscc.out 1>&2
```

Suggestions for faster compilation in isCOBOL IDE

Put project and sources on local disk instead of network disk, if possible. If you need to share the project among multiple PCs, consider using SVN or some other type of source version control instead of putting the files on a network drive.

If sources need to stay on network disk for some reason, try to have at least the other necessary files on the client (like .list in list folder, .class in output folder). You would create a workspace on your PC, and point to the folders there when creating objects and listings, for instance. This minimizes some network traffic, while keeping the important files safe on the network.

Turn off the option to refresh project when compiling:

1. Click on *Window* in the menu bar and choose *Preferences*
2. Expand the *isCOBOL* tree and choose *Compile/Build*
3. Change the value of *Refresh Project Automatically* to 'Do Not Refresh Automatically'

Turn off automatic build:

1. Click on *Project* in the menu bar
2. Uncheck *Build Automatically*

Turn off the real time reconciling feature:

1. Click on *Window* in the menu bar and choose *Preferences*
2. Expand the *isCOBOL* tree and choose *Editor*
3. Change the *Enable reconciling* value from 'Always' to 'When opening file' or 'Never'

Turn off the option to build before launching:

1. Click on *Window* in the menu bar and choose *Preferences*
2. Expand the *Run/Debug* tree and choose *Launching*
3. Uncheck the option *Build (if required) before launching*

Increase the memory used by IDE if the machine has enough RAM. Tune the -Xms and -Xmx options set in *isIDE.ini* file. You can find the *isIDE.ini* file in the isCOBOL IDE installation folder. You'll need to experiment with the amount of memory, depending on your specific environment. Try doubling it at first.

Check if the Antivirus activity causes IDE slow performance. It's good practice to exclude the isCOBOL IDE installation folder and the location of the workspace from the list of folders checked by your Antivirus.

Guidelines for better runtime performance

Lite objects and high speed are always very appreciated in a COBOL application, but compiling and running programs with default settings does not always provide them. This chapter is intended to help you in tuning performance and optimize programs in order to obtain the best from your COBOL application.

The most important actions to be performed in order to optimize performance are:

- use *-server* Java option, if available
- avoid debug information in the programs
- avoid logging the Runtime activity

In the next pages you will find further information about performance tuning.

Compile-time optimizations

The first area on which to focus the attention when tuning performance is the compilation of the source files. Starting with small and optimized classes is an important step that will help improving performance at run time.

Latest Compiler

The isCOBOL Compiler is constantly improved so, after you upgrade your isCOBOL SDK, it's good practice to re-compile your programs using the current Compiler.

Smaller classes

Smaller classes are loaded faster that means less time is required to start the program the first time.

The size of a Java class file is influenced by additional information that are stored for debug purposes. In order to obtain the smaller class files you can choose to remove these information. The disadvantage is that error messages and exception stacks produced by the Framework at run time will be more poor. To remove debug information:

- avoid using -d and -dx compiler option

This will cause debug information to be excluded from the class. If you were using -big option along with -d and -dx because of a "too many constants" javac error, try avoiding -big as well. The “too many constants” error might not occur anymore when the class is smaller due to the lack of debug information.

- avoid using -big option if not necessary

Generally speaking the -big option should be used only when necessary. If and only if the compilation of a program fails with the "too many constants" or "code too large" Java errors, then -big should be considered. Compiling a program with -big despite it’s not necessary generates additional classes and makes the program slower at run time. Consider that the -big option can be included at the top of the source in the programs that require it, using the IMP OPTION directive.

```
       >>IMP OPTION "-big"
IDENTIFICATION DIVISION.
PROGRAM-ID. HugeProgram.
```

In this way you avoid putting -big in your compiler scripts with the risk of using this option also for programs that don’t need it.

- avoid using -g compiler option

This will cause COBOL source references to be excluded from the class.

- add -jo=-g:none to the compiler options

This will cause Java source references to be excluded from the class.

- add -ostrip to the compiler options

This will cause COBOL variables description to be excluded from the class.

However the time spent for the class loading is the sum between the time spent to read the class file from disk and the time spent to register class definitions in the metaspace. With that said, you should optimize not only the size in KB on disk, but also the amount of items and methods in the class. This kind of information can be obtained by processing the class with javap.

Smaller methods

Java performs better if methods are small. It’s better to have a lot of small methods rather than few huge methods. This rule doesn’t affect only object oriented programming; consider that each paragraph of a standard COBOL program becomes a method in the compiled Java class. If your program is made of huge paragraphs, in order to split them in multiple small methods, you might consider compiling with -sns=Statements option. Use rather low values, like 200.

Relaxed size checking on computational data-items

The -dz compiler option improves performance by relaxing size checking on computational numeric items.

The price for the best compatibility on comparisons and moves

The -cudc compiler option provides more compatibility with other COBOLs by treating numeric USAGE DISPLAY data as characters in comparisons and moves. On the other hand, programs compiled with this option are usually slower.

Proper data types

isCOBOL handles COBOL data types by creating an object for each one of them. There are two cases in which you can make isCOBOL handle data in a more optimized way: Strings management and Arithmetic operations on integers.

Strings management

Operations on alphanumeric items, such as INSPECT and UNSTRING, can be optimized by compiling the COBOL program with -b option. This option causes isCOBOL to handle the data item using a byte array instead of an object, causing the string management to be faster. However, the -b option avoids you to use national data items.

EVALUATE statements that test string literals can be optimized by compiling with -oe option. In this case the compiler translates the EVALUATE statement to a Java SWITCH statement instead of calling the EVALUATE implementation in the isCOBOL runtime.

Example:

```
iscc -b -oe string-test.cbl
```

Arithmetic operations on integers

If you need to perform additions or subtractions on integer numeric items (for example incrementing a counter), define your item as:

```
77 num INT.
```

instead of:

```
77 num PIC 9(4).
```

This will cause isCOBOL to use a native int to store the data instead of creating an object instance for it, causing the arithmetic operation to be faster.

Better run time performance

Before tuning run time performances, ensure that you're using optimized class files. For example, programs compiled in debug mode are slower than programs compiled without debug information. Read Compile-time optimizations for details.

Run time performance are influenced by the memory. Operations made in memory without swapping to disk are better performing. Java tries to gain memory by periodically performing a cleaning procedure called Garbage Collector; during this procedure performance are slowed down. The more memory you provide to Java and isCOBOL, the more operations will be made in memory and Java will not invoke the Garbage Collector if enough memory is available.

No logging

Logging the Runtime activity to file has a cost in terms of disk i/o. If the Runtime must record every operation on a disk file, it will loose time doing it. Therefore, it's strongly suggested to disable the isCOBOL logging by changing the configuration in one of the following three ways:

- avoid having *iscobol.tracelevel* set in the configuration or
- set *iscobol.tracelevel*=0 in the configuration or
- put an hash sign before *iscobol.tracelevel* in the configuration file

Tuning the JVM memory

The JVM memory is controlled mainly by four options:

-Xms	Set initial Java heap size
-Xmx	Set maximum Java heap size
-Xss	Set java thread stack size
-XX:MaxPermSize (until Java7) -XX:MaxMetaspaceSize (from Java8)	Set the metaspace size

For example, in order to specify a memory limit of 512 MB, you use:

```
iscrun -J-Xmx512m PROGRAM_NAME
```

It's suggested to run on 64 bit systems, so the memory amount for Java can be increased over 2GB, that is the maximum amount of memory per process on 32 bit systems.

In order to know what are the default values for heap size and thread stack size as well as check if the above options changed these values, you can rely on the *-XX:+PrintFlagsFinal* Java option. E.g.

```
iscrun -J-Xmx512m -J-XX:+PrintFlagsFinal PROGRAM_NAME
```

Look for these entries in the resulting console output: *InitialHeapSize*, *MaxHeapSize* and *ThreadStackSize*.

Giving Java more memory it's not always the best option. The proper amount of memory for the JVM depends on the environment. To be precise, it depends on the number of concurrent JVMs running and their work. For example, in an environment where several JVMs are running but they do a small work, it's good practice to keep the memory usage low, because having all the JVMs allocate a lot of memory may saturate the operating system memory. Known scenarios with several JVMs running are:

- a Linux server where several clients connect via Telnet or SSH and run character based programs with Charva,
- a Windows server where several clients connect via Terminal Server and run either an isCOBOL runtime or an isCOBOL Client,
- a web server with webClient, as webClient starts a JVM for every session.

Vice versa, in an environment where only one JVM is running and it does a huge work, it's OK to keep the memory usage high. The typical scenario of this kind is a server where the isCOBOL Server is running and several isCOBOL Clients connect to it from other computers in the network.

Reduce the number of running JVMs, if possible

Some applications manage tasks by instantiating dedicated runtime processes. This approach is good when the runtime is light, that is not the case of the isCOBOL runtime. Being written in Java, the isCOBOL runtime requires more time to start. For this reason, you should consider reducing the number of isCOBOL runtimes that are started. If your application instantiates runtimes to run programs, e.g.

```
call "system" using "iscrun PROG1".
call "system" using "iscrun PROG2".
call "system" using "iscrun PROG3".
call "system" using "iscrun PROG4".
```

Consider to use CALL RUN statements instead, e.g.

```
call run "PROG1".
call run "PROG2".
call run "PROG3".
call run "PROG4".
```

Note - the asynchrony of CALL RUN is controlled by the iscobol.call_run.sync (boolean) configuration property.

In this way the whole process runs in the same JVM and you save the time spent in the instantiation of new JVMs.

More memory to isCOBOL

isCOBOL can allocate more memory for arrays and sort works.

To provide more memory for array handling, set the property iscobol.array_cache in the isCOBOL configuration. Veryant recommends to set it to the maximum number of OCCURS in the application. If memory consumption is slowing performance then set to a lower value and in this case, a prime number is recommended to reduce collisions by making a more even hash distribution.

The default is 101. There is no maximum value, but higher values consume more memory.

To provide more memory for sorting procedures, set the property iscobol.sort.memsize in the isCOBOL configuration. Veryant recommends to increase the sort memory to a value of 32 MB:

```
iscobol.sort.memsize=33554432
```

The above setting improves performance of the SORT verb, the ISSORT (External Sort) utility and the C$SORT routine.

Class loading optimizations

Each time a new class is loaded into the JVM, a certain amount of time is spent to perform the necessary operations. While working with isCOBOL a new class is loaded each time a COBOL program is called the first time and each time a COBOL program uses a class (a runtime feature or an external Java class) that has not been loaded yet.

The class loading process can be optimized in two ways:

1. Using the Java option
    - -Xverify:none

With this option you instruct the JVM to not verify the correctness of the bytecode. Avoiding such verification, the JVM loads the classes faster.

2. Setting the CLASSPATH instead of iscobol.code_prefix to tell where your programs are stored.

When the code_prefix is set, the isCOBOL runtime performs some operations to find the class on disk and check if the class needs to be reloaded or not. These operations have a cost. Letting Java load your programs as standard Java classes from the CLASSPATH makes the classes load faster. The disadvantage is that the classes descriptions are kept in memory so you can't update a program by just replacing the class file while the COBOL application is running, you also need to close and restart the JVM (in thin client you need to restart the isCOBOL Server).

Instead of setting the CLASSPATH you may consider setting *iscobol.code_prefix.reload=0* along with the code_prefix. In this way it is still possible to reload a updated class from disk without restarting the JVM, but the process is not automatic. It's your duty to unload the program class from the JVM memory calling the C$UNLOAD library routine. After it, the class will be reloaded from disc at the next CALL. This approach makes the runtime access the disc only when actually necessary and not every time a CALL is performed.

3. store your classes in jar libraries instead of folders. See JAR Files and the Jar Utility for details.
4. preload the jar libraries by calling the C$PRELOAD routine.

Disable check for numeric content

Making the runtime check for the content of numeric data items may slow down performance a little. For this reason, the property iscobol.check.numeric_content * should be enabled only during development and test, and then disabled for production.

Additional JVM optimizations

Where available, you should think about using the following Java options:

- -server

This option is usually available only with the JDK (the JRE doesn't provide it). It causes Java to work in server mode instead of the default client mode. The server mode causes background operations (such as file i/o) to be better performing.

- -Xnoclassgc

This option disables the garbage collection. By default the JVM unloads a class from memory when there are no live instances of that class left, but this can degrade performance. Turning off class garbage collection eliminates the overhead of loading and unloading the same class multiple times. If a class is no longer needed, the space that it occupies on the heap is normally used for the creation of new objects. However, if you have an application that handles requests by creating a new instance of a class and if requests for that application come in at random times, it is possible that when the previous requester is finished, the normal class garbage collection will clean up this class by freeing the heap space it occupied, only to have to re-instantiate the class when the next request comes along. In this situation you might want to use this option to disable the garbage collection of classes.

With the JVM provided by IBM, the following option can also be used:

- -Xshareclasses

This option make classes shared between JVM processes. This feature improves performance since class byte code needs to be loaded only once. Class byte code is loaded into a shared cache. This cache is then accessed by multiple JVMs to run the class bytecode. Currently, in Windows and UNIX, a cache is implemented as a

memory mapped file. Because the byte code is cached in a file, it reduces demand for memory. Every cache has a name. A JVM must attach itself to a cache to share classes from it. This is done using the -Xshareclasses argument. For example:

