ELECTRIC POWER RESEARCH INSTITUTE

OpenDSS Cheatsheet

Examples of Commonly-Used OpenDSS Commands

Command	Description
New Circuit.MyCircuit	Makes a new circuit with a 115 kV Vsource connected to SourceBus
New Circuit.MyCircuit BasekV=69 pu=1.05	Makes a new circuit and changes the voltage to 69 kV, 1.05 pu
New Circuit.MyCircuit ~ BasekV=345 pu=1.05 ~ puZ1=.05 pu Z0=.08	Makes a new circuit with a 345 kV source, 1.05 pu,2000 MVA 3-phase fault (default Base MVA for puZ1 property is 100 MVA)
Solve	Solves the present circuit with the present solution mode
Solve Mode=Yearly	Sets the mode to Yearly and executes the default 8760-hour yearly solution model algorithm Note: Changing the Mode resets all Monitor and EnergyMeter Object
Solve Mode=Dutycycle Stepsize=2s number=30000	Set the mode to Dutycycle and solve 30000 time steps at a 2 s interval.
Solve Mode=Daily Stepsize=15m number=96	Set the mode to Daily and solve 96 steps at a 15 min interval (one full day)
Set Mode=dynamic Stepsize=.001 number=2000 Solve	In two commands, set the mode to Dynamic and then solve 2000 steps at a 1 ms interval (the default unit on Stepsize option is sec)
Solve Mode=FaultStudy Show Faultstudy	Solve the present circuit in FaultStudy mode and display the fault current. Includes Loads as linear elements if they are defined and enabled.
New Fault.F1 Bus1=MyBus.2 Solve Mode=Dynamic Number=1 Stepsize=0.00001	Add a SLG fault to phase 2 of MyBus (phases=1 by default). Solve 1 timestep in Dynamic mode at a tiny timestep. This captures Generator contribution to fault.

Solution Modes:	1. Snapshot, (DEFAULT mode) 2. Daily, 3. Yearly), 4. DIrect, 5. DUtycycle, 6. Time, (see LoadShapeClass option) 7. DYnamic, 8. Harmonic, 9. M1 (Monte Carlo 1), 10. M2 (Monte Carlo 2), 11. M3 (Monte Carlo 3), 12. Faultstudy, 13. MF (monte carlo fault study) 14. Peakday, 15. LD1 (load-duration 1) 16. LD2 (load-duration 2) 17. AutoAdd (see AddType)
Plot Circuit Power Max=2000 C1=blue 1ph=3 **Plot Circuit Power Max=2000 C1=$00FF0000 1ph=3**	Plots the active circuit (if bus coordinates defined) with line thicknesses proportional to power. 2000 kW = width of 7. Colors may be specified as standard color name or RGB integer number (Hexadecimal format shown: 00BBGGRR)
Standard Color Names	Black Maroon Green Olive Navy Purple Teal Gray Silver Red Lime Yellow Blue Fuchsia Aqua LtGray DkGray White

plot profile phases=primary plot profile phases=all plot profile phases=1	Various Plot Profile command examples. Requires an Energymeter object at the head of the feeder.
New Line.MyLine Bus1=FromBus.1.2.3 Bus2=ToBus.1.2.3 ~ LineCode= 336ACSR Length=2.5 Units=Mi	Typical Line object definition. A good idea to define units on both the LineCode and Line objects. Units don't have to match. OpenDSS will take care of conversions
New Load.MyLoad Bus1=LoadBus.1.2.3 kW=250 PF=.95 ~ kV=12.47	Typical Load definition (3-phase is default). Don't forget kV property. For phases > 1, specify L-L kV.
New Load.My1phLoad Phases=1 Bus1=SecBus.1 kW=25 ~ kV=0.24 PF=0.99	25 kW 240-V single-phase load
New Load.MyLLLoad Phases=1 kV=12.47 ~ Bus1=MyLoadBus.1.2 ~ kW=25 kvar=5	1-phase L-L connected load at 12.47kV level
New Monitor.M1 Element=Line.MyLine Terminal=1 New Monitor.M1 Line.MyLine 1 New Monitor.M1 Line.MyLine	Equivalent syntax for defining a default Monitor object connected to terminal 1 of the Line object named MyLine. The default mode (0) captures Voltages and Currents in Magnitude/Angle at the terminal.
New Monitor.M1 Line.MyLine 1 VIPolar=No	Voltages and Currents are saved as complex numbers instead of the polar form of magnitude/angle
New Monitor.M1 Line.MyLine 1 Mode=1 PPolar=No	Monitor saves power at the terminal in (kW, kvar) instead of polar form of (kVA, angle).
Monitor Modes	0: Voltages and Currents 1: Powers 2: Transformer taps (Transformer elements only) 3: State Variables (PCElements only) 4: Flicker (Pst) of Voltages only (10 min simulation req'd)

