

IOWA STATE UNIVERSITY

ECpE Department

OpenDSS Tutorial and Cases

GRA: Fankun Bu

Advisor: Dr. Zhaoyu Wang

Department of Electrical and Computer Engineering

Iowa State University

Reference:

R. C. Dugan, The Open Distribution System Simulator (OpenDSS),
Electric Power Research Institute, Palo Alto, CA, USA, 2019.

Part I: OpenDSS Tutorial

Outline

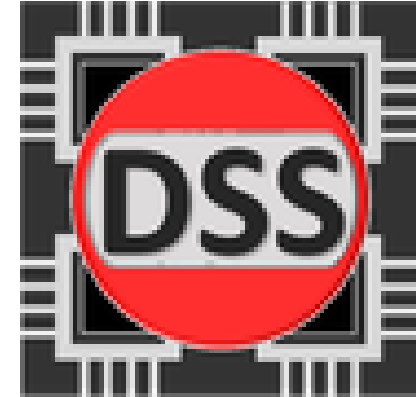
- Introduction
 - What is OpenDSS?
 - What can OpenDSS do?
 - OpenDSS Resources
- Installation
- User Interface
- Workflow of developing a project
 - Define a circuit
 - Set up the circuit options
 - Solve the circuit
 - Perform analysis

Introduction

What is OpenDSS?

The Open Distribution System Simulator (OpenDSS, or simply, DSS) is a comprehensive electrical system simulation tool for electric utility **distribution systems**.

- Open → Open Source
 - DSS → Distribution System Simulator
-
- The Development of OpenDSS began in April 1997. **Roger Dugan** is the principal author of the software.
 - In 2004, the DSS had been acquired by EPRI Solutions.
 - In 2008, EPRI released the software under an open-source license to cooperate with other grid modernization efforts.



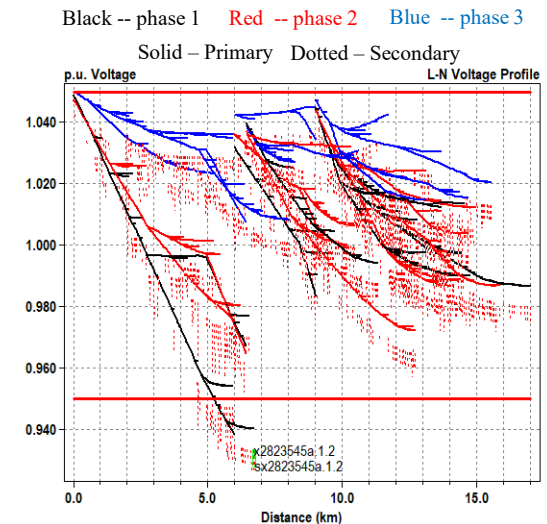
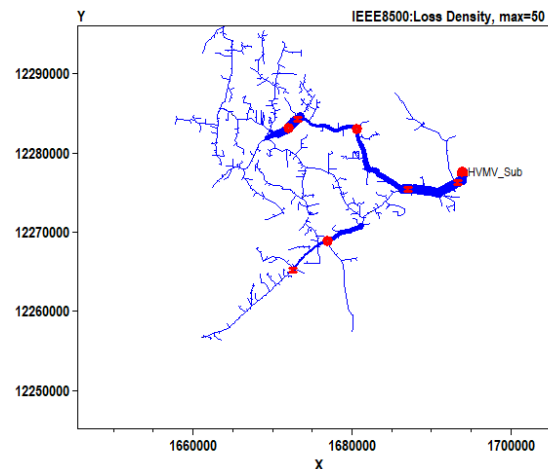
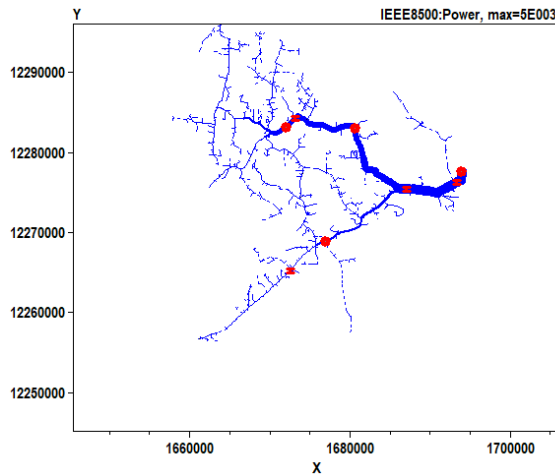
Introduction

What can OpenDSS do?

- Power flow analysis

When a power flow is completed, the losses, nodal voltages, currents and power flows are available for the total system, each component, and certain defined areas.

- OpenDSS can handle both *radial* distribution (MV) circuits and *network* (meshed) systems.
- It can also be used to solve *small-* to *medium-sized* networks with a transmission-style power flow.
- The power flow executes in numerous solution *modes* including the standard single Snapshot mode, and Daily mode, etc.



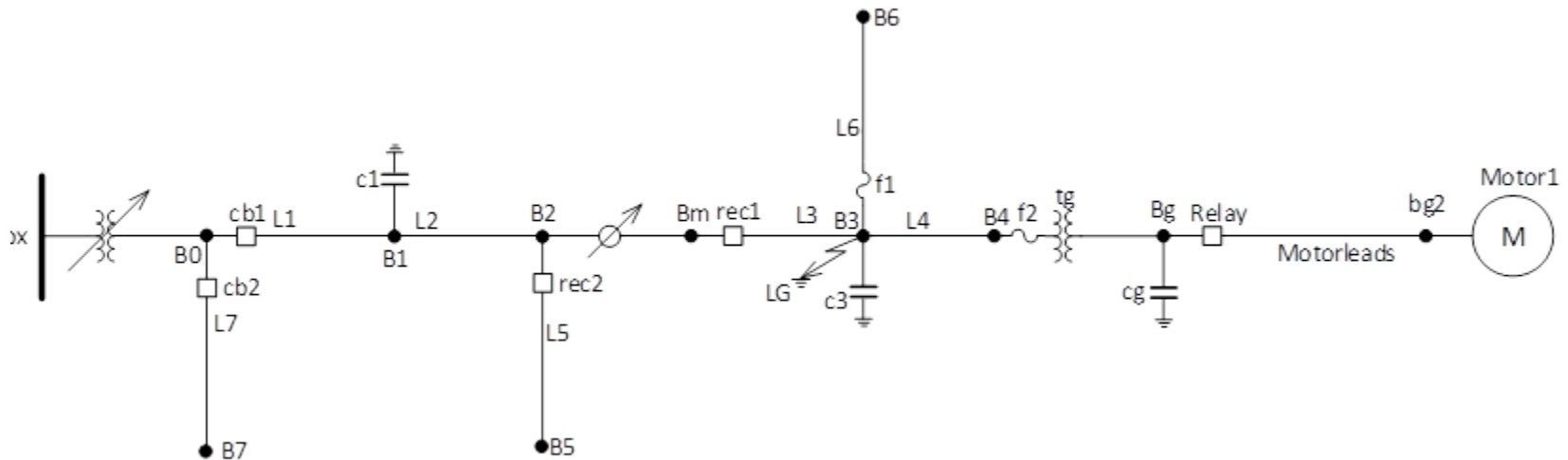
Introduction

What can OpenDSS do?

- Fault study

Fault study can give us the *short-circuit currents* and *voltages*, which can further be used for selecting circuit breakers, setting relays or reclosers, and analyzing the stability of system operation.

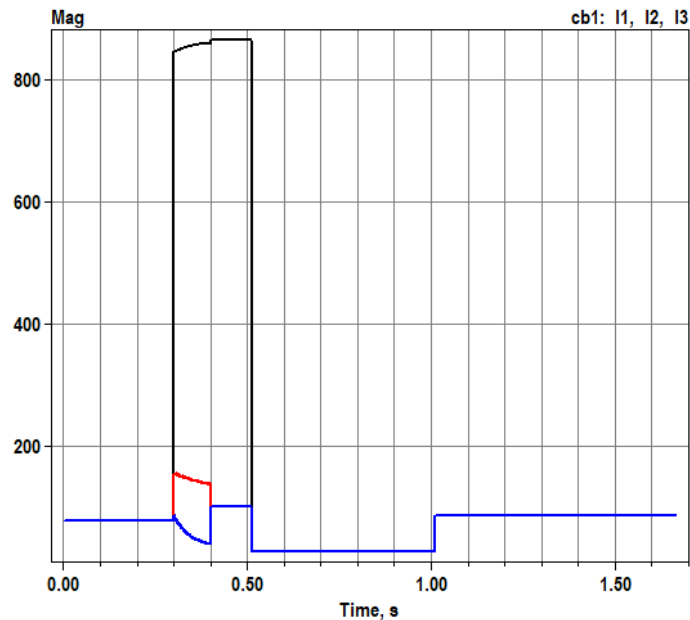
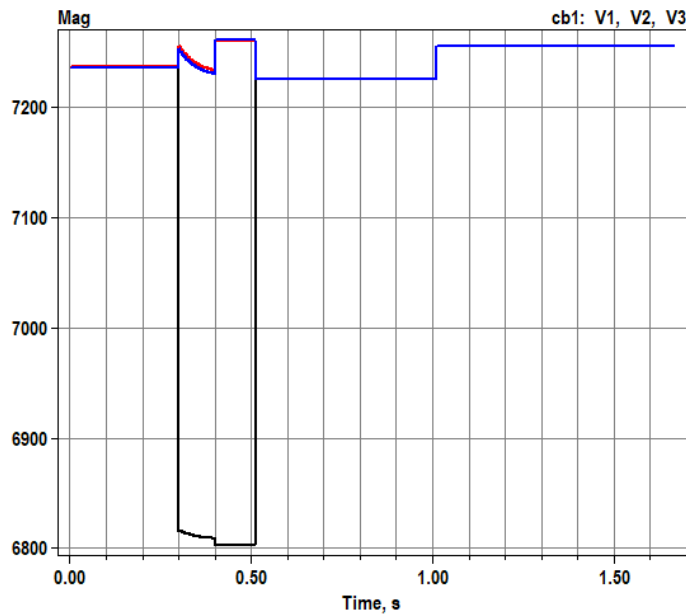
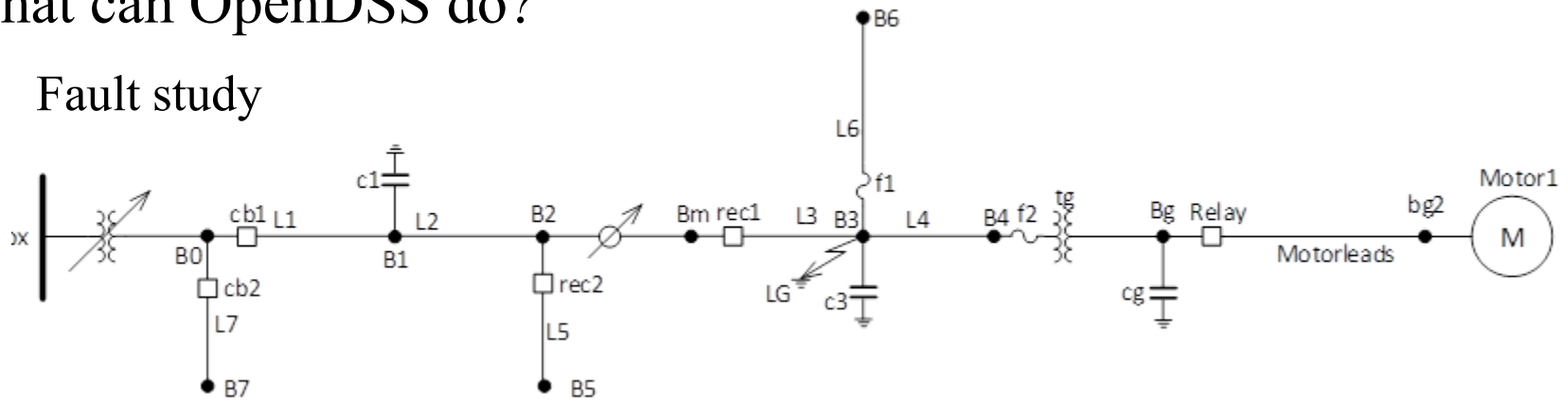
 - OpenDSS can perform fault study for all buses, reporting currents and voltages on all phases for all types of faults, including 3-phase faults, SLG faults on each phase, LL and L-L-G faults.



Introduction

What can OpenDSS do?

- Fault study



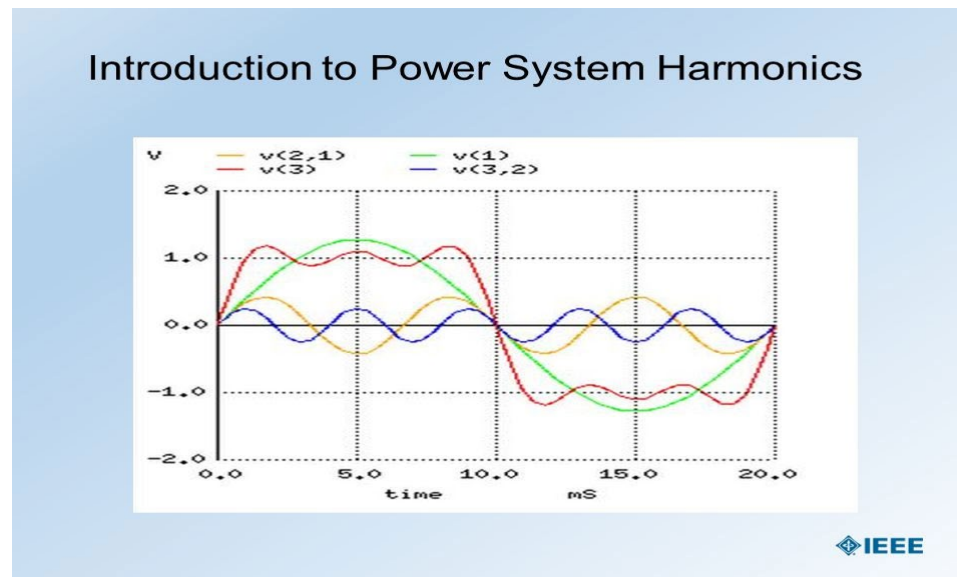
Black -- phase 1
Red -- phase 2
Blue -- phase 3

Introduction

What can OpenDSS do?