```
iscrun -J-Xshareclasses:name=myCache PROGRAM
```

When this command is run for the first time, a cache called myCache is created. Classes (core Java classes and application classes) are cached there. If another JVM is launched with the same command line, classes will be shared between processes. A cache lives beyond the life time of a JVM. It can be explicitly deleted. It is also deleted when the OS is rebooted.

Latest Java version

Oracle implements performance improvements in every new version of Java. For this reason it's suggested to use the latest stable Java version available.

User Interface Optimization

The isCOBOL architecture separates the UI from the back end elaboration using a client/server logic. The UI is managed by the client part while the back end is managed by the server part. Every time the user interaction causes some COBOL code to be executed (e.g. the user leaves a field that has an After Procedure) and everytime the program must update the video or accept the user input, client/server traffic is generated.

When the client part and the server part are executed by two different JVM processes (e.g. in thin client) then the performance may be affected by the client/server communication and the below suggestions beneficial effects will be more evident.

The main objective is to reduce the number of embedded and event procedures handled by the program so that the user interface must not send too much information to the server part while the user is interacting with it. For example, if you toke advantage of Before and After procedures to color the current Entry-Field while the user navigates on the screen, then you may think to instruct the runtime by setting iscobol.gui.curr_bcolor and iscobol.gui.curr_fcolor properties in the configuration instead of coding embedded procedures.

UI changes bufferization

isCOBOL includes an internal optimizer that gathers data of all DISPLAY and MODIFY (if the GIVING clause is omitted) statements and sends this data to the client

- when iscobol.gui.cstimeout * expires
- when iscobol.gui.csmaxbuffersize * is reached
- where either WFLUSH-REFRESH or WFLUSH-ALLOW W$FLUSH op-codes are called
- when an ACCEPT of user input is performed
- when the CBL_READ_SCR_CHATTRS routine is called, or the equivalent statement ACCEPT dest-item FROM SCREEN is performed
- when an INQUIRE is performed, unless WFLUSH-INHIBIT W$FLUSH op-code was called. Note that not all INQUIREs cause network traffic, it depends if the Framework needs to communicate with the UI in order to retrieve the inquired attribute.
- when a MODIFY with GIVING clause is performed, except for TREE-VIEW's ITEM-TO-ADD
- when a MODIFY of VISIBLE or ENABLED properties is performed on a window handle
- when a SET INPUT WINDOW or a SET I-O WINDOW is performed
- when a print file or a file whose class is "com.iscobol.io.RemoteRelative" is open
- when a CALL CLIENT is performed

- when events are generated client-side (it may happen in a multi-thread environment where the user interacts with the screen while another thread is performing MODIFY or INQUIRE that are being gathered by the optimizer)

Event Lists

isCOBOL also offers the ability to discard some events so that when they happen the client doesn't communicate with the server. This feature is obtained by setting the EVENT-LIST and EXCLUDE-EVENT-LIST properties. See Controls Reference for details.

The drag events of Grid control can be disabled also through the configuration property iscobol.gui.grid.no_cell_drag (boolean) * or the style No-Cell-Drag.

Programming Tips

Some tips to write programs optimized for the client/server environment:

- use MODIFY instead of DISPLAY to update the screen. Modify acts on a single property, while DISPLAY redraws the whole control (or screen)
- if possible, avoid using the GIVING clause with MODIFY unless you're using Item-To-Add in TREE-VIEW
- use absolute values for LINE, COLUMN, LINES and SIZE properties
- rely on the MASS-UPDATE feature when you need to load a Combo-Box, a Grid, a List-Box or a Tree-View
- setting iscobol.gui.curr_bcolor and iscobol.gui.curr_fcolor in the configuration is preferable than changing the EntryField colors in its embedded procedures.
- setting Row-Cursor-Color (or Row-Cursor-Background-Color and Row-Cursor-Foreground-Color) in the Screen Section is preferable than changing the Region-Color (or Region-Background-Color and Region-Foreground-Color) property inside Grid event procedures.
- rely on the Search-Options and Search-Text properties instead of scanning the Grid content with a loop of INQUIRE of the CELL-DATA property when you're looking for a text in the Grid.
- rely on ACTION-COPY and ACTION-EXPORT instead of scanning the Grid content with a loop of INQUIRE of the CELL-DATA property if you need to implement the copy of the Grid content to a Excel spreadsheet or to the clipboard.
- huge elaboration cycles that periodically display the progress can be made faster by disabling the update of the UI by calling WFLUSH-DISABLE-UI before the elaboration and then calling WFLUSH-ENABLE-UI when the elaboration is completed.
- if a lot of INQUIREs must be performed (e.g. if you have a cycle that checks the content of each row in a Grid), consider buffering them through WFLUSH-INHIBIT and WFLUSH-ALLOW.
- attach embedded procedures only to those controls where you actually need to do something when the focus is gained or lost and avoid defining embedded procedures on Screen group items as they would be executed for every control in the group.
- Use Format-String on ENTRY-FIELD only if you actually need it and avoid using PIC if the picture doesn't include any kind of editing (e.g. there's no point in having PIC X(10) among ENTRY-FIELD's properties). FORMAT-STRING and PIC generate client/server traffic.
- Delay the NTF-CHANGED event by setting iscobol.gui.entryfield.notify_change_delay * in the configuration (if you wish to affect all the entry-fields) or by modifying the property Notify-Change-Delay (if you wish to affect specific entry-fields). If the user types quickly, the runtime would generate too many NTF-CHANGED events. With this delay you can reduce the number of events generated.

Thin Client optimization

Client/Server performance in Thin Client environment are influenced by the network traffic generated by the communication between the user interface that runs on the client and the back-end part that runs on the server. Refer to User Interface Optimization for suggestions about how to optimize this communication.

In addition, when working in a thin client environment, the following suggestions might be useful.

- when you need to create or read a rather huge sequential file on the client pc, it's better to have a copy of it on the server and work on that copy rather than accessing the file directly on the client with the class "com.iscobol.io.RemoteRelative". Refer to C$COPY documentation for information about how to copy files from server to client and vice versa.
- when you need to create a PDF print file on the client pc, it's better to print the PDF locally on the server and then copy it to the client. Refer to C$COPY documentation for information about how to copy files from server to client.
- if you have a lot of icons and pictures, you may consider to copy them to the client machine and load them from there using WBITMAP-LOAD-FROM-CLIENT instead of WBITMAP-LOAD. The copy could be done during the installation or your application or at the very first runtime session. Loading bitmaps from the client machine reduces the client/server traffic and, as a consequence, improves performance.

isCOBOL offers the ability to compress data that transits on the TCP/IP. Set iscobol.gui.cscompress * properly to take advantage of this feature.

Data Access

This chapter provides suggestions and guidelines for better performance during data access.

Choosing the proper file handler

File connectors like vfc (Vision File Connector) and mfc (Micro Focus File Connector) are useful to test isCOBOL without converting your original data set as well as running isCOBOL along with either ACUCOBOL-GT or Micro Focus working on the same data set. However, file connectors don't provide good performance. If you're looking for good performance on data access, then you should consider to convert your original data set to either c-tree RTG or JIsam.

c-tree RTG should be used instead of JIsam when the i-o activity and the concurrency on indexed files are huge.

Installing the c-tree RTG Server on the server machine where data files are stored is more safe and faster than accessing data files via network drive or UNC paths, regardless of the amount of i-o activity and concurrency on these files.

c-tree RTG is rather fast with the default configuration, but it can be further optimized with some tuning. See c-tree RTG for some advice about how to optimize c-tree RTG.

General advice

The following suggestions are applicable to data access, regardless of the file handler in use:

- Sequential files can be created in memory. If you need a temporary sequential file there is no need to create it on disk. You can obtain better performance by creating the file in memory. The syntax to define a memory file is:

```
INPUT-OUTPUT SECTION.
FILE-CONTROL.
select my-file assign to address memory-area
  ....
WORKING-STORAGE SECTION.
77 memory-area pic x any length.
```

- In thin client and file server environments, avoid setting iscobol.file.lock_manager *, if possible, especially when there are a lot of clients connected simultaneously (e.g. more than 100 clients).
- The files that are used very often should reside on a solid-state drive (SSD) or a RAM disk.

c-tree RTG

The following suggestions are applicable to c-treeRTG:

General

- Avoid accessing files via network drives or UNC paths. Start the c-tree Server on the machine where files are stored instead.

Server-side

- Use the latest c-tree version available.
- Increase the values of DAT_MEMORY and IDX_MEMORY in ctsrvr.cfg.
- Enable the SHAREMEM protocol in ctsrvr.cfg, if not yet enabled.
- In thin client it's better to call C$LOCKPID instead of using the BaseLockManager if you need to know who's locking a record.
- In thin client you can run the c-tree server in the same process as isCOBOL Server. Set iscobol.ctree.bound_server (boolean) to true in the isCOBOL Server configuration. The c-tree server will start as part of the isCOBOL Server process at the first OPEN of an indexed file performed by a Client. Working in this mode, the performance is better than having c-tree server running as a separate process. It's still possible to connect to the c-tree server using external tools, utilities and runtimes.
- If your files are under transaction with logging, consider setting DELAYED_DURABILITY in ctsrvr.cfg.

Client-side

- Take advantage of prefetch and batchaddition where applicable. For details:

Configuration via CTREE_CONF	Configuration via iscobol.properties
<prefetch>	iscobol.file.index.prefetch (boolean) iscobol.file.index.prefetch.allowwriters (boolean) iscobol.file.index.prefetch.records

Configuration via CTREE_CONF	Configuration via iscobol.properties
<batchaddition>	iscobol.file.index.batchaddition (boolean) iscobol.file.index.batchaddition.records

- Avoid the use of a file connector, if possible. Use ctreej instead.
- For temporary files, memory files should be used. For details:

Configuration via CTREE_CONF	Configuration via iscobol.properties
<memoryfile>	iscobol.file.index.memoryfile (boolean)

- Disable c-tree activity logging, so avoid the following settings:

Configuration via CTREE_CONF	Configuration via iscobol.properties
<log>	iscobol.file.index.log.debug.batchaddition (boolean) iscobol.file.index.log.debug.prefetch (boolean) iscobol.file.index.log.error (boolean) iscobol.file.index.log.error.atend (boolean) iscobol.file.index.log.error.notfound (boolean) iscobol.file.index.log.file iscobol.file.index.log.info (boolean) iscobol.file.index.log.profile (boolean)

- Enabling the ctfixed option forces creating fixed-length record data files as fixed-length c-tree files. If you enable ctfixed, you may see a small performance enhancement as there is additional overhead in processing variable-length record data files.

Configuration via CTREE_CONF	Configuration via iscobol.properties
<ctfixed>	iscobol.file.index.fixed_length (boolean)

- If the COBOL application performs several OPEN operations on the same files, consider to add the files to a pool:

Configuration via CTREE_CONF	Configuration via iscobol.properties
<filepool>	iscobol.file.index.filepool

- c-treeRTG allows data compression so that less disk space is used, which results in performance enhancements while reading and writing to disk. Enabling data compression using the RLE algorithm provides the advantages of compressed data with a very small impact on CPU usage. Since most applications are saturated at the I/O level, the slight increase in CPU usage but less overhead on the I/O channel typically results in performance gains.

Configuration via CTREE_CONF	Configuration via iscobol.properties
<datacompress>	iscobol.file.index.datacompress (boolean)

- The optimisticadd option enables adding keys before the data during WRITE operations. When optimisticadd is disabled, c-tree attempts to add unique keys before adding the data record. This eliminates the overhead of deleting a data record when the unique key check fails, speeding up the insert process. Disable optimisticadd if the COBOL application frequently performs WRITE operations conflicting with existing records.

Configuration via CTREE_CONF	Configuration via iscobol.properties
<optimisticadd>	iscobol.file.index.optimisticadd (boolean)

Database Bridge

The follwing tips are applicable when working on databases with isCOBOL Database Bridge.

- Handling locks through the Application Server or through the File Server by setting iscobol.file.lock_manager * to "com.iscobol.as.locking.InternalLockManager" is faster than having locks handled by the database engine. As side effect, the concurrency is managed only between clients connected to the same isCOBOL Server, so this tip is not applicable if you're working on the database with other tools in addition to the COBOL application. In addition, using a lock manager is not suggested when there are a lot of clients connected simultaneously (e.g. more than 100 clients) as it decreases performance in this case.
- EDBI routines check for table existence with a SQL query on database catalogues every time a OPEN is performed. If your application includes several OPEN statements and you're sure that tables exist on the database, you can avoid this check by generating EDBI routines with the -nocheck option.
- When you generate subroutines for Oracle using edbiis, use the -oho option in order to obtain better performance for START on alternate keys on huge tables.
- When you generate subroutines for MySQL using edbiis, , use either -dmld or -dmlu in order to optimize the use of SQL cursors for read operations. Use the -mh option in order to obtain better performance for START on alternate keys on huge tables.
- When you generate subroutines for PostgreSQL using edbiis, use either -dpld or -dplu in order to optimize the use of SQL cursors for read operations.
- You can reduce the number of processed records by setting The EDBI-WHERE-CONSTRAINT external variable.
- If the key you use for the START has a lot of segments, but you need to use only the first few of them, you can rely on the SIZE clause to limit the size of the key to the size of the useful segments only. This will reduce the number of SELECT queries generated by the Database Bridge to simulate the START. **Note -** when using EDBIIS to generate EDBI routines, the -sl option must be used in order to activate the support for START WITH SIZE.

- The performance of the START operation can be improved by setting iscobol.easydb.limit_dropdown.
- For multi-record files, where you have a table for each record definition, you can tell the START statement to use one specific table depending on the record type instead of using all the tables related to the multi-record file. To achieve it, set iscobol.easydb.start_on_specific_table (boolean) to true in the configuration and ensure that the record type field is properly set before the START statement.
- ESQL (SQL logic) is preferable to the COBOL code generated by isCOBOL Database Bridge (COBOL logic) when you need to perform
    - huge update operations (one single complex query is usually faster than multiple small queries generated by multiple REWRITE statements)
    - complex searches (filters in the WHERE condition of SELECT are preferable to reading all records and performing checks with COBOL statements).
    - testing of few fields (specifying the fields you wish to test in SELECT is preferable to reading the whole record including the fields you're not interested in)
- Database Bridge performance is affected also by the JDBC configuration. See Driver configuration for more advice.

JDBC/ESQL

The following tips are applicable to JDBC so they affect both Database Bridge and ESQL.

Driver configuration

- If you're working on Microsoft SQL Server with either the official driver (sqljdbc.jar) or JTDS, then add the JDBC option *sendStringParametersAsUnicode=false* to your connection URL. Example:

```
iscobol.jdbc.url=jdbc:sqlserver://
   localhost:1433\SQLEXPRESS;databasename=master;sendStringParametersAsUnicode=false
```

SQL syntax

- If you're working on Oracle, consider using Optimizer Hints in your queries in order to direct the optimizer to choose the best query execution plan.
- If you're working on either MySQL or PostgreSQL, consider using the LIMIT clause in your queries in order to generate lighter cursors and save memory.

Printing

- The first time isCOBOL inquires the system printers (e.g. the first time you call WIN$PRINTER to enumerate printers or select a printer) it takes more time because it loads printer settings in memory. From the second time, it reads from the memory and it's faster. In order to hide the first

slow printer interaction to the user, you might consider calling the WINPRINT-GET-NO-PRINTERS op-code in a separate thread at the very beginning or your application while the user is inputting login data or exploring the main menu looking for the subprogram to launch.

- By default isCOBOL stores the print job in a temporary file. When the print file is closed, it reads from the temporary file and sends data to the spooler. Since reading/writing memory is faster than reading/writing disk files, you can tell isCOBOL to save temporary information in memory instead of using a temporary file. To achieve it, set the iscobol.print.memory (boolean) configuration entry to true.

Profiling COBOL programs

The isCOBOL Framework provides the ability to profile COBOL programs in order to identify which paragraphs or sections are used most of the CPU time during a elaboration.

Note that this feature is not supported for programs that were compiled with -big or -sysc options and for CLASS-ID programs.

Before profiling your programs, ensure that you already followed the suggestions from Compile-time optimizations and Better run time performance, e.g. programs are not compiled in debug mode and there is no logging of the runtime activity.

The following jar libraries, installed along with isCOBOL, are necessary in the Classpath:

- isprofiler.jar
- javassist.jar

In order to enable the profiling, add the following option to the runtime command-line:

```
-J-javaagent:[path/to/]isprofiler.jar[=list-of-programs]
```

Where:

- *[path/to/]isprofiler.jar* is the name and path of the isprofiler library. If no path is specified, the library is searched in the current directory.
- *list-of-programs* is a optional parameter. It is a comma-separated list of regular expressions defining the programs to profile. The default value is "[A-Z_][A-Z0-9_]*" (all the classes whose name is fully uppercase). This is useful to discard programs with user interface and focus the profiling on programs that perform back-end elaboration.

Command example:

```
iscrun -J-javaagent:/opt/iscobol/lib/isprofiler.jar IO_PERFORMANCE
```

The above command starts the IO_PERFORMANCE sample installed with isCOBOL (it is stored in the folder sample/io-performance of the isCOBOL SDK).

When the runtime session terminates, a similar output is generated:

