IOWA STATE UNIVERSITY
ECpE Department

OpenDSS - DSS Command

Dr. Zhaoyu Wang
1113 Coover Hall, Ames, IA
wzy@iastate.edu

Contents

- DSS command language syntax
- DSS command reference

Dss command language syntax

The DSS is designed such that all functions can be carried out through text-based DSS Command Language scripts. The text streams may come from any of these sources:

1. Selecting and executing a script on a Control Panel window,
2. Through the COM interface, or
3. From a standard text file to which the command interpreter may be temporarily redirected (Compile or Redirect commands).

This makes the DSS an easily accessible tool for users who simply want to key in a small circuit and do a quick study as well as to those who perform quite complicated studies. It also makes the DSS more easily adapted by others who have a great deal invested in their own database and would put forth much effort to conform to another. Text files are the most common means of transferring data from one source into another. The DSS scripts can often be configured to be close to various data transfer formats.

Always refer to the **Help** command for the latest commands and property names that are recognized by the DSS.

Dss command language syntax
-- Command syntax

The command language is of the form:

Command parm1, parm2 parm3 parm 4 ….

Parameters (parm1, etc) may be separated by commas (,) or white space (blank, tab). If a parameter includes a delimiter, enclose it in either

- double quotes (“),
- single quotes(‘), or
- parentheses (…), or..
- brackets [..], or
- braces {..}.

While any of these will work, double or single quotes are preferred for strings. Brackets are preferred for arrays, and curly braces are preferred for in-line math.

Note: Be careful of using parantheses on Windows & for file names containing full path names because Windows uses the string (x86) for 32-bit programs in the default Program Files folders. Use some other delimiter. Double quotes work fine.

Dss command language syntax
-- Parameters

Parameters may be positional or named (tagged). If named, an "=" sign is expected.

1. *Name=value* (this is the named form)
2. *Value* (value alone in positional form)

For example, the following two commands are equivalent.

```
New Object="Line.First Line" Bus1=b1240 Bus2=32 LineCode=336ACSR,
…
New "Line.First Line", b1240 32 336ACSR, …
```

The first example uses named parameters, which are shown in the default order. The second example simply gives the values of the parameters and the parser assumes that they are in the default order. Note that the name of the object contains a blank, which is a standard DSS delimiter character. Therefore, it is enclosed in quotes or parentheses, etc..

Dss command language syntax

-- Parameters

You may mix named parameters and positional parameters. Using a named parameter repositions the parser's positional pointer so that subsequent parameters need not be named. The order of parameters is always given in the DSS help command window. The DSS help window allows for display of commands and properties in either alphabetical order or positional (numerical) order. The properties of elements are by default processed in positional order unless the "=" appears in the field.

Some commands are interpreted at more than one lexical level inside the OpenDSS. In this example, the main DSS command interpreter interprets the New command and essentially passes the remainder of the string to the Executive for adding new circuit elements. It determines the type of element to add (e.g., a Line) and confirms that it is indeed a registered class. It then passes the remainder of the text line on to the module that handles the instantiation and definitions of Line objects. Only the Line model code needs to know how to interpret the property definitions it receives via the parameter list. This design allows great flexibility for future modifications to the Line model that might require adding new properties. This happens with regularity and the main DSS executive does not have to be modified; only the module affected.

Dss command language syntax
-- Parameters

For the New command, the first two parameters are always required and positional:

- The New command itself, and
- The name of the object to add.

For circuit elements, the next one, or two, parameters are normally the bus connection properties, which are processed and stored with the circuit element model. Then the definition of the object being created continues, using an editing function expressly devoted to that class of circuit element.

Dss command language syntax
-- Properties

- The parameters of circuit element editing commands are referred to as “properties”. Properties generally set values of some data field in the targeted object, but may also have some side effects. Properties behave like properties in object-oriented programming languages. They may perform an action as well as setting a value.
- For example, setting the **PF** property of a Load object also causes the **kvar** property to be updated.
- Many objects have multiple properties that essentially set the same internal data value. For example, you can set the kVA rating by either the **kVA** or **MVA** property is some elements.
- This is a different approach than programs using databases typically take where the values are static and the data fields generally fixed.

Dss command language syntax
-- Delimiters and other special characters

The DSS recognizes these delimiters and other special characters:

Array or string delimiter pairs:

- []
- {}
- ()
- “”
- ‘’

Matrix row delimiter: | (vertical bar)

Value delimiters:

- Comma(,)
- Any white space (tab or space)

Keyword / value separator: =

Dss command language syntax
-- Delimiters and other special characters

Continuation of previous line: ~ (actually a synonym for the More command)

Comment line: // or !

In-line comment: !

Query a property: ?

Dss command language syntax

-- Array properties [and quote pairs]

Array parameters are sequences of numbers. Of necessity, delimiters must be present to separate the numbers. To define an array, simply enclose the sequence in any of the quote pairs. Square brackets, [..], are the preferred method, although any of the other quote pairs: “..”, ‘..’, (..), {..} will work. For example, for a 3-winding transformer you could define arrays such as:

```
kvs = [115, 6.6, 22]
kvas=[20000 16000 16000]
```

Dss command language syntax
-- Array properties [and quote pairs]

Standard Ways to Define Array Properties:

Arrays can be defined by the following methods:

Entering the numeric values directly

```
mult=[1, 2, 3, 4, ...]
```

Entering the numbers from a single column text file:

```
mult=[File=MyTextDataFile.CSV]
```

Entering the data from a packed binary file of doubles:

```
mult=[dblFile=MyFileOfDoubles.dbl]
```

Entering the data from a packed binary file of singles:

```
mult=[sngFile=MyFileOfDoubles.sng]
```

Dss command language syntax
-- Array properties [and quote pairs]

Enhanced Syntax:

The 'File=...' capability for text files has been enhanced to allow you to extract a column of data from a multi-column CSV file (like you might get from an Excel spreadsheet or the OpenDSS Export command). The complete syntax is: Entering the numeric values directly

```
mult=[File=myMultiColumnFile.CSV, Column=n, Header=Yes/No]
```

Enter n for the column number you want. The default is always set to 1 for each array.

Specifying Header=Yes causes the first record in the CSV file to be skipped. This allows for a single line of non-numeric data in the first line, as is common in OpenDSS Export files. Default is “Header=No”.

Example

```
New Loadshape.Ramp2 npts=4000 sInterval=1
~ mult= (file=MultiChannelTest.csv, column=3, header=yes)
```

Dss command language syntax
-- Array properties [and quote pairs]

Special Reserved File Name:

The name **`%result%`** is now used to designate the last result file. For example, if you want to export a file (such as a monitor file) and then immediately read in column 5 to do something with, you could do it with the following syntax:

```
mult=[File=%result%, Column=5, Header=Yes]
```

This must be executed before another command is issued that writes a result file.

Dss command language syntax
-- Matrix properties

Matrix parameters are entered by extending the Array syntax: Simply place a vertical bar (|) character between rows, for example:

```
Xmatrix=[1.2 .3 .3 | .3 1.2 3 | .3 .3 1.2] ! (3x3 matrix)
```

Symmetrical matrices like this example may also be entered in lower triangle form to be more compact.

```
Xmatrix=[ 1.2 | .3 1.2 | .3 .3 1.2 ] ! (3x3 matrix lower
triangle)
```

Dss command language syntax
-- String length

The DSS Command strings may be as long as can be reasonably passed through the COM interface. This is very long. They must generally NOT be split into separate "lines" since there is no concept of a line of text in the standard DSS COM interface; only separate commands. However, New and Edit commands can be continued on a subsequent text line with the More or ~ command.

In general, string values in DSS scripts can be as long as desired. There are, of course, limitations to what can be reasonably deciphered when printed on reports. Also, the DSS "hashes" bus names, device names and any other strings that are expected to result in long lists for large circuits. This is done for fast searching. There have been various hashing algorithms implemented and it is difficult to keep the documentation up with the present release. Some algorithms would hash only the first 8 characters. This does not mean that comparisons are only 8 characters; comparisons are done on full string lengths. slow performance if there are thousands of names in the list.

Dss command language syntax
-- String length

Abbreviations are allowed by default in DSS commands and element Property names. However, if the circuit is very large, script processing will actually proceed more efficiently if the names of commands and properties are spelled out completely. It won't matter on small circuits. The reason is that the hash lists are much shorter than the linear lists. If the DSS does not find the abbreviated name in a hash list, it will then search the whole list from top to bottom. This can slow performance if there are thousands of names in the list.

Dss command language syntax
-- Default values

When a New command results in the instantiation of a DSS element, the element is instantiated with reasonable values. Only those properties that need to be changed to correctly define the object need be included in the command string. Commonly, only the bus connections and a few key properties need be defined. Also, a new element need not be defined in one command line. It may be edited as many times as desired with subsequent commands (see Edit, More and ~ commands).

When an element is created or selected by a command, it becomes the *Active* element. Thereafter, property edit commands are passed directly to the active element until another element is defined by the New command or selected by some other command. In that respect, the command language mirrors the basic COM interface.