- Harmonic flow analysis

It can give us power flow results corresponding to each frequency. The user defines various harmonic spectra to represent harmonic sources of interest. The spectra are connected to Load, Generator, voltage source, current source objects and a few other power conversion elements as desired.

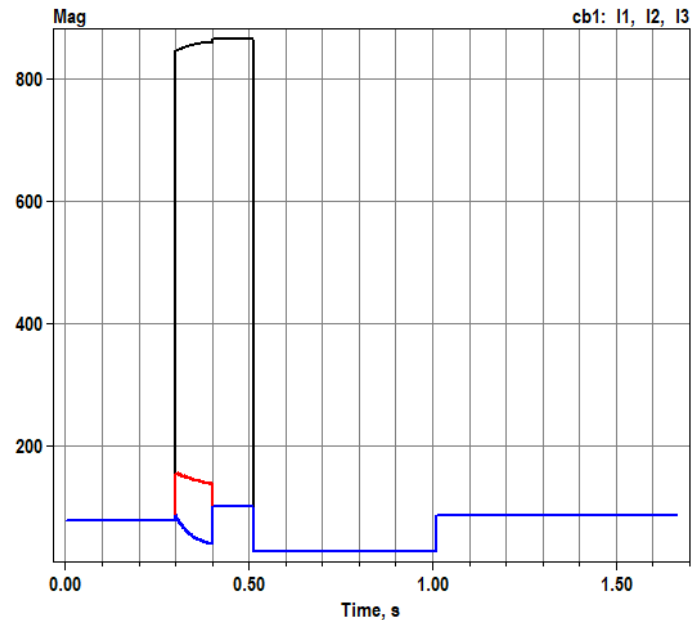
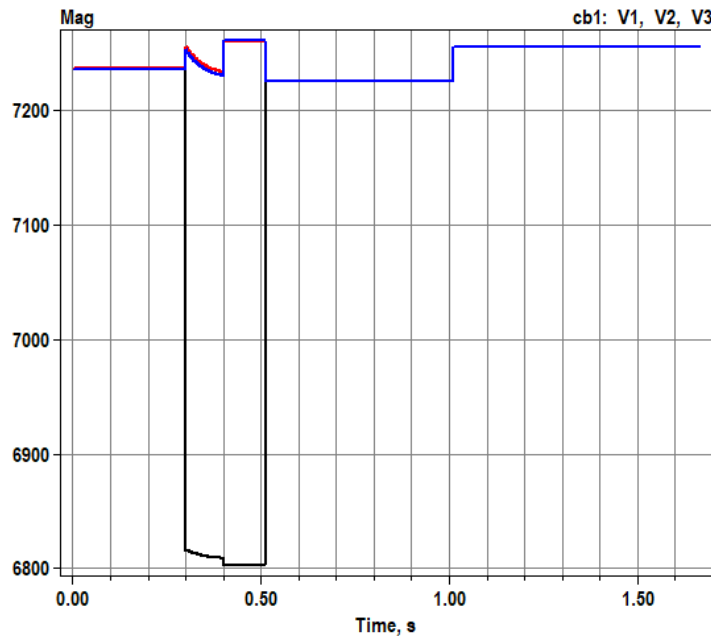


<https://slideplayer.com/slide/4397909/>

Introduction

What can OpenDSS do?

- Dynamics
The OpenDSS can perform basic electromechanical transient, or dynamic simulations.



Black -- phase 1
Red -- phase 2
Blue -- phase 3

Introduction

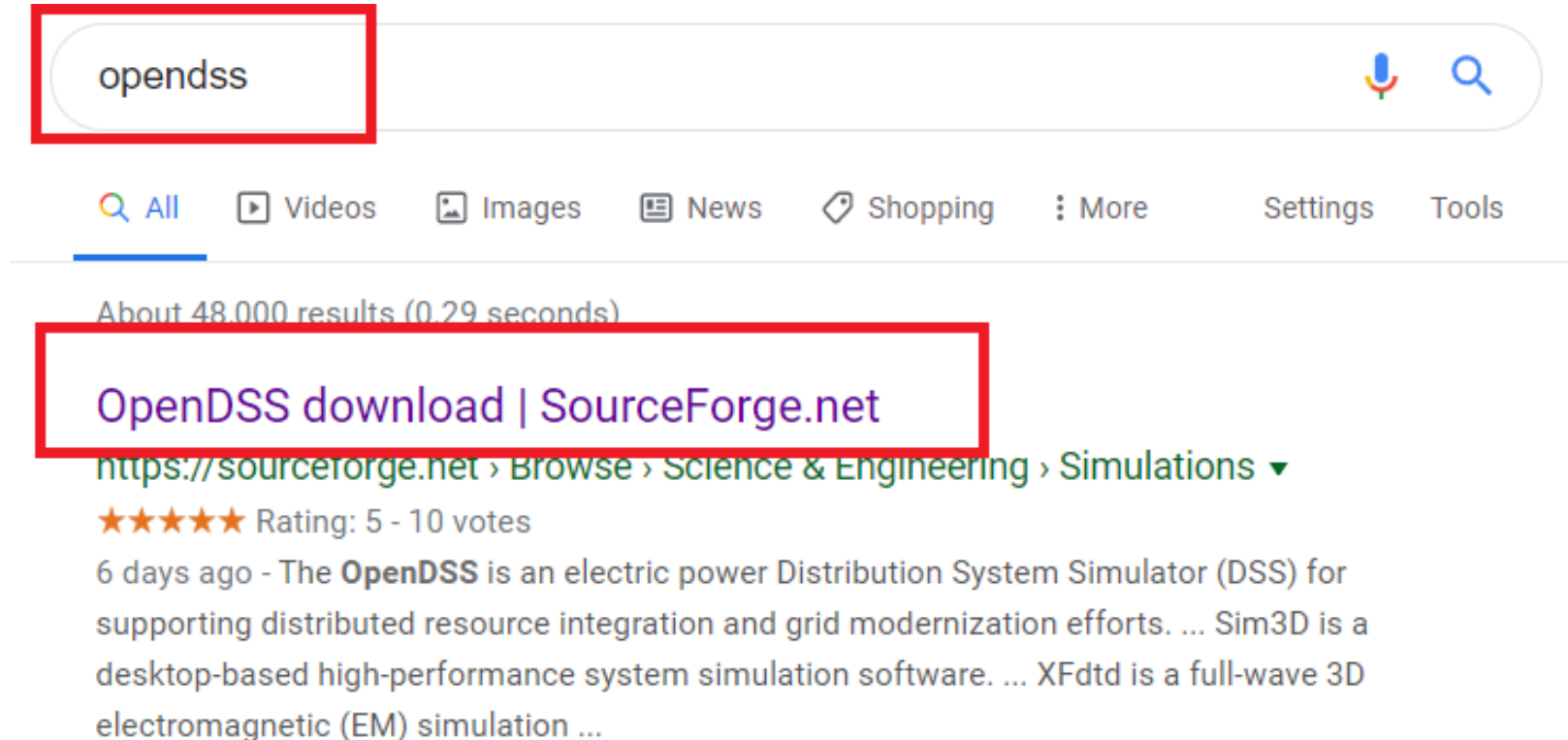
What can OpenDSS do?

- Specific Applications

- Distribution Planning and Analysis
- General Multi-phase AC Circuit Analysis
- Analysis of Distributed Generation Interconnections
- Annual Load and Generation Simulations
- Risk-based Distribution Planning Studies
- Probabilistic Planning Studies
- Solar PV System Simulation
- Wind Plant Simulations
- Nuclear Plant Station Auxiliary Transformer Modeling
- Distribution Automation Control Assessment
- Protection System Simulation
- Storage Modeling
- Distribution Feeder Simulation with AMI Data
- Distribution State Estimation
- Ground Voltage Rise on Transmission Systems
- Geomagnetically-Induced Currents (GIC)
- EV Impacts Simulations
- Co-simulation of Power and Communications Networks
- Analysis of Unusual Transformer Configurations
- Harmonic and Interharmonic Distortion Analysis
- Neutral-to-earth Voltage Simulations
- Development of IEEE Test feeder cases
- Phase Shifter Simulation
- Arc Furnace Simulation
- Impulse Loads (car crushers, etc.)
- And more

Introduction

Where to download and learn OpenDSS?



The image shows a Google search interface. The search bar contains the text "opendss". Below the search bar, there are navigation tabs for "All", "Videos", "Images", "News", "Shopping", "More", "Settings", and "Tools". The search results show "About 48,000 results (0.29 seconds)". The first result is "OpenDSS download | SourceForge.net" with a URL "https://sourceforge.net > Browse > Science & Engineering > Simulations". It has a rating of 5 stars from 10 votes and a description: "6 days ago - The **OpenDSS** is an electric power Distribution System Simulator (DSS) for supporting distributed resource integration and grid modernization efforts. ... Sim3D is a desktop-based high-performance system simulation software. ... XFDTD is a full-wave 3D electromagnetic (EM) simulation ...".

Introduction

Where to download and learn OpenDSS?

Home / Browse / Science & Engineering / Simulations / OpenDSS

OpenDSS

EPRC Distribution System Simulator
Brought to you by: [davismont](#), [rdugan](#), [robertkhenry](#), [tomedrm](#), [wsunderm1](#)

★★★★★ 10 Reviews

Downloads: 212 This Week

Last Update: 6 days ago

Download Get Updates Share This

Windows

- Summary
- Files
- Reviews
- Support
- Wiki
- Feature Requests
- News
- Discussion
- Code

The OpenDSS is an electric power Distribution System Simulator (DSS) for supporting distributed resource integration and grid modernization efforts. See Discussion Forum for latest news.

Project Samples

Introduction

Where to download and learn OpenDSS?

Home / Browse / Science & Engineering / Simulations / OpenDSS / Files

OpenDSS

EPRi Distribution System Simulator
Brought to you by: [davismont](#), [rdugan](#), [robertkhenry](#), [temcdm](#), [wsunderm1](#)

Summary **Files** Reviews Support Wiki Feature Requests News Discussion Code

[Download Latest Version](#)
OpenDSSInstaller861_1.exe (100.0 MB) [Get Updates](#)

Home

Name	Modified	Size	Downloads / Week
OpenDSSCmd	2019-08-23		8
OpenDSS	2019-08-14		178
User_Instructions_for_Parallel_Processing.pdf	2019-07-09	1.0 MB	8
readme.txt	2019-07-09	7.8 kB	6
License.txt	2019-03-19	1.7 kB	1
Getting Started With OpenDSS.pdf	2014-09-02	30.3 kB	11

Introduction

OpenDSS Forum

The screenshot shows the OpenDSS Forum website. At the top, there is a navigation bar with tabs for Summary, Files, Reviews, Support, Wiki, Feature Requests, News, Discussion (highlighted with a red circle), and Code. Below the navigation bar is a search bar labeled "Search Discussion" with a magnifying glass icon. On the left side, there is a sidebar with links for "Create Topic", "Stats Graph", and "Forums". Under "Forums", there are links for "Open Discussion (1315)", "Help (1037)", "Beginners (780)", "Experts (34)", and "Help". Below these is a link for "Formatting Help". The main content area is titled "Discussion" and contains a table with the following columns: FORUM, LATEST POST, and # TOPICS.

FORUM	LATEST POST	# TOPICS
Open Discussion	Voltages coming as zero by Roger Dugan 14 hours ago	1315
Help	Discrepancy between single and three phase load behavior by Roger Dugan 37 minutes ago	1037
Beginners This forum is for users new to OpenDSS	Set capacitor to be zero by Tom 31 seconds ago	780
Experts This forum is for expert users of OpenDSS	Primary load flow by TSI 2019-08-26	34

Introduction

OpenDSS Forum

The screenshot shows the OpenDSS Forum website. At the top, there is a navigation bar with links for Summary, Files, Reviews, Support, Wiki, Feature Requests, News, Discussion (highlighted with a red circle), and Code. Below the navigation bar is a search bar with the text "Search Discussion" and a magnifying glass icon. To the right of the search bar, there is a search bar with the text "single-phase center-tapped" (highlighted with a red box) and a "Search" button. Below the search bar, there are radio buttons for "Search discussion" (selected) and "Search entire project", and a "Search history?" checkbox. To the right of the search bar, there is a "Help" button. Below the search bar, there is a pagination link "1 2 3 .. 132 > >> (Page 1 of 132)". The search results are listed below, with the first three results highlighted by a red box:

- [Center-Tap transformer - phases](#) ^(Thread)
Center-Tap transformer - phases
Last updated: 2017-05-31
- [phase-phase center tap transformers](#) ^(Thread)
phase-phase center tap transformers
Last updated: 2019-01-10
- [Post by dkwappa on Polarities of Nodes](#) ^(Post)
for a **center tapped** transformer? Despite the transformer being defined as a **single phase** from phase 1
Last updated: 2015-08-07

Below the highlighted results, there are two more results:

- [Post by fcgaino on Center-Tap transformer - phases](#) ^(Post)
Having a **single phase** transformer with **center-tap** connected in **phase C "3"** on the primary side
Last updated: 2017-05-31
- [Alternative modeling for center-tapped transformers](#) ^(Thread)
Alternative modeling for **center-tapped** transformers
Last updated: 2019-08-23

Introduction

OpenDSS Forum

Summary Files Reviews Support Wiki Feature Requests News **Discussion** Code

Search Discussion

Create Topic

Stats Graph

Forums

Open Discussion (1315)

Help (1037)

Beginners (780)

Experts (34)

Help

Formatting Help

Center-Tap transformer - phases

Forum: [Beginners](#) Creator: [Fabio Gaino](#) Created: 2017-05-31 Updated: 2017-05-31

[Fabio Gaino](#) - 2017-05-31

Having a single phase transformer with center-tap connected in phase C ".3" on the primary side, is it possible to have phases A and B ".1.2" on the low voltage lines?

New Transformer.Example1-ph phases=1 windings=3
~buses=[bus1.3 bus2.1.0 bus2.0.2]
~kvs=[7.2 .12 .12]
~kvas=[25 25 25]
~cons=[wye wye wye]

New Line.L1 phases=2 Bus1=bus2.1.2.4 Bus2=bus3.1.2.4 Length=0.2 units=km linecode=Arranjo

[Roger Dugan](#) - 2017-05-31

These are node numbers not necessarily phase designations. The node numbers are nearly arbitrary. By convention, nodes 1, 2, 3 correspond to phases ABC on three-phase systems. Of course, OpenDSS can accommodate high phase order models as well. So you can call the nodes anything you want on the LV side of the transformer. Of course, 0 is reserved for the reference voltage.