```
elapsed   time = 2.40
evaluated time = 2.19
overhead1=25; overhead2=35
-------+-------+---------+-----------+-----------------
self % |accum %| seconds |   count   |program:paragraph
-------+-------+---------+-----------+-----------------
 29.38%| 29.38%|     0.64|          1|IO_INDEXED:DELETE_FILE1_TEST
 14.98%| 44.37%|     0.32|          1|IO_INDEXED:UPDATE_FILE1_TEST
 12.75%| 57.12%|     0.27|          1|IO_INDEXED:LOAD_FILE1_TEST
  6.91%| 64.03%|     0.15|          1|IO_SEQUENTIAL:LOAD_FILE1_TEST
  6.33%| 70.37%|     0.13|          1|IO_INDEXED:READ_FILE1_TEST
  6.23%| 76.60%|     0.13|          1|IO_RELATIVE:UPDATE_FILE1_TEST
  5.29%| 81.89%|     0.11|          1|IO_RELATIVE:DELETE_FILE1_TEST
  4.17%| 86.06%|     0.09|          1|IO_SEQUENTIAL:READ_FILE1_TEST
  3.79%| 89.86%|     0.08|          1|IO_RELATIVE:LOAD_FILE1_TEST
  3.12%| 92.99%|     0.06|          1|IO_PERFORMANCE:MAIN_LOGIC
  2.96%| 95.95%|     0.06|          1|IO_RELATIVE:READ_FILE1_TEST
  1.97%| 97.92%|     0.04|          1|IO_INDEXED:MAIN_LOGIC
  0.87%| 98.80%|     0.01|          4|IO_RELATIVE:START_TIMER
  0.53%| 99.34%|     0.01|          4|IO_INDEXED:START_TIMER
  0.49%| 99.83%|     0.01|          2|IO_SEQUENTIAL:START_TIMER
  0.05%| 99.89%|     0.00|          1|IO_SEQUENTIAL:MAIN_LOGIC
  0.05%| 99.95%|     0.00|          1|IO_RELATIVE:MAIN_LOGIC
  0.03%| 99.98%|     0.00|          4|IO_INDEXED:STOP_TIMER
  0.00%| 99.99%|     0.00|          4|IO_RELATIVE:STOP_TIMER
  0.00%|100.00%|     0.00|          2|IO_SEQUENTIAL:STOP_TIMER
```

Note - The snippet above shows the profiling result of the IO_PERFORMANCE sample program using Jlsam as file handler on a recent PC.

The output is stored in a file named *iscobol.hprof.txt*, generated in the working directory.

The information in the file header has the following meaning:

- *elapsed time* is the real time passed between the profiler startup and the report generation.
- *evaluated time* is time spent executing COBOL paragraphs.
- *overhead1* is the estimated overhead in nanoseconds added by the profiler for each paragraph not containing PERFORM/CALL.
- *overhead2* is the further estimated overhead in nanoseconds for each PERFORM/CALL.

A row for each measured block of code (paragraph of a program) is provided.

Columns have the following meaning:

- *self %* is the ratio between the time spent by the block of code and the *evaluated time*.
- *accum %* is the sum between the current *self%* and the previous ones.
- *seconds* is the number of seconds spent by the block of code.
- *count* is the number of times the block of code was executed.
- *program:paragraph* is the name of the block of code.

The profile of a program execution is obtained by measuring the time spent in individual paragraph (excluding the time spent in PERFORM/CALL) and by counting the number of times each paragraph is called.

The profiler adds an overhead that is roughly the same for each paragraph. This overhead is evaluated before the profiling and it is subtracted from the results. However the estimated overhead and the actual overhead can differ from time to time due to the machine status (multitasking, JIT compiler, etc).

If a paragraph has very few statements and it is executed many more times that the other paragraphs, the difference may be relevant and may affect the results accordingly.

The profile will include a row for each instance. It means that if a program is called, cancelled then called again or if a program is called in thread, more instances of the same program will be profiled and it will appear multiple time in the profiler output.

Good practice for an accurate profiling

It's good practice to profile only back-end programs, if possible. Profiling an interactive program may produce an unreliable report as the time spent by the user while interacting with the program is taken into account as well by the isCOBOL profiler. If the COBOL application consists of a set of programs that manage the UI and a set of programs that perform elaborations, then you may consider to profile only the second set of programs.

This may not always be an option, in which case try to be as quick as possible while inputing data to the program.

In order to provide an example, we can use the isCOBOL Demo (isControlSet) installed with isCOBOL. When you click on the "Print" button, a sub program (PRINTPROG) is called to manage the print job. Assuming that we wish to profile the activity of PRINTPROG, we can start the isCOBOL Demo as follows:

```
iscrun -J-javaagent:/opt/iscobol/lib/isprofiler.jar=PRINTPROG ISCONTROLSET
```

When the runtime session terminates, a similar output is generated:

```
elapsed   time = 4.58
evaluated time = 3.45
overhead1=27; overhead2=42
-------+-------+---------+-----------+-----------------
self % |accum %| seconds |   count   |program:paragraph
-------+-------+---------+-----------+-----------------
 70.97%| 70.97%|     2.45|          1|PRINTPROG:MAIN
 20.03%| 91.00%|     0.69|          1|PRINTPROG:SET_ATTRIBUTE
  4.89%| 95.89%|     0.16|          1|PRINTPROG:LOAD_FONT
  2.26%| 98.16%|     0.07|          1|PRINTPROG:ST_ADDRESS
  1.33%| 99.50%|     0.04|          1|PRINTPROG:LOAD_BMP
  0.13%| 99.63%|     0.00|          1|PRINTPROG:ST_UNDER_LOGO
  0.12%| 99.75%|     0.00|         18|PRINTPROG:RESET_COLUMN
  0.10%| 99.85%|     0.00|          3|PRINTPROG:LOAD_LARGE_FONT2
  0.05%| 99.91%|     0.00|          1|PRINTPROG:PRINT_F_PIC
  0.04%| 99.96%|     0.00|        230|PRINTPROG:TEST_GIVING
  0.02%| 99.98%|     0.00|          1|PRINTPROG:SMALL_FONT2
  0.01%| 99.99%|     0.00|          1|PRINTPROG:LOAD_LARGE_FONT3
  0.00%|100.00%|     0.00|          1|PRINTPROG:DESTROY_HANDLE
```

There's no trace of ISCONTROLSET in the profiler output. Only PRINTPROG has been profiled.

Thin client

In a thin client environment it is possible to profile the application server (isCOBOL Server) activity by starting the server process with the same *-javaagent* option used for the runtime. E.g.:

```
iscserver -J-javaagent:/opt/iscobol/lib/isprofiler.jar
```

or

```
iscserver -J-javaagent:/opt/iscobol/lib/isprofiler.jar=PRINTPROG
```

The profiling output is shown when the whole application server is terminated and it includes the profiling of all clients activities mixed together, therefore, if you need to profile some programs in a thin client environment, you should use a dedicated application server with only one client connected.

Appendix M

Troubleshooting

This guide will help you in troubleshooting problems that may occur while working with isCOBOL.

Licensing and Configuration issues

It's very important to keep in mind that the isCOBOL configuration is built up by merging multiple files and settings. A configuration property is searched in the following list of places:

- the configuration file specified on the command line by -Discobol.conf.only=

or, if the above option is not used

1. the system environment variables
2. the iscobol.properties file in the /etc directory
3. the iscobol.properties file in the user home directory
4. the iscobol.properties files found in the directory and jar libraries listed in the Classpath
5. the configuration file specified on the command line by -Discobol.conf=
6. the iscobol.properties file in the folder specified by the ISCOBOL environment variable
7. the single properties specified on the command line by -D*propertyname*=

If a property appears in more than one of the above places, the last occurrence is considered.

When a "missing license" error occurs, you need to search for "iscobol.license.2016" in all of the above places and ensure that the value is a valid license code. Unfortunately there's no way to retrieve license information when the "missing license" error occurs, so, if you're afraid that your license code is invalid or expired, then you should send it to Veryant technicians for analysis.

To monitor the active configuration of the COBOL application you can take advantage of the isCOBOL logging feature. This feature is activated by setting the `iscobol.tracelevel` property to a value different than zero and optionally by setting `iscobol.logfile` to the name of the log file you wish to create.

If you're looking for the list of settings read by configuration and subsequent changes made by the programs, it's enough to set the trace level to the value of 1.

Compiler Issues

The compilation process with isCOBOL is divided in two steps:

1. the cobol source is parsed and, if no error occurs, an intermediate java source is generated
2. the java source is compiled by the JDK compiler and the class file of the COBOL program is created

The most of compiler errors appears in the first step. These errors may be caused by misspelled keywords, wrong usage of features or missing source files. Consult Compiler Errors for a detailed description of the most common compiler errors.

If an error occurs during the second step, it means that the isCOBOL Compiler generated an invalid java source. This kind of problem should never exist. If you reproduce it, then you should report it to Veryant technicians providing all necessary items and information that allow to reproduce the error. The Compiler will be patched to handle the problem in the first step instead of producing an invalid java source in the future.

Runtime Issues

Because the isCOBOL Compiler produces pure Java bytecode objects, mapping COBOL programs to Java classes, engineers can use any monitoring, logging, debugging and diagnostic tool that is available to the Java community. This includes everything discussed in Troubleshooting Guide for Java(TM) SE with HotSpot (TM) VM at https://docs.oracle.com/javase/8/docs/technotes/guides/troubleshoot/index.html.

Here are just some of the tools available: HPROF profiler, JConsole, jdb, jhat, jinfo, jmap, jps, jrunscript, jstack, jstat, jstatd, visualgc, and the recently released JVisualVM.

isCOBOL Evolve also provides the following COBOL-oriented tools and features to help diagnose and resolve problems quickly:

- isCOBOL IDE
- isCOBOL Debugger
- isCOBOL Runtime Framework trace facility
- isCOBOL Compiler option to generate SMAP information for JSR 45 debugging
- isCOBOL Abend Diagnostic Snapshot (ADS)
- Alt+Pause keyboard combination

isCOBOL IDE

Built on Eclipse, the isCOBOL IDE features a COBOL code editor, integrated debugger with remote debugging capability, the ability to debug COBOL and Java together and all of the other features available to the Eclipse community.

isCOBOL Debugger

The isCOBOL graphical source-level debugger is written 100% in Java and is completely portable. The isCOBOL Debugger provides remote debugging capability with COBOL programs deployed in server and transaction processing environments. This keeps your application source code secure by maintaining it on the development machine, while debugging a program running on a remote test or production machine.

isCOBOL Runtime Framework trace facility

Supports different trace levels to allow you to follow the flow of execution with control over the level of detail that you see in the trace log.

isCOBOL Compiler option to generate SMAP information for JSR 45 debugging

The isCOBOL Compiler is compliant with the JSR-45 specification (http://jcp.org/en/jsr/detail?id=45)

Including SMAP information in the COBOL object files allows you to debug COBOL programs using JSR-45 compliant debuggers such as Eclipse and the NetBeans IDE.

isCOBOL Abend Diagnostic Snapshot (ADS)

The isCOBOL Runtime Framework can be configured to produce a detailed report, the Abend Diagnostic Snapshot (ADS), that describes the state of an application at the moment that an abnormal termination occurs. This report can be used to identify the cause of a problem after the fact.

As a matter of policy many enterprise IT organizations do not allow debugging on production machines. In these cases it is necessary to maintain a test environment that is carefully replicated from the production environment. It is critical to have one or more diagnostic capture facilities on the production machine that can provide enough information when a problem occurs to allow replication of that problem in the test environment. The isCOBOL ADS is one of the most important of these facilities because it provides all of the information available to the isCOBOL Runtime Framework at the point where an error occurs.

Together with the TPE Transaction Dump File and other available trace and dump files, the isCOBOL ADS can be used to determine the cause of a problem and as an aid in creating a solution. The ADS is useful even in situations where it is possible to attach a debugger. In dynamic online transaction processing systems such as CICS where there is a lot of volume at the time an ABEND occurs, there is often no other way to diagnose or debug a problem. In addition, some problems occur sporadically at a live site and cannot be reproduced at will so there is no opportunity to attach a debugger.

The ADS report contains the following information:

- The name of the exception that occurred along with the Java package and class that raised the exception
- The reason that the exception occurred as reported by the class that raised the exception
- The numeric code associated with the exception
- The name of the COBOL program, name of the paragraph, name of the COBOL source file and COBOL source line number of the code that was executing when the exception occurred
- Information about the particular COBOL operation that was executing when the exception occurred
- The COBOL call stack at the time the exception occurred, including the names of all COBOL programs in the stack, the names of the associated COBOL source files, and the line numbers in those source files of the particular CALL statement
- The date and time when the exception occurred
- The version of the isCOBOL compiler used to compile the COBOL program
- The command line arguments used to start the COBOL program
- The version of the isCOBOL Runtime Framework in use
- The version of the Java Runtime Environment in use
- The currently loaded COBOL programs
- A dump of COBOL working-storage memory, including each data-item name, offset, length, value and hexadecimal byte dump of the data-item contents

The report contains 3 major sections:

- General information about the program, such as the command-line parameters, the reason for the shutdown, and the line number where the shutdown occurred
- A call stack summary of each thread Detailed information about each program, including all the data values

When an exception arises in a isCOBOL program, a stack trace is generated; before isCOBOL 2008, the Java stack was displayed, but from that version a more comfortable format for COBOL users has been adopted. This new format may include the number of COBOL source lines if the program is compiled using the -g option (note that the informations are inserted directly in the .class file, so the -jj option is incompatible with -g).

To get the stack in the old format, the following boolean property must be set:

```
iscobol.exception.java=1
```

The stack trace can be enriched with important informations about the state of the program at the moment the exception arose: this further informations are called "Abend Diagnostic Snapshot" (ADS) and to get them the following boolean property must be set:

```
iscobol.exception.dump=1
```

The output of the stack trace (and, in the case, of the ADS) is by default shown in a message box, but this behavior can be changed setting the property iscobol.exception.message in the following way:

iscobol.exception.message=0	Exception is shown in a message box (default)
iscobol.exception.message=1	Exception is shown on standard output
iscobol.exception.message=2	Exception is shown on standard error
iscobol.exception.message=3	Exception is written in a file whose name is in the form <prog_name><number>.ads.log

In the last case, the file is created in the current directory; this behavior can be modified setting the property iscobol.exception.prefix that allows to change the destination directory and/or the initial part of the ads file name. For example setting

```
iscobol.exception.prefix=/tmp/
```

will create the ads files under the /tmp directory, while setting:

```
iscobol.exception.prefix=xx
```

will create the ads files under the current directory and their names will start with 'xx'.

This new features allow you to make a report to analyze the cause of an abnormal program shutdown. There are 3 new properties:

iscobol.exception.dump	boolean, enables the dump

iscobol.exception.message=3	creates a file named <progname>xxxx.ads.log with the exception message
iscobol.exception.prefix	add a prefix (possibly containing a directory) to the dump file

This is an example of dump file:

```
java.lang.ArrayIndexOutOfBoundsException caught! (1)
  in program OVERRUN, paragraph MAIN_LOGIC
     program MAINPROG, paragraph MAIN_LOGIC (Top of stack)
   (com.iscobol.rts.Factory.myFill(Unknown Source))
   (com.iscobol.types.CobolVar.setSpace(Unknown Source)) === Dump created: Thu Sep 25
14:10:41 CEST
2008 === Program: OVERRUN, compiled: 440, requires: 426 === Command line arguments:
=== isCOBOL Version: release 2008.2.0 build#440-20080919alpha === Java version: 1.6.0
Sun Microsystems Inc.
=== Current loaded programs ===
 OVERRUN
 MAINPROG
=== Memory dump ===
->01 RETURN-CODE,offset=0,length=8, value=0
00 00 00 00 00 00 00 00                          "........"

