PSS®E 33.4

Release 18:
Saved Case Data Extraction Subroutines

March, 2013

SIEMENS

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

SIEMENS

Table of Contents

Chapter 1 - General Information 1-1

1.1 Overview 1-1

1.2 How to Use the Saved Case Data Extraction Subroutines 1-1

1.3 Restrictions 1-2

Chapter 2 - Subroutine Definitions 2-1

2.1 PSSOPN – Open Case 2-1
2.1.1 About Unit Numbers 2-2

2.2 PSSCLS – Close Case 2-2

2.3 PSSREV – Inquire Release Number 2-2

2.4 PSSSIZ – Get Case Sizes 2-3

2.5 PSSMSC – Retrieve Miscellaneous Data 2-5
2.5.1 Converted Loads 2-5

2.6 PSSTIT – Retrieve Headings and Title Data 2-5

2.7 PSSBUS – Retrieve Bus Data 2-6
2.7.1 Older Cases 2-7
2.7.2 Usage Notes 2-7

2.8 PSSLOD – Retrieve Load Data 2-7
2.8.1 Older Cases 2-8
2.8.2 Usage Notes 2-9

2.9 PSSFSH – Retrieve Fixed Shunt Data 2-9
2.9.1 Older Cases 2-9

2.10 PSSGEN – Retrieve Generator Data 2-9
2.10.1 Older Cases 2-11

2.11 PSSBRN – Retrieve Branch Data 2-11
2.11.1 Older Cases 2-13
2.11.2 Usage Notes 2-13

2.12 PSSTRN – Retrieve Two-Winding Transformer Data 2-14
2.12.1 Older Cases 2-15
2.12.2 Usage Notes 2-16

2.13 PSS3WT - Retrieve Three-Winding Transformer Data 2-16
2.13.1 Usage Notes 2-17

2.14 PSS3IX – Build Indices to Three-Winding Transformer Data 2-17
2.14.1 Usage Notes 2-18

2.15 PSSAIN – Retrieve Area Interchange Data 2-18
2.15.1 Older Cases 2-19

2.16 PSS2DC – Retrieve Two-Terminal Transmission Line Data 2-19
2.16.1 Older Cases 2-22
2.17 PSSVSC - Retrieve Voltage Source Converter dc Line Data 2-22
2.18 PSSWSH – Retrieve Switched Shunt Data 2-24
2.18.1 Older Cases 2-25
2.19 PSSTIC – Retrieve Transformer Impedance Correction Tables 2-25
2.19.1 Usage Notes 2-26
2.20 PSSMDC – Retrieve Multiterminal dc Transmission Line Data 2-27
2.20.1 Older Cases 2-30
2.20.2 Usage Notes 2-30
2.21 PSSMSL – Retrieve Multisection Line Data 2-30
2.21.1 Older Cases 2-31
2.21.2 Usage Notes 2-31
2.22 PSSZNM – Retrieve Zone Names 2-31
2.22.1 Older Cases 2-32
2.23 PSSATR – Retrieve Area Transaction Data 2-32
2.24 PSSOWN – Retrieve Owner Names 2-33
2.24.1 Older Cases 2-33
2.25 PSSFCT – Retrieve FACTS Device Data 2-33
2.25.1 Older Cases 2-36
2.26 PSSALD – Retrieve Nominal Values of Adjusted Load Data 2-36
2.26.1 Usage Notes 2-37
2.27 PSSABX – Retrieve Nominal Values of Adjusted Branch Data 2-38
2.27.1 Usage Notes 2-38
2.28 PSSIND - Retrieve Induction Machine Data 2-38

Access to the USRCAS Subroutine Library A-1

A.1 IBM – PC Microsoft Windows 2000 / Windows XP A-1

Extracting Saved Case Data in Python B-1

B.1 Introduction B-1
B.2 Access Procedure B-1
B.3 Usage Notes B-3

SIEMENS

Chapter 1

General Information

1.1 Overview

The Saved Case Data Extraction Subroutines have been developed to allow the user to write programs that access data in PSS®E saved cases directly. These subroutines are described individually later in this manual. Each subroutine retrieves data of a specific type, and a call would be made to each subroutine corresponding to a type of data which is desired by the user's program.

1.2 How to Use the Saved Case Data Extraction Subroutines

The first subroutine called must always be PSSOPN or PSSREV. All other subroutines in this package will not execute if the saved case file has not been opened. PSSCLS must be called before PSSOPN can be called a second time, i.e., only one saved case file may be accessed at a time. It is recommended that PSSSIZ be called next. This will return the number of the various components in the saved case, e.g., number of buses, branches, etc. More importantly, this will indicate the minimum dimensions of the arrays that must be used to contain the data retrieved by subsequent calls and the actual amount of data returned. Where the dimensions of the calling arguments exceed the necessary amount, the values of the additional array elements are unpredictable, and it is strongly recommended that user programs not be written in such a way as to depend on such values, e.g., bus number NBUS+1, where the case contains NBUS buses, will have an unpredictable value. The order in which the data is retrieved, i.e., whether PSSBUS, for example, is called before or after PSSBRN, is completely insignificant, with the exception of PSS3IX. PSSOPN and PSSCLS are the only subroutines that must be called, all others are optional (PSSSIZ and PSSMSC probably should always be called). There are no data arrays that are returned by more than one subroutine with the exception of PSS3IX; the user must call each subroutine that corresponds to a type of data which is desired. Only PSSOPN and PSS3IX require input values. All the usual rules for FORTRAN subroutine calls apply as well.

When the user has completed coding, and successfully compiled the application program, the next step will be to create an executable module which will include the Saved Case Data Extraction Subroutines. This process varies between different computer systems, and may be called link, load, bind, link edit, or some other name. In each case the process is to take object or relocatable code and create an executable program. Refer to Appendix A for details on how the Saved Case Data Extraction Subroutines can be made available to such a process on your computer.

1.3 Restrictions

There are no short integer or logical variables used. Nor are there any double precision variables used. All arguments are default size INTEGER, LOGICAL, REAL, or COMPLEX type, or default CHARACTER with the length indicated.

Only one (1) saved case may be accessed at a time.

Prior to release 4.0, these subroutines could only access saved cases created by the current release of PSS®E. Beginning with release 4.0, saved cases created by PSS®E-16 or later may be read.

Later releases of this package may require modification of existing user programs if the structure of the saved case file changes.

Note that these subroutines use two common blocks, PSSCMN and PSSCMC, five other internal subroutines, PSSDBG, PSSRWD, PSSIXX, PSSI24 and PSSL24, as well as many routines from the PTI FORTRAN Utility Programs package (FUP - supplied with PSS®E). Application programs may not have subroutines or common blocks with any of these names.

This package will be supplied compiled with the same release of the FORTRAN compiler as the concurrent release of PSS®E. In general, all application programs will need to be compiled with that compiler.

Some arguments are used as temporary work space. It cannot be assumed, for instance, that an array can be used for more than one argument, a technique sometimes used to avoid declaring arrays for uninteresting values.

SIEMENS

Chapter 2

Subroutine Definitions

2.1 PSSOPN – Open Case

```
PSSOPN(CASNAM,LU,IERR)
```

CASNAM — Type: CHARACTER; scalar
Usage: input. Must contain the filename of the saved case to be opened.

LU — Type: INTEGER; scalar
Usage: input/output. Must contain the unit number to be used to open the file whose name is contained in the argument CASNAM. See Section 2.1.1 About Unit Numbers for information about unit numbers.

IERR — Type: INTEGER; scalar
Usage: output. Will contain the return code:

0 No error; file opened and other subroutines in this package may now be used.

1 Error opening file.

2 I/O error reading file after open.

3 File is not a PSS®E saved case file.

4 File is a PSS®E saved case file, but of an older release.
Use PSS®E to read case and resave in the current format.

5 There is already a case open.

6 Any other error.

11 Assertion error: size parameter.

See note below regarding negative values.

PSSOPN must be the first call made for each saved case file to be accessed. Only one case may be accessed at a time and a call to PSSCLS must be made before PSSOPN may be called again. Note that both CASNAM and LU are *input* to this routine. It is the user's responsibility that LU be a valid available unit number, and to verify the return code. The form of the filename contained in CASNAM must follow the same rules as saved case names for PSS®E on that host computer system. Normally, when IERR is not

zero, the file CASNAM will not be open. If an error was detected after the file was opened, and then PSSOPN was unable to close this file, the above error codes will be returned as negative numbers (e.g., -3 means the file was not a PSS®E saved case and that an error occurred while attempting to close the file).

2.1.1 About Unit Numbers

Valid values for unit numbers vary for different computers, and in some cases for different compilers. In our experience, numbers in the range 10-40 tend to work well. If the application using these subroutines does not open any other files, a zero may be used and PSSOPN will select a valid unit number, and return the value used in LU. On some computers PSSOPN can determine which units are in use and therefore LU=0 can always be used. Refer to your FORTRAN documentation for more information about unit numbers.

2.2 PSSCLS – Close Case

```
PSSCLS(IERR)
```

IERR — Type: INTEGER; scalar
Will contain the return code:

0 No error; file has been closed.

1 Case not open.

2 Error closing file.

PSSCLS must be called after all processing of the saved case opened by PSSOPN has been completed. The PSSCLS can only close the file opened by the last execution of PSSOPN, so the only argument is the return code. It is the user's responsibility to verify the return code. Note that after PSSOPN has successfully opened a saved case, PSSCLS must be called before PSSOPN may be called again to open another case. After PSSCLS is called, all the other Case Data Extraction Subroutines will consider the saved case file to be closed, even if PSSCLS returns with IERR equal to 2.

2.3 PSSREV – Inquire Release Number

```
PSSREV(PNAME, MAJREL, MINREL, MODREL, DATE, IBUF)
```

PNAME — Type: CHARACTER *8; scalar
Will return the string 'UserCase'.

MAJREL — Type: INTEGER; scalar
Will return the major release number.

MINREL — Type: INTEGER; scalar
Will return the minor release number.

MODREL — Type: INTEGER; scalar
Will return the modification level.

DATE Type: CHARACTER *23; scalar
Will return the release date.

IBUF Type: CHARACTER *80; scalar
Not in use; will return spaces.

2.4 PSSSIZ – Get Case Sizes

```
PSSSIZ(NBUS,NLOAD,NBUSHN,NGEN,NLIN,NTRFMR,N3WNDT,NSHUNT,NDCL,
NVSC,NMTDCL,NMSLIN,NSECTN,NTRNAC,NFACTS,NWNDMC,NAREAS,NZONES,
NOWNRS,NINDMC,NGNES,MAXARE,MAXZNM,MAXTIC,MAXOWN,ADJLOD, ADJ-
BRN,IERR)
```

All variables are type INTEGER; scalar.

NBUS Number of buses.

LOAD Number of loads.

NBUSHN Number of fixed shunts.

NGEN Number of generators (machines).

NLIN Number of lines (branches and transformers).

NTRFMR Number of two-winding transformers.

N3WNDT Number of three-winding transformers.

NSHUNT Number of switched shunts.

NDCL Number of two-terminal dc lines.

NVSC Number of Voltage Source Converter dc lines.

NMTDCL Number of multiterminal dc lines.

NMSLIN Number of multisection line groupings.

NSECTN Number of multisection line sections (total).

NTRNAC Number of area transactions.

NFACTS Number of FACTS devices.

NWNDMC Number of wind machines.