Please review bus and node definitions from the user manual.

[Log in](#) to post a comment.

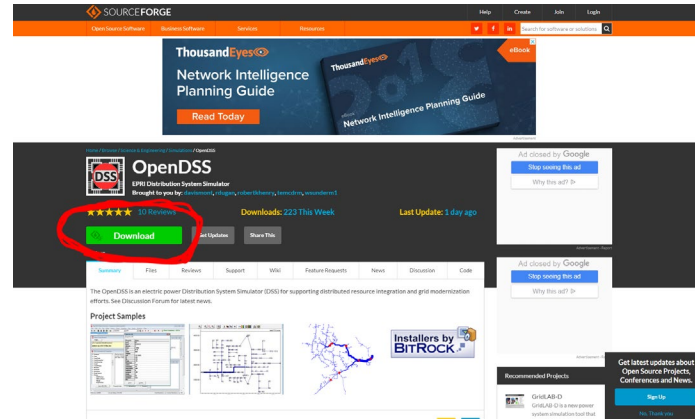
Introduction

Tutorial Videos

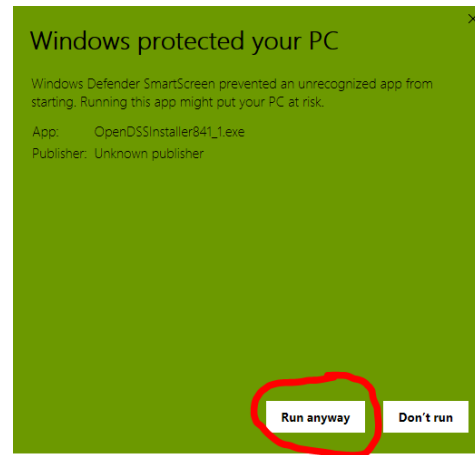
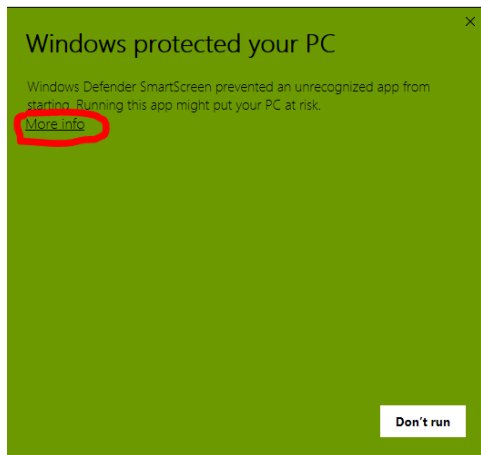
- Introduction:
https://www.youtube.com/watch?v=RGbflfhGcRg&list=PLcOap2oqW_gEMEVH9dg2HoXJ4NvydfsZM
- Line:
https://www.youtube.com/watch?v=jUSWAC0jNDU&list=PLcOap2oqW_gEMEVH9dg2HoXJ4NvydfsZM&index=7
- Linecode:
https://www.youtube.com/watch?v=k8l3Lo10dgk&list=PLcOap2oqW_gEMEVH9dg2HoXJ4NvydfsZM&index=8
- Transformer:
https://www.youtube.com/watch?v=z9EbQCmaWBo&list=PLcOap2oqW_gEMEVH9dg2HoXJ4NvydfsZM&index=12
- Capacitor:
https://www.youtube.com/watch?v=cggpjOixWUI&list=PLcOap2oqW_gEMEVH9dg2HoXJ4NvydfsZM&index=11
- Load:
https://www.youtube.com/watch?v=FANmMQPnDPY&list=PLcOap2oqW_gEMEVH9dg2HoXJ4NvydfsZM&index=10

Installation

- Enter the link <https://sourceforge.net/projects/electricdss/> in your internet browser.
- Click “Download” button in the figure below, the software will be downloaded automatically.

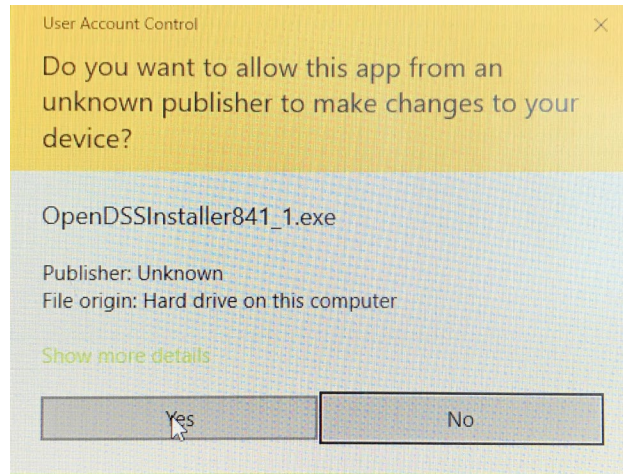


- Open the downloaded OpenDSSInstaller.
(If your PC displays the reminder as below, click “more info”, then click “Run anyway”.)

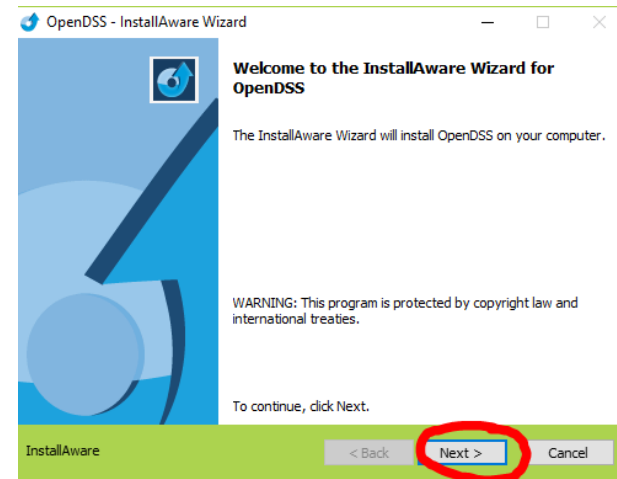


Installation

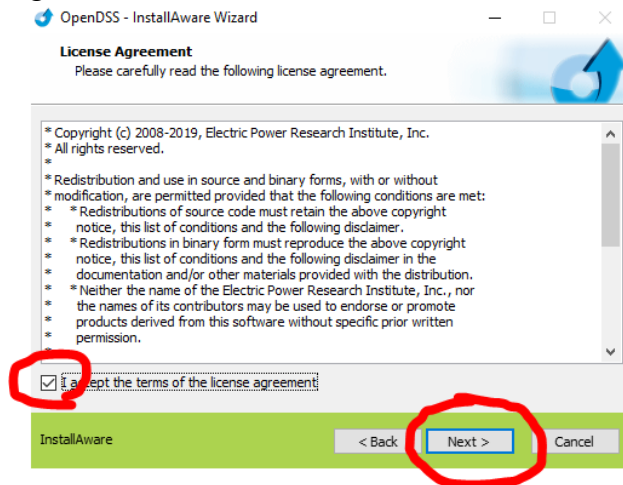
- Click “Yes” in the popup dialog box.



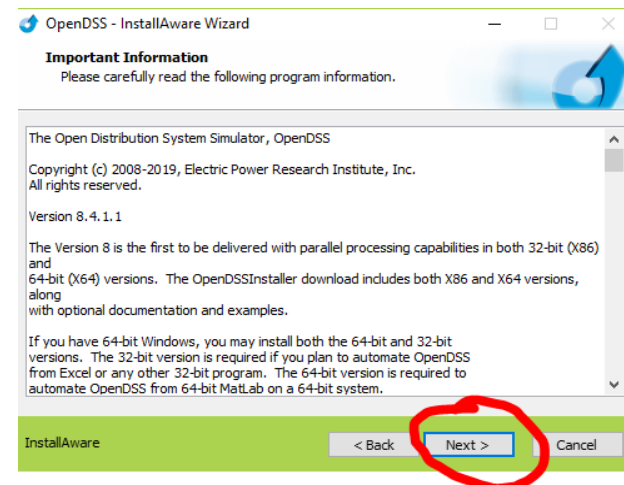
- Click “Next” button.



- Select “I accept the terms of the license agreement” and click “Next”.

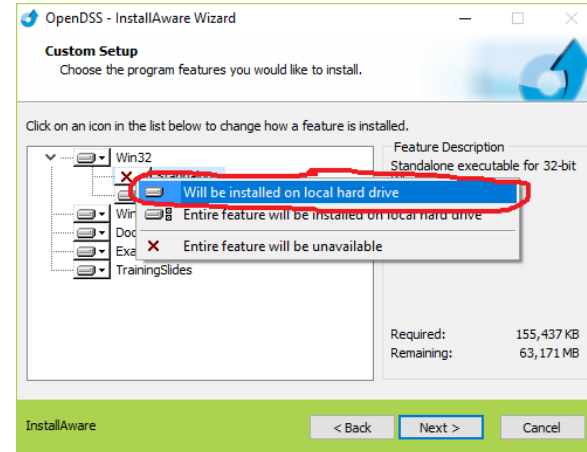
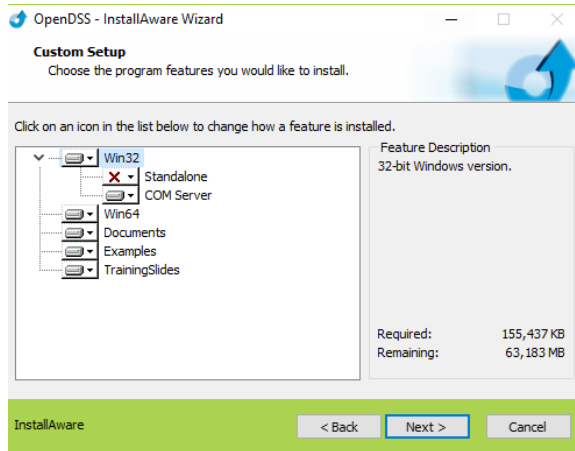


- Click “Next”.

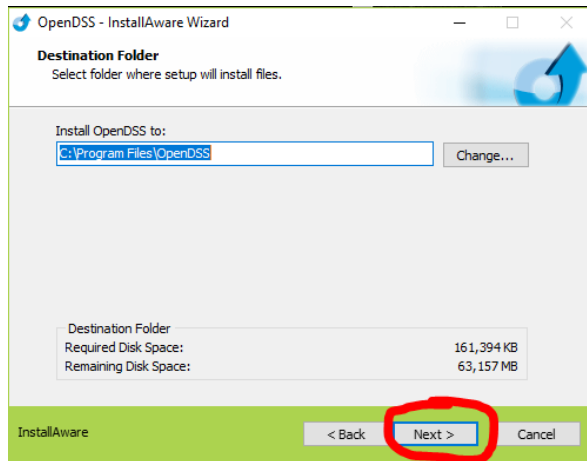


Installation

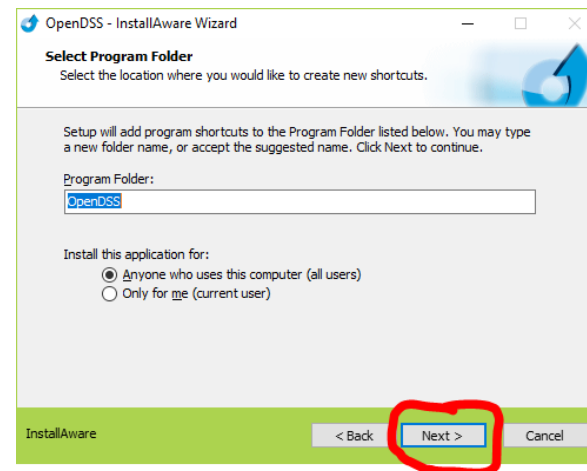
- Then, a dialog box like the picture on the left will appear. Click the red cross and select “Will be installed on local hard drive”.



- Choose the directory where you want to install and click “Next”.

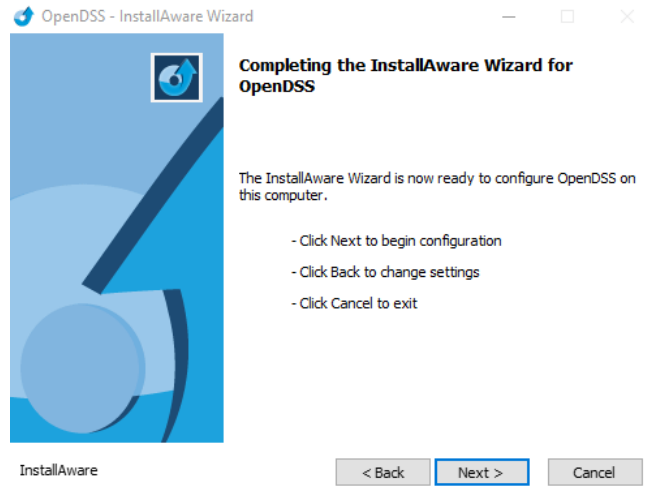


- Click “Next”.

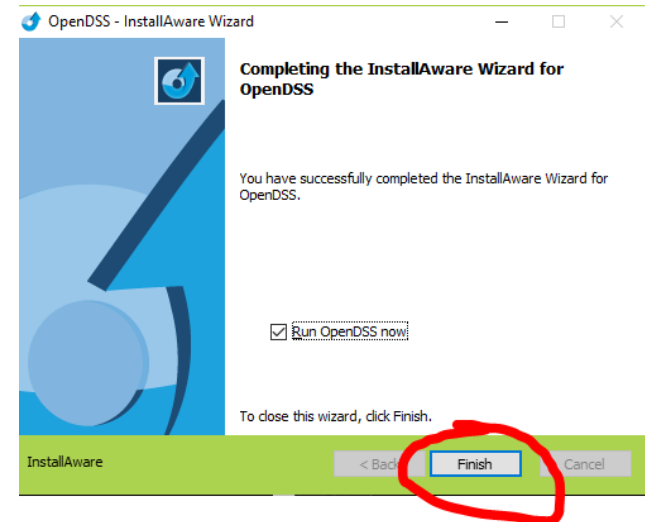


Installation

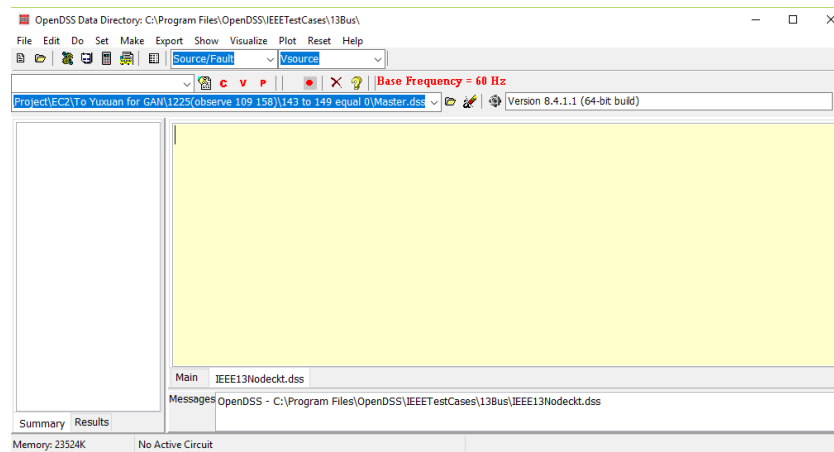
- Click “Next”.