->01 TRANSACTION-STATUS,offset=0,length=2
20 20                                            "  "

-->02 TABLE-1-ITEM(1),offset=0,length=1
20                                               " "
```

Alt+Pause keyboard combination

By pressing Alt+Pause when a graphical window is active, a message box is shown to provide information about the focus. When the user closes the message box, the Framework restores the focus on the window. This kind of operation is often useful to get rid of hanged screens.

If the configuration property iscobol.gui.native_name is set to true, the Screen Section name of the controls is shown in the message box produced by Alt+Pause. In thin client environment, the property must be set server-side.

If `iscobol.tracelevel` is set to a value of 512 or greater, complementary information is written to the log file specified by `iscobol.logfile` when Alt+Pause is pressed. It's good practice to set `iscobol.logfile.append` to true. In thin client environment the client-side log file is updated, so the three properties must be set client-side.

How it works

The isCOBOL runtime is divided in server and client, also when it works in stand-alone mode.

Every event that occurs on the client is sent to the server, that manages it. For example, when the user clicks on a push button, the client sends a GOTO event and a CLICKED event to the server. To avoid that multiple events are sent in the same time (i.e. if the user clicks quickly on more push-buttons) the isCOBOL runtime locks the keyboard before sending the event to the server and unlocks the keyboard when the event has been managed from the server.

When `iscobol.tracelevel` is set to a value of 512 or greater, the isCOBOL runtime traces the activation and deactivation of the keyboard in the log. This kind of information is important to address focus errors where the active window appears frozen. In these circumstances, pressing ALT+PAUSE a special message box

appears and shows some information about the focus; the same information is written in the log file. After ALT+PAUSE is pressed, the runtime starts to log also the focus changes on the windows and the events that are sent to the client.

Alternative key

The Pause button may not be available on your keyboard. In this case, it's possible to assign the same function to another key. The chosen key must be assigned with the exception value 65535. For example, in order to use F5 as replacement for Alt+Pause, include the following entry in the configuration:

```
iscobol.key.f5=exception=65535
```

Data Access Issues

Problems related to data access are identified by a file status value different than zero after a specific I/O operation. See File Status Codes for possible file status values and their meaning.

In order to retrieve additional information on the cause of failure you can take advantage of the isCOBOL logging feature. This feature is activated by setting the `iscobol.tracelevel` property to a value different than zero and optionally by setting `iscobol.logfile` to the name of the log file you wish to create.

If you are troubleshooting problems during the open of a file, it's enough to set the trace level to the value of 8. When the I/O error occurs, you will find useful information in the log file including the path where the file was searched and the file handler that isCOBOL used to open the file.

If you're troubleshooting other kind of I/O errors, you may find interesting to know the content of the record or the key that causes the error. In this case you should set the trace level to the value of 32.

When the file handler is c-treeRTG, two additional log files can be consulted to retrieve more information.

- the file CTSTATUS.FCS that is automatically updated in the server process directory. This file is particularly useful to diagnose startup errors. You may check it to ensure that the server process has been correctly started.
- the client log file that can be activated either with the iscobol.file.index.log.file configuration property or with the <log> ctree.conf entry.

If the file handler is Database Bridge then you may find additional information in the JDBC log. Consult the database documentation or contact your database vendor to know how to trace the JDBC activity.

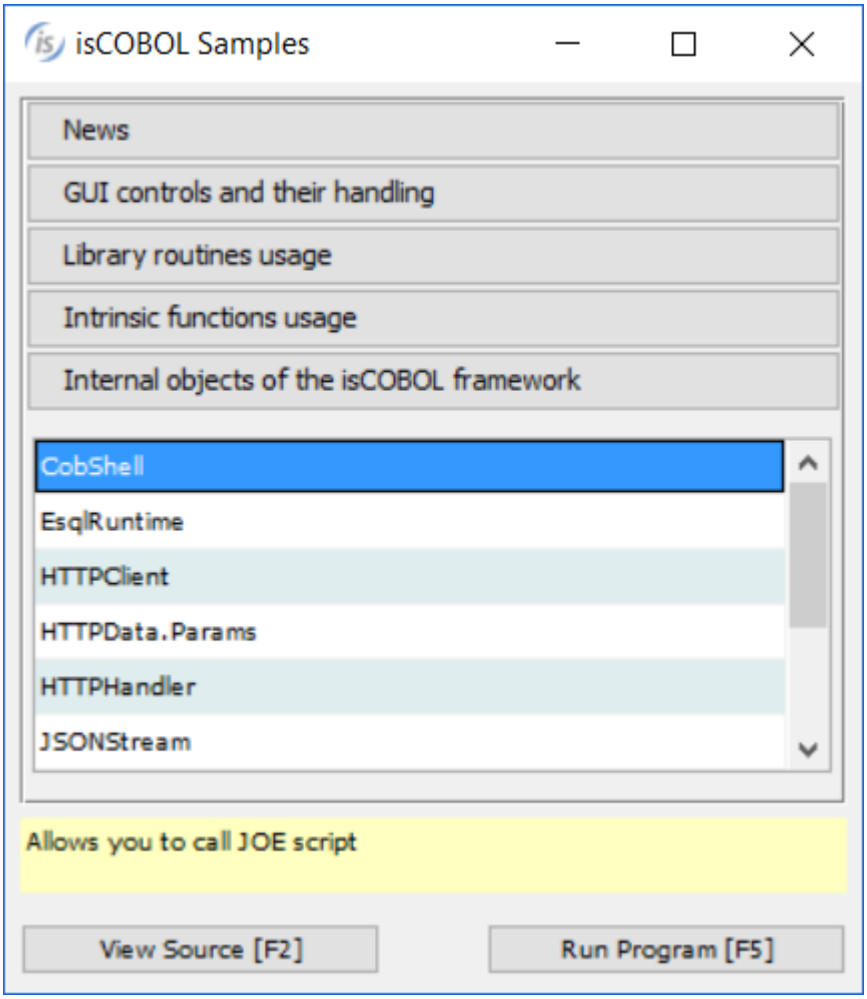
Appendix N

Internal Objects

isCOBOL includes the following internal objects:

- CobolGUIJavaBean Class (com.iscobol.gui.server.CobolGUIJavaBean)
- CobShell Class (com.iscobol.cobshell.CobShell)
- CurrentDate Class (com.iscobol.rts.CurrentDate)
- efdParser Class (com.iscobol.lib.efdParser)
- EsqlRuntime (com.iscobol.rts.EsqlRuntime)
- HTTPClient, HTTPData.params and HTTPHandler Classes
- JSONStream Class (com.iscobol.rts.JSONStream)
- Logger (com.iscobol.logger.Logger)
- LoggerFactory (com.iscobol.logger.LoggerFactory)
- Slf4jLogger class (com.iscobol.logger.Slf4jLogger)
- SpoolPrinter class (com.iscobol.rts.print.SpoolPrinter) and the Print Preview
- StoreProcedure Class (com.iscobol.lib.StoreProcedure)
- XMLStream Class (com.iscobol.rts.XMLStream)

Sample programs for the above objects are available among the isCOBOL Samples.



CobolGUIJavaBean Class (com.iscobol.gui.server.CobolGUIJavaBean)

CobolGUIJavaBean is an internal class that allows to manage java-beans in Screen Section as well as invoking static methods client side and server side in thin client environment.

callMethod

Invokes a method of the java-bean in Screen Section. See JAVA-BEAN for more information on java-beans in Screen Section.

General format

Object **callMethod** (name)
Object **callMethod** (name[, object-parameter-1][, object-parameter-2][, object-parameter-3][, object-parameter-4][, object-parameter-5])
Object **callMethod** (name, object-parameters)
Object **callMethod** (name, signature, object-parameters)

Syntax rules

1. *name* is an alphanumeric data item or literal
2. *object-parameter-1* to *object-parameter-5* are object reference to java.lang.Object
3. *signature* is an alphanumeric data item or literal
4. *object-parameters* is an object reference to java.lang.Object[]

General Rules

1. *name* is case sensitive.
2. *signature* is a comma separated list of Java data types that describe the parameters that will follow, e.g. ("int, int, String").
3. The object returned can be intercepted using COBOL data items if it's a number or a string.

callStaticMethod

Invokes a static method of a given class client side in thin client environment.

General format

Object **callStaticMethod** (class, name)
Object **callStaticMethod** (class, name[, object-parameter-1][, object-parameter-2][, object-parameter-3][, object-parameter-4][, object-parameter-5])
Object **callStaticMethod** (class, name, object-parameters)
Object **callStaticMethod** (class, name, signature, object-parameters)

Syntax rules

1. *class* is an alphanumeric data item or literal
2. *name* is an alphanumeric data item or literal
3. *object-parameter-1* to *object-parameter-5* are object reference to java.lang.Object
4. *signature* is an alphanumeric data item or literal
5. *object-parameters* is an object reference to java.lang.Object[]

General Rules

1. *class* and *name* is case sensitive.
2. *signature* is a comma separated list of Java data types that describe the parameters that will follow, e.g. ("int, int, String").
3. The object returned can be intercepted using COBOL data items if it's a number or a string.

callStaticMethodOnServer

Invokes a static method of a given class server side using the remote calls technology. See Remote objects for more information about remote calls.

General format

Object **callStaticMethodOnServer** (class, name)
Object **callStaticMethodOnServer** (class, name[, object-parameter-1][, object-parameter-2][, object-parameter-3][, object-parameter-4][, object-parameter-5])
Object **callStaticMethodOnServer** (class, name, object-parameters)
Object **callStaticMethodOnServer** (class, name, signature, object-parameters)

Syntax rules

1. *class* is an alphanumeric data item or literal
2. *name* is an alphanumeric data item or literal
3. *object-parameter-1* to *object-parameter-5* are object reference to java.lang.Object
4. *signature* is an alphanumeric data item or literal
5. *object-parameters* is an object reference to java.lang.Object[]

General Rules

1. *class* and *name* is case sensitive
2. *signature* is a comma separated list of Java data types that describe the parameters that will follow, e.g. ("int, int, String")
3. The runtime invokes the specified method locally, first. If it does not find the class or the method locally and the configuration property iscobol.remote.code_prefix is set to a valid value, thenit invokes the method remotelly.
4. For security reasons, the method will be invoked on the server side only if it is defined as alias name.See Using Aliases for more information about aliases. The method alias name must be in the format *className.methodName*, case-sensitive.
5. The object returned can be intercepted using COBOL data items if it's a number or a string.

getProperty

Retrieves the value of a property of the java-bean in Screen Section. See JAVA-BEAN for more information on java-beans in Screen Section.

General format

Object **getProperty** (name)

Syntax rules

1. *name* is an alphanumeric data item or literal.

General Rules

1. *name* is case sensitive.
2. The object returned by getProperty() can be intercepted using COBOL data items if it's a number or a string.

setProperty

Sets the value of a property of the java-bean in Screen Section. See JAVA-BEAN for more information on java-beans in Screen Section.

General format

```
int setProperty (name, value)
```

Syntax rules

1. *name* is an alphanumeric data item or literal.
2. *value* is an object reference to java.lang.Object. For numeric and alphanumeric values, COBOL data items or literals can be used.

General Rules

1. *name* is case sensitive.

- setProperty() returns 0 on success and -1 on failure.

CobShell Class (com.iscobol.cobshell.CobShell)

CobShell is the interpreter of the JOE script language. COBOL programs can invoke JOE scripts through this class.

Constructor

Creates a new instance of the CobShell class.

Format 1

```
CobShell (Script-Name)
```

Syntax rules

- *Script-Name* is a alphanumeric data item or string literal.

General rules

- *Script-Name* should point to a disk file. Relative paths are resolved according to the JVM working directory.

Code example

```
       configuration section.
       repository.
           class cobshell as "com.iscobol.cobshell.CobShell"
       ...
       working-storage section.
       77  joe object reference cobshell.
       ...
       procedure division.
       ...
           try
              set joe  to cobshell:>new ("validator.joe")
           catch exception
              display exception-object
           end-try.
```

Format 2

```
CobShell (Script-Name, Arguments)
```

Syntax rules

- *Script-Name* is a alphanumeric data item or string literal.
- *Arguments* is a variable number of java.lang.Object instances. COBOL data items and literals can be used as well.

General rules

- *Script-Name* should point to a disk file. Relative paths are resolved according to the JVM working directory.
- *Arguments* are the values that you would pass if you run the script on the command-line.

Code example

```
       configuration section.
       repository.
           class cobshell as "com.iscobol.cobshell.CobShell"
       ...
       working-storage section.
       77  joe object reference cobshell.
       ...
       procedure division.
       ...
           try
              set joe  to cobshell:>new ("script1.joe", 1, "XYZ")
           catch exception
              display exception-object
           end-try.
```

execBlock

Executes a named block of code in the JOE script.

Format 1

```
java.lang.Object execBlock (Block-Name)
```

Syntax rules

- *Block-Name* is a alphanumeric data item or string literal.

General rules

- *Block-Name* should point to the name of one of the Blocks included in the script.

Code example

```
       configuration section.
       repository.
           class cobshell as "com.iscobol.cobshell.CobShell"
       ...
       working-storage section.
       77  joe object reference cobshell.
       ...
       procedure division.
       ...
           try
              joe:>execBlock ("showVersion")
           catch exception
              display exception-object
           end-try.
```

Format 2

```
java.lang.Object execBlock (Block-Name, Arguments)
```

Syntax rules

- *Block-Name* is a alphanumeric data item or string literal.
- *Arguments* is a variable number of java.lang.Object instances. COBOL data items and literals can be used as well.

General rules

- *Block-Name* should point to the name of one of the Blocks included in the script.
- *Arguments* are passed to the block.

Code example

```
       configuration section.
       repository.
           class cobshell as "com.iscobol.cobshell.CobShell"
       ...
       working-storage section.
       77  joe object reference cobshell.
       77  joe-result        pic x any length.
       77  w-id              pic x any length.
       ...
       procedure division.
       ...
           try
              set joe-result to joe:>execBlock ( "checkID",
                                                 w-id )
              display joe-result
           catch exception
              display exception-object
           end-try.
```

CurrentDate Class (com.iscobol.rts.CurrentDate)

The com.iscobol.rts.CurrentDate class allows the specification of an artificial date.

When an ACCEPT FROM CENTURY-DATE, ACCEPT FROM TIME or a FUNCTION CURRENT-DATE is performed, the Framework runs the internal class com.iscobol.rts.CurrentDate that returns the current date and time by invoking the "now" method of the GregorianCalendar Java object.

If you wish to make the Framework return a datetime that is different than the current one, you can write a main program that inherits the CurrentDate class, uses its “set” method to set a custom date and then calls the main program of the COBOL application. For the whole runtime session, each inquire on date and time will return the custom values.

A working sample is supplied with isCOBOL and is available in the $ISCOBOL/sample/date-simulator directory.

The program CHDATE is a sort of custom runtime that sets the custom datetime and launches a COBOL program. The usage is:

```
iscrun CHDATE custom_date program_name
```

Where custom_date is a 8 digits or 14 digits number that identifies a datetime in the format YYYYMMDDhhnnss. The hhnnss part is optional; if omitted, the current time is used. Program_name is the name of the COBOL program to start.

Example:

```
iscrun CHDATE 20010212 MYPROG
```

All inquiries on the current datetime made by MYPROG and the programs it calls will return 12th February 2001, whatever the current system datetime is.

efdParser Class (com.iscobol.lib.efdParser)

The efdParser is an internal class that allows you to retrieve information from EFD dictionaries.

The efdParser object is deprecated and supported only for compatibility with previous versions. The C$PARSEEFD routine should be used instead.

Constructor

Creates a new instance of the efdParser class.

General format

```
efdParser
```

Code example

```
...
configuration section.
repository.
    class efdParser  as "com.iscobol.lib.efdParser"
    .
...
working-storage section.
...
77 efd-parser object reference efdParser.
...
procedure division.
...
set efd-parser to efdParser:>new()
...
```

Note - if your program is compiled with `-cp` option, use *com.iscobol.lib_n.efdParser* instead of *com.iscobol.lib.efdParser*.

efdDescr

Parses an EFD file and returns general information about it.

General format

```
efd-description efdDescr ( efd-name )
```

Syntax rules

1. *efd-name* is an alphanumeric data-item or string literal.

2. *efd-description* is a structure defined in the SQLCA copybook as follows.:

```
01  efd-description.
  03  efd-version                          pic x    comp-x.
  03  efd-select-name                      pic x(30).
  03  efd-filename                         pic x(30).
  03  efd-filetype                         pic x    comp-x.
  03  efd-max-record-size                  pic x(4) comp-x.
  03  efd-min-record-size                  pic x(4) comp-x.
  03  efd-number-of-keys                   pic x    comp-x.
  03  efd-number-conditions                pic xx   comp-x.
  03  efd-number-fields                    pic xx   comp-x.
  03  efd-total-number-fields              pic xx   comp-x.
  03  efd-total-number-allfields              pic xx   comp-x.
  03  efd-key-index                        pic xx   comp-x.
  03  efd-field-index                      pic xx   comp-x.
  03  save-efd-field-index                 pic xx   comp-x.
  03  min-efd-field-index                  pic xx   comp-x.
  03  max-efd-field-index                  pic xx   comp-x.
  03  efd-cond-index                       pic xx   comp-x.
  03  efd-max-field-name-len               pic xx   comp-x.
  03  efd-num-key-flds                     pic x    comp-x occurs 120 times.
```

Code example