NAREAS Number of areas.

NZONES Number of zones.

NOWNRS Number of owners.

NINDMC Number of induction machines.

NGNES Number of genetic network elements.

MAXARE Maximum area number in the saved case.

MAXZNM Maximum zone number in the saved case.

MAXTIC Maximum transformer impedance correction table number in the saved case.

MAX2DC Maximum two-terminal dc line number in older saved cases (equal to NDCL in current saved cases).

MAXMDC Maximum multi-terminal dc line number in older saved cases (equal to NMTDCL in current saved cases).

MAXOWN Maximum owner number in the saved case.

MAXFCT Maximum owner number in older saved cases (equal to NFACTS in current saved cases).

ADJLOD Number of loads with a load adjustment table assigned.

ADJBRN Number of branches with a branch adjustment table assigned.

IERR Will contain the return code:

0 No error.

1 Case not open.

2 Error detected, values returned are undependable.

Most of the above values correspond to the necessary dimensions for the arrays used by other Case Data Extraction Subroutines that can vary from case to case. Referring to data retrieval subroutines (other than PSSSIZ), some arrays have variable dimensions (e.g., NBUS), and some arrays have fixed dimensions. For those arrays that have variable length dimensions, only that many (the number returned by PSSSIZ) values will be returned. For those arrays that have fixed dimensions, that fixed number of values will be returned regardless of how many real data items were read into PSS®E for that case.
Some of the above values are maximums, e.g. MAXARE. NAREAS will be the number of areas defined in the saved case (and the number of values returned in the arrays by PSSAIN) and MAXARE is the largest numeric value of any area number. In some cases a maximum value is used for dimensioning. The values above still indicate the number of data items to be returned.
NBUS includes the hidden star buses created for three-winding transformers (NBUS = number of raw data buses + number of three-winding transformers).
NTRMFR includes three two-winding transformer records representing each three-winding transformer (NTRMFR = number of raw data two-winding transformers + three times the number of three-winding transformers).
NLIN includes transformers as well as nontransformer branches (NLIN = number of raw nontransformer branches + NTRMFR).

2.5 PSSMSC – Retrieve Miscellaneous Data

```
PSSMSC(THRSHZ,SBASE,XFRRAT,NXFRAT,BAS-
FRQ,LCONV,GCONV,HIBUS,PSSVER,IERR)
```

THRSHZ — Type: REAL; scalar
Will contain the threshold value for zero impedance lines. Nontransformer branches with zero resistance and whose magnitude of reactance falls below this threshold are treated as zero impedance lines in PSS®E.

SBASE — Type: REAL; scalar
Will contain the system base MVA.

XFRRAT — Type: INTEGER; scalar
Will contain the units of transformer ratings.

NXFRAT — Type: INTEGER; scalar
Will contain the units of non-transformer branch ratings.

BASFRQ — Type: REAL; scalar
Will contain the system base frequency.

LCONV — Type: LOGICAL; scalar
True if loads have been converted in the saved case, false otherwise; see Section 2.5.1 Converted Loads.

GCONV — Type: LOGICAL; scalar
True if generators have been converted in the saved case, false otherwise.

HIBUS — Type: INTEGER; scalar
Highest bus number allowed for user input.

PSSVER — Type: REAL; scalar
Version number of oldest PSSE release associated with this version of the Saved Case. Form is major.minor.

IERR — Type: INTEGER; scalar
Will contain the return code:

0 No error.

1 Case not open.

2 Other error; values returned are undependable.

PSSMSC must be used if the application needs to retrieve these values.

2.5.1 Converted Loads

The concept of converted loads is obsolete and only has meaning for saved cases prior to PSS®E-24 (current cases will always return TRUE for this value); see PSSLOD.

2.6 PSSTIT – Retrieve Headings and Title Data

```
PSSTIT(HEDING,TITLE,IERR)
```

HEDING Type: CHARACTER∗60; dimension (2)
Will return the two heading lines.

TITLE Type: CHARACTER∗72; dimension (16)
Will return the long case title.

IERR Type: INTEGER; scalar
Will contain the return code:

0 No error.

1 Case not open.

2 Other error; values returned are undependable.

All this data (120 bytes for HEDING, 1152 bytes for TITLE) is returned, even if it is all or largely spaces.

2.7 PSSBUS – Retrieve Bus Data

```
PSS-
BUS(NUM,IDE,AREA,VM,VA,NAME,BASKV,ZONE,OWNER,NMAXV,NMINV,FMA
XV,EMINV,IERR)
```

NUM Type: INTEGER; dimension NBUS
Bus numbers.

IDE Type: INTEGER; dimension NBUS
Bus type code.

AREA Type: INTEGER; dimension NBUS
Area to which the bus is assigned.

VM Type: REAL; dimension NBUS
Bus voltage magnitude, in pu.

VA Type: REAL; dimension NBUS
Bus voltage angle, in degrees.

NAME Type: CHARACTER∗12; dimension NBUS
Alphanumeric bus identifier.

BASKV Type: REAL; dimension NBUS
Bus base voltage, in kV.

ZONE Type: INTEGER; dimension NBUS
Zone to which the bus is assigned.

OWNER Type: INTEGER; dimension NBUS
Owner to which the bus is assigned.

NMAXVType: REAL
Normal voltage magnitude high limit; entered in pu. NVHI=1.1 by default

NMINVType: REAL
Normal voltage magnitude low limit, entered in pu. NVLO=0.9 by default

FMAXVType: REAL
Emergency voltage magnitude high limit; entered in pu. EVHI=1.1 by default

EMINVType: REAL
Emergency voltage magnitude low limit; entered in pu. EVLO=0.9 by default

IERR — Type: INTEGER; scalar
Will contain the return code:
0 No error.
1 Case not open.
2 Other error; values returned are undependable.

The bus data is not retrieved in any particular order, i.e., there is no relationship between the buses that can be determined from their order or position in the above arrays. PSSBUS will attempt to place NBUS values in each data array; it is the users' responsibility to ensure that each array used for any argument is properly dimensioned. NBUS can be retrieved by PSSSIZ prior to calling PSSBUS. Refer to *PSS®E Program Operation Manual*, Section 5.2, Reading Power Flow Raw Data into the Working Case, for more details on the data contained in these arrays.

2.7.1 Older Cases

Cases prior to PSS®E-20 could have zone values of zero. If the case contains zone values of zero, they are set to 9999.

For cases prior to PSS®E-25, OWNER will be set to 1.

For cases prior to PSS®E-31, bus data also contained fixed shunt data. They will be made part of a fixed shunt data type. Refer PSSFSH routine.

2.7.2 Usage Notes

Cases containing three-winding transformers will have bus records added for the hidden star bus. These buses are returned by PSSBUS (and included in the count of buses, NBUS, returned by PSSSIZ). They can be identified by having a bus number in excess of the value HIBUS, which can be retrieved by PSSMSC.

2.8 PSSLOD – Retrieve Load Data

```
PSSLOD(NUM,ID,STATUS,AREA,ZONE,LOAD,OWNER,LDSCALE,LDINT,IERR)
```

NUM — Type: INTEGER; dimension NLOAD
Bus number.

ID — Type: CHARACTER∗2; dimension NLOAD
Two character load identifier.

STATUS — Type: INTEGER; dimension NLOAD
Load in-service status. One for in-service; zero for out-of-service.

AREA — Type: INTEGER; dimension NLOAD
Area to which the load is assigned.

ZONE Type: INTEGER; dimension NLOAD
Zone to which the load is assigned.

LOAD Type: REAL; dimension (6, NLOAD)
Components of each load, in pu:

1. Real power component of constant MVA load.

2. Reactive power component of constant MVA load.

3. Real power component of constant current load.

4. Reactive power component of constant current load.

5. Real power component of constant admittance load.

6. Reactive power component of constant admittance load.

OWNER Type: INTEGER; dimension NLOAD
OWNER to which the load is assigned.

LDSCALE Type: LOGICAL; dimension NLOAD
True if this load participates in scaling operations.

LDINT Type: LOGICAL; dimension NLOAD
True if this load is interruptible in contingency analysis.

IERR Type: INTEGER; scalar
Will contain the return code:

0 No error.

1 Case not open.

2 Other error; values returned are undependable.

The load data is not retrieved in any particular order, i.e., there is no relationship between the loads that can be determined from their order or position in the above arrays. PSSLOD will attempt to place NLOAD values in each data array (or 6∗NLOAD for LOAD); it is the users' responsibility to ensure that each array used for any argument is properly dimensioned. NLOAD can be retrieved by PSSSIZ prior to calling PSSLOD. Refer to *PSS®E Program Operation Manual*, Section 5.2, Reading Power Flow Raw Data into the Working Case, for more details on the data contained in these arrays.

2.8.1 Older Cases

For cases prior to PSS®E-24, ID will be blank, STATUS will be one, and AREA and ZONE will be set to the values from the bus record for bus NUM. Normally these cases had unconverted loads, i.e., all loads were represented as constant MVA loads only. PSSLOD will place those values in the constant MVA elements of the LOAD array, setting the other elements to zero. Cases with converted loads will return values for all 6 elements of LOAD for each load.

For cases prior to PSS®E-25, OWNER will be set to 1.

2.8.2 Usage Notes

The array LOAD contains the actual values used in the power flow solution, including any adjustments. For nominal values, see PSSALD.

2.9 PSSFSH – Retrieve Fixed Shunt Data

```
PSSFSH(NUM,ID,STATUS,SHUNT,IERR)
```

NUM — Type: INTEGER; dimension NBUSHN
Bus numbers.

ID — Type: CHARACTER*2; dimension NBUSHN
Two character shunt identifier.

STATUS — Type: INTEGER; dimension NBUSHN
Shunt in-service status. One for in-service; zero for out-of-service.

SHUNT — Type: COMPLEX; dimension NBUSHN
Shunt admittance to ground, in pu.

IERR — Type: INTEGER; scalar
Will contain the return code:

0 No error.

1 Case not open.

2 Other error; values returned are undependable.

The fixed shunt data is not retrieved in any particular order, i.e., there is no relationship between the buses that can be determined from their order or position in the above arrays. PSSFSH will attempt to place NBUSHN values in each data array; it is the users' responsibility to ensure that each array used for any argument is properly dimensioned. NBUSHN can be retrieved by PSSSIZ prior to calling PSSFSH. Refer to *PSS®E Program Operation Manual*, Section 5.2, Reading Power Flow Raw Data into the Working Case, for more details on the data contained in these arrays.

2.9.1 Older Cases

For cases prior to PSS®E-31, there is no data for the array's ID or STATUS. The ID is set to 1 (only one per bus was allowed) and STATUS is set according to defaults in PSS®E activity CASE.

2.10 PSSGEN – Retrieve Generator Data

```
PSSGEN(NUM,IDE,PG,QG,QT,QB,VS,IREG,MBASE,ZSORCE,XTRAN,GTAP,STAT,
RMPCT,PT,PB,OWNER,OWNPCT,WMOD,WPF,IERR)
```

NUM — Type: INTEGER; dimension NGEN
Bus number.

IDE Type: CHARACTER∗2; dimension NGEN
Single character machine identifier.

PG Type: REAL; dimension NGEN
Generator real power output, in pu.

QG Type: REAL; dimension NGEN
Generator reactive power output, in pu.

QT Type: REAL; dimension NGEN
Maximum generator reactive power output, in pu.