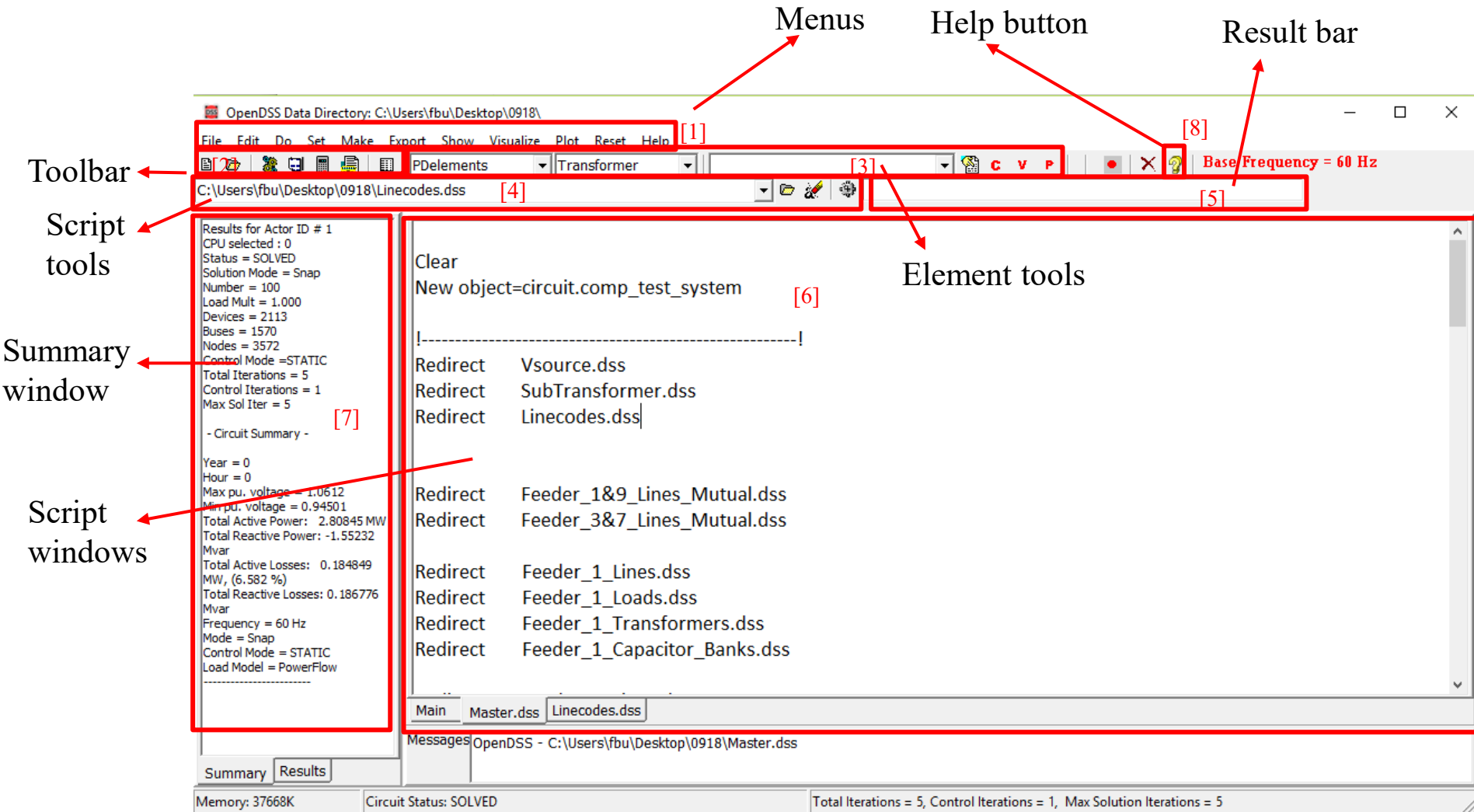
- Click “Finish” and check if the software has been successfully installed.



- If successful, an interface like below will appear.

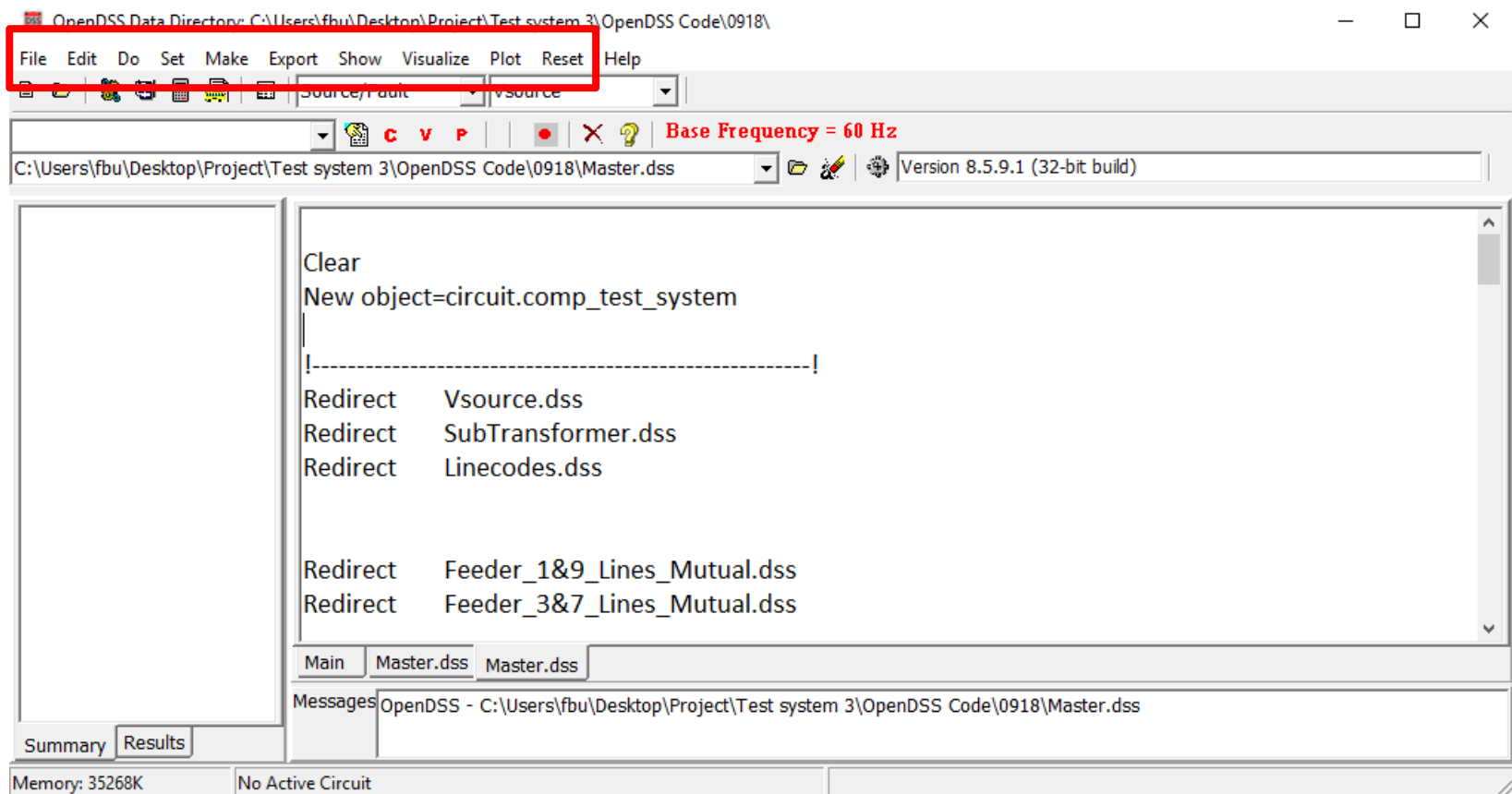


User Interface



User Interface

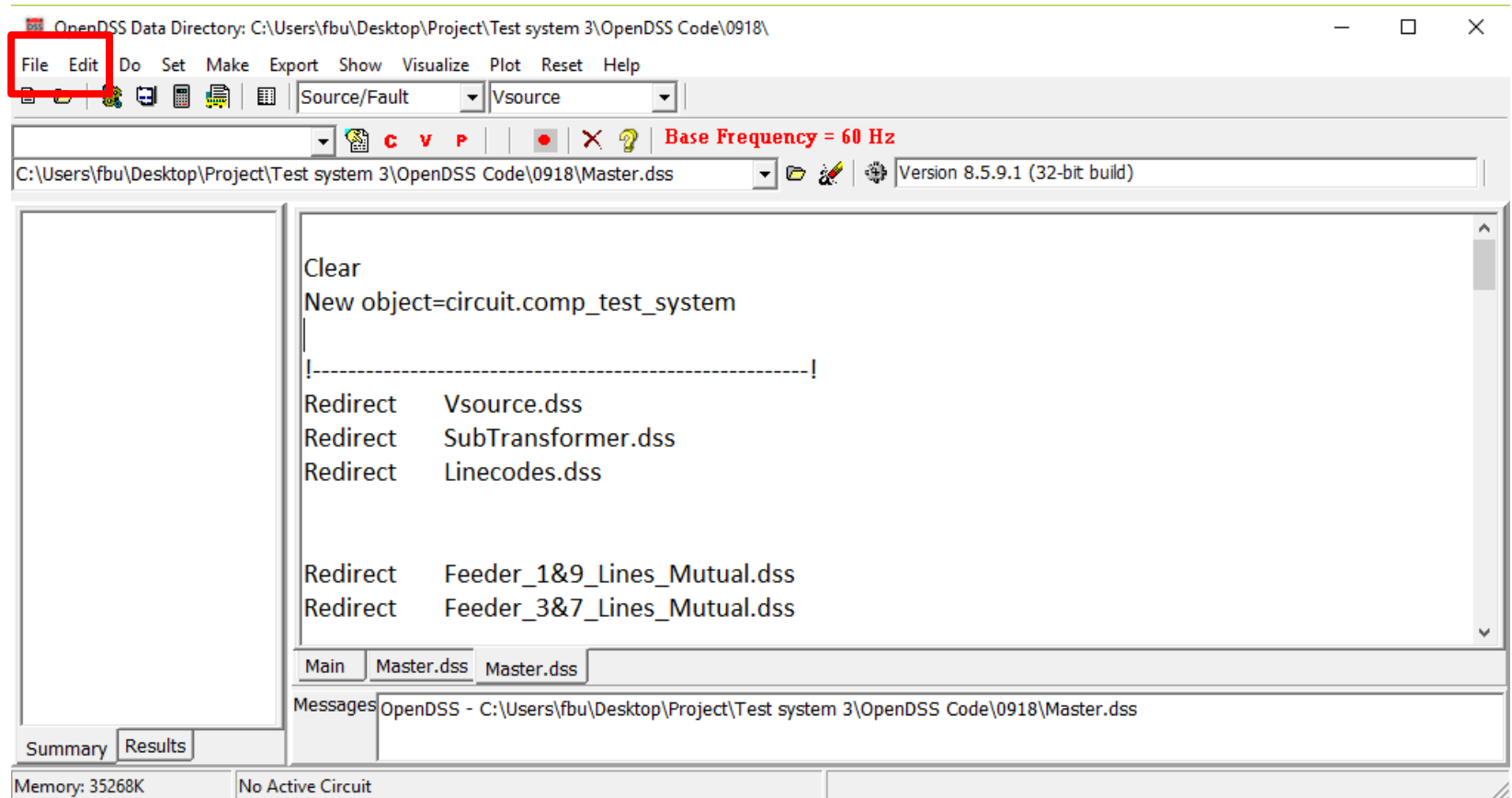
- [1] Menus: drive most of the workflow in OpenDSS.



Note that all these menus can be executed via writing alternative commands.