Dss command language syntax
-- Default values

All changes are persistent. That is, a parameter changed with one command remains as it was defined until changed by a subsequent command. This might be a source of misunderstanding with novices using the program who might expect values to reset to a base case as they do in some other programs. When this is a concern, issue a "clear" command and redefine the circuit from scratch.

Changes made "on the fly" are NOT saved back to the original script files. If you wish to capture the present state of the circuit for future analysis, execute the Save Circuit command, which saves the present circuit to a separate folder. It will not overwrite an existing folder. Some manual fixup may be required to get the saved circuit to compile properly.

Dss command language syntax
-- In-line math

The DSS scripting language uses a form of Reverse Polish Notation (RPN) to accommodate in-line math. This feature may be used to pre-process input data before simulation. The parser will evaluate RPN expressions automatically if you enclose the expression in any of the quoted formats (quotation marks or any of the matched parentheses, brackets, etc.).

- **RPN Expressions**

Basically, you enter the same keystrokes as you would with an RPN calculator (such as an HP 48). One difference is that you separate numbers, operators, and functions by a space or comma. The RPN calculator in OpenDSS has a "stack" of 10 registers just like some calculators. In the function descriptions that follow, the first stack register is referred to as "X" and the nex higher as "Y" as on the HP calculator. When a new number is entered, the existing registers are rolled up and the new number is inserted into the X register.

…

Dss command language syntax

-- In-line math

- **RPN Expressions**

…

These functions operate on the first two registers, X and Y:

```
+ Add the last two operands (X and Y registers; result in X; X = X + Y )
- X = Y - X
* X = X * Y
/ X = Y / X
^ (exponentiation) X = Y to the X power
```

These unitary functions operate on the X register and leave the result in X.

```
sqr X=X*X
sqrt take the square root of X
inv (inverse of X = 1/X)
ln (natural log of X)
exp (e to the X)
log10 (Log base 10 of X)
sin for X in degrees, take the sine
cos for X in degrees, take the cosine
```

Dss command language syntax

-- In-line math

- **RPN Expressions**

…

tan for X in degrees, take the tangent

asin take the inverse sine, result in degrees

acos take the inverse cosine, result in degrees

atan take the arc tangent, result in degrees

atan2 take the arc tangent with two arguments, Y=rise and X = run, result in degrees over all four quadrants

The following functions manipulate the stack of registers

Rollup shift all registers up

Rolldn shift all registers dn

Swap swap X and Y

There is one constant preprogrammed: **pi**

Dss command language syntax
-- In-line math

- **RPN Examples**

For example, the following DSS code will calculate X1 for a 1-mH inductance using inline RPN:

```
// convert 1 mH to ohms at 60 Hz, note the last * operator
line.L1.X1 = (2 pi * 60 * .001 *)
```

The expression in parentheses is evaluated left-to-right. '2' is entered followed by 'pi'. Then the two are multiplied together yielding 2π. The result is then multiplied by 60 to yield ω(2πf). Finally, the result is multiplied by 1 mH to yield the reactance at 60 Hz. To specify the values of an array using in-line math, simply nest the quotes or parentheses:

```
// Convert 300 kvar to 14.4 kV, 2 steps
Capacitor.C1.kvar = [(14.4 13.8 / sqr 300 *), (14.4 13.8 / sqr
300 *)]
```

…

Dss command language syntax
-- In-line math

- **RPN Examples**

…

The Edit | RPN Evaluator menu command brings up a modal form in which you can enter an RPN expression and compute the result. Clicking the OK button on this form automatically copies the result to the clipboard. The purpose of this feature is to enable you to interpret RPN strings you find in the DSS script or to simplify the above script by computing the result and replacing the RPN string with the final value if you choose:

```
Capacitor.C1.kvar = [ 326.65, 326.65] ! 300 kvar converted to
14.4 kV, 2 steps
```

This next example shows how to use RPN expressions inside an array. Two different delimiter types are necessary to differentiate the array from the expressions.

```
// set the winding kvs to (14.4 20)
New Transformer.t kvs=("24.9 3 sqrt /" "10 2 *")
```

Contents

- DSS command language syntax
- DSS command reference

Dss command reference

Nearly all DSS commands and parameter names may be abbreviated. This is for convenience when typing commands in directly. However, there is not necessarily any speed benefit to abbreviating for machine-generated text. (See String Length above.) Thus, commands and parameter names should be spelled out completely when placed in script files that are auto-generated from other computer data sources. Abbreviate only when manually typing commands to the DSS.

Dss command reference

-- Specifying objects

Any object in the DSS, whether a circuit element or a general DSS object, can be referenced by its complete name:

```
Object=Classname.objname
```

For example,

```
object=vsource.source.
```

In nearly all circumstances, the 'object=' may be omitted as it can for any other command line parameter. The object name is almost always expected to be the first parameter immediately following the command verb.

If the 'classname' prefix is omitted (i.e., no dot in the object name), the previously used class (the active DSS class) is assumed. For example,

```
New line.firstline . . .
New secondline . . .
```

Recommendation: Always used the full element name (classname.elementname) to avoid confusion at a later date. It is better to make the text human readable than exploit shortcuts the program allows.

Dss command reference

-- Command reference

The following documents selected command definitions as of this writing. There are **98** commands as of this writing. Newer builds of the DSS may have additional properties and commands. Execute the Help command while running the DSS to view the present commands available in your version.

// (comment) and ! (inline comment)

The appearance of "//" in the command position indicates that this statement is a comment line. It is ignored by the DSS. If you wish to place an in-line comment at the end of a command line, use the "!" character. The parser ignores all characters following the ! character.

```
// This is a comment line
New line.line4 linecode=336acsr length=2.0 ! this is an in-line
comment
```

Dss command reference

-- Command reference

/* … */ Block Comments

New at version 7.6, you can now comment out whole sections (whole lines of script) using the block comment capability. The block comment must begin with /* in the FIRST column of the line. The block comment terminates after the appearance of */ anywhere in a line or with the end of a script file or selection in a script window. Example:

```
Compile "C:\Users\prdu001\DSSData\CDR\Master3.DSS"
New Monitor.Line1-PQ Line.LINE1 1 mode=1 ppolar=no
New Monitor.Line1-VI Line.LINE1 1 mode=0 VIpolar=Yes
/* comment out the next two monitors
New Monitor.Source-PQ Vsource.source 1 mode=1 ppolar=no
New Monitor.source-VI Vsource.source 1 mode=0 VIpolar=Yes
****/ End of block comment
New Monitor.Tran2-VI Transformer.PHAB 2 mod=0 VIPolar=no
New Monitor.Tran3-VI Transformer.PHAB 3 mod=0 VIPolar=no
Solve
```

Dss command reference
-- Command reference

Customizing Solution Processes

The next seven commands, all beginning with an underscore ('_') character, allow you to script your own solution process by providing access to the different steps of the solution process.

_DoControlActions

For step control of solution process: Pops control actions off the control queue according to the present control mode rules. Dispatches contol actions to proper control element "DoPendingAction" handlers.

_InitSnap

For step control of solution process: Intialize iteration counters, etc. that normally occurs at the start of a snapshot solution process.

_SampleControls

For step control of solution process: Sample the control elements, which push control action requests onto the control queue.

…

Dss command reference
-- Command reference

Customizing Solution Processes

…

_ShowControlQueue

For step control of solution process: Show the present control queue contents.

_SolveDirect

For step control of solution process: Invoke direct solution function in DSS. Non-iterative solution of Y matrix and active sources only.

_SolveNoControl

For step control of solution process: Solves the circuit in present state but does not check for control actions.

_SolvePFlow

For step control of solution process: Invoke iterative power flow solution function of DSS directly.

Dss command reference
-- Command reference

Customizing Solution Processes

…

_ShowControlQueue

For step control of solution process: Show the present control queue contents.

_SolveDirect

For step control of solution process: Invoke direct solution function in DSS. Non-iterative solution of Y matrix and active sources only.

_SolveNoControl

For step control of solution process: Solves the circuit in present state but does not check for control actions.

_SolvePFlow

For step control of solution process: Invoke iterative power flow solution function of DSS directly.

Dss command reference
-- Command reference

AddBusMarker

Add a marker to a bus in a circuit plot. Markers must be added before issuing the Plot command. Effect is persistent until circuit is cleared or ClearBusMarkers command is issued. Example:

```
ClearBusMarkers !...Clears any previous bus markers
AddBusMarker Bus=Mybusname1 code=5 color=Red size=3
AddBusMarker Bus=Mybusname2 code=5 color=Red size=3
...
```

You can use any of the standard color names or RGB numbers. See Help on C1 property in Plot command.

To clear the present definitions of bus markers, issue the ClearBusMarkers command. If you specify a busname that doesn't exist, it is simply ignored. See Set Markercode= description below to see a list of marker codes presently implemented.

Dss command reference
-- Command reference

AllocateLoads

Estimates the allocation factors for loads that are defined using the XFKVA property. Requires that energymeter objects be defined with the PEAKCURRENT property set. Loads that are not in the zone of an energymeter cannot be allocated. This command adjusts the allocation factors for the appropriate loads until the best match possible to the meter values is achieved. Loads are adjusted by phase. Therefore all single-phase loads on the same phase will end up with the same allocation factors.