QB Type: REAL; dimension NGEN
Minimum generator reactive power output, in pu.

VS Type: REAL; dimension NGEN
Regulated voltage setpoint, in pu.

IREG Type: INTEGER; dimension NGEN
Bus number of a remote bus whose voltage is regulated by this plant to the value specified by VS. IREG=0 indicates that the plant regulates its own voltage.

MBASE Type: REAL; dimension NGEN
Total base MVA of the units represented by this machine.

ZSORCE Type: COMPLEX; dimension NGEN
Machine impedance, in pu, on MBASE base.

XTRAN Type: COMPLEX; dimension NGEN
Step-up transformer impedance, in pu, on MBASE base.

GTAP Type: REAL; dimension NGEN
Step-up transformer off-nominal turns ratio, in pu.

STAT Type: INTEGER; dimension NGEN
Machine status of one, for in-service, and zero for out-of-service.

RMPCT Type: REAL; dimension NGEN
Percent of total Mvar required to hold the voltage at bus IREG that are to be contributed by the generation at bus NUM. RMPCT=100 if there is no other generator bus controlling remote bus IREG.

PT Type: REAL; dimension NGEN
Maximum generator real power output, in pu.

PB Type: REAL; dimension NGEN
Minimum generator real power output, in pu.

OWNER Type INTEGER; dimension (4,NGEN)
Owner(s) to which the generator is assigned. There may be up to 4 such owners; OWNER(i)=0 means that there are only i-1 owners.

OWNPCT Type REAL; dimension (4,NGEN)
Percentage of the generator owned by that owner.

WMOD Type: INTEGER; dimension NGEN
Wind machine control mode, WMOD is used to indicate whether a machine is a wind machine, and, if it is, the type of reactive power limits to be imposed:

0 for a machine which is not a wind machine;
1 for a wind machine whose reactive power limits are specified by QT and QB;
2 for a wind machine whose reactive power limits are determined from the machine's active power output and WPF, limits are of equal magnitude and opposite sign;
3 for a wind machine with a fixed reactive power setting determined from the machine's active power output and WPF; when WPF is positive, the machine's reactive power has the same sign as its active power; when WPF is negative, the machine's reactive power has the opposite sign of its active power.

WPF — Type: REAL; dimension NGEN
Power factor used in calculating reactive power limits or output when WMOD is 2 or 3.

IERR — Type: INTEGER; scalar
Will contain the return code:

0 No error.

1 Case not open.

2 Other error; values returned are undependable.

3 Allocation error.

4 Deallocation error.

The generator data is not retrieved in any particular order, i.e., there is no relationship between the generators that can be determined from their order or position in the above arrays. PSSGEN will attempt to place NGEN values in each data array (or 4∗NGEN for OWNER, OWNPCT); it is the users' responsibility to ensure that each array used for any argument is properly dimensioned. NGEN can be retrieved by PSSSIZ prior to calling PSSGEN. Refer to *PSS®E Program Operation Manual*, Section 5.2, Reading Power Flow Raw Data into the Working Case, for more details on the data contained in these arrays.

2.10.1 Older Cases

For cases prior to PSS®E-25, OWNER(1,NGEN) will be set to 1 and OWNPCT(1,NGEN) will be set to 100%.
For cases prior to PSS®E-31, WMOD is set to 0 and WPF is set to 1.0.

2.11 PSSBRN – Retrieve Branch Data

```
PSSBRN(FRMBUS,TOBUS,METBUS,CKT,RX,B,RATEA,RATEB,RATEC,GBI,GBJ,
STAT,LINLEN,INDX2W,OWNER,OWNPCT,IERR)
```

FRMBUS — Type: INTEGER; dimension NLIN
Bus number of from bus; for a transformer this is the winding one bus.

TOBUS — Type: INTEGER; dimension NLIN
Bus number of to bus; for a transformer this is the winding one bus.

METBUS Type: INTEGER; dimension NLIN
Bus number which has been designated as the metered end for area interchange and loss zone calculations.

CKT Type: CHARACTER*2; dimension NLIN
Two character circuit identifier.

RX Type: COMPLEX; dimension NLIN
Branch resistance and reactance, in pu.

B Type: REAL; dimension NLIN
Total branch charging susceptance, in pu, will be zero for transformers.

RATEA Type: REAL; dimension NLIN
First branch current rating, in MVA.

RATEB Type: REAL; dimension NLIN
Second branch current rating, in MVA.

RATEC Type: REAL; dimension NLIN
Third branch current rating, in MVA.

GBI Type: COMPLEX; dimension NLIN
Complex admittance of the line shunt at the from bus end of the branch, in pu. A negative reactive component indicates a line-connected reactor.

GBJ Type: COMPLEX; dimension NLIN
Complex admittance of the line shunt at the to bus end of the branch, in pu. A negative reactive component indicates a line-connected reactor will be zero for transformers.

STAT Type: INTEGER; dimension NLIN
Branch in-service status. One for in-service, and zero for out-of-service.

LINLEN Type: REAL; dimension NLIN
Length of line will be zero for transformers.

INDX2W Type: INTEGER; dimension NLIN
Index to the two-winding transformer data. If INDX2W is zero, the branch is not a transformer. If INDX2W is not zero, it provides an index to the arrays returned by PSSTRN.

OWNER Type INTEGER; dimension (4,NLIN)
Owner(s) to which the line is assigned. There may be up to four such owners; OWNER(i)=0 means that there are only i-1 owners.

OWNPCT Type REAL; dimension (4,NLIN)
Percentage of line owned by that owner.

IERR Type: INTEGER; scalar
Will contain the return code:

0 No error.

1 Case not open.

2 Other error; values returned are undependable.

3 Allocation error.

4 Deallocation error.

The branch data is not retrieved in any particular order, i.e., there is no relationship between the branches that can be determined from their order or position in the above arrays. PSSBRN will attempt to place NLIN values in each data array or (or 4∗NLIN for OWNER, OWNPCT); it is the users' responsibility to ensure that each array used for any argument is properly dimensioned. NLIN can be retrieved by PSSSIZ prior to calling PSSBRN. Refer to *PSS®E Program Operation Manual*, Section 5.2, Reading Power Flow Raw Data into the Working Case, for more details on the data contained in these arrays.

2.11.1 Older Cases

For saved cases produced by versions earlier than PSS®E-24, there is no data for the array LINLEN and PSSBRN will return values of zero. For cases prior to PSS®E-25, OWNER(1,NGEN) will be set to 1 and OWNPCT(1,NGEN) will be set to 100%.

2.11.2 Usage Notes

For cases containing transformers with nonzero values for the arrays B and LINLEN, those entries will be set to zero. For cases containing transformers with nonzero values for the array GBJ, those values will be added to the corresponding GBI entry, and the GBJ entry will be set to zero.

Branch data includes branches that are transformers. Cases containing three-winding transformers will have three branch records added for each three-winding transformer. These branches are returned by PSSBRN (and included in the count of branches, NLIN, returned by PSSSIZ). They connect each of the buses identified for the three-winding transformer to the hidden star bus. The hidden star bus can be identified by having a bus number in excess of the value HIBUS, which can be retrieved by PSSMSC.

RX will contain the actual values used in the power flow solutions. For nominal values see array RXTRAN of PSSTRN for transformer adjustments, and PSSABX for branch adjustments.

2.12 PSSTRN – Retrieve Two-Winding Transformer Data

```
PSSTRN(WIND1,WIND2,NOMV1,NOMV2,ANG1,SBASE1,TRNAME,CONBUS,RASW,
RMAX,RMIN,VMAX,VMIN,NTAPS,RXTRAN,TABLE,CNTL,XFRCMP,ANGW,
INDX3W,VECGRP,IERR)
```

WIND1 Type: REAL; dimension NTRFMR
The winding 1 off-nominal turns ratio, in pu, of the from bus base voltage.

WIND2 Type: REAL; dimension NTRFMR
The winding 2 off-nominal turns ratio, in pu, of the to bus base voltage.

NOMV1 Type: REAL; dimension NTRFMR
The nominal (rated) winding 1 voltage, in kV.

NOMV2 Type: REAL; dimension NTRFMR
The nominal (rated) winding 2 voltage, in kV.

ANG1 Type: REAL; dimension NTRFMR
The winding 1 phase shift angle, in degrees.

SBASE1 Type: REAL; dimension NTRFMR
The winding 1 base MVA of the transformer.

TRNAME Type: CHARACTER*12; dimension NTRFMR
An alphanumeric identifier assigned to the transformer.

CONBUS Type: INTEGER; dimension NTRFMR
Bus number of the bus whose voltage is controlled by this transformer.

RASW Type: LOGICAL; dimension NTRFMR
RASW will be true if and only if CONBUS is not the from or the to bus of the branch, and ratio adjustment should be performed as if the bus were on the tapped side of the transformer, rather than the impedance side.

RMAX Type: REAL; dimension NTRFMR
Upper limit of (a) off-nominal ratio for voltage- or Mvar-controlling transformers (when /CNTL/ is 1 or 2), in pu; or (b) phase shift angle for MW controlling transformers (when /CNTL/ is 3), in degrees.

RMIN Type: REAL; dimension NTRFMR
Lower limit, corresponding to RMAX.

VMAX Type: REAL; dimension NTRFMR
Upper limit of (a) controlled bus voltage (when |CNTL| is 1), in pu; or (b) real power through phase shifter (at tapped side) (when |CNTL| is 3), in MW; or (c) reactive power flow through the transformer (at tapped side) (when |CNTL| is 2), in Mvar.

VMIN Type: REAL; dimension NTRFMR
Lower limit, corresponding to VMAX.

NTAPS Type: INTEGER; dimension NTRFMR
Number of tap positions available.

RXTRAN Type: COMPLEX; dimension NTRFMR
Initial branch resistance and reactance before transformer impedance correction (see TABLE), in pu.

TABLE Type: INTEGER; dimension NTRFMR
Zero, or number of transformer impedance correction table.

CNTL Type: INTEGER; dimension NTRFMR
The transformer control mode for automatic adjustments of the first winding tap or phase shift angle during power flow solutions: 0 for no control (fixed tap and phase shift); ±1 for voltage control; ±2 for reactive power flow control; ±3 for active power flow control; ±4 for control of a dc line quantity. A positive control mode enables automatic adjustment of this transformer when the corresponding adjustment is activated during power flow solutions; a negative control mode suppresses the automatic adjustment of this transformer.

XFRCMP Type: COMPLEX; dimension NTRFMR
Load drop compensation impedance for voltage controlling transformers, in pu.

ANGW Type: REAL; dimension NTRFMR
Winding connection angle, in degrees.

INDX3W Type: INTEGER; dimension NTRFMR
Index to the three-winding transformer data. If INDX3W is zero, the transformer is not part of a three-winding transformer. If INDX3W is not zero, it provides an index to the arrays returned by PSS3WT.

VECGRPType: CHARACTER * 12; dimension NTRFMR
Vector Group identifier

IERR Type: INTEGER; scalar
Will contain the return code:

0 No error.

1 Case not open.

2 Other error; values returned are undependable.

The transformer data is not retrieved in any particular order, i.e., there is no relationship between the transformers that can be determined from their order or position in the above arrays. PSSTRN will attempt to place NTRFMR values in each data array; it is the users' responsibility to ensure that each array used for any argument is properly dimensioned. NTRFMR can be retrieved by PSSSIZ prior to calling PSSTRN. Refer to *PSS®E Program Operation Manual*, Section 5.2, Reading Power Flow Raw Data into the Working Case, for more details on the data contained in these arrays.