User Interface

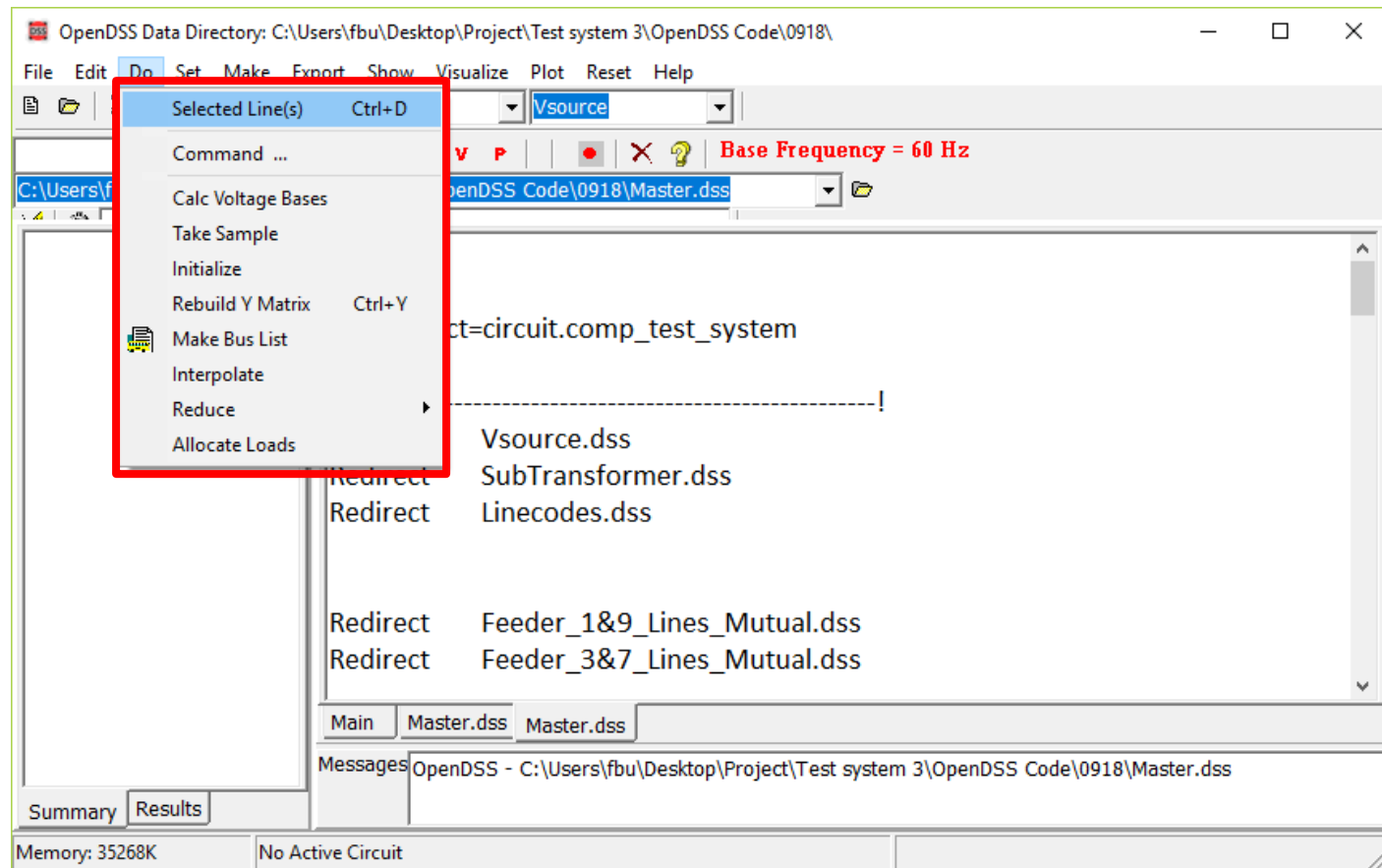
- [1] Menus
-- File and Edit: create, open, save and edit project files.



User Interface

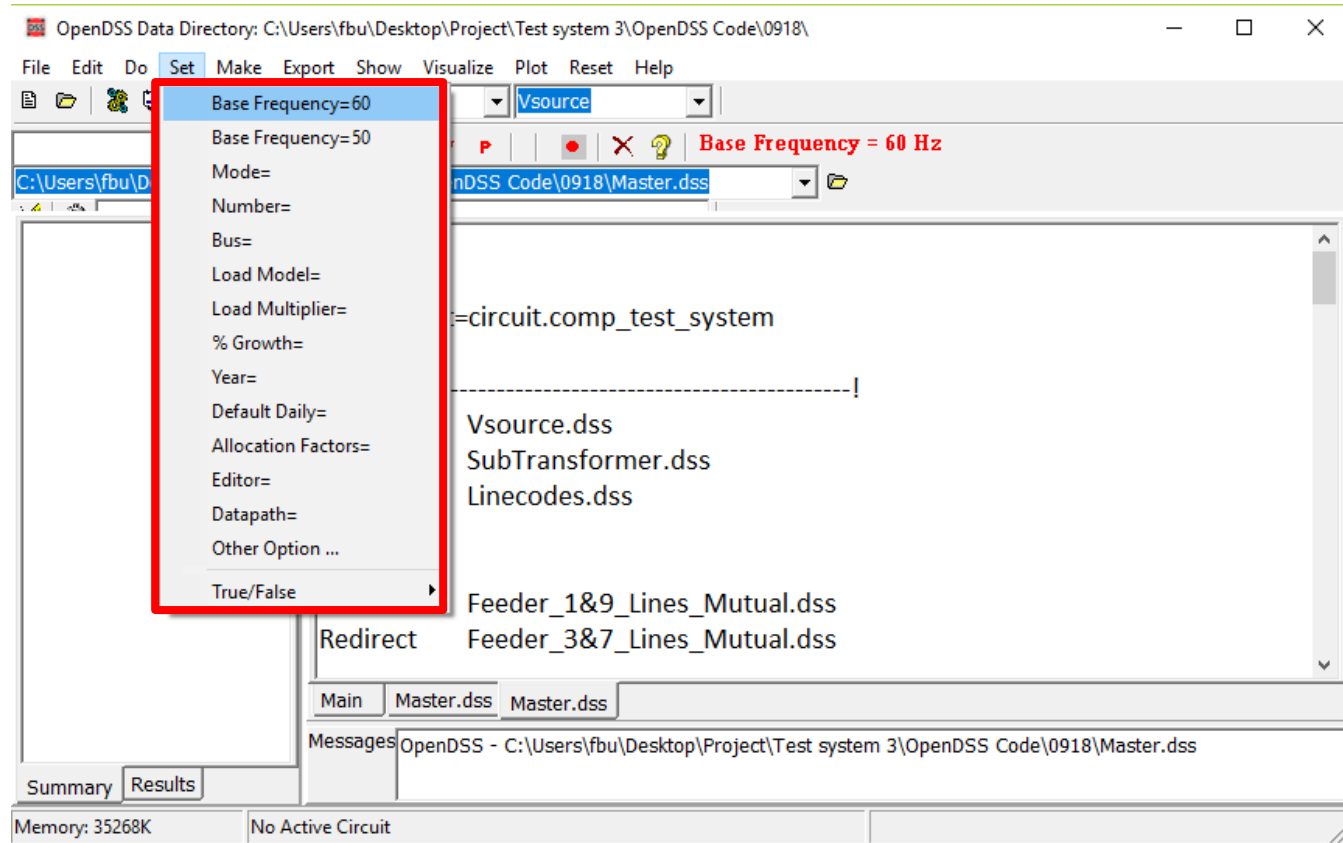
- [1] Menus
 - Do: contains the commands which are used for executing OpenDSS text strings.

The command “Selected Line(s)” is commonly used, it is like the general command “Run”.



User Interface

- [1] Menus
 - Set: allows us to set any solution parameters that can be set via the option scripting commands.



User Interface

- [1] Menus
-- Set

The image shows two screenshots of the OpenDSS software interface. The left screenshot shows the 'Set' menu with 'Base Frequency=60' highlighted. The right screenshot shows the command window with 'Set Frequency=60' entered. A double-headed arrow labeled 'Equivalent' connects the two screenshots.

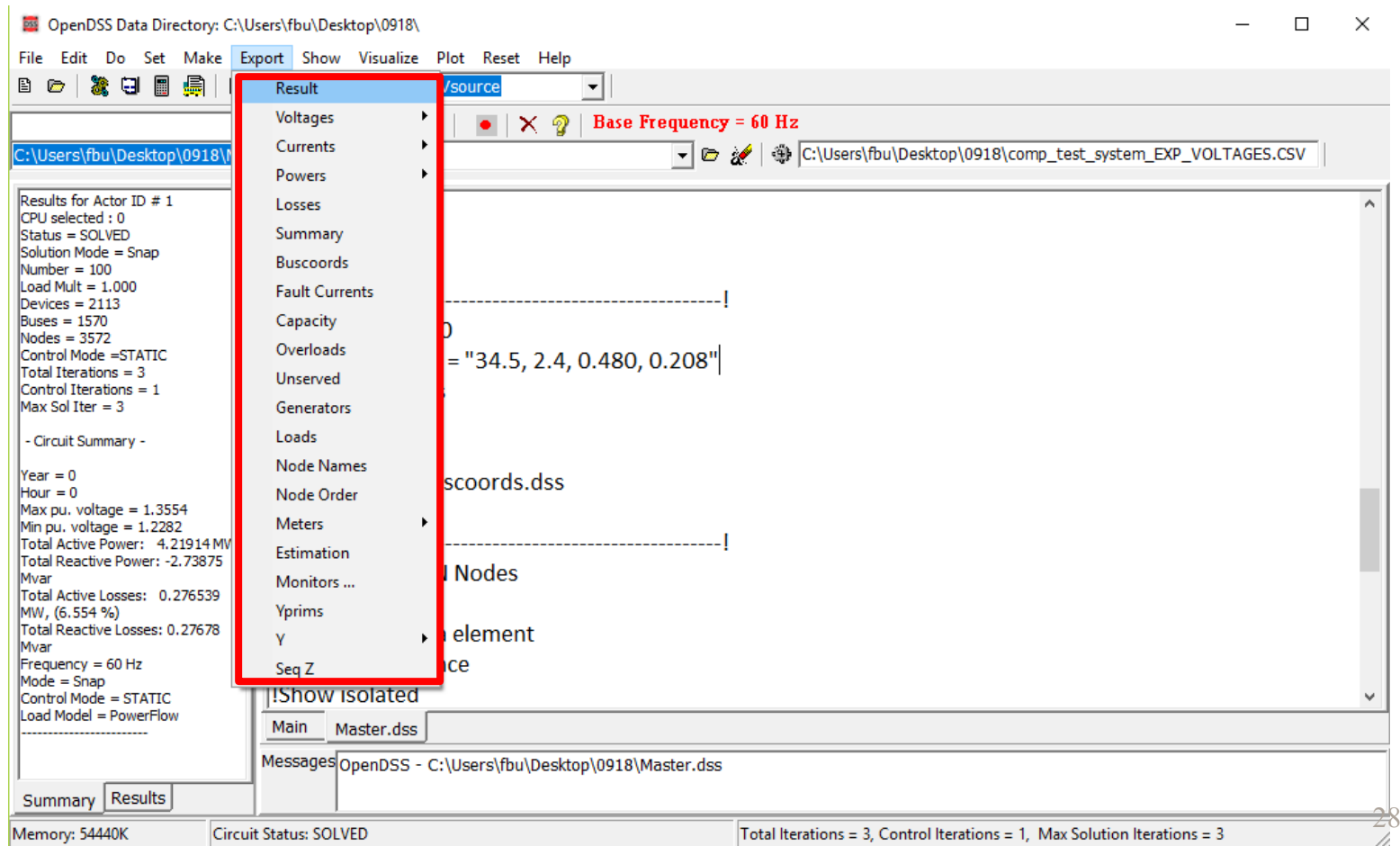
Left Screenshot: OpenDSS Data Directory: C:\Users\fbu\Desktop\EE. The 'Set' menu is open, showing options like 'Base Frequency=60', 'Base Frequency=50', 'Mode=', 'Number=', 'Bus=', 'Load Model=', 'Load Multiplier=', '% Growth=', 'Year=', 'Default Daily=', 'Allocation Factors=', 'Editor=', 'Datapath=', 'Other Option ...', and 'True/False'. The 'Base Frequency=60' option is highlighted with a red box.

Right Screenshot: OpenDSS Data Directory: C:\Users\fbu\Desktop\0918\Master.dss. The command window shows the command 'Set Frequency=60' entered, highlighted with a red box. The output shows 'Results for Actor ID # 1' and 'Control Mode =STATIC'. The status bar at the bottom indicates 'Circuit Status: SOLVED'.

Equivalent

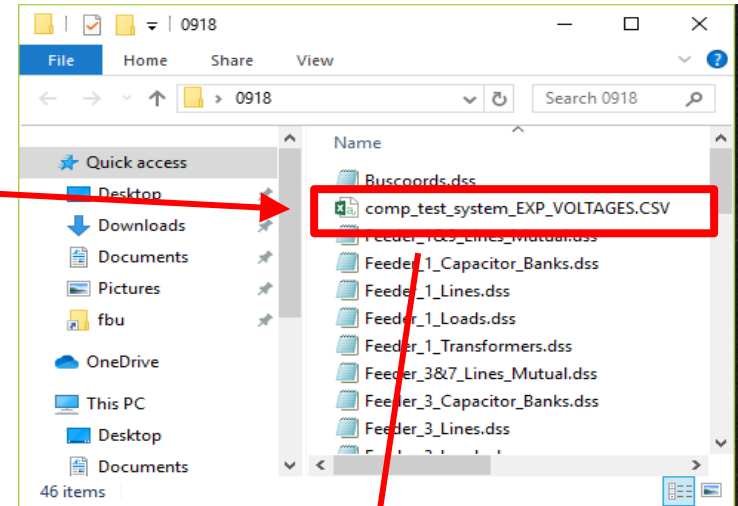
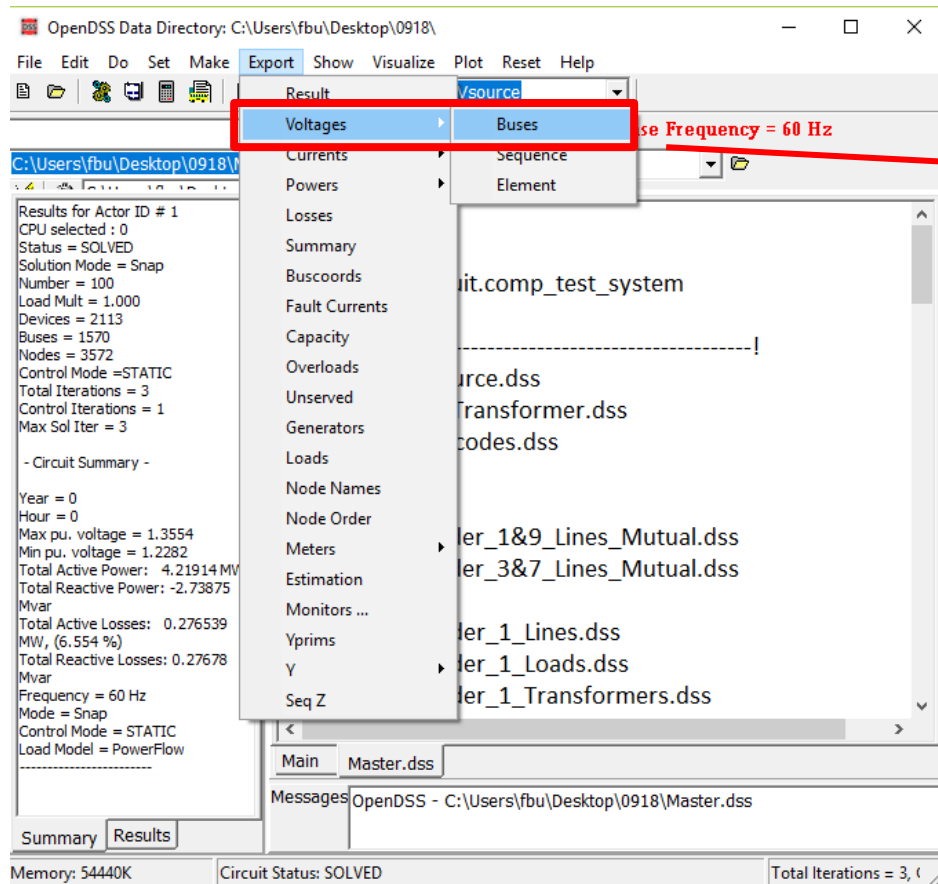
User Interface

- [1] Menus
 - Export: allows us to save various reports to csv files.