If loads are not defined with the XKVA property, they are ignored by this command.

Dss command reference
-- Command reference

BusCoords

Define x,y coordinates for buses. Execute after Solve or MakeBusList command is executed so that bus lists are defined.Reads coordinates from a CSV file with records of the form:

```
busname, x, y.
```

You may use spaces and tabs as well as commas for value separators.

Example:

```
BusCoords [file=]xxxx.csv
```

See also **`LatLongCoords`**.

Dss command reference

-- Command reference

CalcVoltageBases

Estimates the voltage base for each bus based on the array of voltage bases defined with a "**`SetVoltagebases=...`**" command. Performs a zero-current power flow considering only the series power-delivery elements of the system. No loads, generators, or other shunt elements are included in the solution. The voltage base for each bus is then set to the nearest voltage base specified in the voltage base array.

Alternatively, you may use the SetkVBase command to set the voltage base for each bus individually. Note that the OpenDSS does not need the voltage base for most calculations, but uses it for reporting. Exceptions include processes like the AutoAdd solution mode where the program needs to specify the voltage rating of capacitors and generators it will automatically add to the system. Also, some controls may need the base voltage to work better.

…

Dss command reference

-- Command reference

CalcVoltageBases

…

It is useful to show the bus voltages after the execution of this command. This will help confirm that everything in the circuit is connected as it should be. It is especially useful for devices with unusual connections. Supplement this check with a Faultstudy mode solution to verify that the system impedances are also properly specified.

Dss command reference
-- Command reference

CktLosses

Returns the total losses for the active circuit in the Result string in kW, kvar.

Clear

Clears all circuit element definitions from the DSS. This statement is recommended at the beginning of all Master files for defining DSS circuits.

ClearBusMarkers

Clear all bus markers created with the AddBusMarker command.

Currents

Returns the currents for each conductor of ALL terminals of the active circuit element in the Result string. (See Select command.) Returned as comma-separated magnitude and angle.

Dss command reference

-- Command reference

Disable [Object]

Disables object in active circuit. All objects are Enabled when first defined. Use this command if you wish to temporarily remove an object from the active circuit, for a contingency case, for example. If this results in isolating a portion of the circuit, the voltages for those buses will be computed to be zero. (Also see Open, Close commands.)

Edit [Object] [Edit String]

Edits the object specified. The object Class and Name fields are required and must designate a valid object (previously instantiated by a New command) in the problem. Otherwise, nothing is done and an error is posted.

The edit string is passed on to the object named to process. The DSS main program does not attempt to interpret property values for circuit element classes. These can and do change periodically.

Dss command reference
-- Command reference

Enable [Object]

Cancels a previous **`Disable`** command. All objects are automatically Enabled when first defined. Therefore, the use of this command is unnecessary until an object has been first disabled. (Also see Open, Close commands.)

Export <Quantity> [Filename or switch]

Writes a text file (.CSV) of the specified quantity for the most recent solution. Defaults to Export Voltages. The purpose of this command is to produce a file that is readily readable by other programs such as MATLAB (use csvread), spreadsheet programs, or database programs.

The first record is a header record providing the names of the fields. The remaining records are for data. For example, the voltage export looks like this:

…

Dss command reference

-- Command reference

Export <Quantity> [Filename or switch]

…

```
Bus, Node Ref., Node, Magnitude, Angle, p.u., Base kV
sourcebus , 1, 1, 6.6395E+0004, 0.0, 1.000, 115.00
sourcebus , 2, 2, 6.6395E+0004, -120.0, 1.000, 115.00
sourcebus , 3, 3, 6.6395E+0004, 120.0, 1.000, 115.00
subbus , 4, 1, 7.1996E+0003, 30.0, 1.000, 12.47
subbus , 5, 2, 7.1996E+0003, -90.0, 1.000, 12.47
subbus , 6, 3, 7.1996E+0003, 150.0, 1.000, 12.47
```

This format is common for many spreadsheets and databases, although databases may require field types and sizes for direct import. The columns are aligned for better readability.

Valid syntax for the command can be one of the following statement prototypes in bold. If the Filename is omitted, the file name defaults to the name shown in italics in parantheses. (This list is incomplete. Check Help for the Export commands on line. When in doubt, execute the export and observe the result.)

…

Dss command reference
-- Command reference

Export <Quantity> [Filename or switch]

…

Export Voltages [Filename] (EXP_VOLTAGES.CSV). Exports voltages for every bus and active node in the circuit. (Magnitude and angle format).

Export SeqVoltages [Filename] (EXP_SEQVOLTAGES.CSV) Exports the sequence voltage magnitudes and the percent of negative- and zero-sequence to positive sequence.

Export Currents [Filename] (EXP_CURRENTS.CSV) Exports currents in magnitude and angle for each phase of each terminal of each device.

Export Overloads [Filename] EXP_OVERLOADS.CSV) Exports positive sequence current for each device and the percent of overload for each power delivery element that is overloaded.

Export SeqCurrents [Filename] (EXP_SEQCURRENTS.CSV) Exports the sequence currents for each terminal of each element of the circuit.

…

Dss command reference
-- Command reference

Export <Quantity> [Filename or switch]

…

Export Powers [MVA] [Filename] (EXP_POWERS.CSV). Exports the powers for each terminal of each element of the circuit. If the MVA switch is specified, the result are specified in MVA. Otherwise, the results are in kVA units.

Export Faultstudy [Filename] (EXP_FAULTS.CSV) Exports a simple report of the 3- phase, 1-phase and max L-L fault at each bus.

Export Loads [Filename] (EXP_LOADS.CSV) Exports the follow data for each load object in the circuit: Connected KVA, Allocation Factor, Phases, kW, kvar, PF, Model.

Export Monitors monitorname (file name is assigned) Automatically creates a separate filename for each monitor. Exports the monitor record corresponding the monitor's mode. This will vary for different modes.

…

Dss command reference

-- Command reference

Export <Quantity> [Filename or switch]

…

Export Meters [Filename | /multiple] (EXP_METERS.CSV)

Export Generators [Filename | /multiple] (EXP_GENMETERS.CSV)

EnergyMeter and Generator object exports are similar. Both export the time and the values of the energy registers in the two classes of objects. In contrast to the other Export options, each invocation of these export commands appends a record to the file.

For Energymeter and Generator, specifying the switch "/multiple" (or /m) for the file name will cause a separate file to be written for each meter or generator. The default is for a single file containing all meter or generator elements.

Export Yprims [Filename] (EXP_Yprims.CSV). Exports all primitive Y matrices for the present circuit to a CSV file.

…

Dss command reference

-- Command reference

Export <Quantity> [Filename or switch]

…

Export Y [Filename] (EXP_Y.CSV). Exports the present system Y matrix to a CSV file. Useful for importing into another application. Note: This file can be HUGE!

Export SeqZ [Filename] (EXP_SEQZ.CSV). Exports the equivalent sequence short circuit impedances at each bus. Should be preceded by a successful "Solve Mode=Faultstudy" command. This will initialize the short circuit impedance matrices at each bus.

Dss command reference

-- Command reference

Get [Opt1] [opt2] etc.

Basically, the opposite of the SET command. Returns DSS property values for options set using the Set command. Result is returned in the Result property of the Text interface.

VBA Example:

```
DSSText.Command = "Get mode"
Answer = DSSText.Result
```

Multiple properties may be requested on one get. The results are appended and the individual values separated by commas. Array values are returned separated by commas.

Dss command reference
-- Command reference

LatLongCoords

Define x,y coordinates for buses using Latitude and Longitude values (decimal numbers). Similar to **`BusCoords`** command. Execute after Solve command or MakeBusList command is executed so that bus lists are defined. Reads coordinates from a CSV file with records of the form:

```
busname, Latitude, Longitude.
```

Example:

```
LatLongCoords [file=]xxxx.csv
```

Note: Longitude is mapped to x coordinate and Latitude is mapped to y coordinate.

Losses

Returns the **`total`** losses for the active circuit element (see Select command) in the Result string in kW, kvar.

Dss command reference
-- Command reference

MakeBusList

Updates the buslist, if needed, using the currently enabled circuit elements. (This happens automatically for Solve command.) See **`ReprocessBuses`**.

New [Object] [Edit String]

Adds an element described on the remainder of the line to the active circuit. The first parameter (Object=…) is required for the New command. Of course, "Object=" may be omitted and often is for aesthetics.

The remainder of the command line is processed by the editing function of the specified element type. All circuit objects are instantiated with a reasonable set of values so that they can likely be included in the circuit and solved without modification. Therefore, the Edit String need only include definitions for property values that are different than the defaults.

…

Dss command reference

-- Command reference

New [Object] [Edit String]

…

```
Examples:
New Object=Line.Lin2 ! Min required
New Line.Lin2 ! Same, sans object= …
!Line from Bs1 to Bs2
New Line.Lin2 Bs1 Bs2 R1=.01 X1=.5 Length=1.3
```

The Edit String does not have to be complete at the time of issuing the New command. The object instantiated may be edited again at any later time by invoking the Edit command or by continuing with the next command line (see More or ~). The 'later time' does not have to occur immediately after definition. One can make up a script later that edits one or more object properties.