2.12.1 Older Cases

For saved cases produced by versions earlier than PSS®E-17, there is no data for the array CNTL. Values will be assigned according to defaults in PSS®E activity CASE.
For saved cases produced by versions earlier than PSS®E, there is no data for the array XFRCMP. Values will be assigned according to defaults in PSS®E activity CASE.

For saved cases produced by versions earlier than PSS®E-27, there is no data for WIND1, WIND2, NOMV1, NOMV2, ANG1, SBASE1, and TRNAME. Values will be assigned according to defaults in PSS®E activity CASE.

2.12.2 Usage Notes

The array INDX2W, retrieved by PSSBRN, provides a correspondence between the branch data arrays and these transformer data arrays, i.e., if, for a branch I, the value of INDX2W(I) is J, then the Jth element of all the arrays returned by PSSTRN pertain to the branch I.
Cases containing three-winding transformers will have three two-winding transformer records added for each three-winding transformer. These transformers are returned by PSSTRN (and included in the count of two-winding transformers, NTRFMR, returned by PSSSIZ). They connect each of the buses identified for the three-winding transformer to the hidden star bus. The hidden star bus can be identified by having a bus number in excess of the value HIBUS, which can be retrieved by PSSMSC.

2.13 PSS3WT - Retrieve Three-Winding Transformer Data

```
PSS3WT (BUS1ST,BUS2ND,BUS3RD,BUSSTAR,STATUS,NMETER,CKT,TRNAME,
VECGRP,IERR)
```

BUS1ST — Type: INTEGER; dimension N3WNDT
Bus number of the bus to which the first winding is connected.

BUS2ND — Type: INTEGER; dimension N3WNDT
Bus number of the bus to which the second winding is connected.

BUS3RD — Type: INTEGER; dimension N3WNDT
Bus number of the bus to which the third winding is connected.

BUSSTAR — Type: INTEGER; dimension N3WNDT
Bus number of the hidden star bus.

STATUS — Type: INTEGER; dimension N3WNDT
Branch in-service status. Zero for out-of-service, one for in-service, two for only winding two out-of-service, three for only winding three out-of-service, and four for only winding one out-of-service.

NMETER — Type: INTEGER; dimension N3WNDT
Bus number of the nonmetered end of the three-winding transformer.

CKT — Type: CHARACTER*2; dimension N3WNDT
Two-character circuit identifier.

TRNAME — Type: CHARACTER*12; dimension N3WNDT
An alphanumeric identifier assigned to the transformer.

VECGRPType: CHARACTER * 12; dimension NT3WDT
Vector Group identifier

IERR — Type: INTEGER; scalar
Will contain the return code:

0 No error.
1 Case not open.
2 Other error; values returned are undependable.

The transformer data is not retrieved in any particular order, i.e., there is no relationship between the transformers that can be determined from their order or position in the above arrays. PSS3WT will attempt to place N3WNDT values in each array; it is the user's responsibility to ensure that each array used for any argument is properly dimensioned. N3WNDT can be retrieved by PSSSIZ, prior to calling PSS3WT. Refer to *PSS®E Program Operation Manual*, Section 5.2, Reading Power Flow Raw Data into the Working Case, for more details on the data contained in these arrays.

2.13.1 Usage Notes

The Ith entry in each array corresponds to a separate three-winding transformer. Each three-winding transformer, when entered into PSS®E, generates the creation of a bus record (for the hidden star bus), as well as three two-winding transformer records (returned by these routines as three branch entries and three corresponding two-winding transformer entries).
There is no power flow data returned by this routine that is not available from the routines PSSBRN and PSSTRN. The three branch/transformers indicated by these routines (BUS1ST(I)-BUSSTAR(I), BUS2ND(I)-BUSSTAR(I), and BUS3RD(I)-BUSSTAR(I)) will all have the same status (STATUS(I)), circuit-identifier (CKT(I)), and transformer name (TRNAME(I)), and the branch metered end will be consistent with NMETER(I).

2.14 PSS3IX – Build Indices to Three-Winding Transformer Data

```
PSS3IX (FRMBUS,TOBUS,INDX2W,INDX3W,BUS1ST,BUS2ND,BUS3RD,BUSSTAR,
IDX1BR,IDX2BR,IDX3BR,IDXBUS,IERR)
```

FRMBUS — Input Array (see PSSBRN for description)
The FRMBUS array retrieved by using PSSBRN.

TOBUS — Input Array (see PSSBRN for description)
The TOBUS array retrieved by using PSSBRN.

INDX2W — Input Array (see PSSBRN for description)
The INDX2W array retrieved by using PSSBRN.

INDX3W — Input Array (see PSSTRN for description)
The INDX3W array retrieved by using PSSTRN.

BUS1ST — Input Array (see PSS3WT for description)
The BUS1ST array retrieved by using PSS3WT.

BUS2ND — Input Array (see PSS3WT for description)
The BUS2ND array retrieved by using PSS3WT.

BUS3RD — Input Array (see PSS3WT for description)
The BUS3RD array retrieved by using PSS3WT.

BUSSTAR Input Array (see PSS3WT for description)
The BUSSTAR array retrieved by using PSS3WT.

IDX1BR Type: INTEGER; dimension N3WNDT
An index to the branch arrays for the two-winding transformer placed between the first winding bus and the hidden star bus of the three-winding transformer.

IDX2BR Type: INTEGER; dimension N3WNDT
An index to the branch arrays for the two-winding transformer placed between the second winding bus and the hidden star bus of the three-winding transformer.

IDX3BR Type: INTEGER; dimension N3WNDT
An index to the branch arrays for the two-winding transformer placed between the third winding bus and the hidden star bus of the three-winding transformer.

IDXBUS Type: INTEGER; dimension N3WNDT
An index to the bus arrays for the hidden star bus of the three-winding transformer.

IERR Type: INTEGER; scalar
Will contain the return code:

0 No error.

1 Case not open.

2 Other error; values returned are undependable.

In order to use this routine, PSSBRN and PSS3WT must be called first, as they will supply the source for the first eight arrays, which are input to this routine.
PSS3IX will attempt to place N3WNDT values in each returned array; it is the user's responsibility to ensure that each array used for any argument is properly dimensioned. N3WNDT can be retrieved using PSSSIZ, prior to calling PSS3IX.

2.14.1 Usage Notes

The purpose of this routine is to allow easy reference to the data corresponding to a given three-winding transformer. The Ith entry in the IDX1BR, IDX2BR, and IDX3BR arrays each indicates the branch array entries that contain the data for the Ith three-winding transformer. Transformer data can then be accessed using the INDX2W array for that branch. The Ith entry in the IDXBUS array indicates the bus array entries that contain the data for the hidden star bus for the Ith three-winding transformer.

2.15 PSSAIN – Retrieve Area Interchange Data

```
PSSAIN(ARNUM,ISW,PDES,PTOL,ARNAM,IERR)
```

ARNUM Type: INTEGER; dimension (MAXARE)
Area number.

ISW Type: INTEGER; dimension (MAXARE)
Number of area slack bus for area interchange control.

PDES Type: REAL; dimension (MAXARE)
Desired net interchange leaving the area, in pu.

PTOL Type: REAL; dimension (MAXARE)
Interchange tolerance band width, in pu.

ARNAM Type: CHARACTER∗12; dimension (MAXARE)
Alphanumeric identifier for the area.

IERR Type: INTEGER; scalar
Will contain the return code:

0 No error.

1 Case not open.

2 Other error; values returned are undependable.

The area interchange data is not retrieved in any particular order, i.e., there is no relationship between the areas that can be determined from their order or position in the above arrays. PSSAIN will attempt to use MAXARE elements and return NAREAS values in each array; it is the user's responsibility to ensure that each array used for any argument is properly dimensioned. NAREAS and MAXARE can be retrieved using PSSSIZ prior to calling PSSAIN. Refer to *PSS®E Program Operation Manual*, Section 5.2, Reading Power Flow Raw Data into the Working Case, for more details on the data contained in these arrays.

2.15.1 Older Cases

For cases written by versions of PSS®E prior to version 25 only areas where the value of at least one array besides ARNUM is not the default value are returned. For later saved case versions areas that contain any equipment in the case are also returned.

2.16 PSS2DC – Retrieve Two-Terminal Transmission Line Data

```
PSS2DC(NAME,MDC,RDC,SETVL,VSCHD,VCMOD,RCOMP,DELTI,METER,DCVMIN,
CITMX,CACC,IP,NB,MX,MN,RC,XC,EBAS,TR,TAP,TPMX,TPMN,TSTP,IC,IFR,
ITO,ID,XCAP,PAC,QAC,IERR)
```

This group of arrays is dimensioned by the number of dc lines.

NAME Type: CHARACTER∗12; dimension (MAX2DC)
The non-blank alphanumeric identifier assigned to the dc line.

MDC Type: INTEGER; dimension (MAX2DC)
Control mode: 0 for blocked, 1 for power, 2 for current.

RDC Type: REAL; dimension (MAX2DC)
dc line resistance, in ohms.

SETVL Type: REAL; dimension (MAX2DC)
Current (amps) or power (MW) demand. When MDC is one, a positive value of SETVL specifies desired power at the rectifier and a negative value specifies inverter power.

VSCHD Type: REAL; dimension (MAX2DC)
Scheduled compounded dc voltage, in kV.

VCMOD Type: REAL; dimension (MAX2DC)
Mode switch dc voltage, in kV. When MDC=1, if dc voltage falls below this value indicates control should switch to current mode (MDC=2).

RCOMP Type: REAL; dimension (MAX2DC)
Compounding resistance, in ohms. Used to calculate compound voltage.

DELTI Type: REAL; dimension (MAX2DC)
Margin in per unit of desired dc power or current (fraction by which order is reduced when ALPHA is at its minimum and the inverter is controlling the line current).

METER Type: CHARACTER∗1; dimension (MAX2DC)
Indicates metered end. 'R' for rectifier, 'I' for inverter.

DCVMIN Type: REAL; dimension (MAX2DC)
Minimum compounded dc voltage, in kV, used in constant gamma operation.

CITMX Type: INTEGER; dimension (MAX2DC)
Iteration limit for CCC Newton solution procedure.

CACC Type: REAL; dimension (MAX2DC)
Acceleration factor for CCC Newton solution procedure.

The remaining arrays are dimensioned (2, MAX2DC) where entry (1,I) and (2,I) pertain to the rectifier end and the inverter end of dc line number I, respectively.

IP Type: INTEGER
Bus number.

NB Type: INTEGER
Number of bridges in series.

MX Type: REAL
Nominal maximum firing angle, in degrees.

MN Type: REAL
Minimum steady state firing angle, in degrees.

RC Type: REAL
Commutating transformer resistance per bridge, in ohms.

XC Type: REAL
Commutating transformer reactance per bridge, in ohms.

EBAS Type: REAL
Primary base ac voltage, in kV.

TR Type: REAL
Transformer ratio.

TAP Type: REAL
Tap setting.

TPMX Type: REAL
Maximum tap setting.

TPMN Type: REAL
Minimum tap setting.

TSTP Type: REAL
Tap step.

IC Type: INTEGER
Firing angle measuring bus. If 0 (zero), firing angle measured at converter bus.