Modified Monitor Voltage and Power Modes	Monitor Objects can capture sequence quantities directly by a bitmask adder to the Mode number. Combine with adders below to achieve other results for terminal quantities: +16 = Sequence quantities +32 = Magnitude only +64 = Positive sequence only or avg of all phases e.g., Mode=17 will return sequence powers
Scripting Variables (var command) *Introduced at version 7.6.3.32* **var** *(shows all scripting variables)* **var @lastfile** *(reports value in Result form/interface)* *Setting scripting variable values*: **var @dots=no @labels=yes** …etc… (*see example* →)	Scripting variable names must begin with "**@**". When encountered as a token in the script, the variable name is replaced with the present value of the variable. **Example**: plot Circuit 6 max=0 dots=**@dots** labels=**@labels** subs=n object=**@lastexportfile** C1=$00FF0000 C2=$000000FF Some intrinsic variables: @lastfile *(generally set to the name of the last output file written)* @lastExportfile *(set to the name of the last file written by the Export command)* @lastShowfile *(set to the name of the last file written by the Show command)* @lastPlotfile *(set to the name of the last file written by the Plot command)*

One-Timers

A couple of things you need to do one time. OpenDSS stores these values in the Windows registry so you don't need to do them more than once.

Command	Description
Set Editor=MyFavoriteEditorFullPathName.exe	The default text editor is Notepad, which is expected on all Windows installations. Use this command to change it to another editor. All text file reports will be automatically opened with this editor. Note that for creating scripts, the editor must be capable of producing plain text files in ANSI/ASCII encoding. It is nice to have a more powerful editor for working on large scripts and reports. Column selection capability is a great help.
Set DefaultBaseFrequency=60 Set DefaultBaseFrequency=50	Set Default Base Frequency, Hz. Sets solution Frequency and default Circuit Base Frequency. VSOURCE objects will default to this frequency and impedances specified for LINE, REACTOR, TRANSFORMER, etc objects are assumed to be on this frequency base unless otherwise specified.

COM Interface Tips

Starting the In-Process COM Server

Language	Command
VBA	Requires setting References to OpenDSSEngine.DSS Public DSSObj as OpenDSSEngine.DSS Set DSSObj= New OpenDSSEngine.DSS DSSObj.Start(0) -or- Dim DSSObj as Variant Set DSSObj = CreateObject(“OpenDSSEngine.DSS”) DSSObj.Start(0)
Matlab	DSSObj = actxserver(‘OpenDSSEngine.DSS’); Start=DSSObj.Start(0);
Python	import win32com.client … Class DSS: self.engine = win32com.client.Dispatch("OpenDSSEngine.DSS") self.engine.Start("0")
Delphi	(Import OpenDSSEngine Type Library into IDE) DSSObject: IDSS; DSSObject := coDSS.Create; If DSSObject.Start(0) then …;

Starting the In-Process COM Server, cont'd

Language	Command
C#	(Project>Add Reference … (Select OpenDSSEngine) … using OpenDSSEngine; … public DSS DSSObj; … DSSObj = new DSS(); if (!(DSSObj.Start(0))) …
Python	import win32com.client … Class DSS: self.engine = win32com.client.Dispatch("OpenDSSEngine.DSS") self.engine.Start("0")

Setting Some Commonly-Used Interface Variables

When the Start(0) function is called, the OpenDSS COM server creates all the interfaces. You can assign variables to this after starting the server to make access more efficient and your code more readable. VBA will be used as the example language, but it looks pretty much the same in the other languages after deleting the Set command. The Set command is fairly unique to VB and is how you instantiate *Objects.* Other languages are able to figure this out in the normal language syntax. Also, see examples for setting DSSText in the table below.

Public DSSText as Text
Set DSSText = DSSObj.Text
 	
Var DSSText: IText;
DSSText : OpenDSSEngine.Text; // **Delphi**
DSSText := DSSObj.Text;
 	
Public DSSText as Text; // **C#**
DSSText = DSSObj.Text;
 	
class DSS: # **Python**
self.text = self.engine.Text
	This variable will allow you easy access to the text interface Command and Result properties, for example:
	
	DSSText.Command = "? Line.Myline.Bus1"
	Bus1Name = DSSText.Result
	
	You can execute any of the OpenDSS commands with this variable.