…

Dss command reference

-- Command reference

New [Object] [Edit String]

…

Immediately after issuing the New command, the instantiated object remains the active object and the Edit command does not have to be given to select the object for further editing. You can simply issue the **`More`** command, or one of its abbreviations (~), and continue to send editing instructions. Actually, the DSS command interpreter defaults to editing mode and you may simply issue the command

```
Property=value …(and other editing statements)
```

The "=" is required when using this format. When the DSS parser sees this, it will assume you wish to continue editing and are not issuing a separate DSS Command. To avoid ambiguity, which is always recommended for readability, you may specify the element completely:

```
Class.ElementName.Property = Value
```

…

Dss command reference

-- Command reference

New [Object] [Edit String]

…

More than one property may be set on the same command, just as if you had issued the New or Edit commands.

```
Class.ElementName.Property1 = Value1 property2=value2 …
```

Note that all DSS objects have a Like property inherited from the base class. Users are somewhat at the mercy of developers to ensure they have implemented the Like feature, but this has been done on all DSS objects to date. When another element of the same class is very similar to a new one being created, use the Like parameter to start the definition then change only the parameters that differ. Issue the Like=nnnn property first. Command lines are parsed from left to right with the later ones taking precedence. Throughout the DSS, the design goal is that a property persistently remains in its last state until subsequently changed.

…

Dss command reference

-- Command reference

New [Object] [Edit String]

…

```
New Line.Lin3 like=Lin2 Length=1.7
```

While all devices have a Like property, for Line objects and Transformer objects, users generally prefer to use the Linecode and Xfmrcode properties instead when objects have the same properties.

Dss command reference

-- Command reference

[Object Property Name] = value

This syntax permits the setting of any published property value of a DSS circuit element. Simply specify the complete element and property name, "=", and a value. For example,

```
Monitor.mon1.mode=48
```

Sets the monitor mode queried in the previous example. The DSS command interpreter defaults to the Edit command. If it does not recognize the command it looks for the "=" and attempts to edit a property as specified. Thus, in this example, this method simply invokes the Monitor object's editor and sets the value. If there is more text on the string, the editor continues editing. For example,

```
Line.line1.R1=.05 .12 .1 .4
```

will set the R1, X1, R0, X0 properties of Line.line1 in sequence using positional property rules.

This is a convenient syntax to use to change properties in circuit elements that have already been defined.

Dss command reference
-- Command reference

Open [Object] [Term] [Cond]

Opens a specified terminal conductor switch. All conductors in the terminals of all circuit elements have an inherent switch. This command can be used to open one or more conductors in a specified terminal. If the 'Cond=' field is 0 or omitted, all phase conductors are opened. Any other conductors there might be are unaffected. Otherwise, open one conductor at a time (one per command). For example:

```
Open object=line.linxx term=1 ! opens all phase conductors of
terminal 1 of linxx
Open line.linxx 2 3 ! opens 3rd conductor of 2nd terminal of
linxx
line object
Open load.LD3 1 4 ! opens neutral conductor of wye-connected 3-
phase
load
```

No action is taken if either the terminal or conductor specifications are invalid.

…

voltage computes to zero. But you have to worry less about it causing a floating point exception.

Dss command reference

-- Command reference

Open [Object] [Term] [Cond]

…

Note, this action disconnects the terminal from the node to which it is normally connected. The node remains in the problem. If it becomes isolated, a tiny conductance is attached to it and the voltage computes to zero. But you have to worry less about it causing a floating point exception.

PhaseLosses

Returns the losses for the active circuit element (see Select command) for each PHASE in the Result string in comma-separated kW, kvar pairs.

Dss command reference

-- Command reference

Plot (options …)

Check the Help in the EXE version for additions to the Plot command. There are many options to the Plot command and it has been moved to its own branch in the Help tree.

Plot is a rather complex command that displays a variety of results in a variety of manners on graphs. You should use the control panel to execute the plot command with the recorder on to see examples of how to construct the plot command. Implemented options include (in order):

…

Dss command reference

-- Command reference

Plot (options …)

…

Type = {Circuit | Monitor | Daisy | Zones | AutoAdd | General (bus data) | Loadshape | Tshape | Priceshape |Profile}. A Circuit plot requires that the bus coordinates be defined. By default the thickness of the circuit lines is drawn proportional to power. A Daisy plot is a special circuit plot that shows a unique symbol for generators. (When there are many generators at the same bus, the plot resembles a daisy.) The Monitor plot plots one or more channels from a Monitor element. The Zones plot draws the energymeter zones. Autoadd shows autoadded elements on the circuit plot. General expects a CSV file of bus data with bus name and a number of values. Specify which value to plot in Quantity= property. Bus colors are interpolated based on the specification of C1, C2, and C3.

…

Dss command reference
-- Command reference

Plot (options …)

…

Quantity = {Voltage | Current | Power | Losses | Capacity | (Value Index for General, AutoAdd, or Circuit[w/ file]) } Specify quantity or value index to be plotted.

Max = {0 | value corresponding to max scale or line thickness}

Dots = {Y | N} Turns on/off the dot symbol for bus locations on the circuit plot.

Labels = {Y | N} Turns on/off the bus labels on the circuit plot (with all bus labels displayed, the

plot can get cluttered – zoom in to see individual names)

Object = [metername for Zone plot | Monitor name | File Name for General bus data or Circuit branch data | Loadshape name]. Specifies what object to plot: meter zones, monitor or CSV file, or previously defined Loadshape. Note: for Loadshape plot, both the Mult and Qmult properties, if defined, are plotted.

…

Dss command reference
-- Command reference

Plot (options …)

…

ShowLoops = {Y | N} (default=N). Shows the loops in meter zone in red. Note that the DSS has no problem solving loops; This is to help detect unintentional loops in radial circuits.

R3 = pu value for tri-color plot max range [0.85] (Color C3)

R2 = pu value for tri-color plot mid range [0.50] (Color C2)

C1, C2, C3 = {RGB color number}. Three color variables used for various plots.

Channels = (array of channel numbers for monitor plot). More than one monitor channel can be plotted on the same graph. Ex: Channels=[1, 3, 5]

Bases = (array of base values for each channel for monitor plot). Default is 1.0 for each. This will per-unitize the plot to the specified bases. Set Bases=[…] after defining channels.

Subs = {Y | N} (default=N) (show substations) See Transformer definition

…

Dss command reference
-- Command reference

Plot (options …)

…

Thickness = max thickness allowed for lines in circuit plots (default=7). Useful for controlling aesthetics on circuit plots.

Buslist = {Array of Bus Names | File=filename } This is for the Daisy plot.

Plot daisy power max=5000 dots=N Buslist=[file=MyBusList.txt]

A "daisy" marker is plotted for each bus in the list. Bus names may be repeated, which results in multiple markers distributed in a circle around the bus location. This gives the appearance of a daisy if there are several symbols at a bus. Not needed for plotting active generators.

Phases = {default* | ALL | PRIMARY | LL3ph | LLALL | LLPRIMARY | (phase number)} For Profile plot. Specify which phases you want plotted.

default = plot only nodes 1-3 at 3-phase buses (default)

ALL = plot all nodes

…

Dss command reference
-- Command reference

Plot (options …)

…

```
PRIMARY = plot all nodes -- primary only (voltage > 1kV)
LL3ph = 3-ph buses only -- L-L voltages)
LLALL = plot all nodes -- L-L voltages)
LLPRIMARY = plot all nodes -- L-L voltages primary only)
(phase number) = plot all nodes on selected phase
Note: Only nodes downline from an energy meter are plotted.
```

Loadshapes may be plotted from the control panel of the user interface.

Power and Losses are specified in kW. C1 used for default color. C2, C3 used for gradients, tricolor plots. Scale determined automatically of Max = 0 or not specified. Some examples:

…

Dss command reference

-- Command reference

Plot (options …)

…

```
Plot circuit quantity=7 Max=.010 dots=Y Object=branchdata.csv
Plot General Quantity=2 Object=valuefile.csv
Plot type=circuit quantity=power
Plot Circuit Losses 1phlinestyle=3
Plot Circuit quantity=3 object=mybranchdata.csv
Plot daisy power 5000 dots=N
Plot daisy power max=5000 dots=N Buslist=[file=MyBusList.txt]
Plot General quantity=1 object=mybusdata.csv
Plot Loadshape object=myloadshape
Plot Tshape object=mytemperatureshape
Plot Priceshape object=mypriceshape
Plot Profile
Plot Profile Phases=Primary
```

Dss command reference
-- Command reference

Powers

Returns the powers (complex) going into each conductors of ALL terminals of the active circuit element in the Result string. (See Select command.) Returned as comma-separated kW and kvar.

Sample

Force all monitors and Energymeters to take a sample for the most recent solution. Keep in mind that Energymeters will perform integration each time they take a sample.

SeqCurrents

Returns the sequence currents into all terminals of the active circuit element (see Select command) in Result string. Returned as comma-separated magnitude only values. Order of returned values: 0, 1, 2 (for each terminal).