IFR Type: INTEGER
Tapped side from bus number of ac transformer branch. If 0 (zero) then dc tap is adjusted to control quantities inside the dc line, and the next two arrays should be ignored.

ITO Type: INTEGER
Untapped side to bus number of ac transformer branch.

ID Type: CHARACTER∗2
Circuit identifier for ac transformer branch.

XCAP Type: REAL
Commutating capacitor reactance magnitude per bridge, in ohms.

PAC Type: REAL
Real power flowing into the dc line at converter bus, in pu.

QAC Type: REAL
Reactive power flowing into the dc line at converter bus, in pu.

This last argument is not an array.

IERR Type: INTEGER; scalar
Will contain the return code:

0 No error.

1 Case not open.

2 Other error; values returned are undependable.

3 Allocation error.

4 Deallocation error.

The two-terminal transmission line data is not retrieved in any particular order, i.e., there is no relationship between the lines that can be determined from their order or position in the above arrays. PSS2DC will attempt to use MAX2DC elements and return NDCL values in each data array; it is the users' responsibility to ensure that each array used for any argument is properly dimensioned. NDCL and MAX2DC can be retrieved by PSSSIZ prior to calling PSS2DC. Refer to *PSS®E Program Operation Manual*, Section 5.2, Reading Power Flow Raw Data into the Working Case, for more details on the data contained in these arrays.

2.16.1 Older Cases

For saved cases produced by versions earlier than PSS®E-18, there is no data for the arrays IC, IFR, ITO, DCVMIN and ID, and for cases prior to PSS®E-26, there is no data for arrays CITMX, CACC, and XCAP. Values are assigned according to defaults in PSS®E activity CASE.
For cases prior to PSS®E-31, NAME is set to the character value of the dc line number (names were not used), e.g., the name for dc line #3 will be 3.

2.17 PSSVSC - Retrieve Voltage Source Converter dc Line Data

```
PSSVSC(VNAME,CNTL,RDC,OWNER,OWNPCT,CNVBUS,ITYPE,MODE,DCSET,ACSET,
ALOSS,BLOSS,MNLOSS,SMAX,IMAX,PWF,MAXQ,MINQ,REMBUS,REMPCT,PAC,QAC,
IERR)
```

VNAME — Type: CHARACTER*12; dimension NVSC
An alphanumeric identifier assigned to the dc line.

CNTL — Type: INTEGER; dimension NVSC
Control mode: 0 for out-of-service, 1 for in-service.

RDC — Type: REAL; dimension NVSC
Dc line resistance, in ohms.

OWNER — Type: INTEGER; dimension (4,NVSC)
Owner(s) to which the dc line is assigned. There may be up to 4 such owners; OWNER(i)=0 mean that there are only i-1 owners.

OWNPCT — Type: REAL; dimension (4,NVSC)
Percentage of the dc line owned by that owner.

The following arrays are dimensioned (2,NVSC) where entry (1,i) and (2,i) refer to data for converter bus 1 and converter bus 2, respectively.

CNVBUS — Type: INTEGER; dimension (2,NVSC)
Converter bus number.

ITYPE — Type: INTEGER; dimension (2,NVSC)
Type of converter dc control: 1 for dc voltage control or 2 for MW control.

MODE — Type: INTEGER; dimension (2,NVSC)
Converter ac control mode: 1 for ac voltage control or 2 for fixed ac power factor.

DCSET — Type: REAL; dimension (2,NVSC)
Converter dc setpoint. For TYPE=1 it is the scheduled dc voltage on the dc side of the converter bus, in kV. For TYPE=2 it is the power demand at CNVBUS, in MW (positive value indicates feeding power to the ac network at CNVBUS, negative value indicates withdrawing power from the ac network at CNVBUS).

ACSET — Type: REAL; dimension (2,NVSC)
Converter ac setpoint. For MODE=1 it is the regulated ac voltage setpoint, in pu. For MODE=2 it is the power factor setpoint.

ALOSS — Type: REAL; dimension (2,NVSC)

BLOSS Type: REAL; dimension (2,NVSC)
Coefficients of the linear equation used to calculate converter losses:

$KW_{conv\ loss} = ALOSS + I_{dc} * BLOSS$

MNLOSS Type: REAL; dimension (2,NVSC)
Minimum converter losses, in kW.

SMAX Type: REAL; dimension (2,NVSC)
Converter MVA rating, in MVA (zero indicates unlimited converter MVA loading).

IMAX Type: REAL; dimension (2,NVSC)
Converter ac rating, in amps (zero indicates unlimited converter current loading).

PWF Type: REAL; dimension (2,NVSC)
Power weighting factor fraction (0.0 <= PWF <= 1.0) used in reducing the active power order and either the reactive power order (when MODE=2) or the reactive power limits (when MODE=1) when the converter MVA or current rating is violated. When PWF is 0.0, only the active power is reduced; when PWF is 1.0, only the reactive power is reduced; otherwise a weighted reduction of both active and reactive power is applied.

MAXQ Type: REAL; dimension (2,NVSC)

MINQ Type: REAL; dimension (2,NVSC)
Reactive power upper limit, and reactive power lower limit (positive value of reactive power indicates reactive power flowing into the ac network from the converter; negative value of reactive power indicates reactive power withdrawn from the ac network). Not used if MODE=2.

REMBUS Type: INTEGER; dimension (2,NVSC)
Bus number of remote bus whose voltage is to be regulated by this converter to the value specified by ACSET. Not used if MODE=2.

RMPCT Type: REAL; dimension (2,NVSC)
Percent of total Mvar required to hold the voltage at the bus controlled by bus CNVBUS that are to be contributed by this VSC. Only needed if REMBUS specifies a bus controlled by more than one VSC. Not used if MODE=2.

PAC Type: REAL; dimension (2,NVSC)
Real power flowing into the dc line at CNVBUS, in pu.

QAC Type: REAL; dimension (2,NVSC)
Reactive power flowing into the dc line at CNVBUS, in pu.

This last argument is not an array.

IERR Type: INTEGER; scalar
Will contain the return code:

0 no error.

1 case not open.

2 other error; values returned are undependable.

The VSC dc line data is not retrieved in any particular order, i.e., there is no relationship between the VSC dc lines that can be determined from their order or position in the above

arrays. PSSVSC will attempt to place NVSC values in each data array (or 2*NVSC if the array is dimensioned (2,NVSC), or 4*NVSC if the array is dimensioned (4,NVSC)); it is the user's responsibility to ensure that each array used for any argument is properly dimensioned. NVSC can be retrieved using PSSSIZ, prior to calling PSSVSC. Refer to *PSS®E Program Operation Manual*, Section 5.2, Reading Power Flow Raw Data into the Working Case, for more details on the data contained in these arrays.

2.18 PSSWSH – Retrieve Switched Shunt Data

```
PSSWSH(NUM,MODSW,ADJM,STAT,VSWHI,VSWLO,SWREM,RMPCT,BINIT,RMINDX,
NI,BI,IERR)
```

NUM — Type: INTEGER; dimension NSHUNT
Bus number.

MODSW — Type: INTEGER; dimension NSHUNT
Control mode:
0 - locked
1 - discrete adjustment, controlling voltage locally or at bus SWREM
2 - continuous adjustment, controlling voltage locally or at bus SWREM
3 - discrete adjustment, controlling reactive power output of the plant at bus SWREM
4 - discrete adjustment, controlling reactive power output of the VSC dc line converter at bus SWREM of the VSC dc line whose name is specified as RMINDX
5 - discrete adjustment, controlling admittance setting of the switched shunt at bus SWREM
6 - discrete adjustment, controlling reactive power output of the shunt element of the FACTS device whose name is specified as RMINDX

ADJM — Type: INTEGER; dimension NSHUNT
Adjustment method:
0 - steps and blocks are switched on in input order, and off in reverse input order; this adjustment method was the only method available prior to PSS®E-32.0.
1 - steps and blocks are switched on and off such that the next highest (or lowest, as appropriate) total admittance is achieved.

STAT — Type: INTEGER; dimension NSHUNT
Initial switched shunt status of one for in-service and zero for out-of-service.

VSWHI — Type: REAL; dimension NSHUNT
Controlled voltage upper limit, in pu.

VSWLO — Type: REAL; dimension NSHUNT
Controller voltage lower limit, in pu.

SWREM — Type: INTEGER; dimension NSHUNT
Bus number of remote bus whose voltage is controlled by this switched shunt. Zero indicates device controls its own voltage.

RMPCT — Type: REAL; dimension NSHUNT
Percent of total Mvar required to hold the voltage at the bus NUM that are to be

contributed by this switched shunt. Only needed if SWREM specifies a bus controlled by more than one voltage controlling device. Only used if MODSW = 1 or 2.

BINIT — Type: REAL; dimension NSHUNT
Actual switched shunt admittance, in pu.

RMINDX — Type: INTEGER; dimension NSHUNT
Index to VSC dc line data when MODSW is 4, indicates the VSC dc line whose converter bus is specified in SWREM.

NI — Type: INTEGER; dimension (8,NSHUNT)
Number of steps per block, up to 8 blocks.

BI — Type: REAL; dimension (8,NSHUNT)
Admittance increment per step per block, up to 8 blocks.

IERR — Type: INTEGER; scalar
Will contain the return code:

0 No error.

1 Case not open.

2 Other error; values returned are undependable.

The switched shunt data is not retrieved in any particular order, i.e., there is no relationship between the switched shunts that can be determined from their order or position in the above arrays. PSSWSH will attempt to place NSHUNT values in each data array (or 8∗NSHUNT for NI, BI); it is the users' responsibility to ensure that each array used for any argument is properly dimensioned. NSHUNT can be retrieved by PSSSIZ prior to calling PSSWSH. Refer to *PSS®E Program Operation Manual*, Section 5.2, Reading Power Flow Raw Data into the Working Case, for more details on the data contained in these arrays.

2.18.1 Older Cases

For cases prior to PSS®E-29, there is no data for array RMINDX. For cases prior to PSS®E-30, there is no data for array RMPCT. These values are assigned according to defaults in PSS®E activity CASE.

2.19 PSSTIC – Retrieve Transformer Impedance Correction Tables

```
PSSTIC(TBLCNT,TI,FI,PHTR,IERR)
```

TBLCNT — Type: INTEGER; dimension (MAXTIC)
Number of entries in each table.

TI — Type: REAL; dimension (11, MAXTIC)
Either winding one off-nominal turns ratio in pu or phase shift angle in degrees.

FI — Type: REAL; dimension (11, MAXTIC)
Scaling factor by which transformer nominal impedance is to be multiplied to obtain the actual transformer impedance for the corresponding TI.

PHTR Type: LOGICAL; dimension (MAXTIC)
If true then TI is a phase shift angle, else TI is a turns ratio.

IERR Type: INTEGER; scalar
Will contain the return code:

0 No error.

1 Case not open.

2 Other error; values returned are undependable.

The Ith row (e.g., TI(x,I) or TBLCNT(I)) in each array corresponds to table number I. When no data for table number I has been entered into PSS®E and saved in the saved case, TBLCNT(I) will be zero, for I less than MAXTIC. PSSTIC will attempt to place 11*MAXTIC values in arrays TI and FI, and MAXTIC values in arrays PHTR and TBLCNT; it is the users' responsibility to ensure that each array used for any argument is properly dimensioned. MAXTIC can be retrieved by PSSSIZ prior to calling PSSTIC. Refer to *PSS®E Program Operation Manual*, Section 5.2, Reading Power Flow Raw Data into the Working Case, for more details on the data contained in these arrays.