```
...
configuration section.
repository.
    class efdParser  as "com.iscobol.lib.efdParser"
    .
...
working-storage section.
...
copy "efdParser.def"
77 efd-parser object reference efdParser.
...
procedure division.
...
set efd-description to efd-parser:>efdDescr("fd1.xml")
...
```

efdCondInfo

Parses an EFD file and returns information about a specific condition.

General format

```
efd-condition-description efdCondInfo ( efd-name, cond-index)
```

Syntax rules

1. *efd-name* is an alphanumeric data-item or string literal.
2. *cond-index* must be defined as PIC X(2) COMP-X.

3. *efd-condition-description* is a structure defined in the efdParser.def copybook as follows.:

```
01  efd-condition-description.
    03  efd-condition-type                  pic x comp-x.
        88  efd-equal-condition             value 1.
        88  efd-and-condition               value 2.
        88  efd-other-condition             value 3.
        88  efd-gt-condition                value 4.
        88  efd-ge-condition                value 5.
        88  efd-lt-condition                value 6.
        88  efd-le-condition                value 7.
        88  efd-ne-condition                value 8.
        88  efd-or-condition                value 9.
        88  efd-comparison-condition        values 1, 4 through 8.
    03  efd-condition-flag                  pic x.
        88  efd-true-condition              value 'y' false 'n'.
    03  efd-other-conditions.
        05  efd-other-fieldnum              pic xx comp-x.
        05  efd-other-fieldname             pic x(30).
        05  efd-other-field-val             pic x(50).
        05  efd-other-field-nums            redefines efd-other-field-val.
            07  efd-cond-val-1              pic s9(18).
            07  efd-cond-val-2              pic s9(18).
    03  efd-and-conditions                  redefines efd-other-conditions.
        05  efd-condition-1                 pic xx comp-x.
        05  efd-condition-2                 pic xx comp-x.
    03  efd-condition-tablename             pic x(30).
```

General rules

1. *cond-index* must be greater than zero and specifies the ordinal position of the condition in the EFD file.

Code example

```
...
configuration section.
repository.
    class efdParser  as "com.iscobol.lib.efdParser"
    .
...
working-storage section.
...
copy "efdParser.def"
77 efd-parser object reference efdParser.
77 cond-idx   pic xx comp-x.
77 buf-type   pic x(2).
...
procedure division.
...
move 1 to cond-idx.
set efd-condition-description to efd-parser:>efdCondInfo("fd1.xml",
                                                         cond-idx)
...
```

efdFieldInfo

Parses an EFD file and returns information about a specific field. Only fields that are not marked as "hidden" are returned.

General format

```
efd-field-description efdFieldInfo ( efd-name, field-index)
```

Syntax rules

1. *efd-name* is an alphanumeric data-item or string literal.
2. *field-index* is a numeric data-item or numeric literal.

3. *efd-field-description* is a structure defined in the efdParser.def copybook as follows.:

```
01  efd-field-description.
    03  efd-field-offset                    pic x(4) comp-x.
    03  efd-field-length                    pic x(4) comp-x.
    03  efd-field-type                      pic x comp-x.
        88  efd-signed-field                values efd-NumSignSep
                                                   efd-NumSigned
                                                   efd-NumSepLead
                                                   efd-NumLeading
                                                   efd-CompSigned
                                                   efd-PackedSigned
                                                   efd-BinarySigned
                                                   efd-NativeSigned.
        88  efd-numeric-field               values efd-NumEdited thru efd-
NativeUnsigned.
        88  efd-float-field                 value  efd-Flt.
        88  efd-ascii-field                 values efd-Alphanum  thru efd-Group.
        88  efd-national-field              values efd-Nat-type  thru efd-
NatEdited.
        88  efd-wide-field                  values efd-Wide-type thru efd-
WideEdited.
    03  efd-field-digits                    pic x(4) comp-x.
    03  efd-field-scale                     pic s99 comp-4.
    03  efd-field-user-type                 pic xx comp-x.
    03  efd-field-condition                 pic xx comp-x.
    03  efd-field-level                     pic x comp-x.
    03  efd-field-name                      pic x(30).
    03  efd-field-occurs-depth              pic x comp-x.
    03  efd-field-occurs-table              occurs efdMaxNumKeyFields times
                                            indexed by efd-field-occurs-level.
        05  efd-field-occ-max-idx           pic xx comp-x.
        05  efd-field-occ-offset            pic xx comp-x.
```

General rules

1. *field-index* must be greater than zero and not greater than `efd-total-number-fields` (returned by efdDescr) and specifies the ordinal position of the field in the EFD file.

Code example

```
...
configuration section.
repository.
    class efdParser  as "com.iscobol.lib.efdParser"
    .
...
working-storage section.
...
copy "efdParser.def"
77 efd-parser object reference efdParser.
77 field-idx  pic 9(3).
77 buf-type   pic x(20).
...
procedure division.
...
move 1 to field-idx.
set efd-field-description to efd-parser:>efdFieldInfo("fd1.xml",
                                                      field-idx)
...
```

efdFieldAllInfo

Parses an EFD file and returns information about a specific field including fields marked as "hidden".

General format

efd-field-description **efdFieldAllInfo** (efd-name, field-index)

Syntax rules

1. *efd-name* is an alphanumeric data-item or string literal.
2. *field-index* is a numeric data-item or numeric literal.

3. *efd-field-description* is a structure defined in the efdParser.def copybook as follows.:

```
01  efd-field-description.
    03  efd-field-offset                    pic x(4) comp-x.
    03  efd-field-length                    pic x(4) comp-x.
    03  efd-field-type                      pic x comp-x.
        88  efd-signed-field                values efd-NumSignSep
                                                   efd-NumSigned
                                                   efd-NumSepLead
                                                   efd-NumLeading
                                                   efd-CompSigned
                                                   efd-PackedSigned
                                                   efd-BinarySigned
                                                   efd-NativeSigned.
        88  efd-numeric-field               values efd-NumEdited thru efd-
NativeUnsigned.
        88  efd-float-field                 value  efd-Flt.
        88  efd-ascii-field                 values efd-Alphanum  thru efd-Group.
        88  efd-national-field              values efd-Nat-type  thru efd-
NatEdited.
        88  efd-wide-field                  values efd-Wide-type thru efd-
WideEdited.
    03  efd-field-digits                    pic x(4) comp-x.
    03  efd-field-scale                     pic s99 comp-4.
    03  efd-field-user-type                 pic xx comp-x.
    03  efd-field-condition                 pic xx comp-x.
    03  efd-field-level                     pic x comp-x.
    03  efd-field-name                      pic x(30).
    03  efd-field-occurs-depth              pic x comp-x.
    03  efd-field-occurs-table              occurs efdMaxNumKeyFields times
                                            indexed by efd-field-occurs-level.
        05  efd-field-occ-max-idx           pic xx comp-x.
        05  efd-field-occ-offset            pic xx comp-x.
```

General rules

1. *field-index* must be greater than zero and not greater than `efd-total-number-allfields` (returned by efdDescr) and specifies the ordinal position of the field in the EFD file.

Code example

```
...
configuration section.
repository.
    class efdParser  as "com.iscobol.lib.efdParser"
    .
...
working-storage section.
...
copy "efdParser.def"
77 efd-parser object reference efdParser.
77 field-idx  pic 9(3).
77 buf-type   pic x(20).
...
procedure division.
...
move 1 to field-idx.
set efd-field-description to efd-parser:>efdFieldAllInfo("fd1.xml",
                                                         field-idx)
...
```

efdKeyInfo

Parses an EFD file and returns information about a specific field.

General format

```
efd-key-description efdKeyInfo ( efd-name, key-index )
```

Syntax rules

1. *efd-name* is an alphanumeric data-item or string literal.
2. *field-index* must be defined as PIC 9(3).
3. *efd-key-description* is a structure defined in the efdParser.def copybook as follows.:

```
01 efd-key-description-group.
    03  efd-key-description.
        05  efd-number-of-segments              pic 99.
        05  efd-allow-dup-flag                  pic 9.
            88  efd-allow-duplicates            value 1 false 0.
        05  efd-segment-description             occurs efdmax-
segs times indexed by efd-seg-idx.
            07  efd-segment-length              pic x    comp-x.
            07  efd-segment-offset              pic x(4) comp-x.
        05  efd-num-of-key-fields               pic x    comp-x.
        05  efd-key-
fields                      occurs efdMaxNumKeyFields times indexed by efd-key-field-
idx.
            07  efd-key-field-name              pic x(30).
            07  efd-key-field-num               pic xx   comp-x.
    03  efd-key-index-buf                       pic xx   comp-x.
```

General rules

1. *key-index* must be greater than zero and specifies the ordinal position of the field in the EFD file.

Code example

```
...
configuration section.
repository.
    class efdParser  as "com.iscobol.lib.efdParser"
    .
...
working-storage section.
...
copy "efdParser.def"
77 efd-parser object reference efdParser.
77 key-idx    pic 9(3).
...
procedure division.
...
move 1 to key-idx.
set efd-key-description to efd-parser:>efdKeyInfo("fd1.xml",
                                                   key-idx)
...
```

EsqlRuntime (com.iscobol.rts.EsqlRuntime)

The EsqlRuntime class returns the java.sql.Connection instance of the current JDBC connection.

Constructor

The method exposed by this class is static, so no constructor is available.

getCurrConnection

Returns the Connection instance of the current JDBC connection.

General format

```
java.sql.Connection getCurrConnection ()
```

General rules

1. A java.sql.Connection object is returned. If no connection exists, then null is returned.

Code example

```
...
configuration section.
repository.
    class EsqlRuntime as "com.iscobol.rts.EsqlRuntime"
    class JSQLConnection as "java.sql.Connection"
...
working-storage section.
...
77 jcon object reference JSQLConnection.
...
procedure division.
...
    exec sql
     connect
    end-exec.
    if sqlcode = 0
     set jcon to EsqlRuntime:>getCurrConnection()
    else
     display "Connection error: " sqlcode
     display sqlerrmc
    end-if.
...
```

HTTPClient, HTTPData.params and HTTPHandler Classes

isCOBOL includes a set of classes for the communication between COBOL and HTTP.

Refer to isCOBOL EIS Appendices for details.

JSONStream Class (com.iscobol.rts.JSONStream)

The JSONStream is an internal class that allows JSON streams to be easily read and written.

Constructor

Creates a new instance of the JSONStream class.

Format 1

```
JSONStream ( Record-Definition )
```

Format 2

```
JSONStream ( Record-Definition, hasDummyRoot )
```

Format 3

```
JSONStream ( Record-Definition, hasDummyRoot, encoding )
```

Syntax rules

1. *Record-Definition* is a level 01 group data item for which the IS IDENTIFIED clause has been specified.
2. *hasDummyRoot* is an alphanumeric data item or literal hosting a boolean value (e.g. "0", "1", "true", "false", "yes", "no", "on" or "off"). If the boolean value is TRUE, then the top level item of Record-Definition is discarded and will not appear in the JSON stream . This parameter can be NULL, in such case, FALSE is assumed.
3. *encoding* is an alphanumeric data item or literal that specifies the character set of the JSON stream. All the canonical names listed in the following Java documentation can be used as value for this property: http://java.sun.com/javase/6/docs/technotes/guides/intl/encoding.doc.html.

General rules

1. When a new instance of JSONStream is created, the data item identified by *Record-Definition* is associated with the new object.

Code example

```
...
configuration section.
repository.
    class jsonStream  as "com.iscobol.rts.JSONStream"
    .
...
working-storage section.
...
77 objJsonStream object reference jsonStream.
01 Record-Definition identified by "Record-Definition".
   (JSON fields)
...
procedure division.
...
set objJsonStream to jsonStream:>new(Record-Definition)
...
```

getPrintWriter

Returns the PrintWriter associated to the JSONStream object.

General format

```
java.io.PrintWriter getPrintWriter ()
```

General rules

1. A java.io.PrintWriter object is returned. You can use this method to add text to the content generated by other JSONStream write methods.

Code example

```
...
configuration section.
repository.
    class jsontream  as "com.iscobol.rts.JSONStream"
    class printWriter  as "java.io.PrintWriter"
    .
...
working-storage section.
...
77 objJsonStream object reference jsonStream.
77 objPrintWriter object reference printWriter.
...
procedure division.
...
set objPrintWriter to objJsonStream:>getPrintWriter().
```

read

Reads a JSON stream or file and puts its content in the data item associated with the JSONStream object.

General format

```
void read ( Json-Source )
```

Syntax rules

1. *Json-Source* can be either a Data Item or a Nonnumeric Literal.

General rules

1. *Json-Source* refers to a regular disk file.
2. When the read method is invoked, the whole content of *Json-Source* is read and the data item associated with the JSONStream object is updated.

   NOTE - If the number of occurrences of a JSON field is not known at the time the program is written, specifying the `DYNAMIC` phrase of the OCCURS clause in the data item associated with the JSONStream object is recommended.
3. The following escapes are allowed: \b, \f, \n, \r, \t, \" and \\. Any other character preceeded by a backslash is considered invalid and makes the read fail unless you set iscobol.jsonstream.allow_backslash_escaping_any_character (boolean) to true in the configuration.

Code example

```
...
configuration section.
repository.
    class jsonStream  as "com.iscobol.rts.JSONStream"
    .
...
working-storage section.
...
77 objJsonStream object reference jsonStream.
...
procedure division.
...
objJsonStream:>read("Sample.json")
...
```

readFromFile

Reads a JSON file and puts its content in the data item associated with the JSONStream object.

General format

```
void readFromFile ( Json-Source )
```

Syntax rules

1. *Json-Source* can be either a Data Item or a Nonnumeric Literal.

General rules

1. *Json-Source* refers to a regular disk file.
2. When the read method is invoked, the whole content of *Json-Source* is read and the data item associated with the JSONStream object is updated.

   NOTE - If the number of occurrences of an JSON field is not known at the time the program is written, specifying the `DYNAMIC` phrase of the OCCURS clause in the data item associated with the JSONStream object is recommended.
3. The following escapes are allowed: \b, \f, \n, \r, \t, \" and \\. Any other character preceeded by a backslash is considered invalid and makes the read fail unless you set iscobol.jsonstream.allow_backslash_escaping_any_character (boolean) to true in the configuration.

Code example

```
...
configuration section.
repository.
    class jsonStream  as "com.iscobol.rts.JSONStream"
    .
...
working-storage section.
...
77 objJsonStream object reference jsonStream.
...
procedure division.
...
objJsonStream:>readFromFile("Sample.json")
...
```

readFromString

Reads a JSON stream and puts its content in the data item associated with the JSONStream object.

Format 1

void **readFromString** (Json-Source)

Format 2

void **readFromString** (Json-Source, encoding)

Syntax rules

1. *Json-Source* is an object reference to java.lang.String.
2. *encoding* is a string literal or data item that specified the character set to be used while parsing the JSON stream. It accepts the same values as the iscobol.encoding * configuration property.

General rules

1. When the read method is invoked, the whole content of *Json-Source* is read and the data item associated with the JSONStream object is updated.

   NOTE - If the number of occurrences of a JSON field is not known at the time the program is written, specifying the `DYNAMIC` phrase of the OCCURS clause in the data item associated with the JSONStream object is recommended.
2. The following escapes are allowed: \b, \f, \n, \r, \t, \" and \\. Any other character preceeded by a backslash is considered invalid and makes the read fail unless you set iscobol.jsonstream.allow_backslash_escaping_any_character (boolean) to true in the configuration.

Code example