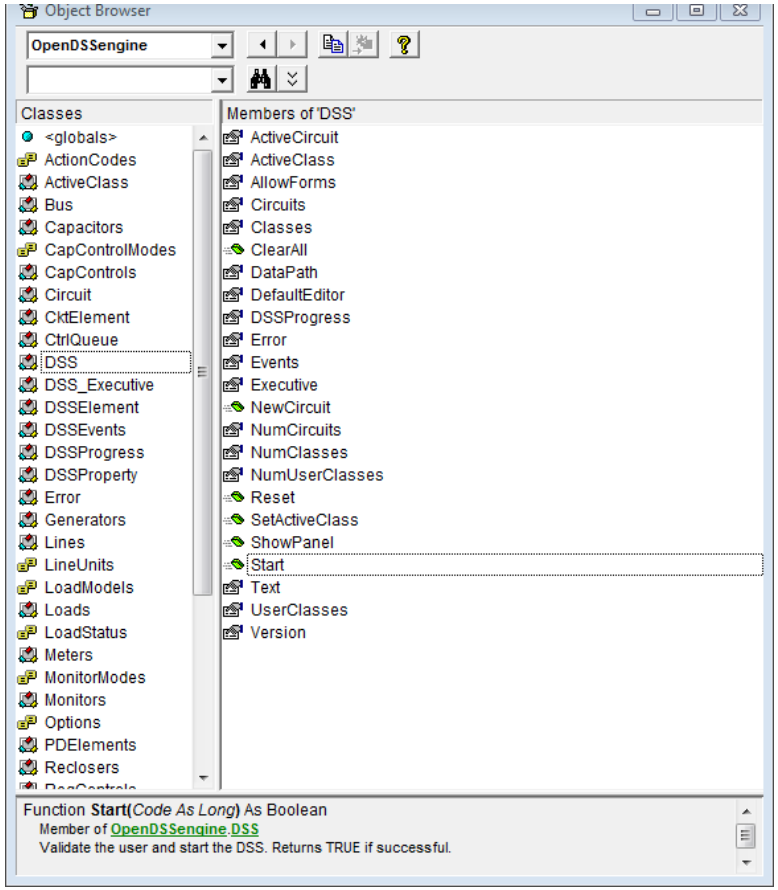
Set DSSCircuit = DSSObj.ActiveCircuit	The ActiveCircuit interface is one of the most important for getting to various quantities via the COM interface.
Set DSSCktElement = DSSCircuit.ActiveCktElement	When any action in the program sets a circuit element active, this variable will give you access to the properties of this element.

Set DSSSolution = DSSCircuit.Solution	A common use of this Solution interface is to solve the circuit: DSSSolution.Solve This invokes the appropriate solve function in the program directly and should generally be the fastest way to execute a solution command.
Set DSSBus = DSSCircuit.ActiveBus	This variable will give you rhe properties of the active bus.
Set DSSControlQueue = DSSCircuit.CtrlQueue	This variable will give you access to the Control Queue in the program. You can see what is on the queue and push messages onto it.
Set DSSLines = DSSCircuit.Lines	This variable allows access to the Lines collection in the active circuit. You can use it to cycle through all the lines in the circuit, for example:. iLine = DSSLines.First While iLine>0 Do …. Do something …. iLine = DSSLines.Next Loop
Set DSSLoads = DSSCircuit.Loads	This variable allows access to the Loads collection in the active circuit.
Set DSSGenerators = DSSCircuit.Generators	This variable allows access to the Generators collection in the active circuit.

While many users choose to drive OpenDSS with Matlab programs, the Object Browser in Microsoft tools, including Office, are generally better for exploring the contents of the OpenDSS type library and learning how to use the COM interface. The code completion feature in VBA and Visual Studio is particularly useful in showing what options are available after typing the name of the variable. Some users draft their code in VBA or C# and then translate to Matlab. There are also 3rd party Type Library or IDL

browsers/documenters that are helpful. If you are familiar with IDL, see the OpenDSSEngine.ridl file in the DLL folder under the SOURCE folder in the code.

DSS Interface Properties and Methods (from Excel)



Circuit Interface Properties and Methods