Dss command reference

-- Command reference

SeqPowers

Returns the sequence powers into all terminals of the active circuit element (see Select command) in Result string. Returned as comma-separated kw, kvar pairs.Order of returned values: 0, 1, 2 (for each terminal).

SeqVoltages

Returns the sequence voltages at all terminals of the active circuit element (see Select command) in Result string. Returned as comma-separated magnitude only values.Order of returned values: 0, 1, 2 (for each terminal).

Set [option1=value1] [option2=value2] (Options)

The Set command sets various global variables and options having to do with solution modes, user interface issues, and the like. Works like the Edit command except that you don't specify object type and name.

…

Dss command reference
-- Command reference

…

See the Options Reference for descriptions of the various options you can set with the Set command.

There are new Set command values added periodically. Check the Help on the DSS while it is running.

SetkVBase [bus=…] [kvll=..]

Command to explicitly set the base voltage for a bus. Bus must be previously defined. This will override the definitions determined by CalcVoltageBases. When there are only a few voltage bases in the problem, and they are very distinct, the CalcvoltageBases command will work nearly every time. Problems arise if there are two voltage bases that are close together, such as 12.47 kV and 13.2 kV. Use a script composed of SetkVBase commands to remove ambiguity.

…

Dss command reference

-- Command reference

SetkVBase [bus=…] [kvll=..]

…

Parameters in order are:

Bus = {bus name}

kVLL = (line-to-line base kV)

kVLN = (line-to-neutral base kV)

kV base is normally given in line-to-line kV (phase-phase). However, it may also be specified by line-to-neutral kV. The following exampes are equivalent:

```
setkvbase Bus=B9654 kVLL=13.2

setkvbase B9654 13.2

setkvbase B9654 kvln=7.62
```

Dss command reference

-- Command reference

Show <Quantity>

See the separate help on the Show command in the OpenDSS executable. Show commands are added frequently, often making this document out of date.

The Show commands generally writes a text file report of the specified quantity for the most recent solution and opens a viewer (the default Editor -- e.g., Notepad or some other editor) to display the file. Defaults to Show Voltages – so if you mistype the name of the quantity you want, you will get the sequence voltages.

> **Quantity** can be one of:
>
> **Currents** - Shows the currents into each device terminal.
>
> **Monitor <monitor name>** - Shows a text (CSV) file with the voltages and currents presently stored in the specified monitor.
>
> **Faults** - Shows results of Faultstudy mode solution: all-phase, one-phase, and adjacent 2-phase fault currents at each bus.

…

Dss command reference
-- Command reference

Show <Quantity>

...

Elements - Shows all the elements in the active circuit.

Buses Shows all buses in the active circuit.

Panel - Same as Panel Command. Opens the internal DSS control panel.

Meter - shows the present values in the energy meter registers in the active circuit.

Generators - Each generator has its own energy meter. Shows the present values in each generator energy meter register in the active circuit.

Losses Loss summary

Powers [MVA|kVA*] [Seq* | Elements] Show the power flow in various units. Default

(*) is sequence power in kVA.

....

Dss command reference
-- Command reference

Show <Quantity>

…

Voltages [LL |LN*] [Seq* | Nodes | Elements]. Shows the voltages in different ways. Default (*) is sequence quantities line-to-neutral.

Zone EnergyMeterName [Treeview] Different ways to show the selected energymeter zone.

AutoAdded (see AutoAdd solution mode)

Taps shows the taps on regulated transformers

Overloads Overloaded PD elements report

Unserved [UEonly] Unserved energy report. Loads that are unserved.

EVentlog Show the event log (capacitor switching, regulator tap changes)

….

Dss command reference

-- Command reference

Solve [see set command options …]

Executes the solution mode specified by the Set Mode = command. It may execute a single solution or hundreds of solutions. The Solution is a DSS object associated with the active circuit. It has several properties that you may set to define which solution mode will be performed next. This command invokes the Solve method of the Solution object, which proceeds to execute the designated mode. You may also specify the Mode and Number options directly on the Solve command line if you wish:

```
Solve Mode=M1 Number=1000,
```

for example. Note that there is also a Solve method in the Solution interface in the DSS COM serverimplementation. This is generally faster when driving the DSS from user-written code for a custom solution algorithm that must execute many solutions.

…

Dss command reference
-- Command reference

Solve [see set command options …]

…

You may use the same options on the Solve command line as you can with the Set command. In fact, the Solve command is simply the Set command that performs a solution after it is done setting the options. For example:

```
Solve mode=daily stepsize=15m number=96
```

Summary

Returns a power flow summary of the most recent solution in the global result string.

Totals

Total of all EnergyMeter objects in the circuit. Reports register totals in the result string.

Dss command reference

-- Command reference

Variable

Syntax:

```
Variable [name=] MyVariableName [Index=] IndexofMyVariable
```

Returns the value of the specified state variable of the active circuit element, if a PCelement. Applies only to PCelements (Load, Generator, etc) that contain state variables. Returns the value as a string in the Result window or the Text.Result interface if using the COM server.

You may specify the variable by name or by its index. You can determine the index using the **`VarNames`** command. If any part of the request is invalid, the Result is null. These may change from time to time, so it is always good to check.

Dss command reference
-- Command reference

Varnames

Returns all variable names for active element if PC element. Otherwise, returns null.

VarValues

Returns all variable values for active element if PC element. Otherwise, returns null.

Visualize

[What=] {Currents* | Voltages | Powers} [element=]full_element_name (class.name).

Shows the currents, voltages, or powers for selected element on a drawing in phasor quantities.

For example:

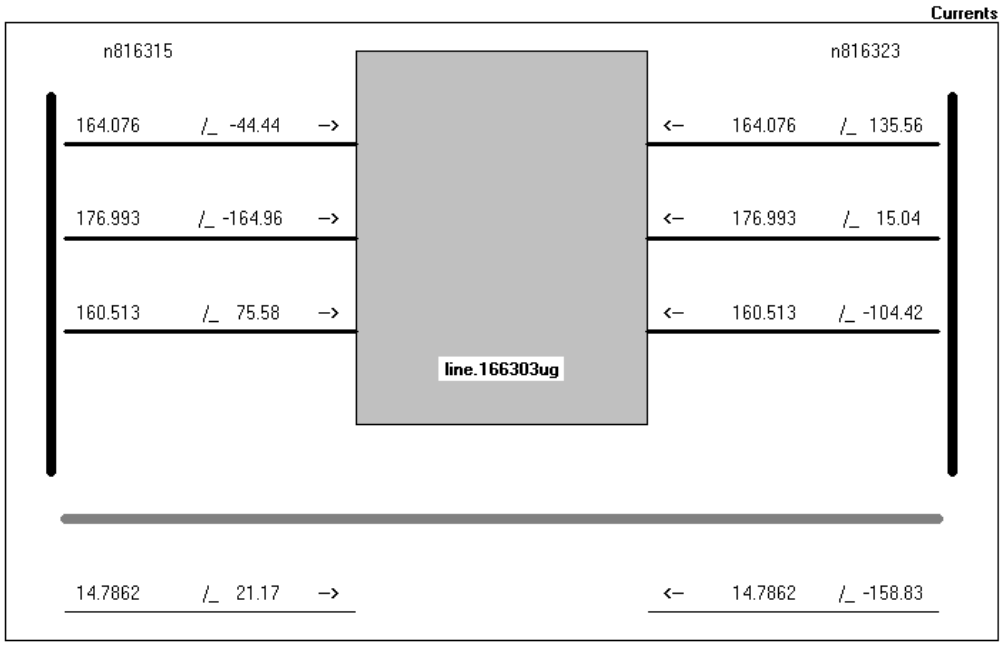
…

Dss command reference

-- Command reference

Visualize

. . .



Dss command reference
-- Command reference

Voltages

Returns the voltages for the ACTIVE TERMINAL ONLY of the active circuit element in the Result string. For setting the active circuit element, see the Select command or the Set Terminal = property. Returned as magnitude and angle quantities, comma separated, one set per conductor of the terminal.

Ysc

Returns full short circuit admittance, Ysc, matrix for the ACTIVE BUS in comma-separated complex number form G + jB.

Zsc

Returns full Short circuit impedance, Zsc, matrix for the ACTIVE BUS in comma-separated complex number form.

Dss command reference

-- Command reference

Zsc10

Returns symmetrical component short-circuit impedances, Z1, Z0 for the ACTIVE BUS in comma-separated R+jX form.

Dss command reference
-- Option reference

DSS options are set using either the Set command or the Solve command. The Solve command first executes the Set and then executes a solution. This allows for more concise syntax for some cases. For example, the sequence

```
Set mode=snapshot
Solve
Set mode=harmonics
Solve
```

May be accomplished with just 2 lines:

```
Solve mode=snapshot
Solve mode=harmonics
```

Selected Options currently implemented are described below. Options are added frequently. Check the on-line Help.

Dss command reference
-- Option reference

%growth =
Set default annual growth rate, percent, for loads with no growth curve specified. Default is 2.5.

%mean =
Percent mean to use for global load multiplier. Default is 65%.

%Normal =
Sets the Normal rating of all lines to a specified percent of the emergency rating. Note: This action takes place immediately. Only the in-memory value is changed for the duration of the run.