```
...
configuration section.
repository.
    class jsonStream  as "com.iscobol.rts.JSONStream"
    class JString     as "java.lang.String"
    .
...
working-storage section.
...
77 objJsonStream object reference jsonStream.
77 objString    object reference JString.
...
procedure division.
...
objJsonStream:>readFromString(objString)
...
```

readFromStream

Reads a JSON stream and puts its content in the data item associated with the JSONStream object.

General format

void **readFromStream** (Json-Source)

Syntax rules

1. *Json-Source* is an object reference to java.io.InputStream.

General rules

1. When the read method is invoked, the whole content of *Json-Source* is read and the data item associated with the JSONStream object is updated.

   NOTE - If the number of occurrences of a JSON field is not known at the time the program is written, specifying the `DYNAMIC` phrase of the OCCURS clause in the data item associated with the JSONStream object is recommended.

2. The following escapes are allowed: \b, \f, \n, \r, \t, \" and \\. Any other character preceeded by a backslash is considered invalid and makes the read fail unless you set iscobol.jsonstream.allow_backslash_escaping_any_character (boolean) to true in the configuration.

Code example

```
...
configuration section.
repository.
    class jsonStream  as "com.iscobol.rts.JSONStream"
    class inpStream  as "java.io.InputStream"
    .
...
working-storage section.
...
77 objJsonStream object reference jsonStream.
77 objInpStream object reference inpStream.
...
procedure division.
...
objJsonStream:>readFromStream(objInpStream)
...
```

setPrintWriter

Associates a PrintWriter to the JSONStream object. It will be used by the write method.

General format

void **setPrintWriter** (Print-Writer)

Syntax rules

1. *Print-Writer* is an object reference to java.io.PrintWriter.

Code example

```
...
configuration section.
repository.
    class jsonStream  as "com.iscobol.rts.JSONStream"
    .
...
working-storage section.
...
77 objJsonStream object reference jsonStream.
...
procedure division.
...
objJsonStream:>setPrintWriter("out.json")
...
```

write

General format

```
void write ( )
```

General rules

1. When the write method is invoked, the whole content of the data item associated with the JSONStream object is sent to the PrintWriter object associated by the method setPrintWriter. If no PrintWriter has been associated, the stream is printed on the system output.

   NOTE - If the number of occurrences of a JSON fields is not known at the time the program is written, specifying the `DYNAMIC` phrase of the OCCURS clause is recommended.

Code example

```
...
configuration section.
repository.
    class jsonStream  as "com.iscobol.rts.JSONStream"
    .
...
working-storage section.
...
77 objJsonStream object reference jsonStream.
...
procedure division.
...
objJsonStream:>write()
...
```

writeToFile

Puts the content of the data item associated with a JSONStream object into a file.

General format

```
void writeToFile ( Json-Destination )
```

Syntax rules

1. *Json-Destination* can be either a Data Item or a Nonnumeric Literal.

General rules

1. *Json-Destination* refers to a regular disk file.
2. When the write method is invoked, the whole content of the data item associated with the JSONStream object is put into *Json-Destination*.

> **NOTE** - If the number of occurrences of a JSON fields is not known at the time the program is written, specifying the `DYNAMIC` phrase of the OCCURS clause is recommended.

Code example

```
...
configuration section.
repository.
    class jsonStream  as "com.iscobol.rts.JSONStream"
    .
...
working-storage section.
...
77 objJsonStream object reference jsonStream.
...
procedure division.
...
objJsonStream:>writeToFile("Sample.json")
...
```

writeToPrintWriter

Puts the content of the data item associated with an JSONStream object into a PrintWriter object.

General format

```
void writeToPrintWriter ( Json-Destination )
```

Syntax rules

1. *Json-Destination* is an object reference to java.io.PrintWriter.

General rules

1. When the write method is invoked, the whole content of the data item associated with the JSONStream object is put into *Json-Destination*.

> **NOTE** - If the number of occurrences of a JSON fields is not known at the time the program is written, specifying the `DYNAMIC` phrase of the OCCURS clause is recommended.

Code example

```
...
configuration section.
repository.
    class jsonStream  as "com.iscobol.rts.JSONStream"
    .
...
working-storage section.
...
77 objJsonStream object reference jsonStream.
...
procedure division.
...
objJsonStream:>writeToPrintWriter(objJsonStream:>getPrintWriter)
...
```

writeToStream

Puts the content of the data item associated with an JSONStream object into an OutputStream object.

General format

void **writeToStream** (Json-Destination)

Syntax rules

1. *Json-Destination* is an object reference to java.io.OutputStream.

General rules

1. When the write method is invoked, the whole content of the data item associated with the JSONStream object is put into *Json-Destination*.

   NOTE - If the number of occurrences of a JSON fields is not known at the time the program is written, specifying the `DYNAMIC` phrase of the OCCURS clause is recommended.

Code example

```
...
configuration section.
repository.
    class jsonStream  as "com.iscobol.rts.JSONStream"
    class outStream  as "java.io.OutputStream"
    .
...
working-storage section.
...
77 objJsonStream object reference jsonStream.
77 objOutStream object reference outStream.
...
procedure division.
...
set objOutStream to outStream:>new()
objJsonStream:>writeToStream(objOutStream)
...
```

writeToStringBuffer

Puts the content of the data item associated with an JSONStream object into an StringBuffer object.

General format

void **writeToStringBuffer** (Json-Destination)

Syntax rules

1. *Json-Destination* is an object reference to java.lang.StringBuffer.

General rules

1. When the write method is invoked, the whole content of the data item associated with the JSONStream object is put into *Json-Destination*.

   NOTE - If the number of occurrences of a JSON fields is not known at the time the program is written, specifying the `DYNAMIC` phrase of the OCCURS clause is recommended.

Code example

```
...
configuration section.
repository.
    class jsonStream  as "com.iscobol.rts.JSONStream"
    class strBuffer  as "java.lang.StringBuffer"
    .
...
working-storage section.
...
77 objJsonStream object reference jsonStream.
77 objStrBuffer object reference strBuffer.
...
procedure division.
...
set objStrBuffer to strBuffer:>new()
objJsonStream:>writeToStringBuffer(objStrBuffer)
...
```

Logger (com.iscobol.logger.Logger)

The Logger class writes messages to the isCOBOL log.

Constructor

General Format

```
Logger()
```

General Rules

1. Use LoggerFactory (com.iscobol.logger.LoggerFactory) to retrieve the instance of this class.

Code example

```
...
configuration section.
repository.
    class LoggerFactory  as "com.iscobol.logger.LoggerFactory"
    class Logger  as "com.iscobol.logger.Logger"
    .
...
working-storage section.
...
77 objLogger object reference Logger.
...
procedure division.
...
set objLogger to LoggerFactory:>getCurrLog()
...
```

info

Writes a message of type INFO to the log. This is the same result you obtain by calling the C$WRITELOG routine.

General Format

```
info (message)
```

Syntax Rules

1. *message* can be any data item.

General Rules

1. The string representation of *message* is added as a new line in the log. The current timestamp and the type INFO are printed before it.

Code example

```
...
configuration section.
repository.
    class LoggerFactory  as "com.iscobol.logger.LoggerFactory"
    class Logger  as "com.iscobol.logger.Logger"
    .
...
working-storage section.
...
77 objLogger object reference Logger.
...
procedure division.
...
set objLogger to LoggerFactory:>getCurrLog()
objLogger:>info("this is an informational message")
...
```

severe

Writes a message of type SEVERE to the log.

General Format

```
severe (message)
```

Syntax Rules

1. *message* can be any data item.

General Rules

1. The string representation of *message* is added as a new line in the log. The current timestamp and the type SEVERE are printed before it.

Code example

```
...
configuration section.
repository.
    class LoggerFactory  as "com.iscobol.logger.LoggerFactory"
    class Logger  as "com.iscobol.logger.Logger"
    .
...
working-storage section.
...
77 objLogger object reference Logger.
...
procedure division.
...
set objLogger to LoggerFactory:>getCurrLog()
objLogger:>severe("this is a severe error message")
...
```

warning

Writes a message of type WARNING to the log.

General Format

```
warning (message)
```

Syntax Rules

1. *message* can be any data item.

General Rules

1. The string representation of *message* is added as a new line in the log. The current timestamp and the type WARNING are printed before it.

Code example

```
...
configuration section.
repository.
    class LoggerFactory  as "com.iscobol.logger.LoggerFactory"
    class Logger  as "com.iscobol.logger.Logger"
    .
...
working-storage section.
...
77 objLogger object reference Logger.
...
procedure division.
...
set objLogger to LoggerFactory:>getCurrLog()
objLogger:>warning("this is a warning message")
...
```

LoggerFactory (com.iscobol.logger.LoggerFactory)

The LoggerFactory class returns the instance of the isCOBOL Logger. Use this instance to add custom messages of different types to the isCOBOL log.

Constructor

The method exposed by this class is static, so no constructor is available.

getCurrLog

The getCurrLog method returns the instance of the current logger. The property iscobol.tracelevel must be set to a value greater than zero, otherwise no logger exists.

General format

```
com.iscobol.logger.Logger getCurrLog ()
```

General rules

1. A Logger (com.iscobol.logger.Logger) object is returned. If no log exists because iscobol.tracelevel is set to zero, then null is returned.

Code example

```
...
configuration section.
repository.
    class LoggerFactory  as "com.iscobol.logger.LoggerFactory"
    class Logger  as "com.iscobol.logger.Logger"
    .
...
working-storage section.
...
77 objLogger object reference Logger.
...
procedure division.
...
set objLogger to LoggerFactory:>getCurrLog()
...
```

Slf4jLogger class (com.iscobol.logger.Slf4jLogger)

The isCOBOL runtime can trace the runtime activity into a log file. There are different trace levels available. Depending on the trace level environment settings, called programs, file i/o and SQL can be included in the trace or discarded. The trace level is set through the property iscobol.tracelevel.

By default the trace is saved to a disk file whose name it's specified by the configuration property iscobol.logfile. It's possible to make the runtime send all the trace information to a class tha implements the *com.iscobol.logger.Logger* interface. In this case it will be a duty of this class to manage the trace by saving it to file or by performing other actions.

The class name must be specified through the configuration property iscobol.logclass. When this property is set, the runtime sends the trace information to the class specified by the property instead of writing the information to the file indicated by the iscobol.logfile property.

Currently there is only one class that implements the *com.iscobol.logger.Logger* interface; this class is included in the isCOBOL runtime library, it's named *com.iscobol.logger.Slf4jLogger* and it's a bridge to the Self4J logger.

In order to use this class, set:

```
iscobol.logclass=com.iscobol.logger.Slf4jLogger
```

Example using Log4J 2

In the following example we make the isCOBOL runtime send the trace information to the Slf4jLogger class that will produce a log splitted into multiple gzipped files. It creates a new file every time the log size reaches 1 MB. The writing to the log is asynchronous.

Classpath setting

The following libraries must appear in the Classpath for a correct result:

Library	Description
iscobol.jar	The isCOBOL runtime.

Library	Description
slf4j-api-1.7.19.jar	The Slf4j API.
log4j-slf4j-impl-2.8.2.jar	The Slf4j bridge to Log4J.
log4j-api-2.8.2.jar log4j-core-2.8.2.jar	The Log4J 2 API.
apache-log4j-extras-1.2.17.jar	Log4J extra functions like rolling and zipping.

Note - slf4j and log4j libraries are not distributed along with isCOBOL. They must be downloaded separately from their web sites.

Log4J 2 configuration

A file named *log4j2.xml* must appear in the Classpath. Put the following content into it in order to activate rolling and zipping of the log file:

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration>
  <Appenders>
    <RollingRandomAccessFile name="RandomAccessFile" fileName="isc.log" filePattern="i
sc-%d{yyyy-MM-dd}-%i.gz" immediateFlush="false" append="true">
      <PatternLayout>
        <Pattern>%d{yyyy-MM-dd HH:mm:ss} - %m%n</Pattern>
      </PatternLayout>
      <Policies>
        <SizeBasedTriggeringPolicy size="1 MB"/>
      </Policies>
    </RollingRandomAccessFile>
  </Appenders>
  <Loggers>
    <Root level="info" includeLocation="false">
      <AppenderRef ref="RandomAccessFile"/>
    </Root>
  </Loggers>
</Configuration>
```

Note - Refer to Log4J 2 documentation for more information about the above entries and other possible entries.

Runtime

For this example, we're going to run the isCOBOL I/O performance test, installed along with isCOBOL. You can find it in the isCOBOL installation directory under the *sample/io-performance* subdirectory.

Run the following command:

```
iscrun -J-Discobol.tracelevel=11 -J-Discobol.logclass=com.iscobol.logger.Slf4jLogger
IO_INDEXED
```

If everything was configured correctly, you should find a similar list of files in the current folder:

```
isc-2019-02-12-1.gz
isc-2019-02-12-2.gz
```

Note - the file name might be a little different, depeding on the date and time when you run the test.

Example using Log4J

In the following example we make the isCOBOL runtime send the trace information to the Slf4jLogger class that will produce a log splitted into multiple gzipped files, one per second.

Classpath setting

The following libraries must appear in the Classpath for a correct result:

Library	Description
iscobol.jar	The isCOBOL runtime.
slf4j-api-1.7.19.jar	The Slf4j API.
slf4j-log4j12-1.7.19.jar	The Slf4j bridge to Log4J.
log4j-1.2.17.jar	The Log4J API.
apache-log4j-extras-1.2.17.jar	Log4J extra functions like rolling and zipping.

Note - slf4j and log4j libraries are not distributed along with isCOBOL. They must be downloaded separately from their web sites.

Log4J configuration

A file named *log4j.properties* must appear in the Classpath. Put the following content into it in order to activate rolling and zipping of the log file:

```
log4j.rootCategory=WARN, iscobolAppender
log4j.appender.WARN=org.apache.log4j.FileAppender
log4j.appender.WARN.File=myLogInfo.log
log4j.appender.WARN.layout=org.apache.log4j.PatternLayout
log4j.logger.com.iscobol.logger.Slf4jLogger=iscobolAppender, WARN
log4j.appender.iscobolAppender=org.apache.log4j.rolling.RollingFileAppender
log4j.appender.iscobolAppender.RollingPolicy=org.apache.log4j.rolling.TimeBasedRolling
Policy
log4j.appender.iscobolAppender.RollingPolicy.FileNamePattern=msg.%d{yyyyMMdd.HHmmssSS}
.gz
log4j.appender.iscobolAppender.layout=org.apache.log4j.PatternLayout
```

Note - Refer to Log4J documentation for more information about the above entries and other possible entries.

Runtime

For this example, we're going to run the isCOBOL I/O performance test, installed along with isCOBOL. You can find it in the isCOBOL installation directory under the *sample/io-performance* subdirectory.

Run the following command:

```
iscrun -J-Discobol.tracelevel=11 -J-Discobol.logclass=com.iscobol.logger.Slf4jLogger
IO_INDEXED
```

If everything was configured correctly, you should find a similar list of files in the current folder:

```
msg.20160331.112746589.gz
msg.20160331.11274778.gz
msg.20160331.11274800.gz
msg.20160331.11274956.gz
msg.20160331.11275000.gz
msg.20160331.11275100.gz
msg.20160331.11275200.gz
msg.20160331.11275300.gz
msg.20160331.11275400.gz
msg.20160331.11275500.gz
msg.20160331.11275600.gz
msg.20160331.11275700.gz
msg.20160331.11275800.gz
msg.20160331.11275901.gz
msg.20160331.11280000.gz
msg.20160331.11280131.gz
msg.20160331.11280200.gz
msg.20160331.11280300.gz
msg.20160331.11280401.gz
msg.20160331.11280500.gz
```

Note - the file name might be a little different, depeding on the date and time when you run the test.