User Interface

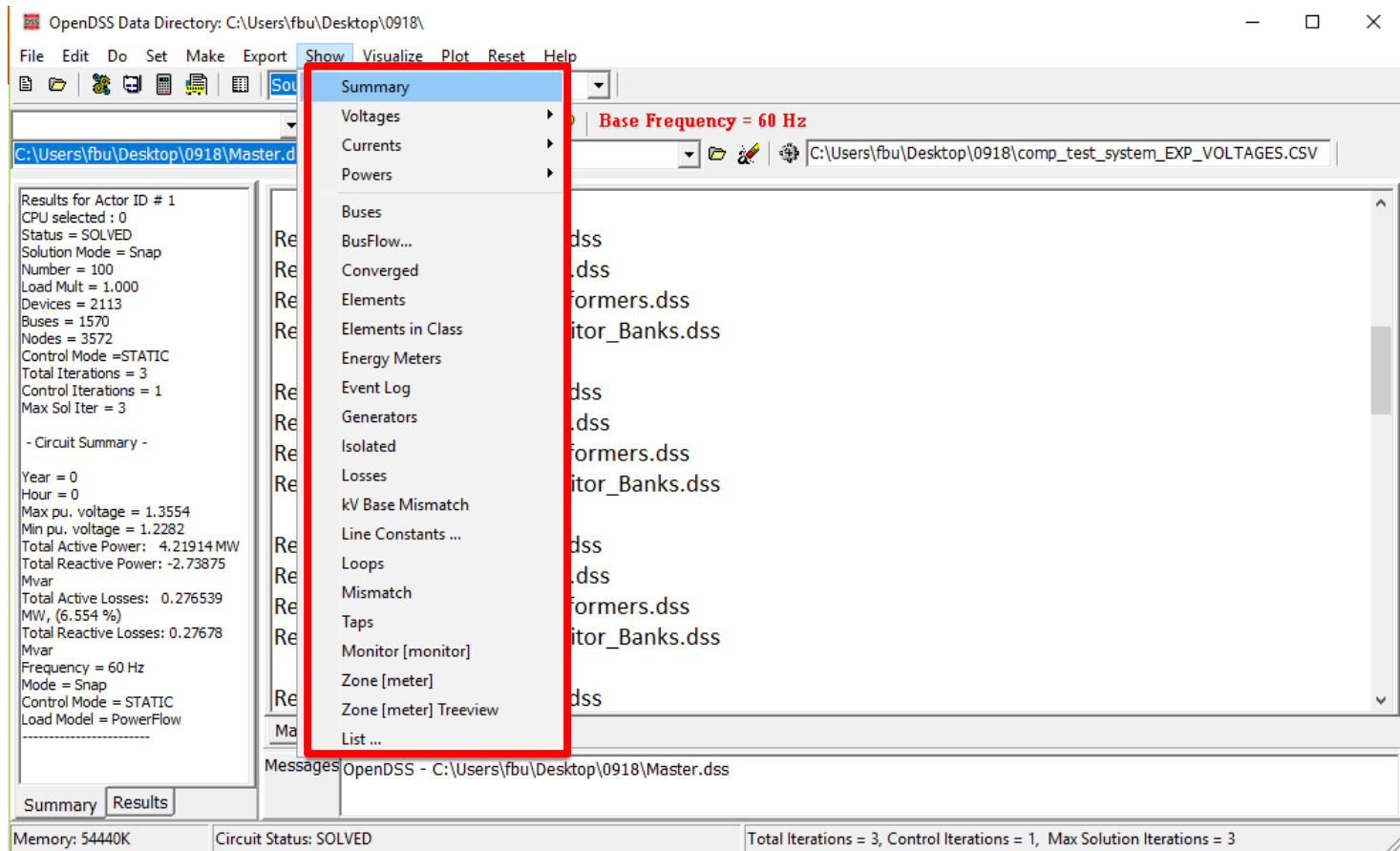
- [1] Menus
- Export



	A	B	C	D	E
1	Bus	BasekV	Node1	Magnitude1	Angle1
2	EQ_SOURCE_BUS	34.5	1	26003.7	-1.6
3	FDR-STRT-1	2.4	1	1816.92	-32.2
4	FDR-STRT-2	2.4	1	1812.48	-32
5	1&9-B28T	2.4	1	1813.12	-152.1
6	1&9-A	2.4	1	1812.43	-152.1
7	1&9-B	2.4	1	1812.2	-152.1
8	1&9-1	2.4	1	1810.93	-152.1
9	1&9-2	2.4	1	1809.84	-152.2
10	1&9-3	2.4	1	1808.86	-152.2

User Interface

- [1] Menus
 - Show: contains much of the same information as the exported reports using the Export menu, but displays them directly in the GUI.



User Interface

- [1] Menus
-- Show

OpenDSS Data Directory: C:\Users\fbu\Desktop\0918\

File Edit Do Set Make Export Show Visualize Plot Reset Help

Summary

Voltages LN

popup

Results for Actor ID # 1
CPU selected : 0
Status = SOLVED
Solution Mode = Snap
Number = 100
Load Mult = 1.000
Devices = 2113
Buses = 1570
Nodes = 3572
Control Mode =STATIC
Total Iterations = 3
Control Iterations = 1
Max Sol Iter = 3

- Circuit Summary -
Year = 0
Hour = 0
Max pu. voltage = 1.3554
Min pu. voltage = 1.2282
Total Active Power: 4.21914 MW
Total Reactive Power: -2.73875 Mvar
Total Active Losses: 0.276539 MW, (6.554 %)
Total Reactive Losses: 0.27678 Mvar
Frequency = 60 Hz
Mode = Snap
Control Mode = STATIC
Load Model = PowerFlow

OpenDSS - C:\Users\fbu\Desktop\0918\Master.dss

Memory: 54440K Circuit Status: SOLVED Total Itera

comp_test_system_VLN.Txt - Notepad

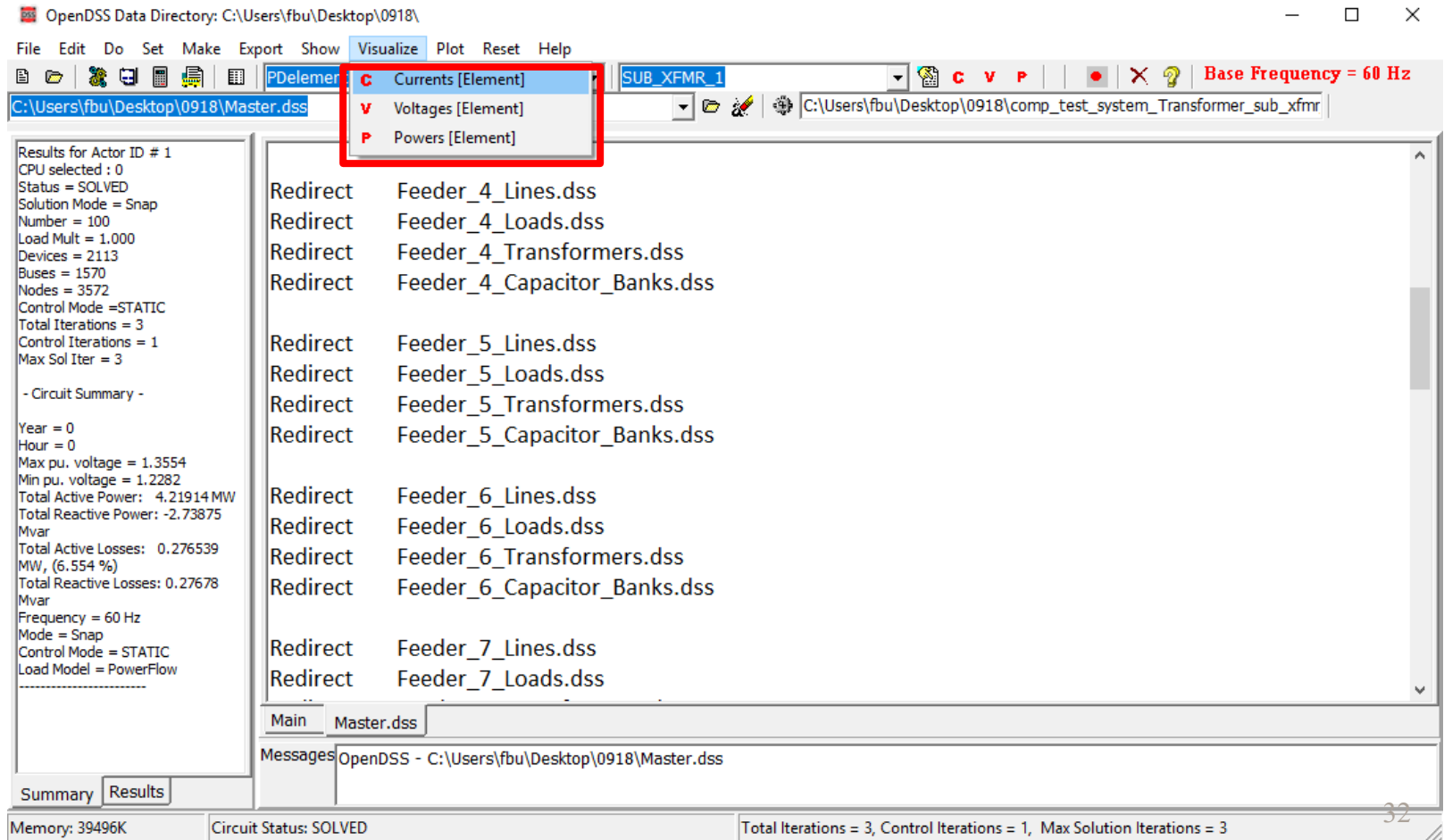
File Edit Format View Help

SYMMETRICAL COMPONENT VOLTAGES BY BUS (for 3-phase buses)

Bus	Mag:	V1 (kV)	p.u.	V2 (kV)	%V2/V1	V0 (kV)	%V0/V1
eq_source_bus	26.01	1.306	0.01092	0.04199	1.684E-009	6.477E-009	
fdr-str1-1	1.818	1.312	0.001295	0.07124	0.0005336	0.02935	
fdr-str1-2	1.813	1.309	0.0007208	0.03975	0.0001336	0.007367	
189-b28t	0.0007858	0.0005671	1.812	2.306E005	0.0003349	42.63	
189-a	0.0008766	0.0006326	1.811	2.066E005	0.0006157	70.24	
189-b	0.0009113	0.0006577	1.811	1.987E005	0.0007116	78.08	
189-1	0.001133	0.0008157	1.809	1.601E005	0.001245	110.1	
189-2	0.001333	0.0009623	1.807	1.356E005	0.001664	124.8	
189-3	0.001524	0.0011	1.806	1.185E005	0.002041	133.9	
189-4	0.001721	0.001242	1.805	1.049E005	0.002419	140.6	
189-5	0.001973	0.001424	1.803	9.137E004	0.002893	146.6	
189-6	0.002229	0.001609	1.801	8.08E004	0.003347	150.1	
189-8	0.002719	0.001962	1.798	6.614E004	0.004184	153.9	
189-9	0.002926	0.002112	1.797	6.14E004	0.004538	155.1	
189-10	0.003185	0.002299	1.795	5.636E004	0.004949	155.4	
189-11	0.003315	0.002392	1.794	5.413E004	0.005154	155.5	
189-12	0.003522	0.002542	1.793	5.09E004	0.005464	155.1	
189-13	0.003755	0.00271	1.792	4.771E004	0.00584	155.5	
189-14	0.004314	0.003113	1.789	4.148E004	0.00674	156.3	
189-15	0.00458	0.003305	1.788	3.904E004	0.007166	156.5	
189-16	0.005144	0.003712	1.786	3.471E004	0.008071	156.9	
189-17	0.005411	0.003905	1.784	3.298E004	0.008484	156.8	
189-18	0.005708	0.00412	1.783	3.124E004	0.008944	156.7	
189-19	0.006006	0.004335	1.782	2.967E004	0.009405	156.6	
189-20	0.006096	0.004399	1.781	2.922E004	0.009544	156.6	
189-21	0.006304	0.004549	1.78	2.824E004	0.009994	158.5	
189-21x	0.006539	0.004719	1.779	2.721E004	0.0105	160.6	
387-a	1.818	1.312	0.001553	0.08547	0.001114	0.06131	
387-1	1.817	1.312	0.001901	0.1046	0.001711	0.09417	
387-2	1.817	1.311	0.00208	0.1144	0.001987	0.1094	
387-3	1.817	1.311	0.00232	0.1277	0.002342	0.1289	
387-4	1.817	1.311	0.002656	0.1462	0.002862	0.1575	
387-5	1.817	1.311	0.003056	0.1682	0.003477	0.1914	

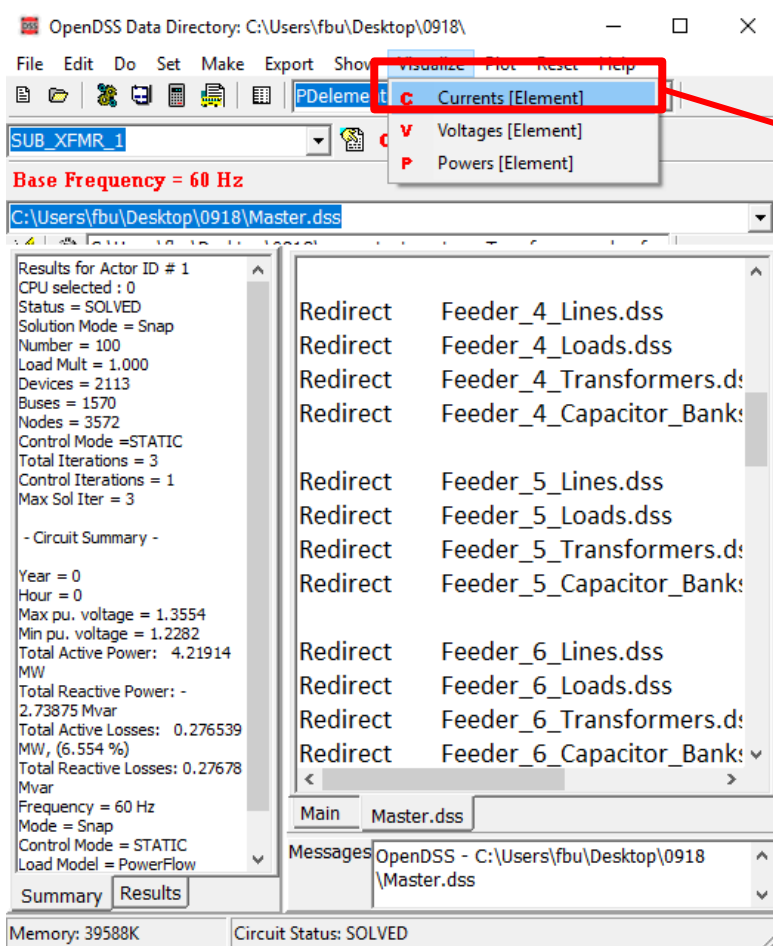
User Interface

- [1] Menus
 - Visualize: provides a graphical output of the device selected via the element selector.