%stddev =
Percent Standard deviation to use for global load multiplier. Default is 9%.

Addtype =
{Generator | Capacitor} Default is Generator. Type of device for AutoAdd Mode.

Dss command reference
-- Option reference

Algorithm **=**

{Normal | Newton} Solution algorithm type. Normal is a fixed point current-injection iteration that is a little quicker (about twice as fast) than the Newton iteration. Normal is adequate for most distribution systems. Newton is more robust for circuits that are difficult to solve.

AllocationFactors **=**

Sets all allocation factors for all loads in the active circuit to the value given. Useful for making an initial guess or forcing a particular allocation of load. The allocation factors may be set automatically by the energy meter elements by placing energy meters on the circuit, defining the PEAKCURRENT property, and issuing the ALLOCATELOADS command. The factors are applied to the XFkVA property of Load objects.

Dss command reference
-- Option reference

AllowDuplicates =

{YES/TRUE | NO/FALSE} Default is No. Flag to indicate if it is OK to have devices of same name in the same class. If No, then a New command is treated as an Edit command but adds an element if it doesn't exist already. If Yes, then a New command will always result in a device being added.

AutoBusList =

Array of bus names to include in AutoAdd searches. Or, you can specify a text file holding the names, one to a line, by using the syntax (file=filename) instead of the actual array elements. Default is null, which results in the program using either the buses in the EnergyMeter object zones or, if no EnergyMeters, all the buses, which can make for lengthy solution times.

```
Examples:
Set autobuslist=(bus1, bus2, bus3, ... )
Set autobuslist=(file=buslist.txt)
```

Dss command reference
-- Option reference

Basefrequency =
Default = 60. Set the fundamental frequency for harmonic solution and the default base frequency for all impedance quantities. Side effect: also changes the value of the solution frequency. See also DefaultBaseFrequency.

Bus =
Set Active Bus by name. Can also be done with Select and SetkVBase commands and the "Set Terminal=" option. The bus connected to the active terminal becomes the active bus. See Zsc and Zsc012 commands.

capkVAR =
Size of capacitor, kVAR, to automatically add to system. Default is 600.0.

casename =
Name of case for yearly simulations with demand interval data. Becomes the name of the subdirectory under which all the year data are stored. Default = circuit name. Side Effect: Sets the prefix for output files.

Dss command reference
-- Option reference

CapMarkerCode **=**

Numeric marker code for capacitors. Default is 37. See MarkerCode option.

CapMarkerSize **=**

Size of Capacitor device marker. Default is 3.

Cfactors**=**

Similar to Set Allocationfactors= except this applies to the kWh billing property of Load objects. Sets all Load Cfactor properties to the same value. A typical value is 4. See online help on the Load object.

circuit **=**

Set the active circuit by name. (Current version allows only one circuit at a time, so this option is basically ignored at present.)

Dss command reference
-- Option reference

CktModel =
{Multiphase | Positive} Default = Multiphase. Designates whether circuit model is to interpreted as a normal multi-phase model or a positive-sequence only model. If Positive sequence, all power quantities are mulitiplied by 3 in reports and through any interface that reports a power quantity. Any line with sequence parameter inputs will use the long-line equivalent pi section.

Class =
Synonym for Type=. sets class (type) for the Active DSS Object. This becomes the Active DSS Class.

ControlMode =
{OFF | STATIC |EVENT | TIME} Default is "STATIC". Control mode for the solution. Set to OFF to prevent controls from changing.
…

Dss command reference
-- Option reference

ControlMode **=**

…

STATIC = Time does not advance. Control actions are executed in order of shortest time to act until all actions are cleared from the control queue. Use this mode for power flow solutions which may require several regulator tap changes per solution. This is the default for the standard Snapshot mode as well as Daily and Yearly simulations where the stepsize is typically greater than 15 min.

EVENT = solution is event driven. Only the control actions nearest in time are executed and the time is advanced automatically to the time of the event.

TIME = solution is time driven. Control actions are executed when the time for the pending action is reached or surpassed. Use this for duty-cycle mode and dynamic mode.

…

Dss command reference
-- Option reference

ControlMode **=**

…

Controls may reset and may choose not to act when it comes their time to respond.

Use **TIME** mode when modeling a control externally to the DSS and a solution mode such as DAILY or DUTYCYCLE that advances time, or set the time (hour and sec) explicitly from the external program.

Datapath **=**

Set the data path for files written or read by the DSS. Defaults to the startup path. May be Null. Executes a CHDIR to this path if non-null. Does not require a circuit defined. You can also use the "cd" command from a script.

Dss command reference
-- Option reference

DefaultBaseFrequency=

Set Default Base Frequency, Hz. The default value when first installed is 60 Hz. Side effect: Sets the solution Frequency and default Circuit Base Frequency. This value is saved in the Windows Registry when the DSS closes down. Therefore, it need only be set one time. This is useful for users studying 50 Hz systems.

DefaultDaily =

Default daily load shape name. Default value is "default", which is a 24-hour curve defined when the DSS is started.

DefaultYearly =

Default yearly load shape name. Default value is "default", which is a 24-hour curve defined when the DSS is started. If no other curve is defined, this curve is simply repeated when in Yearly simulation mode.

Dss command reference

-- Option reference

DemandInterval =

{YES/TRUE | NO/FALSE} Default = no. Set for keeping demand interval data for daily, yearly, etc, simulations. Side Effect: Resets all meters!!!

DIVerbose =

{YES/TRUE | NO/FALSE} Default = FALSE. Set to Yes/True if you wish a separate demand interval (DI) file written for each meter. Otherwise, only the totalizing meters are written.

EarthModel =

One of {Carson | FullCarson | Deri*}. Default is Deri, which is a fit to the Full Carson that works well into high frequencies. "Carson" is the simplified Carson method that is typically used for 50/60 Hz power flow programs. Applies only to Line objects that use LineGeometry objects to compute impedances.

Dss command reference
-- Option reference

Editor=
Set the command string required to start up the editor preferred by the user. Defaults to Notepad. This is used to display certain reports from the DSS. Use the complete path name for any other Editor. Does not require a circuit defined. This value is saved in the Windows Registry and need only be specified one time. EPRI uses the EditPlus editor.

Element =
Sets the active DSS element by name. You can use the complete object spec (class.name) or just the name. if full name is specifed, class becomes the active class, also. See also the Select command.

Emergvmaxpu =
Maximum permissible per unit voltage for emergency (contingency) conditions. Default is 1.08.

Dss command reference
-- Option reference

Emergvminpu =
Minimum permissible per unit voltage for emergency (contingency) conditions. Default is 0.90.

Frequency =
sets the frequency for the next solution of the active circuit.

Genkw =
Size of generator, kW, to automatically add to system. Default is 1000.0

GenMult =
Global multiplier for the kW output of every generator in the circuit. Default is 1.0. Applies to Snapshot, Daily, and DutyCycle solution modes. Ignored if generator is designated as Status=Fixed.

Dss command reference
-- Option reference

Genpf =
Power factor of generator to assume for automatic addition. Default is 1.0.

h =
Alternate name for time step size (see Stepsize).

Harmonics =
{ALL | (list of harmonics) } Default = ALL. Array of harmonics for which to perform a solution in Harmonics mode. If ALL, then solution is performed for all harmonics defined in spectra currently being used. Otherwise, specify a more limited list such as: Set Harmonics=(1 5 7 11 13)

Hour=
sets the hour to be used for the start time of the solution of the active circuit. (See also Time)

Dss command reference
-- Option reference

KeepList =
Array of bus names to keep when performing circuit reductions. You can specify a text file holding the names, one to a line, by using the syntax (file=filename) instead of the actual array elements. Command is cumulative (reset keeplist first). Reduction algorithm may keep other buses automatically.

```
Examples:
Reset Keeplist (sets all buses to FALSE (no keep))
Set KeepList=(bus1, bus2, bus3, ... )
Set KeepList=(file=buslist.txt)
```

LDcurve =
name of the Loadshape object to use for the global circuit Load-Duration curve. Used in solution modes LD1 and LD2 (see below). Must be set before executing those modes. Simply define the load-duration curve as a loadshape object. Default = nil.

Dss command reference

-- Option reference

LoadModel=

{"POWERFLOW" | "ADMITTANCE"} Sets the load model. If POWERFLOW (abbreviated P), loads do not appear in the System Y matrix. For iterative solution types (Mode □ Direct) loads (actually all PC Elements) are current injection sources. If ADMITTANCE, all PC elements appear in the System Y matrix and solution mode should be set to Direct (below) because there will be no injection currents.

LoadMult =

global load multiplier to be applied to all "variable" loads in the circuit for the next solution. Loads designated as "fixed" are not affected. Note that not all solution modes use this multiplier, but many do, including all snapshot modes. See Mode below. The default LoadMult value is 1.0. Remember that it remains at the last value to which it was set. Solution modes such as Monte Carlo and Load-Duration modes will alter this multiplier. Its value is usually posted on DSS control panels. Loads defined with "status=fixed" are not affected by load multipliers. (The default for loads is "status=variable".)