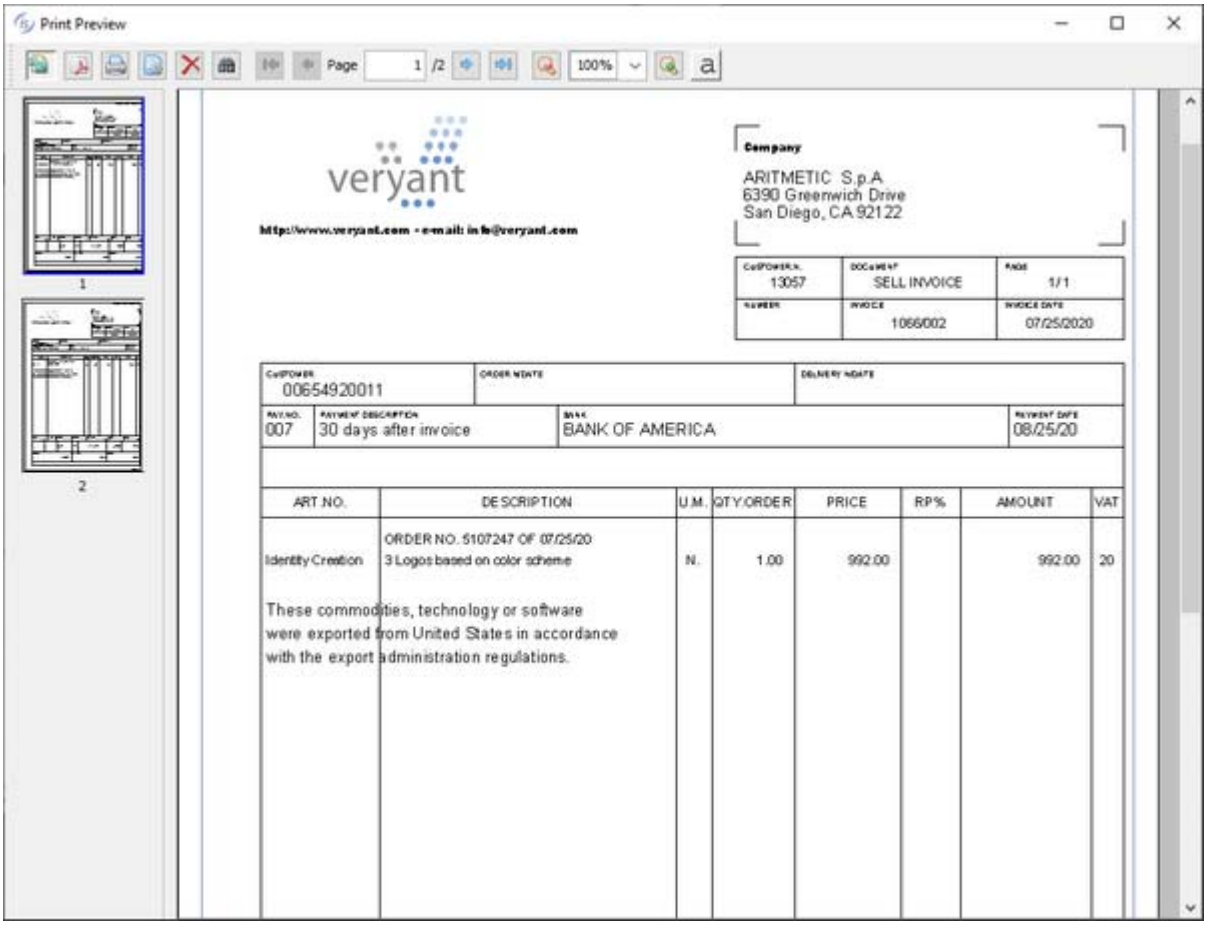
SpoolPrinter class (com.iscobol.rts.print.SpoolPrinter) and the Print Preview

The isCOBOL Framework provides the ability to obtain a print preview.

This feature is applicable to all print files. In order to obtain a print preview you need to define the file as follows:

```
SELECT print-prev ASSIGN TO PRINT "-P PREVIEW"
       ORGANIZATION LINE SEQUENTIAL.
```

The preview dialog will show up when the print file is closed.



The tool-bar on top of the dialog allows you to:

- show or hide page preview thumbnails
- save the job as PDF file
- print the job with a physical printer
- close the dialog
- find text (available also by pressing CTRL+F)
- navigate through pages (if the job contains more than one page)
- zoom in and out for better reading
- apply aliasing on the text for better reading

The dialog title can be set through the configuration property iscobol.print.preview.title.

The dialog icon can be changed through the configuration property iscobol.print.preview.icon.

The com.iscobol.rts.print.SpoolPrinter class

The print preview dialog can be customized by calling methods of the internal factory class com.iscobol.rts.print.SpoolPrinter. These methods must be called before the print file is closed. The following table shows the javadoc of the existing methods.

static String	**getDialogTitle**() Returns the title of the preview window
static javax.swing.ImageIcon	**getImageIcon**() Returns the current icon shown in the preview dialog title bar
static java.awt.Point	**getPreviewLocation**() Returns the location in pixels of the preview dialog
static double	**getPreviewScale**() Returns the current scale
static java.awt.Dimension	**getPreviewSize**() Returns the size in pixels of the preview dialog
static java.awt.Color	**getPrintableAreaBoxColor**() Returns the color of the frame in which the preview is displayed
static String	**getSaveDefaultDirectory**() Returns the directory proposed by the Save As dialog
static String	**getSaveDefaultFilename**() Returns the file name proposed by the Save As dialog
static boolean	**isPreviewAliasing**() Returns the status of the Aliasing button
static boolean	**isPreviewMaximized**() Returns if the preview dialog is maximized
static boolean	**isShowPrintButton**() Returns if the Print button is visible or not in the tool-bar
static boolean	**isShowPrintDialog**() Returns if the *Printer Setup* dialog must be shown before printing
static boolean	**isShowPrintSetupButton**() Returns if the Print Setup button is visible or not in the tool-bar
static boolean	**isShowSaveButton**() Returns if the Save button is visible or not in the tool-bar
static boolean	**isShowThumbnailsButton**() Returns if the Thumbnails button is visible or not in the tool-bar
static void	**setDialogTitle**(String *title*) Sets the title of the preview window
static void	**setImageIcon**(javax.swing.ImageIcon *icon*) Sets the icon to show in the preview dialog title bar
static void	**setPreviewAliasing**(boolean *aliasing*) Sets the status of the Aliasing button

static void **setPreviewLocation**(int *x*, int *y*)
Sets the location in pixels of the preview dialog

static void **setPreviewMaximized**(boolean *maximize*)
Choose if the preview dialog must be maximized or not

static void **setPreviewMaximized**(boolean *maximize*)
Choose if the preview dialog must be maximized or not

static void **setPreviewScale**(double *scale*)
Sets the current scale

static void **setPreviewSize**(int *width*, int *height*)
Sets the size in pixels for the print preview dialog

static void **setPrintableAreaBoxColor**(java.awt.Color *color*)
Sets the color of the frame in which the preview is displayed

static void **setSaveDefaultDirectory**(String *path*)
Sets the directory proposed by the Save As dialog

static void **setSaveDefaultFilename**(String *name*)
Sets the file name proposed by the Save As dialog

static void **setShowPrintButton**(boolean *showPrintButton*)
Sets the visibility of the Print button in the tool-bar

static void **setShowPrintDialog**(boolean *showPrintDialog*)
Choose if the *Printer Setup* dialog must be shown before printing

static void **setShowPrintSetupButton**(boolean *showPrintSetupButton*)
Sets the visibility of the Print Setup button in the tool-bar

static void **setShowSaveButton**(boolean *showSaveButton*)
Sets the visibility of the Save button in the tool-bar

static void **setShowThumbnailsButton**(boolean *showThumbnailsButton*)
Sets the visibility of the Thumbnails button in the tool-bar

The following sample shows how to make the preview dialog appear with the zoom set to 50%:

```
CONFIGURATION SECTION.
REPOSITORY.
    class is-spoolprinter as "com.iscobol.rts.print.SpoolPrinter"
                .
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    select print-job assign to printer "-P PREVIEW"
           organization line sequential.

FILE SECTION.
FD  print-job.
01  print-record pic x(83).

PROCEDURE DIVISION.
main.
    open output print-job.
    write print-record from "test zoom".
    is-spoolprinter:>setPreviewScale(50 as double).
    close print-job.
    goback.
```

Thin Client

In a Thin Client environment, print jobs are managed client side so the print preview dialog must be configured and invoked client side. In order to invoke com.iscobol.rts.print.SpoolPrinter methods client side, you can rely on the CobolGUIJavaBean Class (com.iscobol.gui.server.CobolGUIJavaBean).

The following sample shows how to make the preview dialog appear client side with the zoom set to 50%:

```
CONFIGURATION SECTION.
REPOSITORY.
    class is-java-bean as "com.iscobol.gui.server.CobolGUIJavaBean"
    class j-double     as "java.lang.Double"
                .
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    select print-job assign to printer "-P PREVIEW"
           organization line sequential.

FILE SECTION.
FD  print-job.
01  print-record pic x(83).

PROCEDURE DIVISION.
main.
    open output print-job.
    write print-record from "test zoom".
    is-java-bean:>callStaticMethod("com.iscobol.rts.print.SpoolPrinter",
"setPreviewScale", j-double:>new(50 as double)).
    close print-job.
    goback.
```

Note - The com.iscobol.preview.PreviewDialogSettings JavaBean has been deprecated. It's still supported for backward compatibility but it lacks the latest methods implemented in com.iscobol.rts.print.SpoolPrinter.

StoreProcedure Class (com.iscobol.lib.StoreProcedure)

The StoreProcedure class allows to call remote COBOL subroutines in a File Server environment.

Calling remote COBOL subroutines is permitted only after the connection to the File Server has been enstablished, that means after opening the first remote file.

See isCOBOL File Server for details.

call

Returns an object of class StoreProcedure ready to call the remote subroutine whose name is supplied as a parameter.

General format

```
static StoreProcedure call (name)
```

Syntax rules

1. *name* is an alphanumeric data item or literal
2. *StoreProcedure* is an object reference to com.iscobol.lib.StoreProcedure

end

Run the call.

General format

```
int end ()
```

General rules

1. If the call is successful it returns the return code of the called subroutine; if the call fails then it returns one of the following conventional codes:

-1000	No connection with a File Server is available
-1001	The invoked subroutine is not available
-1002	The call raised an exception on the server

input

Defines an input parameter for the subroutine.

General format

```
StoreProcedure input (parm)
```

Syntax rules

1. *parm* is any COBOL data item

2. *StoreProcedure* is an object reference to com.iscobol.lib.StoreProcedure

General rules

3. An input parameter can be read but not modified by the COBOL subroutine.

inout

Defines an input-output parameter forthe subroutine.

General format

```
StoreProcedure inout (parm)
```

Syntax rules

1. *parm* is any COBOL data item
2. *StoreProcedure* is an object reference to com.iscobol.lib.StoreProcedure

General rules

3. An input-output parameter can be read and modified by the COBOL subroutine.

output

Defines an output parameter for the subroutine.

General format

```
StoreProcedure output (parm)
```

Syntax rules

1. *parm* is any COBOL data item
2. *StoreProcedure* is an object reference to com.iscobol.lib.StoreProcedure

General rules

An output parameter can be modified but not read by the COBOL subroutine.

XMLStream Class (com.iscobol.rts.XMLStream)

The XMLStream is an internal class that allows XML files or streams to be easily read and written.

Constructor

Creates a new instance of the XMLStream class.

General format

```
XMLStream ( Record-Definition )
```

Syntax rules

1. *Record-Definition* is a level 01 group data item for which the IS IDENTIFIED clause has been specified.

General rules

1. When a new instance of XMLStream is created, the data item identified by *Record-Definition* is associated with the new object.

Code example

```
...
configuration section.
repository.
    class xmlStream  as "com.iscobol.rts.XMLStream"
    .
...
working-storage section.
...
77 objXmlStream object reference xmlStream.
01 Record-Definition identified by "Record-Definition".
   (XML fields and/or attributes)
...
procedure division.
...
set objXmlStream to xmlStream:>new(Record-Definition)
...
```

Note: *Record-Definition* can be obtained with the help of STREAM2WRK utility.

getPrintWriter

Returns the PrintWriter associated to the XMLStream object.

General format

```
java.io.PrintWriter getPrintWriter ()
```

General rules

1. A java.io.PrintWriter object is returned. You can use this method to add text to the content generated by other XMLStream write methods. See the below example, that adds a line specifying the XML stylesheet.

Code example

```
...
configuration section.
repository.
    class xmlStream  as "com.iscobol.rts.XMLStream"
    .
...
working-storage section.
...
77 objXmlStream object reference xmlStream.
...
procedure division.
...
    objXmlStream:>getPrintWriter:>println
                ('<?xml-stylesheet type="text/xsl" href="custom.xsl"?>'
                 as string).
```

read

Reads an XML stream or file and puts its content in the data item associated with the XMLStream object.

General format

```
void read ( Xml-Source )
```

Syntax rules

1. *Xml-Source* can be either a Data Item or a Nonnumeric Literal.

General rules

1. *Xml-Source* refers to a regular disk file.
2. When the read method is invoked, the whole content of *Xml-Source* is read and the data item associated with the XMLStream object is updated.

   NOTE - If the number of occurrences of an XML field is not known at the time the program is written, specifying the `DYNAMIC` phrase of the OCCURS clause in the data item associated with the XMLStream object is recommended.

Code example

```
...
configuration section.
repository.
    class xmlStream  as "com.iscobol.rts.XMLStream"
    .
...
working-storage section.
...
77 objXmlStream object reference xmlStream.
...
procedure division.
...
objXmlStream:>read("Sample.xml")
...
```

readFromFile

Reads an XML file and puts its content in the data item associated with the XMLStream object.

General format

void **readFromFile** (Xml-Source)

Syntax rules

1. *Xml-Source* can be either a Data Item or a Nonnumeric Literal.

General rules

1. *Xml-Source* refers to a regular disk file.
2. When the read method is invoked, the whole content of *Xml-Source* is read and the data item associated with the XMLStream object is updated.

   NOTE - If the number of occurrences of an XML field is not known at the time the program is written, specifying the `DYNAMIC` phrase of the OCCURS clause in the data item associated with the XMLStream object is recommended.

Code example

```
...
configuration section.
repository.
    class xmlStream  as "com.iscobol.rts.XMLStream"
    .
...
working-storage section.
...
77 objXmlStream object reference xmlStream.
...
procedure division.
...
objXmlStream:>readFromFile("Sample.xml")
...
```

readFromString

Reads an XML stream and puts its content in the data item associated with the XMLStream object.

Format 1

void **readFromString** (Xml-Source)

Format 2

void **readFromString** (Xml-Source, encoding)

Syntax rules

1. *Xml-Source* is an object reference to java.lang.String.
2. *encoding* is a string literal or data item that specified the character set to be used while parsing the XML stream. It accepts the same values as the iscobol.encoding * configuration property.

General rules

1. When the read method is invoked, the whole content of *Xml-Source* is read and the data item associated with the XMLStream object is updated.

   NOTE - If the number of occurrences of an XML field is not known at the time the program is written, specifying the `DYNAMIC` phrase of the OCCURS clause in the data item associated with the XMLStream object is recommended.

Code example

```
...
configuration section.
repository.
    class xmlStream  as "com.iscobol.rts.XMLStream"
    class JString    as "java.lang.String"
    .
...
working-storage section.
...
77 objXmlStream object reference xmlStream.
77 objString    object reference JString.
...
procedure division.
...
objXmlStream:>readFromString(objString)
...
```

readFromStream

Reads an XML stream and puts its content in the data item associated with the XMLStream object.

General format

void **readFromStream** (Xml-Source)

Syntax rules

1. *Xml-Source* is an object reference to java.io.InputStream.

General rules

1. When the read method is invoked, the whole content of *Xml-Source* is read and the data item associated with the XMLStream object is updated.