2.19.1 Usage Notes

TBLCNT(I) indicates the number of entries in each table, for I less than MAXTIC. If this value is zero, then there is no data for table number I.

2.20 PSSMDC – Retrieve Multiterminal dc Transmission Line Data

```
PSSMDC(NAME,NCONV,NDCBS,NDCLN,MDC,VCONVP,VCONVN,VCMOD,IB,N,ANGMX,
ANGMN,RC,XC,EBAS,TR,TAP,TPMX,TPMN,TSTP,SETVL,DCPF,MARG,CNVCOD,AC,
IA,ZONE,DCNAM,IDC2,RGRND,OWNER,FROM,TO,DCCKT,RDC,LDC,METEND,PAC,
QAC,IERR)
```

This group of arrays is dimensioned by the number of multi-terminal dc lines.

NAME — Type: CHARACTER∗12; dimension (MAXMDC)
The non-blank alphanumeric identifier assigned to the multiterminal dc line.

NCONV — Type: INTEGER; dimension (MAXMDC)
Number of ac converter station buses in multiterminal dc line; will range from 3 to 12. This will also be the number of elements returned in those arrays below dimensioned by (NCONV, dc line #) that contain actual data for each line.

NDCBS — Type: INTEGER; dimension (MAXMDC)
Number of dc buses in multiterminal dc line; will range from NCONV to 20. This will also be the number of elements returned in those arrays below dimensioned by (NDCBS, dc line #) that contain actual data for each line.

NDCLN — Type: INTEGER; dimension (MAXMDC)
Number of dc links in multiterminal dc line; will range from 2 to 20. This will also be the number of elements returned in those arrays below dimensioned by (NDCLN, dc line #) that contain actual data for each line.

MDC — Type: INTEGER; dimension (MAXMDC)
Control mode: 0 for blocked, 1 for power, 2 for current.

VCONVP — Type: INTEGER; dimension (MAXMDC)
Bus number of the ac converter station bus which controls dc voltage on the positive pole of the multiterminal dc line (will be a positive pole inverter).

VCONVN — Type: INTEGER; dimension (MAXMDC)
Bus number of the ac converter station bus which controls dc voltage on the negative pole of the multiterminal dc line. Will be zero if negative pole not modeled.

VCMOD — Type: REAL; dimension (MAXMDC)
Mode switch dc voltage, in kV. When MDC=1, if dc voltage falls below this value indicates control should switch to current mode (MDC=2).

This group of arrays is dimensioned by the maximum number of ac converter station buses, and the number of multi-terminal dc lines (NCONV, dc line #).

IB — Type: INTEGER; dimension (12,MAXMDC)
The ac converter bus number.

N — Type: INTEGER; dimension (12,MAXMDC)
Number of bridges in series.

ANGMX — Type: REAL; dimension (12,MAXMDC)
Nominal maximum ALPHA or GAMMA angle, in degrees.

ANGMN — Type: REAL; dimension (12,MAXMDC)
Minimum steady state ALPHA or GAMMA angle, in degrees.

RC Type: REAL; dimension (12,MAXMDC)
Commutating resistance per bridge, in ohms.

XC Type: REAL; dimension (12,MAXMDC)
Commutating reactance per bridge, in ohms.

EBAS Type: REAL; dimension (12,MAXMDC)
Primary base ac voltage, in kV.

TR Type: REAL; dimension (12,MAXMDC)
Actual transformer ratio.

TAP Type: REAL; dimension (12,MAXMDC)
Tap setting.

TPMX Type: REAL; dimension (12,MAXMDC)
Maximum tap setting.

TPMN Type: REAL; dimension (12,MAXMDC)
Minimum tap setting.

TSTP Type: REAL; dimension (12,MAXMDC)
Tap step.

SETVL Type: REAL; dimension (12,MAXMDC)
Converter setpoint. When IB equals VCONVP or VCONVN above, then SETVL specifies the scheduled dc voltage magnitude, in kV, across the converter. Otherwise, SETVL contains the converter current, in amps, or the power demand, in MW; a positive value of SETVL indicates that IB is a rectifier and a negative value indicates an inverter.

DCPF Type: REAL; dimension (12,MAXMDC)
Converter participation factor. When the order at any rectifier in the multiterminal dc line is reduced, the orders at the remaining converters on the same pole are modified according to their DCPF's.

MARG Type: REAL; dimension (12,MAXMDC)
Rectifier margin, in per unit of desired dc power or current. The converter order reduced by this fraction, (1.-MARG)*SETVL, defines the minimum order for this rectifier. MARG is used only at rectifiers.

CNVCOD Type: INTEGER; dimension (12,MAXMDC)
Converter code.

This group of arrays is dimensioned by the maximum number of dc buses, and the number of multi-terminal dc lines (NDCBS, dc line #).

AC Type: INTEGER; dimension (20,MAXMDC)
The ac converter bus number, or zero. Each converter station bus specified in IB above must occur as a value of ac for that multiterminal dc line; ac is zero otherwise and indicates a dc bus connected only to other dc buses by dc links.

IA Type: INTEGER; dimension (20,MAXMDC)
Area number, range from 1 to 100.

ZONE Type: INTEGER; dimension (20,MAXMDC)
Loss zone, range from 1 to 999.

DCNAM Type: CHARACTER∗12; dimension (20,MAXMDC)
Alphanumeric identifier for this dc bus.

IDC2 Type: INTEGER; dimension (20,MAXMDC)
Second dc bus to which converter ac is connected. Zero if connected directly to ground.

RGRND Type: REAL; dimension (20,MAXMDC)
Resistance to ground. Used during solutions only for those dc buses that were specified as IDC2 on other dc bus records.

OWNER Type: INTEGER; dimension (20, MAXMDC)
Owner to which dc bus is assigned.

This group of arrays is dimensioned by the maximum number of dc links, and the number of multi-terminal dc lines (NDCLN, dc line #).

FROM Type: INTEGER; dimension (20,MAXMDC)
Branch from bus dc bus number.

TO Type: INTEGER; dimension (20,MAXMDC)
Branch to bus dc bus number.

DCKT Type: CHARACTER∗1; dimension (20,MAXMDC)
Branch circuit identifier.

RDC Type: REAL; dimension (20,MAXMDC)
The dc link resistance, in ohms.

LDC Type: REAL; dimension (20,MAXMDC)
The dc link inductance, in mH.

METEND Type: INTEGER; dimension (20,MAXMDC)
Metered end of branch for area interchange and loss zone calculations, must be FROM or TO.

These last two arrays are dimensioned by (NCONV, dc line #) (see above).

PAC Type: REAL; dimension (12,MAXMDC)
Real power flowing into the dc line at the converter bus, in pu.

QAC Type: REAL; dimension (12,MAXMDC)
Reactive power flowing into the dc line at the converter bus, in pu.

This last argument is not an array.

IERR Type: INTEGER; scalar
Will contain the return code:

0 No error.

1 Case not open.

2 Other error; values returned are undependable.

The multi-terminal dc transmission line data is not retrieved in any particular order, i.e., there is no relationship between the lines that can be determined from their order or position in the above arrays. PSSMDC will attempt to use MAXMDC elements and return

NMTDCL values in each data array; it is the users' responsibility to ensure that each array used for any argument is properly dimensioned. NMTDCL and MAXMDC can be retrieved by PSSSIZ prior to calling PSSMDC. Refer to *PSS®E Program Operation Manual*, Section 5.2, Reading Power Flow Raw Data into the Working Case, for more details on the data contained in these arrays.

2.20.1 Older Cases

For saved cases produced by versions earlier than PSS®E-19, there is no data for the arrays VCONVN, IDC2, RGRND and LDC. Values are set according to defaults in PSS®E activity CASE.

For cases prior to PSS®E-25, OWNER value will be set 1.

For cases prior to PSS®E-31, NAME is set to MDCLINE#, where # is a sequence number (see PSS2DC, Section 2.16 PSS2DC – Retrieve Two-Terminal Transmission Line Data).

2.20.2 Usage Notes

There are four groups of data returned by PSSMDC: firstly the arrays indexed solely by multiterminal dc line number (i.e., (NMTDCL)), secondly, those indexed by converter and dc line number (i.e., (12,NMTDCL)), thirdly, those indexed by dc bus number and dc line number (i.e., (20,NMTDCL), AC through RGRND), and fourthly, those indexed by dc link index and dc line number (i.e., (20,NMTDCL), FROM through METEND). NCONV(I) contains the actual number of converter stations for that dc line, even though the arrays must be dimensioned to allow for the maximum, which is 12. NDCBS, for dc buses, and NDCLN, for dc link indices, apply similarly to the other arrays.

2.21 PSSMSL – Retrieve Multisection Line Data

```
PSSMSL(FRMBUS,TOBUS,LINEID,CKT,DUMBUS,METBUS,IERR)
```

FRMBUS — Type: INTEGER; dimension (NMSLIN)
From bus number of multisection line grouping.

TOBUS — Type: INTEGER; dimension (NMSLIN)
To bus number of multisection line grouping.

LINEID — Type: CHARACTER∗2; dimension (NSECTN)
Multisection line grouping identifier. Only NMSLIN values are returned; the additional dimensional requirement is to provide work space for PSSMSL.

CKT — Type: CHARACTER∗2; dimension (10,NMSLIN)
Branch circuit identifiers of branches which are members of this multisection line grouping. Unused values are unchanged. One plus the number of non-zero values in DUMBUS will be set.

DUMBUS — Type: INTEGER; dimension (9,NMSLIN)
Bus numbers of the dummy buses connected by the branches which comprise this multisection line grouping. If there are less than 9 dummy buses (i.e., less than 10

sections), the unused DUMBUS values for that multisection line grouping are set to zero.

METBUS — Type: INTEGER; dimension (NMSLIN)
Bus number which has been designated as the metered end.

IERR — Type: INTEGER; scalar
Will contain the return code:

0 No error.

1 Case not open.

2 Other error; values returned are undependable.

The multisection line data is not retrieved in any particular order, i.e., there is no relationship between the multisection lines that can be determined from their order or position in the above arrays. PSSMSL will attempt to place NMSLIN values in each data array (9∗NMSLIN for array DUMBUS, 10∗NMSLIN for array CKT, NSECTN for array LINEID). It is the users' responsibility to ensure that each array used for any argument is properly dimensioned. NMSLIN and NSECTN can be retrieved by PSSSIZ prior to calling PSSMSL. Refer to *PSS®E Program Operation Manual*, Section 5.2, Reading Power Flow Raw Data into the Working Case, for more details on the data contained in these arrays.

2.21.1 Older Cases

For saved cases produced by PSS®E-19 there is no data for the array METBUS. Values will set according to the defaults in PSS®E activity CASE.

2.21.2 Usage Notes

There is no significance to the from bus and to bus designations.

2.22 PSSZNM – Retrieve Zone Names

```
PSSZNM(ZONUM,ZONAME,IERR)
```

ZONUM — Type: INTEGER; dimension (MAXZNM)
Zone number.

ZONAME — Type: CHARACTER∗12; dimension (MAXZNM)
Zone names.