Dss command reference
-- Option reference

Log =
{YES/TRUE | NO/FALSE} Default = FALSE. Significant solution events are added to the Event Log, primarily for debugging.

LossRegs =
Which EnergyMeter register(s) to use for Losses in AutoAdd Mode. May be one or more registers. if more than one, register values are summed together. Array of integer values > 0. Defaults to 13 (for Zone kWh Losses).
For a list of EnergyMeter register numbers, do the "Show Meters" command after defining a circuit.

LossWeight =
Weighting factor for Losses in AutoAdd functions. Defaults to 1.0. Autoadd mode

```
Minimizes (Lossweight * Losses + UEweight * UE).
```

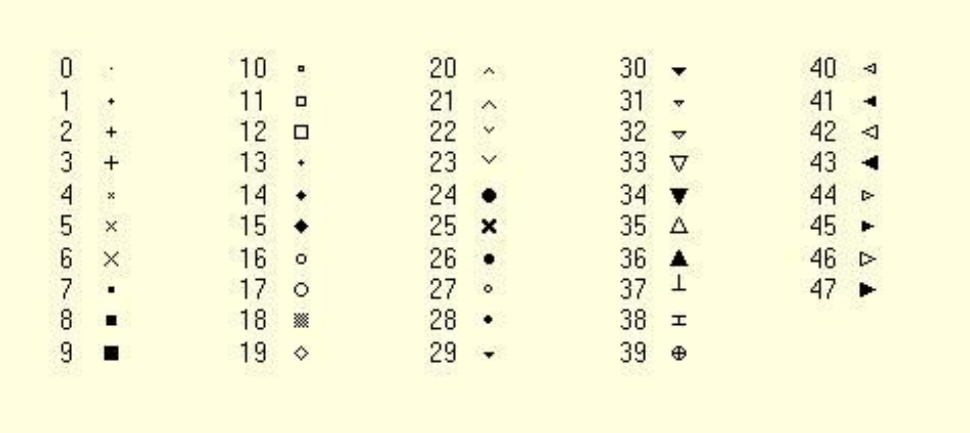
If you wish to ignore Losses, set to 0. This applies only when there are EnergyMeter objects. Otherwise, AutoAdd mode minimizes total system losses.

Dss command reference

-- Option reference

Markercode =

Number code for node marker on circuit plots (these are currently the SDL Components MarkAt options with this version). Marker codes are:



Dss command reference
-- Option reference

Markswitches =
{YES/TRUE | NO/FALSE} Default is NO. Mark lines that are switches or are isolated with a symbol. See SwitchMarkerCode.

MarkCapacitors =
{YES/TRUE | NO/FALSE} Default is NO. Mark Capacitor object locations with a symbol. See CapMarkerCode. The first bus coordinate must exist.

MarkPVSystems =
{YES/TRUE | NO/FALSE} Default is NO. Mark PVSystem locations with a symbol. See PVMarkerCode. The bus coordinate must exist.

MarkRegulators =
{YES/TRUE | NO/FALSE} Default is NO. Mark RegControl object locations with a symbol. See RegMarkerCode. The bus coordinate for the controlled transformer winding must exist.

Dss command reference
-- Option reference

MarkStorage =
{YES/TRUE | NO/FALSE} Default is NO. Mark Storage device locations with a symbol. See StoreMarkerCode. The bus coordinate must exist.

Marktransformers =
{YES/TRUE | NO/FALSE} Default is NO. Mark transformer locations with a symbol. See TransMarkerCode. The coordinate of one of the buses for winding 1 or 2 must be defined for the symbol to show.

Maxcontroliter =
Max control iterations per solution. Default is 10.

Maxiter =
Sets the maximum allowable iterations for power flow solutions. Default is 15.

Dss command reference

-- Option reference

Mode=

Specify the solution mode for the active circuit. Mode can be one of (unique abbreviation will suffice, as with nearly all DSS commands):

Mode=Snap:

Solve a single snapshot power flow for the present conditions. Loads are modified only by the global load multiplier (LoadMult) and the growth factor for the present year (Year).

Mode=Daily:

Do a series of solutions following the daily load curves. The Stepsize defaults to 3600 sec (1 hr). Set the starting hour and the number of solutions (e.g., 24) you wish to execute. Monitors are reset at the beginning of the solution. The peak of the daily load curve is determined by the global load multiplier (LoadMult) and the growth factor for the present year (Year).

Dss command reference
-- Option reference

Mode=Direct:
Solve a single snapshot solution using an admittance model of all loads. This is noniterative; just a direct solution using the currently specified voltage and current sources.

Mode=Dutycycle:
Follow the duty cycle curves with the time increment specified. Perform the solution for the number of times specified by the Number parameter (see below).

Mode=Dynamics:
Sets the solution mode for a dynamics solution. Must be preceded by a successful power flow solution so that the machines can be initialized. Changes to a default time step of 0.001s and ControlMode = TIME. Generator models are changed to a voltage source behind the value specified for transient reactance for each generator and initialized to give approximately the same power flow as the existing solution. Be sure to set the number of time steps to solve each time the Solve command is given.

Dss command reference

-- Option reference

Mode=FaultStudy:

Do a full fault study solution, determining the Thevenin equivalents for each bus in the active circuit. Prepares all the data required to produce fault study report under the Show Fault command.

Mode=Harmonics:

Sets the solution mode for a Harmonics solution. Must be preceded by a successful power flow solution so that the machines and harmonics sources can be initialized. Loads are converted to harmonic current sources and initialized based on the power flow solution according to the Spectrum object associated with each Load. Generators are converted to a voltage source behind subtransient reactance with the voltage spectrum specified for each generator. A Direct solution is performed for each harmonic frequency (more precisely, non-power frequency). The system Y matrix is built for each frequency and solved with the defined injections from all harmonic sources. A solution is performed for each frequency found to be defined in all the spectra being used in the circuit. Note that to perform a frequency scan of a network, you would define a Spectrum object with a small frequency increment and assign it to either an Isource or Vsource object, as appropriate.

Dss command reference

-- Option reference

Mode=Yearly:

Do a solution following the yearly load curves. The solution is repeated as many times as the specified by the Number= option. Each load then follows its yearly load curve. Load is determined solely by the yearly load curve and the growth multiplier. The time step in past revisions was always 1 hour. However, it may now be any value.. Meters and Monitors are reset at the beginning of solution and sampled after each solution. If the yearly load curve is not specified, the daily curve is used and simply repeated if the number of solutions exceeds 24 hrs. This mode is nominally designed to support 8760- hr simulations of load, but can be used for any simulation that uses an hourly time step and needs monitors or meters.

Mode=LD1

(Load-Duration Mode 1): Solves for the joint union of a load-duration curve (defined as a Loadshape object) and the Daily load shape. Nominally performs a Daily solution (24- hr) for each point on the Load-duration (L-D) curve. Thus, the time axis of the L-D curve represents days at that peak load value. L-D curves begin at zero (0) time. Thus, a yearly L-D curve would be defined for 0..365 days. A monthly L-D

…

Dss command reference
-- Option reference

Mode=LD1

…

defined for 0..31 days. Energy meters and monitors are reset at the beginning of the solution. At the conclusion, the energy meter values represent the total of all solutions. If the L-D curve represent one year, then the energy will be for the entire year. This mode is intended for those applications requiring a single energy number for an entire year, month, or other time period. Loads are modified by growth curves as well, so set the year before proceeding. Also, set the L-D curve (see Ldcurve option).

Mode=LD2

(Load-Duration Mode 2): Similar to LD1 mode except that it performs the Load-duration solution for only a selected hour on the daily load shape. Set the desired hour before executing the Solve command. The meters and monitors are reset at the beginning of the solution. At the conclusion, the energy meters have only the values for that hour for the year, or month, or whatever time period the L-D curve represents. The solver simply solves for each point on the L-D curve, multiplying the

…

Dss command reference
-- Option reference

Mode=LD2

…

load at the selected hour by theL-D curve value. This mode has been used to generate a 3-D plot of energy vs. month and hour of the day.

Mode=M1

(Monte Carlo Mode 1): Perform a number of solutions allowing the loads to vary randomly. Executes number of cases specified by the Number option (see below). At each solution, each load is modified by a random multiplier -- a different one for each load. In multiphase loads, all phases are modified simultaneously so that the load remains balanced. The random variation may be uniform or gaussian as specified by the global Random option (see below). If uniform, the load multipliers are between 0 and 1. If gaussian, the multipliers are based on the mean and standard deviation of the Yearly load shape specified for the load. Be sure one is specified for each load.

Mode=M2

(Monte Carlo Mode 2): This mode is designed to execute a number of Daily simulations with the global peak load multiplier (LoadMult) varying randomly. Set

…

Dss command reference

-- Option reference

Mode=M2

…

time step size (h) and Number of solutions to run. "h" defaults to 3600 sec (1 Hr). Number of solutions refers to the number of DAYS. For Random = Gaussian, set the global %Mean and %Stddev variables, e.g. "Set %Mean=65 %Stddev=9". For Random=Uniform, it is not necessary to specify %Mean, since the global load multiplier is varied from 0 to 1. For each day, the global peak load multiplier is generated and then a 24 hour Daily solution is performed at the specified time step size.