   NOTE - If the number of occurrences of an XML field is not known at the time the program is written, specifying the `DYNAMIC` phrase of the OCCURS clause in the data item associated with the XMLStream object is recommended.

Code example

```
...
configuration section.
repository.
    class xmlStream  as "com.iscobol.rts.XMLStream"
    class inpStream  as "java.io.InputStream"
    .
...
working-storage section.
...
77 objXmlStream object reference xmlStream.
77 objInpStream object reference inpStream.
...
procedure division.
...
objXmlStream:>readFromStream(objInpStream)
...
```

setOutputProperty

Puts the content of the data item associated with an XMLStream object into an XML stream or file.

General format

void **setOutputProperty** (Key-Name, Key-Value)

Syntax rules

1. *Key-Name* is a field of the java class javax.xml.transform.OutputKeys. Please refer to the javax.xml.transform.OutputKeys api reference for further details. At the time this document is written, the following fields are available:

   - CDATA_SECTION_ELEMENTS
   - DOCTYPE_PUBLIC
   - DOCTYPE_SYSTEM
   - ENCODING
   - INDENT
   - MEDIA_TYPE
   - METHOD
   - OMIT_XML_DECLARATION
   - STANDALONE
   - VERSION

Note: The ENCODING property specifies the encoding that must be used to read the XML file. This information is for the programs that will read the file and doesn't affect the way the COBOL program writes data into it. To write data with a particular encoding into the XML file, you need to set the file.encoding Java property. For example, in order to make TEST_PROG write UTF-8 data into the XML file, you will launch it in this way:

```
iscrun -J-Dfile.encoding=UTF-8 TEST_PROG
```

Code example

```
...
configuration section.
repository.
    class xmlStream  as "com.iscobol.rts.XMLStream"
    class outputKeys as "javax.xml.transform.OutputKeys"
    .
...
working-storage section.
...
77 objXmlStream object reference xmlStream.
...
procedure division.
...
objXmlStream:>setOutputProperty (outputKeys:>ENCODING "ISO-8859-1")
...
```

write

Puts the content of the data item associated with an XMLStream object into an XML stream or file.

Format 1

```
void write ( Xml-Destination )
```

Format 2

```
void write ( Xml-Destination,  writeQualifiedTagNames )
```

Syntax rules

1. *Xml-Destination* can be either a Data Item or a Nonnumeric Literal.
2. *writeQualifiedTagNames* is an alphanumeric data item or literal hosting a boolean value (e.g. "0", "1", "true", "false", "yes", "no", "on" or "off").

General rules

1. When the write method is invoked, the whole content of the data item associated with the XMLStream object is put into *Xml-Destination*.

   NOTE - If the number of occurrences of an XML fields is not known at the time the program is written, specifying the `DYNAMIC` phrase of the OCCURS clause is recommended.
2. The XMLStream write method will indent the output according to the value of the property iscobol.xmlstream.indent_number. The default is -1 which disables indentation. For example, to indent each level 2 columns set this property as follows:

```
iscobol.XMLStream.indent_number=2
```

3. To omit empty elements from the XML output, set the property iscobol.xmlstream.omit_empty_elements (boolean) to true. The default value of this property is false, that means empty elements are generated.
4. If *writeQualifiedTagNames* is TRUE, namespaces as generated as prefix of the element name, in the form "ns#"

where # is a progressive number. If *writeQualifiedTagNames* is FALSE or omitted, namespaces are generated as attribute of the element, in the form "xmlns=url".

Code example

```
...
configuration section.
repository.
    class xmlStream  as "com.iscobol.rts.XMLStream"
    .
...
working-storage section.
...
77 objXmlStream object reference xmlStream.
...
procedure division.
...
objXmlStream:>write("Sample.xml")
...
```

writeToFile

Puts the content of the data item associated with an XMLStream object into an XML file.

Format 1

```
void writeToFile ( Xml-Destination )
```

Format 2

```
void writeToFile ( Xml-Destination,  writeQualifiedTagNames )
```

Syntax rules

1. *Xml-Destination* can be either a Data Item or a Nonnumeric Literal.
2. *writeQualifiedTagNames* is an alphanumeric data item or literal hosting a boolean value (e.g. "0", "1", "true", "false", "yes", "no", "on" or "off").

General rules

1. *Xml-Destination* refers to a regular disk file.
2. When the write method is invoked, the whole content of the data item associated with the XMLStream object is put into *Xml-Destination*.

   NOTE - If the number of occurrences of an XML fields is not known at the time the program is written, specifying the `DYNAMIC` phrase of the OCCURS clause is recommended.
3. The XMLStream write method will indent the output according to the value of the property iscobol.xmlstream.indent_number. The default is -1 which disables indentation. For example, to indent each level 2 columns set this property as follows:

```
iscobol.XMLStream.indent_number=2
```

4. To omit empty elements from the XML output, set the property iscobol.xmlstream.omit_empty_elements

(boolean) to true. The default value of this property is false, that means empty elements are generated.

5. If *writeQualifiedTagNames* is TRUE, namespaces as generated as prefix of the element name, in the form "ns#" where # is a progressive number. If *writeQualifiedTagNames* is FALSE or omitted, namespaces are generated as attribute of the element, in the form "xmlns=url".

Code example

```
...
configuration section.
repository.
    class xmlStream  as "com.iscobol.rts.XMLStream"
    .
...
working-storage section.
...
77 objXmlStream object reference xmlStream.
...
procedure division.
...
objXmlStream:>writeToFile("Sample.xml")
...
```

writeToPrintWriter

Puts the content of the data item associated with an XMLStream object into a PrintWriter object.

Format 1

```
void writeToPrintWriter ( Xml-Destination )
```

Format 2

```
void writeToPrintWriter ( Xml-Destination,  writeQualifiedTagNames )
```

Syntax rules

1. *Xml-Destination* is an object reference to java.io.PrintWriter.
2. *writeQualifiedTagNames* is an alphanumeric data item or literal hosting a boolean value (e.g. "0", "1", "true", "false", "yes", "no", "on" or "off").

General rules

1. When the write method is invoked, the whole content of the data item associated with the XMLStream object is put into *Xml-Destination*.

   NOTE - If the number of occurrences of an XML fields is not known at the time the program is written, specifying the `DYNAMIC` phrase of the OCCURS clause is recommended.

2. The XMLStream write method will indent the output according to the value of the property iscobol.xmlstream.indent_number. The default is -1 which disables indentation. For example, to indent each level 2 columns set this property as follows:

```
iscobol.XMLStream.indent_number=2
```

3. To omit empty elements from the XML output, set the property iscobol.xmlstream.omit_empty_elements (boolean) to true. The default value of this property is false, that means empty elements are generated.
4. If *writeQualifiedTagNames* is TRUE, namespaces as generated as prefix of the element name, in the form "ns#" where # is a progressive number. If *writeQualifiedTagNames* is FALSE or omitted, namespaces are generated as attribute of the element, in the form "xmlns=url".

Code example

```
...
configuration section.
repository.
    class xmlStream  as "com.iscobol.rts.XMLStream"
    .
...
working-storage section.
...
77 objXmlStream object reference xmlStream.
...
procedure division.
...
objXmlStream:>writeToPrintWriter(objXmlStream:>getPrintWriter)
...
```

writeToStream

Puts the content of the data item associated with an XMLStream object into an OutputStream object.

Format 1

```
void writeToStream ( Xml-Destination )
```

Format 2

```
void writeToStream ( Xml-Destination,  writeQualifiedTagNames )
```

Syntax rules

1. *Xml-Destination* is an object reference to java.io.OutputStream.
2. *writeQualifiedTagNames* is an alphanumeric data item or literal hosting a boolean value (e.g. "0", "1", "true", "false", "yes", "no", "on" or "off").

General rules

1. When the write method is invoked, the whole content of the data item associated with the XMLStream object is put into *Xml-Destination*.

   NOTE - If the number of occurrences of an XML fields is not known at the time the program is written, specifying the `DYNAMIC` phrase of the OCCURS clause is recommended.
2. The XMLStream write method will indent the output according to the value of the property iscobol.xmlstream.indent_number. The default is -1 which disables indentation. For example, to indent each level 2 columns set this property as follows:

```
iscobol.XMLStream.indent_number=2
```

3. To omit empty elements from the XML output, set the property iscobol.xmlstream.omit_empty_elements (boolean) to true. The default value of this property is false, that means empty elements are generated.
4. If *writeQualifiedTagNames* is TRUE, namespaces as generated as prefix of the element name, in the form "ns#" where # is a progressive number. If *writeQualifiedTagNames* is FALSE or omitted, namespaces are generated as attribute of the element, in the form "xmlns=url".

Code example

```
...
configuration section.
repository.
    class xmlStream  as "com.iscobol.rts.XMLStream"
    class outStream  as "java.io.OutputStream"
    .
...
working-storage section.
...
77 objXmlStream object reference xmlStream.
77 objOutStream object reference outStream.
...
procedure division.
...
set objOutStream to outStream:>new()
objXmlStream:>writeToStream(objOutStream)
...
```

writeToStringBuffer

Puts the content of the data item associated with an XMLStream object into an StringBuffer object.

Format 1

void **writeToStringBuffer** (Xml-Destination)

Format 2

void **writeToStringBuffer** (Xml-Destination, writeQualifiedTagNames)

Syntax rules

1. *Xml-Destination* is an object reference to java.lang.StringBuffer.
2. *writeQualifiedTagNames* is an alphanumeric data item or literal hosting a boolean value (e.g. "0", "1", "true", "false", "yes", "no", "on" or "off").

General rules

1. When the write method is invoked, the whole content of the data item associated with the XMLStream object is put into *Xml-Destination*.

   NOTE - If the number of occurrences of an XML fields is not known at the time the program is written, specifying the `DYNAMIC` phrase of the OCCURS clause is recommended.
2. The XMLStream write method will indent the output according to the value of the property

iscobol.xmlstream.indent_number. The default is -1 which disables indentation. For example, to indent each level 2 columns set this property as follows:

```
iscobol.XMLStream.indent_number=2
```

3. To omit empty elements from the XML output, set the property iscobol.xmlstream.omit_empty_elements (boolean) to true. The default value of this property is false, that means empty elements are generated.
4. If *writeQualifiedTagNames* is TRUE, namespaces as generated as prefix of the element name, in the form "ns#" where # is a progressive number. If *writeQualifiedTagNames* is FALSE or omitted, namespaces are generated as attribute of the element, in the form "xmlns=url".

Code example

```
...
configuration section.
repository.
    class xmlStream  as "com.iscobol.rts.XMLStream"
    class strBuffer  as "java.lang.StringBuffer"
    .
...
working-storage section.
...
77 objXmlStream object reference xmlStream.
77 objStrBuffer object reference strBuffer.
...
procedure division.
...
set objStrBuffer to strBuffer:>new()
objXmlStream:>writeToStringBuffer(objStrBuffer)
...
```

Usage example

This small program reads an XML file representing an RSS feed, adds an item and creates a new XML file.

Example XML (Rss.xml):

```
    <rss version="2.0">
      <channel>
        <title>Liftoff News</title>
        <link> http://liftoff.msfc.nasa.gov/</link>
        <description>Liftoff to Space Exploration.</description>
        <language>en-us</language>
        <pubDate>Tue, 10 Jun 2003 04:00:00 GMT</pubDate>
        <lastBuildDate>Tue, 10 Jun 2003 09:41:01 GMT</lastBuildDate>
        <docs> http://blogs.law.harvard.edu/tech/rss</docs>
        <generator>Weblog Editor 2.0</generator>
        <managingEditor>editor@example.com</managingEditor>
        <webMaster>webmaster@example.com</webMaster>

        <item>
          <title>Star City</title>
          <link> http://liftoff.msfc.nasa.gov/news/2003/news-starcity.asp </link>
          <description>How do Americans get ready to work with Russians aboard the
            International Space Station? They take a crash course in culture, language
            and protocol at Russia's Star City.</description>
          <pubDate>Tue, 03 Jun 2003 09:39:21 GMT</pubDate>
          <guid> http://liftoff.msfc.nasa.gov/2003/06/03.html#item573</guid>
        </item>

        <item>
          <title>Space Exploration</title>
          <link> http://liftoff.msfc.nasa.gov/</link>
          <description>Sky watchers in Europe, Asia, and parts of Alaska and Canada
            will experience a partial eclipse of the Sun on Saturday, May 31st.</
description>
          <pubDate>Fri, 30 May 2003 11:06:42 GMT</pubDate>
          <guid> http://liftoff.msfc.nasa.gov/2003/05/30.html#item572</guid>
        </item>

        <item>
          <title>The Engine That Does More</title>
          <link> http://liftoff.msfc.nasa.gov/news/2003/news-VASIMR.asp </link>
          <description>Before man travels to Mars, NASA hopes to design new engines
            that will let us fly through the Solar System more quickly.  The proposed
            VASIMR engine would do that.</description>
          <pubDate>Tue, 27 May 2003 08:37:32 GMT</pubDate>
          <guid> http://liftoff.msfc.nasa.gov/2003/05/27.html#item571</guid>
        </item>

        <item>
          <title>Astronauts' Dirty Laundry</title>
          <link> http://liftoff.msfc.nasa.gov/news/2003/news-laundry.asp </link>
          <description>Compared to earlier spacecraft, the International Space
            Station has many luxuries, but laundry facilities are not one of them.
            Instead, astronauts have other options.</description>
          <pubDate>Tue, 20 May 2003 08:56:02 GMT</pubDate>
          <guid> http://liftoff.msfc.nasa.gov/2003/05/20.html#item570</guid>
        </item>
      </channel>
    </rss>
```

Copybook generated by STREAM2WRK (RSS.wrk):

```
          >>SOURCE FORMAT FREE
    *> XML File: RSS.xml
    01 rss identified by "rss".
       03 attr-version identified by "version" is attribute pic x any length.
       03 channel identified by "channel".
          05 title identified by "title" .
             07 title-data pic x any length.
          05 link identified by "link".
             07 link-data pic x any length.
          05 description identified by "description".
             07 description-data pic x any length.
          05 language identified by "language" .
             07 language-data pic x any length.
          05 pubDate identified by "pubDate" .
             07 pubDate-data pic x any length.
          05 lastBuildDate identified by "lastBuildDate" .
             07 lastBuildDate-data pic x any length.
          05 docs identified by "docs" .
             07 docs-data pic x any length.
          05 generator identified by "generator".
             07 generator-data pic x any length.
          05 managingEditor identified by "managingEditor".
             07 managingEditor-data pic x any length.
          05 webMaster identified by "webMaster".
             07 webMaster-data pic x any length.
          05 item identified by "item" occurs dynamic capacity item-count.
             07 title identified by "title" .
                09 title-data pic x any length.
             07 link identified by "link" .
                09 link-data pic x any length.
             07 description identified by "description" .
                09 description-data pic x any length.
             07 pubDate identified by "pubDate" .
                09 pubDate-data pic x any length.
             07 guid identified by "guid" .
                09 guid-data pic x any length.
           >>SOURCE FORMAT PREVIOUS
```

Example application:

```
program-id. ReadWriteRSS.
configuration section.
repository.
    class xmlStream  as "com.iscobol.rts.XMLStream"
    .

working-storage section.
77  objXmlStream object reference xmlStream.

copy "RSS.wrk".

procedure division.
main.

    set objXmlStream to xmlStream:>new(rss).

    objXmlStream:>read ("RSS.xml").

    display message box "RSS.xml contains " item-count " items.".

    add 1 to item-count.
    move "New title" to title-data of item(item-count).
    move "New link" to link-data of item(item-count).
    move "New description" to description-data of item(item-count).
    move "New pubDate" to pubDate-data of item(item-count).
    move "New guid" to guid-data of item(item-count).

    objXmlStream:>write ("RSS-New.xml").

    initialize rss.

    objXmlStream:>read ("RSS-New.xml").

    display message box "RSS-New.xml contains " item-count " items.".

    goback.
```