IERR — Type: INTEGER; scalar
Will contain the return code:

0 No error.

1 Case not open.

2 Other error; values returned are undependable.

2 Other error; values returned are undependable.

The zone name data is not retrieved in any particular order, i.e., there is no relationship between the zones that can be determined from their order or position in the above arrays. PSSZNM will attempt to use MAXZNM elements and return NZONES values in each array; it is the user's responsibility to ensure that each array used for any argument is properly dimensioned. NZONES and MAXZNM can be retrieved using PSSSIZ prior to calling PSSZNM. Refer to *PSS®E Program Operation Manual*, Section 5.2, Reading Power Flow Raw Data into the Working Case, for more details on the data contained in these arrays.

2.22.1 Older Cases

For cases written by versions of PSS®E prior to version 25 only zones where the value of at least one array besides ZONUM is not the default value are returned. For later saved case versions zones that contain any equipment in the case are also returned.

2.23 PSSATR – Retrieve Area Transaction Data

```
PSSATR(ARFROM,ARTO,PTRAN,TRANID,IERR)
```

ARFROM — Type: INTEGER; dimension (NTRNAC)
Seller's area number.

ARTO — Type: INTEGER; dimension (NTRNAC)
Buyer's area number.

PTRAN — Type: REAL; dimension (NTRNAC)
Amount of transaction, in MW.

TRANID — Type: CHARACTER∗1; dimension (NTRNAC)
Alphanumeric identifier for the transaction.

IERR — Type: INTEGER; scalar
Will contain the return code:

0 No error.

1 Case not open.

2 Other error; values returned are undependable.

The transaction data is not retrieved in any particular order, i.e., there is no relationship between the transactions that can be determined from their order or position in the above arrays. PSSATR will attempt to place NTRNAC values in each data array; it is the users' responsibility to ensure that each array used for any argument is properly dimensioned. NTRNAC can be retrieved by PSSSIZ prior to calling PSSATR. Refer to *PSS®E Program Operation Manual*, Section 5.2, Reading Power Flow Raw Data into the Working Case, for more details on the data contained in these arrays

2.24 PSSOWN – Retrieve Owner Names

```
PSSOWN(OWNUM,OWNAME,IERR)
```

OWNUM — Type: INTEGER; dimension (MAXOWN)
Owner number.

OWNAME — Type: CHARACTER∗12; dimension (MAXOWN)
OWNER names.

IERR — Type: INTEGER; scalar
Will contain the return code.

0 No error.

1 Case not open.

2 Other error; values returned are undependable.

The owner name data is not retrieved in any particular order, i.e., there is no relationship between the owners that can be determined from their order or position in the above arrays. PSSZNM will attempt to use MAXOWN and return NOWNRS values in each array; it is the user's responsibility to ensure that each array used for any argument is properly dimensioned. NOWNRS and MAXOWN can be retrieved using PSSSIZ prior to calling PSSOWN. Refer to *PSS®E Program Operation Manual*, Section 5.2, Reading Power Flow Raw Data into the Working Case, for more details on the data contained in these arrays.

2.24.1 Older Cases

For cases written by versions of PSS®E prior to version 25 only owners where the value of at least one array besides OWNUM is not the default value are returned. For later saved case versions owners that contain any equipment in the case are also returned. .

2.25 PSSFCT – Retrieve FACTS Device Data

```
PSSFCT(NAME,SBUS,TBUS,MODE,PDES,QDES,VSET,SHMX,TRMX,VTMN,VTMX,
VSMX,IMX,LINX,RMPCT,OWNER,SET1,SET2,VSREF,REMOT,MNAME,PBRDG,
QSHNT,PSEND,QSEND,PTERM,QTERM,IERR)
```

NAME — Type: CHARACTER∗12; dimension (MAXFCT)
The non-blank alphanumeric identifier assigned to the FACTS device.

SBUS — Type: INTEGER; dimension MAXFCT
Sending end bus number.

TBUS — Type: INTEGER; dimension MAXFCT
Terminal end bus number; 0 for STATCON.

MODE — Type: INTEGER; dimension MAXFCT
For a STATCON (i.e., a FACTS devices with a shunt element but no series element), J must be 0 and MODE must be either 0 or 1):

0 - out-of-service (i.e., shunt link open)
1 - shunt link operating.

For a FACTS device with a series element (i.e., J is not 0), MODE may be:
0 - out-of-service (i.e., series and shunt links open)
1 - series and shunt links operating.
2 - series link bypassed (i.e., like a zero impedance line) and shunt link operating as a STATCON.
3 - series and shunt links operating with series link at constant series impedance.
4 - series and shunt links operating with series link at constant series voltage.
5 - master device of an IPFC with P and Q setpoints specified; another FACTS device must be designated as the slave device (i.e., its MODE is 6 or 8) of this IPFC.
6 - slave device of an IPFC with P and Q setpoints specified; the FACTS device specified in MNAME must be the master device (i.e., its MODE is 5 or 7) of this IPFC. The Q setpoint is ignored as the master device dictates the active power exchanged between the two devices.
7 - master device of an IPFC with constant series voltage setpoints specified; another FACTS device must be designated as the slave device (i.e., its MODE is 6 or 8) of this IPFC.
8 - slave device of an IPFC with constant series voltage setpoints specified; the FACTS device specified in MNAME must be the master device (i.e., its MODE is 5 or 7) of this IPFC. The complex $V_d + jV_q$ setpoint is modified during power flow solutions to reflect the active power exchange determined by the master device.

PDES Type: REAL; dimension MAXFCT
Desired real power flow arriving at the terminal end bus, in MW.

QDES Type: REAL; dimension MAXFCT
Desired reactive power flow arriving at the terminal end bus, in Mvar

VSET Type: REAL; dimension MAXFCT
Voltage setpoint at the sending end bus, in pu.

SHMX Type: REAL; dimension MAXFCT
Maximum shunt current at the sending bus end, in MVA at unity voltage.

TRMX Type: REAL; dimension MAXFCT
Maximum bridge real power transfer, in MW.

VTMN Type: REAL; dimension MAXFCT
Minimum voltage at the terminal end bus, in pu.

VTMX Type: REAL; dimension MAXFCT
Maximum voltage at the terminal end bus, in pu.

VSMX Type: REAL; dimension MAXFCT
Maximum series voltage, in pu.

IMX Type: REAL; dimension MAXFCT
Maximum series current, or zero for no current limit, in MVA at unity voltage.

LINX Type: REAL; dimension MAXFCT
Reactance of the dummy series element used in certain model solution states, in pu.

RMPCT Type: REAL; dimension MAXFCT
Percent of total Mvar required to hold the voltage at the bus SBUS that are to be contributed by the shunt element of this FACTS device. Only needed if more than one voltage controlling devices are controlling SBUS voltage.

OWNER Type: INTEGER; dimension MAXFCT
OWNER to which the FACTS device is assigned.

SET1 Type: REAL; dimension MAXFCT
If MODE = 3, resistance component of constant series impedance, in pu. If MODE = 4, magnitude of constant series voltage, in pu. If MODE = 7 or 8, real component of the constant series voltage with respect to the quantity referred to VSREF.

SET2 Type: REAL; dimension MAXFCT
Reactive/imaginary component; see SET1.

VSREF Type: INTEGER; dimension MAXFCT
When MODE = 4 or 7 or 8, series voltage reference of SET1 and SET2. 0 for sending end voltage, 1 for series current.

REMOT Type: INTEGER; dimension MAXFCT
Bus number of a remote type 1 or 2 bus whose voltage is to be regulated by the shunt element of this FACTS device to the value specified by VSET. If bus REMOT is other than a type 1 or 2 bus, the shunt element regulates voltage at the sending end bus to the value specified by VSET. REMOT is entered as zero if the shunt element is to regulate voltage at the sending end bus and must be zero if the sending end bus is a type three (swing) bus.

MNAME Type: CHARACTER∗12; dimension (MAXFCT)
The name of the FACTS device which is the IPFC master device when this FACTS device is the slave device of an IPFC (i.e., its MODE is specified as 6 or 8). MNAME must be enclosed in single or double quotes if it contains any blanks or special characters.

PBRDG Type: REAL; dimension MAXFCT
Real power demand to the shunt at the sending bus, in MW.

QSHNT Type: REAL; dimension MAXFCT
Reactive power demand to the shunt at the sending bus, in Mvar.

PSEND Type: REAL; dimension MAXFCT
Real power demand to the series element at the sending bus in MW.

QSEND Type: REAL; dimension MAXFCT
Reactive power demand to the series element at the sending bus in Mvar.

PTERM Type: REAL; dimension MAXFCT
Real power demand at the terminal bus in MW.

QTERM Type: REAL; dimension MAXFCT
Reactive power demand at the terminal bus in Mvar.

IERR Type: INTEGER; scalar
Will contain the return code:

0 No error.
1 Case not open.
2 Other error; values returned are undependable.

The FACTS device data is not retrieved in any particular order, i.e., there is no relationship between the FACTS devices that can be determined from their order or position in the above arrays. PSSFCT will attempt to use MAXFCT elements and return NFACTS values in each data array; it is the users' responsibility to ensure that each array used for any argument is properly dimensioned. NFACTS and MAXFCT can be retrieved by PSSSIZ prior to calling PSSFCT. Refer to *PSS®E Program Operation Manual*, Section 5.2, Reading Power Flow Raw Data into the Working Case, for more details on the data contained in these arrays.

2.25.1 Older Cases

For cases prior to PSS®E-30, there is no data for array RMPCT. Values are assigned according to defaults in PSS®E activity CASE.
For cases prior to PSS®E-31, NAME is set to FACTS#, where # is a sequence number, REMOT is set to 0 and MNAME is set to blank (see PSS2DC, Section 2.16 PSS2DC – Retrieve Two-Terminal Transmission Line Data).

2.26 PSSALD – Retrieve Nominal Values of Adjusted Load Data

```
PSSALD(NUM,ID,ADJFAC,NOMLOD,IERR)
```

NUM	Type: INTEGER; dimension ADJLOD Bus number.
ID	Type: CHARACTER∗2; dimensional ADJLOD Two character load identifier.
ADJFAC	Type: REAL; dimension ADJLOD Adjustment factor.
NOMLOD	Type: REAL; dimension (2, ADJLOD) Nominal constant MVA load.
IERR	Type: INTEGER; scalar Will contain the return code: 0 No error. 1 Case not open. 2 Other error; values returned are undependable.

The data for these loads is not retrieved in any particular order, i.e., there is no relationship between the loads that can be determined from their order or position in the above arrays. PSSALD will attempt to place ADJLOD values in each array (or 2∗ADJLOD for array NOMLOD); it is the user's responsibility to ensure that each array used for any argument is properly dimensioned. ADJLOD can be retrieved using PSSSIZ, prior to calling PSSALD. The *PSS®E Program Operation Manual*, Sections 14.6.3, 14.9.4 and 14.9.5 con-

tain more details on the data contained in these arrays.

2.26.1 Usage Notes

Only the loads with adjustments are returned by PSSALD. The data returned by PSSALD and PSSLOD refer to the same load if, and only if, the arrays NUM, and ID are equal.

2.27 PSSABX – Retrieve Nominal Values of Adjusted Branch Data

```
PSSABX(FRMBUS,TOBUS,CKT,RCTMLT,NMREAC,IERR)
```

FRMBUS — Type: INTEGER; dimension ADJBRN
Bus number of from bus.

TOBUS — Type: INTEGER; dimension ADJBRN
Bus number of to bus.

CKT — Type: CHARACTER*2; dimensional ADJBRN
Two character circuit identifier.