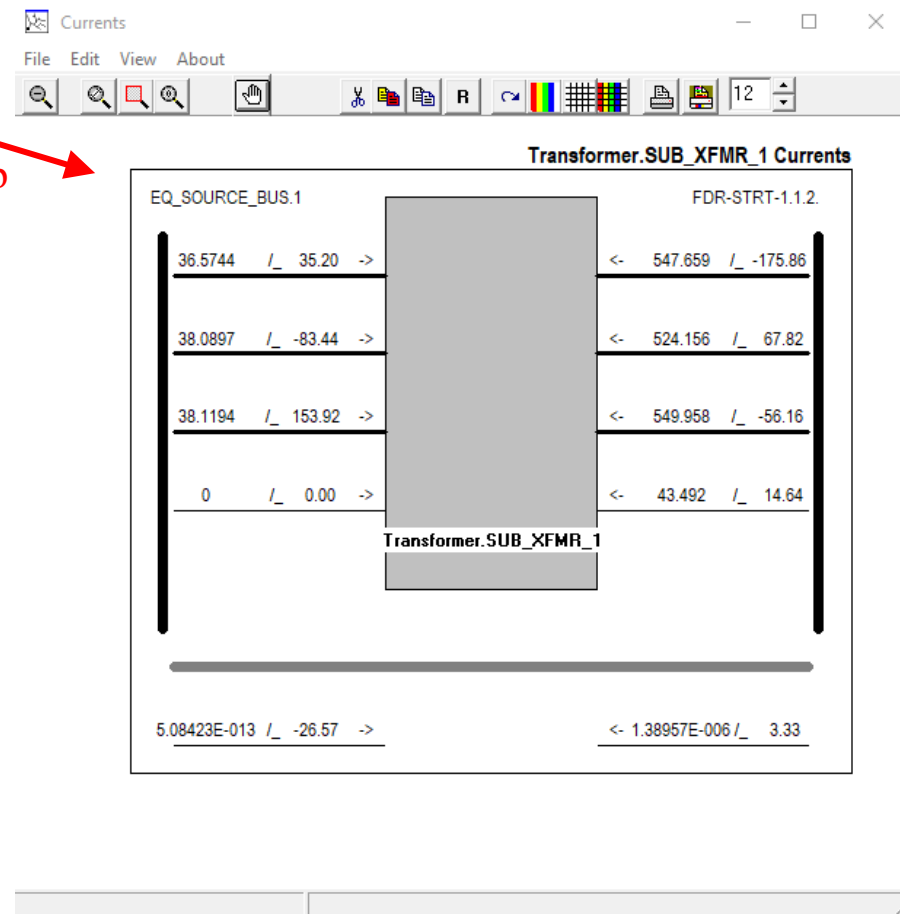


User Interface

- [1] Menus
-- Visualize

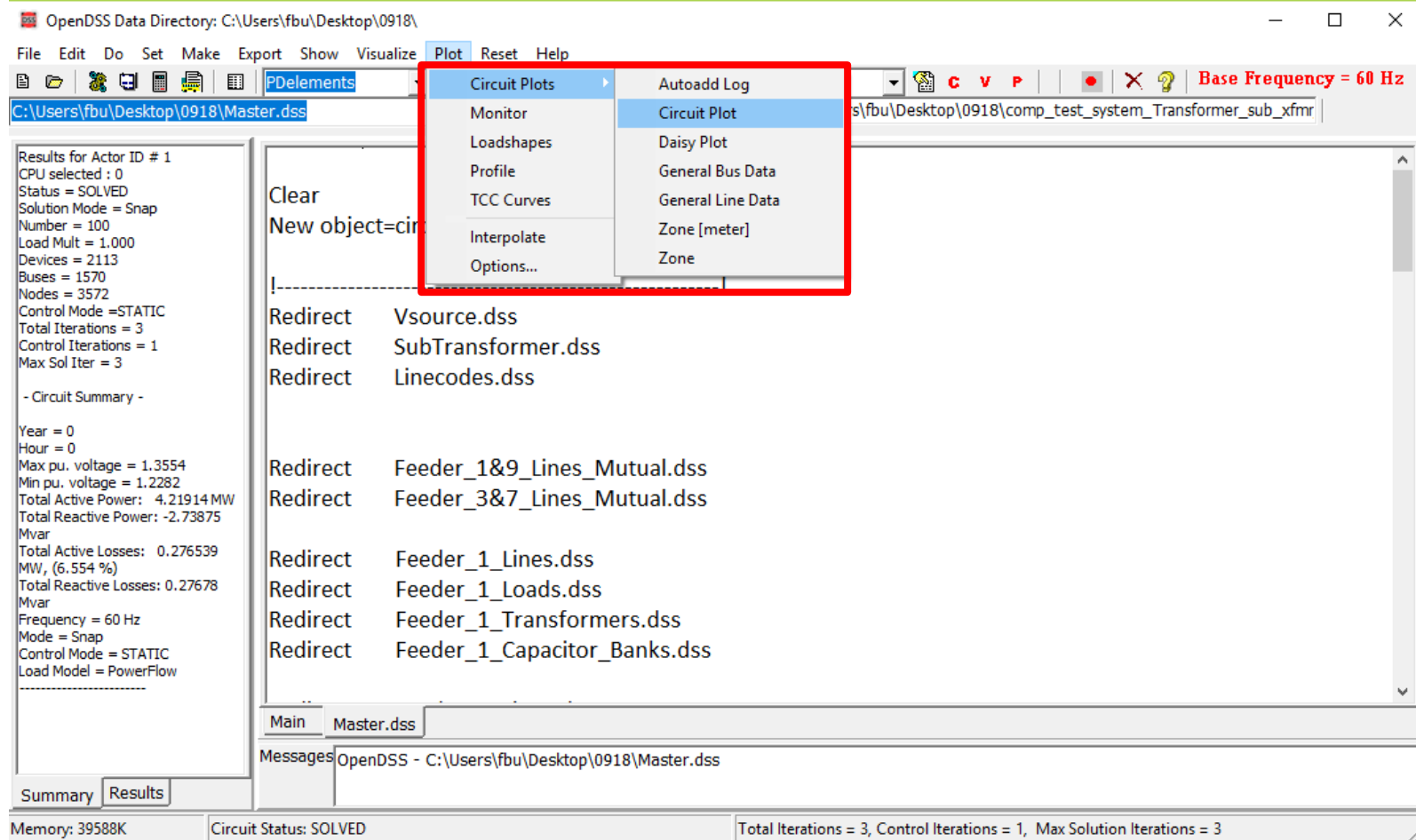


popup



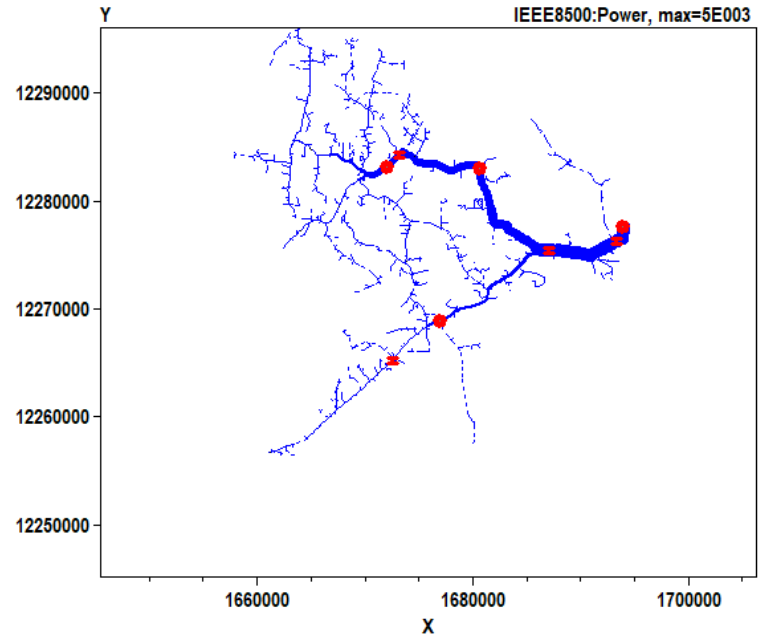
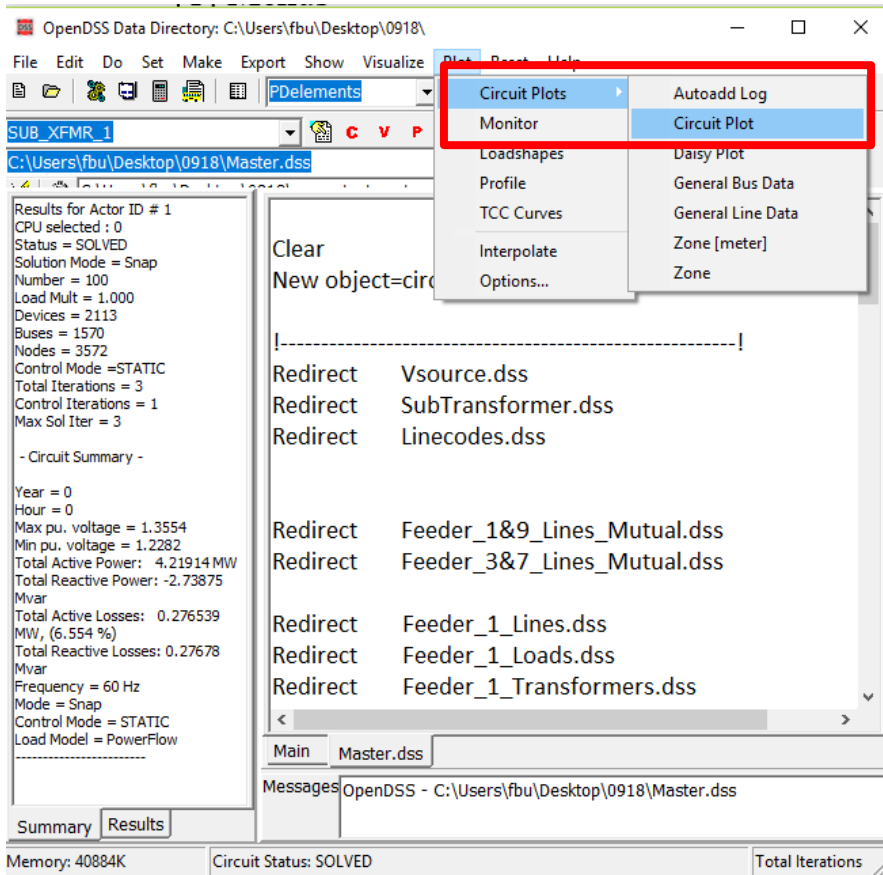
User Interface

- [1] Menus
 - Plot: provides graphical output relevant to the whole system.