Mode=M3

(Monte Carlo Mode 3): This mode is similar to the LD2 mode except that the global load multiplier is varied randomly rather than following a load-duration curve. Set the Hour of the day first (either Set Time=… or Set Hour=…). Meters and monitors are reset at the beginning of the solution. Energy at the conclusion of the solution represents the total of all random solutions. For example, one might use this mode to estimate the total annual energy at a given hour by running only 50 or 100 solutions at each hour (rather than 365) and ratioing up for the full year.

Dss command reference
-- Option reference

Mode=MF
(Monte Carlo Fault mode). One of the faults defined in the active circuit is selected and its resistance value randomized. All other Faults are disabled. Executes number of cases specified by the Number parameter.
Mode=Peakdays:
Do daily solutions (24-hr) only for those days in which the peak exceeds a specified value.

Nodewidth =
Width of node marker in circuit plots. Default=1.

Normvmaxpu =
Maximum permissible per unit voltage for normal conditions. Default is 1.05.

Normvminpu =
Minimum permissible per unit voltage for normal conditions. Default is 0.95.

Dss command reference
-- Option reference

NumAllocIterations **=**
Default is 2. Maximum number of iterations for load allocations for each time the AllocateLoads or Estimate command is given. Usually, 2 are sufficient, but some cases are more difficult. Execute an Export Estimation report to evaluate how well the load allocation has worked.

Number=
specify the number of time steps or solutions to run or the number of Monte Carlo cases to run.

Object (or Name)=
sets the name of the Active DSS Object. Use the complete object specification (classname.objname), or simply the objname, to designate the active object which will be the target of the next command (such as the More command). If 'classname' is omitted, you can set the class by using the Class= field.

Dss command reference
-- Option reference

Overloadreport =
{YES/TRUE | NO/FALSE} Default = FALSE. For yearly solution mode, sets overload reporting on/off. DemandInterval must be set to true for this to have effect.

NeglectLoadY=
{YES/TRUE | NO/FALSE} Default is NO. For Harmonic solution, neglect the Load shunt admittance branch that can siphon off some of the Load injection current. If YES, the current injected from the LOAD at harmonic frequencies will be nearly ideal.

PriceCurve =
Sets the curve to use to obtain for price signal. Default is none (null string). If none, price signal either remains constant or is set by an external process. Curve is defined as a loadshape (not normalized) and should correspond to the type of analysis being performed (daily, yearly, load-duration, etc.).

Dss command reference
-- Option reference

PriceSignal =
Sets the price signal ($/MWh) for the circuit. Initial value is 25.

PVMarkerCode =
Numeric marker code for PVSystem devices. Default is 15. See MarkerCode option.

PVMarkerSize =
Size of PVSystem device marker. Default is 1.

Random=
specify the mode of random variation for Monte Carlo studies: One of [Uniform | Gaussian | Lognormal | None] for Monte Carlo Variables May abbreviate value to "G", “L”, or "U" where G = gaussian (using mean and std deviation for load shape); L = lognormal (also using mean and std dev); U = uniform (varies between 0 and 1 randomly). Anything else: randomization disabled.

Dss command reference

-- Option reference

Recorder =
{YES/TRUE | NO/FALSE} Default = FALSE. Opens DSSRecorder.DSS in DSS install folder and enables recording of all commands that come through the text command interface. Closed by either setting to NO/FALSE or exiting the program. When closed by this command, the file name can be found in the Result. Does not require a circuit defined.

ReduceOption =
{ Default or [null] | Stubs [Zmag=nnn] | MergeParallel | BreakLoops | Switches | TapEnds [maxangle=nnn] | Ends} Strategy for reducing feeders. Default is to eliminate all dangling end buses and buses without load, caps, or taps.
"Stubs [Zmag=0.02]" merges short branches with impedance less than Zmag (default = 0.02 ohms.
"MergeParallel" merges lines that have been found to be in parallel.
"Breakloops" disables one of the lines at the head of a loop.
…

Dss command reference
-- Option reference

ReduceOption =
…
"Tapends [maxangle=15]" eliminates all buses except those at the feeder ends, at tap points and where the feeder turns by greater than maxangle degrees.
"Ends" eliminates dangling ends only.
"Switches" merges switches with downline lines and eliminates dangling switches.
Marking buses with "Keeplist" will prevent their elimination.

RegMarkerCode =
Numeric marker code for Regulators. Default is 47. See MarkerCode option.

RegMarkerSize =
Size of RegControl device marker. Default is 1.

Sec =
sets the seconds from the hour for the start time for the solution of the active circuit. (See also Time)

Dss command reference
-- Option reference

ShowExport=
{YES/TRUE | NO/FALSE} Default = FALSE. If YES/TRUE will automatically show the results of an Export command after it is written. Normally, the result of an Export command (a CSV file) is not automatically displayed.

Stepsize (or h)=
sets the time step size (default unit is sec) for the solution of the active circuit.
Normally, specified for dynamic solution but is also, used for duty-cycle load following solutions. Yearly simulations typically go hour-by-hour and daily simulations follow the smallest increment of the daily load curves. Yearly simulations can also go in any time increment. Reasonable default values are set when you change solution modes. You may specify the stepsize in seconds, minutes, or hours by appending ‘s’, ‘m’, or ‘h’ to the size value. If omitted, ‘s’ is assumed. Example: “set stepsize=15m” is the same as “set stepsize=900.” Do not leave a space between the value and the character.

Dss command reference
-- Option reference

Switchmarkercode =
Numeric marker code for lines with switches or are isolated from the circuit. Default is 4. See markswitches option and markercode option.

Terminal =
Set the active terminal of the active circuit element. May also be done with Select command.

Time=
Specify the solution start time as an array : time="hour, sec" or time = (hour, sec): e.g., time = (23, 370) designate 6 minutes, 10 sec past the 23rd hour.

tolerance=
Sets the solution tolerance. Default is 0.0001.

Dss command reference
-- Option reference

TraceControl =
{YES/TRUE | NO/FALSE} Set to YES to trace the actions taken in the control queue. Creates a file named TRACE_CONTROLQUEUE.CSV in the default directory. The names of all circuit elements taking an action are logged.

TransMarkerCode =
Numeric marker code for transformers. Default is 35. See MarkTransformers option and MarkerCode option.

TransMarkerSize =
Size of transformer marker. Default is 1.

StoreMarkerCode =
Numeric marker code for Storage devices. Default is 9. See MarkerCode option.

Dss command reference
-- Option reference

StoreMarkerSize =
Size of Storage device marker. Default is 1.

Trapezoidal =
{YES/TRUE | NO/FALSE} Default is "No". Specifies whether to use trapezoidal integration for accumulating energy meter registers. Applies to EnergyMeter and Generator objects. Default method simply multiplies the present value of the registers times the width of the interval. Trapezoidal is more accurate when there are sharp changes in a load shape or unequal intervals. Trapezoidal is automatically used for some load-duration curve simulations where the interval size varies considerably. Keep in mind that for Trapezoidal, you have to solve one more point than the number of intervals. That is, to do a Daily simulation on a 24-hr load shape, you would set Number=25 to force a solution at the first point again to establish the last (24th) Interval.

Dss command reference
-- Option reference

Type=
Sets the active DSS class type. Same as Class=...

Ueregs =
Which EnergyMeter register(s) to use for UE in AutoAdd Mode. May be one or more registers. if more than one, register values are summed together. Array of integer values > 0. Defaults to 11 (for Load EEN). For a list of EnergyMeter register numbers, do the "Show Meters" command after defining a circuit.

Ueweight=
Weighting factor for UE/EEN in AutoAdd functions. Defaults to 1.0.
Autoadd mode minimizes (Lossweight * Losses + UEweight * UE).
If you wish to ignore UE, set to 0. This applies only when there are EnergyMeter objects. Otherwise, AutoAdd mode minimizes total system losses.

Dss command reference
-- Option reference

Voltagebases=
Define legal bus voltage bases for this circuit. Enter an array of the legal voltage bases, in phase-to-phase voltages, for example:
set voltagebases=[.208, .480, 12.47, 24.9, 34.5, 115.0, 230.0]
When the CalcVoltageBases command is issued, a snapshot solution is performed with no load injections and the bus base voltage is set to the nearest legal voltage base. The defaults are as shown in the example above.
The DSS does not use per unit values in its solution. You only need to set the voltage bases if you wish to see per unit values on the reports or if you intend to use the AutoAdd feature.

Voltexceptionreport=
{YES/TRUE | NO/FALSE} Default = FALSE. For yearly solution mode, sets voltage exception reporting on/off. DemandInterval must be set to true for this to have effect.

Dss command reference
-- Option reference

Year=
sets the Year to be used for the next solution of the active circuit; the base case is year 0. Used to determine the growth multiplier for each load. Each load may have a unique growth curve (defined as a Growthshape object).

ZoneLock =
{YES/TRUE | NO/FALSE} Default is No. if No, then meter zones are recomputed each time there is a change in the circuit. If Yes, then meter zones are not recomputed unless they have not yet been computed. Meter zones are normally recomputed on Solve command following a circuit change.

Thank you!