RCTMLT — Type: REAL; dimension ADJBRN
Adjustment factor.

NMREAC — Type: REAL; dimension ADJBRN
Nominal reactance, in pu.

IERR — Type: INTEGER; scalar
Will contain the return code:
0 No error.
1 Case not open.
2 Other error; values returned are undependable.

The data for these branches is not retrieved in any particular order, i.e., there is no relationship between the branches that can be determined from their order or position in the above arrays. PSSABX will attempt to place ADJBRN values in each array; it is the user's responsibility to ensure that each array used for any argument is properly dimensioned. ADJBRN can be retrieved by PSSSIZ, prior to calling PSSABX. The *PSS®E Program Operation Manual*, Sections 14.6.4 and 14.9.10 contains more details on the data contained in these arrays.

2.27.1 Usage Notes

Only the branches with adjustments are returned by PSSABX. The data returned by PSSABX and PSSBRN refer to the same branch if, and only if, the arrays FRMBUS, TOBUS, and CKT are equal.

2.28 PSSIND - Retrieve Induction Machine Data

```
PSSIND(NUM,ID,STA-
TUS,SCODE,DCODE,AREA,ZONE,OWNER,TCODE,BCODE,MBASE,RATEKV,PCO
DE,PSET,H,A,B,D,E,RA,XA,XM,R1,X1,R2,X2,X3,E1,SE1,E2,
SE2,IA1,IA2,IAM,IERR)
```

NUMType: INTEGER; dimension NINDMC
Bus number.

IDType: CHARACTER * 2; dimension NINDMC
Two character machine identifier.

STATUSType: INTEGER; dimension NINDMC
Machine in-service status. One for in-service; zero for out-of-service.

SCODEType: INTEGER; dimension NINDMC
Machine standard code.
=1, for NEMA
=2, for IEC

DCODEType: INTEGER; dimension NINDMC
Machine design code. Following are allowed machine design codes.
=0, for Custom design with equivalent circuit reactances specified
=1, for NEMA Design A
=2, for NEMA Design B / IEC Design N
=3, for NEMA Design C / IEC Design H
=4, for NEMA Design D
=5, for NEMA Design E

AREAType: INTEGER; dimension NINDMC
Area to which the induction machine is assigned.

ZONEType: INTEGER; dimension NINDMC
Zone to which the induction machine is assigned.

OWNERType: INTEGER; dimension NINDMC
Owner to which the induction machine is assigned.

TCODEType: INTEGER; dimension NINDMC
Type of mechanical load torque variation.
=1, for the simple power law
=2, for the WECC model.

BCODEType: INTEGER; dimension NINDDC
Machine base power code.
=1, for 1 for mechanical power (MW) output of the machine
=2, for apparent electrical power (MVA) drawn by the machine
BCODE = 1 by default.

MBASEType: REAL; dimension NINDMC
Machine base power; entered in MW or MVA. This value is specified according to BCODE, and could be either the mechanical rating of the machine or the electrical input. It is necessary only that the per unit values entered for the equivalent circuit parameters match the base power.

RATEKVType: REAL; dimension NINDMC
Machine rated voltage; entered in kV line-to-line, or zero to indicate that machine rated voltage is assumed to be identical to the base voltage of bus I.
RATEKV = 0.0 by default.

PCODEType: INTEGER; dimension NINDMC
Scheduled power code
=1, for machanical power (MW) output of the machine
=2, for electrical real power (MW) drawn by the machine.
PCODE = 1 by default.

PSETType: REAL; dimension NINDMC
Scheduled active power for a terminal voltage at the machine of 1.0 pu of the

machine rated voltage; entered in MW. This value is specified according to PCODE, and is either the mechanical power output of the machine or the real electrical power drawn by the machine. The sign convention used is that PSET specifies power supplied to the machine:
A positive value of electrical power means that the machine is operating as a motor; similarly, a positive value of mechanical power output means that the machine is driving a mechanical load and operating as a motor. No default allowed.

HType: REAL; dimension NINDMC
Machine inertia; in per unit on MBASE base.

A, B, D, EType: REAL; dimension NINDMC
Constants that describe the variation of the torque of the mechanical load with speed. If TCODE is 1 (simple power law model), only D is used; if TCODE is 2 (WECC model), all of these constants are used.

RAType: REAL; dimension NINDMC
Armature resistance, ra (> 0.0); in per unit on the power base MBASE and voltage base RATEKV.

XAType: REAL; dimension NINDMC
Armature leakage reactance, Xa (> 0.0); in per unit on the power base MBASE and voltage base RATEKV.

XMType: REAL; dimension NINDMC
Unsaturated magnetizing reactance, Xm (> 0.0); in per unit on the power base MBASE and voltage base RATEKV.

R1Type: REAL; dimension NINDMC
Resistance of the first rotor winding ("cage"), r1 (> 0.0); in per unit on the power base MBASE and voltage base RATEKV.

X1Type: REAL; dimension NINDMC
Reactance of the first rotor winding ("cage"), X1 (> 0.0); in per unit on the power base MBASE and voltage base RATEKV.

R2Type: REAL; dimension NINDMC
Resistance of the second rotor winding ("cage"), r2 (> 0.0); in per unit on the power base MBASE and voltage base RATEKV.

X2Type: REAL; dimension NINDMC
Reactance of the second rotor winding ("cage"), X2 (> 0.0); in per unit on the power base MBASE and voltage base RATEKV.

X3Type: REAL; dimension NINDMC
Third rotor reactance, X3 (> 0.0); in per unit on the power base MBASE and voltage base RATEKV.

E1Type: REAL; dimension NINDMC
First terminal voltage point from the open circuit saturation curve, E1 (> 0.0); entered in per unit on RATEKV base.

SE1Type: REAL; dimension NINDMC
Saturation factor at terminal voltage E1, S(E1).

E2Type: REAL; dimension NINDMC
Second terminal voltage point from the open circuit saturation curve, E2 (> 0.0); entered in per unit on RATEKV base.

SE2Type: REAL; dimension NINDMC
Saturation factor at terminal voltage E2, S(E2).

IA1, IA2Type: REAL; dimension NINDMC
Stator currents in PU specifying saturation of the stator leakage reactance, XA..

IAMType: REAL; dimension NINDMC
Multiplier for the saturated value. Allowed value 0 to 1.0.

IERRType: INTEGER; scalar
Will contain the return code:
0 No error
1 Case not open.
2 Other error; values returned are undependable.

The induction machine data is not retrieved in any particular order, i.e., there is no relationship between the induction machines that can be determined from their order or position in the above arrays. PSSIND will attermpt to place NINDMC values in each data arraay; it is the user's responsibility to ensure that each array used for any argument is properly dimensioned. NINDMC can be retrieved by PSSSIZ prior to calling PSSIND. Refer to the PSS®E Program Operation Manual, Section 5.2, Reading Power Flow Raw Data into the Working Case, for more details on the data contained in these arrays.

SIEMENS

Appendix A

Access to the USRCAS Subroutine Library

A.1 IBM – PC Microsoft Windows 2000 / Windows XP

The PSS®E saved case data extraction subroutines are installed with PSS®E and are located in the PSSBIN directory in a dynamic link library named USRCAS.DLL. Users may access the USRCAS routines by linking USRCAS.LIB, which is located in the PSSLIB directory, with their executables. At run time, the executable program will need access to the USRCAS.DLL, PTIUTILS.DLL, CRUTCH.DLL, and FILEINFO.DLL files. These files are located in the PSSBIN directory. The executable will also need access to the Fortran runtime libraries.

This page intentionally left blank.

SIEMENS

Appendix B

Extracting Saved Case Data in Python

B.1 Introduction

Module Name: caspy

Purpose: Extract PSS®E save case data in Python

The module defines the class Savecase. Creating an instance of this class opens, reads, and closes a PSS®E saved case. That instance object can then be used as follows:

- All retrievable data is stored as dictionary attributes of the instance object
- The attribute names are the documented subroutine names
- The dictionary keys are the documented subroutine arguments
- Arrays are returned as tuples
- The dictionary values can also be referenced as attributes (see example below)
- Case sensitivity has been overridden for the dictionary keys and corresponding attributes (see example below)

B.2 Access Procedure

The process is described in the following examples, using a local saved case file named *sample.sav*.

1. Import the caspy module:

```
>>> import caspy
```

This is only done once per execution.

2. Create the Python object for the PSS®™E saved case:

```
>>> sample = caspy.Savecase('sample')
```

Alternatively:

```
>>> sample = caspy.Savecase(casnam='sample')  OR
>>> sample = caspy.Savecase(casnam='sample.sav', LU=11)
```

The Python class name Savecase is case-sensitive. The casnam and LU arguments are defined for the PSSOPN subroutine, and are also case-sensitive.

3. Once the Savecase object is created, all Saved Case Data Extraction subroutines and their respective argument data can be extracted either as attributes or dictionary items.

   All data of a specific Saved Case Data Extraction subroutine can be extracted as an attribute of the Savecase Python object created in Step 1. The attribute name should be in lowercase. Example:

```
>>> sample.pssopn
{'LU': 11, 'IERR': 0}

>>> sample.pssznm
{'IERR': 0, 'ZONAME': ('NORTH_A1      ', 'MID_A1_A2_A5',
'DISCNT_IN_A1', 'SOUTH_A1_A5 ', 'ALL_A3      ', 'NORTH_A5    ',
'NORTH_A2     ', 'SOUTH_A2     ', 'ALL_A4_A6    ')}

>>> sample.pss3ix
{'IDX2BR': (41, 44, 47, 52), 'IDXBUS': (40, 41, 42, 43),
'IDX1BR': (40, 43, 46, 51), 'IERR': 0, 'IDX3BR': (42, 45, 48,
53)}
```

   Data for a particular argument of a specific subroutine can be extracted as an attribute or as a dictionary item. If the returned data is tuple, it can be further accessed with a tuple index. The attributes or dictionary keys are case-insensitive. For example, any one of the following four lines:

```
>>> sample.pssznm.zoname
>>> sample.pssznm.ZONAME
>>> sample.pssznm['zoname']
>>> sample.pssznm['ZONAME']
```

   will produce return this tuple:

```
('NORTH_A1      ', 'MID_A1_A2_A5', 'DISCNT_IN_A1',
'SOUTH_A1_A5 ', 'ALL_A3      ', 'NORTH_A5    ', 'NORTH_A2
', 'SOUTH_A2     ', 'ALL_A4_A6    ')
```

   Either of the following two lines:

```
>>> sample.pssznm.zoname[0]
>>> sample.pssznm['ZONAME'][0]
```

   will produce this value:

```
'NORTH_A1     '
```

B.3 Usage Notes

Input Arguments

Input arguments will not appear in the dictionaries. Returned values of input/output arguments will.

Multiple Cases

The data for a given case is part of the Savecase instance object, and the saved case file is closed once the object is created. Any number of Savecase instance objects may exist at one time.

Data Order

The user should not depend on the order of the elements in a Python dictionary. The order of elements in a particular tuple will be the same as the order of the corresponding array as documented.

Object Lifetime

As long as a reference to the Savecase instance object exists, the data will remain in memory. Reusing the name (i.e., binding the name to a new object) may result in the data being removed from memory. PSS®E saved cases may be quite large; use the del command to remove references to objects (or delattr for specific object attributes) if you are concerned about the performance impact of retaining the data in memory longer than needed.

This page intentionally left blank.