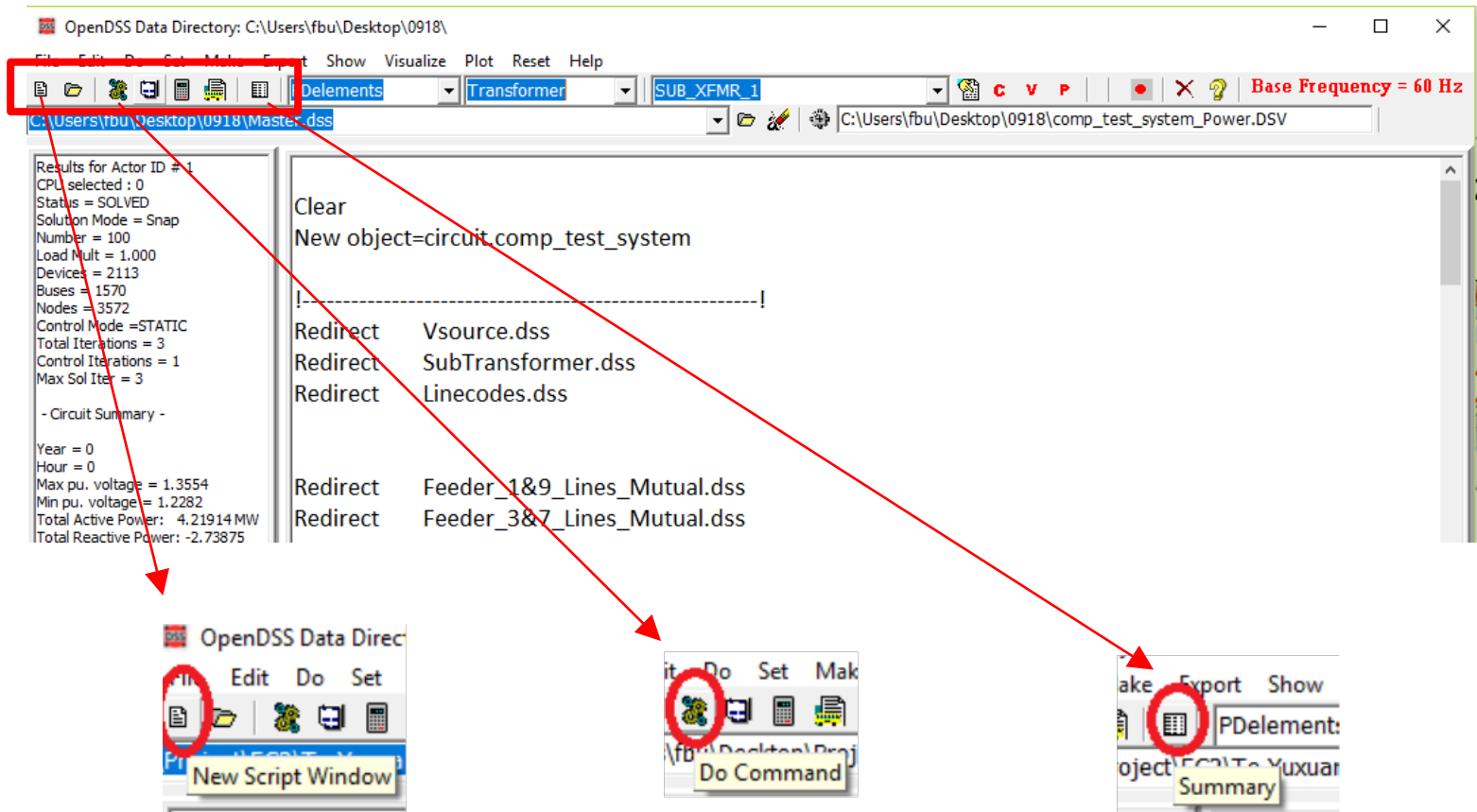
User Interface

- [1] Menus
-- Plot



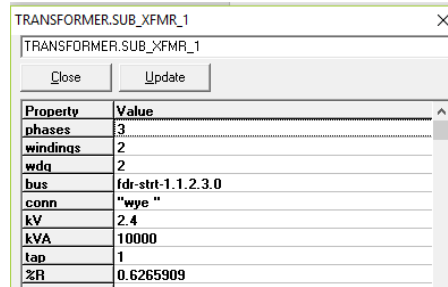
User Interface

- [2] Toolbar: provides direct access to many commonly used OpenDSS commands such as “Solve,” “Summary,” and “Do Command”.



User Interface

- [3] Element tools: allow the user to select which circuit element (by type) to edit or display visualizations for.



Property	Value
phases	3
windings	2
wdq	2
bus	fdr-str1-1.1.2.3.0
conn	"wye "
kV	2.4
kVA	10000
tap	1
%R	0.6265909

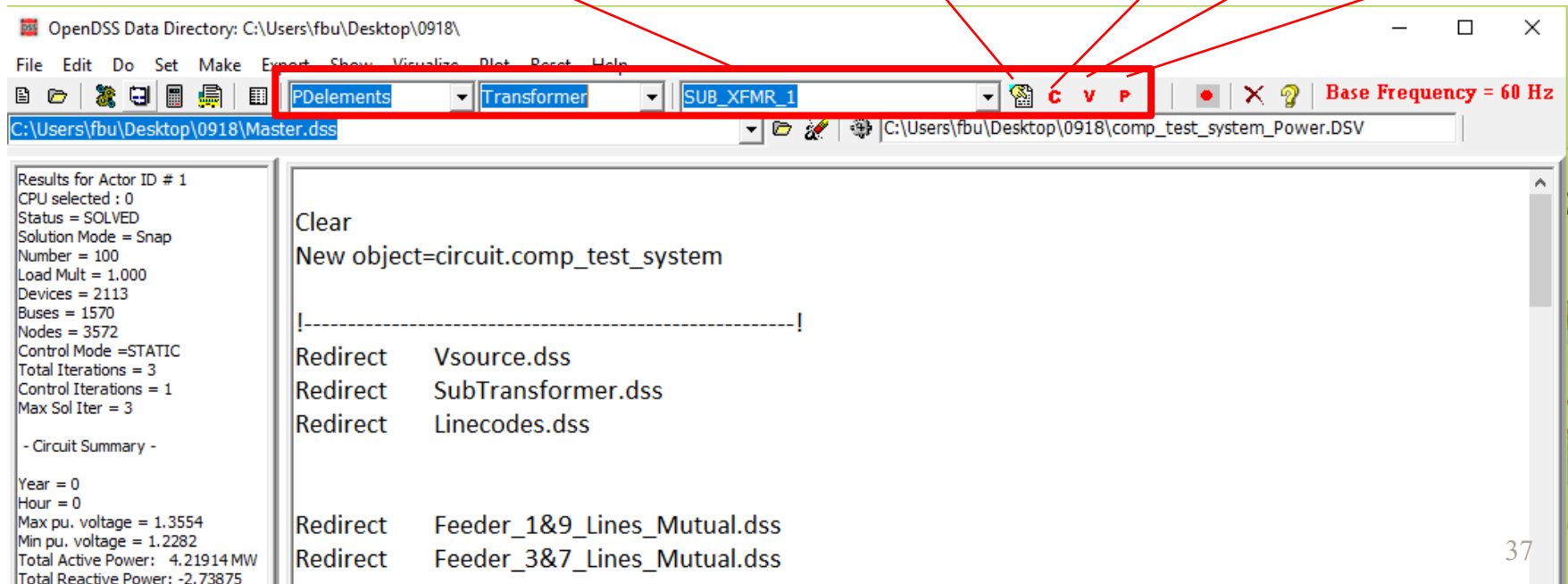
Edit parameter of the selected element.

Name of selected element

Current

Voltage

Power



OpenDSS Data Directory: C:\Users\fbu\Desktop\0918\
File Edit Do Set Make Export Show Visualize Plot Reset Help
PDelements Transformer SUB_XFMR_1 C V P Base Frequency = 60 Hz
C:\Users\fbu\Desktop\0918\Master.dss C:\Users\fbu\Desktop\0918\comp_test_system_Power.DSV

Results for Actor ID # 1
CPU selected : 0
Status = SOLVED
Solution Mode = Snap
Number = 100
Load Mult = 1.000
Devices = 2113
Buses = 1570
Nodes = 3572
Control Mode =STATIC
Total Iterations = 3
Control Iterations = 1
Max Sol Iter = 3
- Circuit Summary -
Year = 0
Hour = 0
Max pu. voltage = 1.3554
Min pu. voltage = 1.2282
Total Active Power: 4.21914 MW
Total Reactive Power: -2.73875

Clear
New object=circuit.comp_test_system
!-----!
Redirect Vsource.dss
Redirect SubTransformer.dss
Redirect Linecodes.dss
Redirect Feeder_1&9_Lines_Mutual.dss
Redirect Feeder_3&7_Lines_Mutual.dss

User Interface

- [4] Script tools: allow us to select which of the current opened scripts to run.

The screenshot displays the OpenDSS software interface. The title bar reads "OpenDSS Data Directory: C:\Users\fbu\Desktop\0918\". The menu bar includes "File", "Edit", "Do", "Set", "Make", "Export", "Show", "Visualize", "Plot", "Reset", and "Help". The toolbar contains icons for file operations and simulation. The main window shows a script tool window with a red border around the file path "C:\Users\fbu\Desktop\0918\Linecodes.dss". The script content includes:

```
Clear
New object=circuit.comp_test_system

!-----!
Redirect Vsource.dss
Redirect SubTransformer.dss
Redirect Linecodes.dss

Redirect Feeder_1&9_Lines_Mutual.dss
Redirect Feeder_3&7_Lines_Mutual.dss

Redirect Feeder_1_Lines.dss
Redirect Feeder_1_Loads.dss
Redirect Feeder_1_Transformers.dss
Redirect Feeder_1_Capacitor_Banks.dss
```

The interface also shows a "Results for Actor ID # 1" panel on the left with various simulation parameters and a "Messages" panel at the bottom. The status bar at the bottom indicates "Memory: 23988K", "Circuit Status: SOLVED", and "Total Iterations = 3, Control Iterations = 1, Max Solution Iterations = 3".

User Interface

- [5] Results bar: provides a condensed version of the results.

The screenshot displays the OpenDSS software interface. The title bar reads "OpenDSS Data Directory: C:\Users\fbu\Desktop\0918\". The menu bar includes "File", "Edit", "Do", "Set", "Make", "Export", "Show", "Visualize", "Plot", "Reset", and "Help". The toolbar contains various icons for file operations and simulation. The main window title is "C:\Users\fbu\Desktop\0918\Linecodes.dss". A red box highlights the file path "C:\Users\fbu\Desktop\0918\comp_test_system_EXP_Result.CSV" in the address bar. The left sidebar shows "Results for Actor ID # 1" with a "Summary" tab selected. The main window displays a "Clear" message and a list of "Redirect" commands for various .dss files. The status bar at the bottom shows "Memory: 37696K", "Circuit Status: SOLVED", and "Total Iterations = 3, Control Iterations = 1, Max Solution Iterations = 3".

OpenDSS Data Directory: C:\Users\fbu\Desktop\0918\

File Edit Do Set Make Export Show Visualize Plot Reset Help

C:\Users\fbu\Desktop\0918\Linecodes.dss

C:\Users\fbu\Desktop\0918\comp_test_system_EXP_Result.CSV

Base Frequency = 60 Hz

Results for Actor ID # 1
CPU selected : 0
Status = SOLVED
Solution Mode = Snap
Number = 100
Load Mult = 1.000
Devices = 2113
Buses = 1570
Nodes = 3572
Control Mode =STATIC
Total Iterations = 3
Control Iterations = 1
Max Sol Iter = 3

- Circuit Summary -
Year = 0
Hour = 0
Max pu. voltage = 1.3554
Min pu. voltage = 1.2282
Total Active Power: 4.21914 MW
Total Reactive Power: -2.73875 Mvar
Total Active Losses: 0.276539 MW, (6.554 %)
Total Reactive Losses: 0.27678 Mvar
Frequency = 60 Hz
Mode = Snap
Control Mode = STATIC
Load Model = PowerFlow

Clear
New object=circuit.comp_test_system

!-----!
Redirect Vsource.dss
Redirect SubTransformer.dss
Redirect Linecodes.dss

Redirect Feeder_1&9_Lines_Mutual.dss
Redirect Feeder_3&7_Lines_Mutual.dss

Redirect Feeder_1_Lines.dss
Redirect Feeder_1_Loads.dss
Redirect Feeder_1_Transformers.dss
Redirect Feeder_1_Capacitor_Banks.dss

Main Master.dss Linecodes.dss

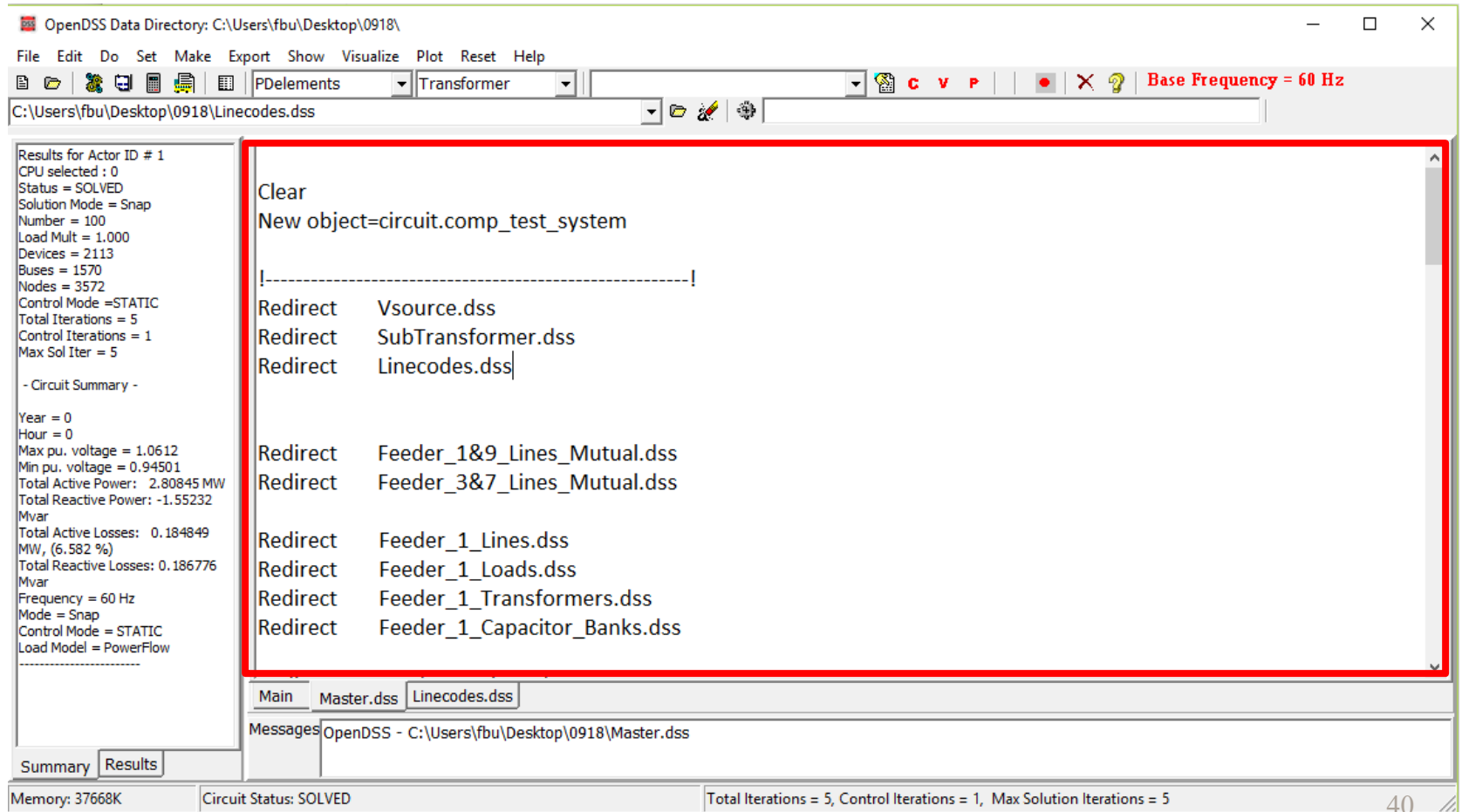
Messages OpenDSS - C:\Users\fbu\Desktop\0918\Master.dss

Summary Results

Memory: 37696K Circuit Status: SOLVED Total Iterations = 3, Control Iterations = 1, Max Solution Iterations = 3

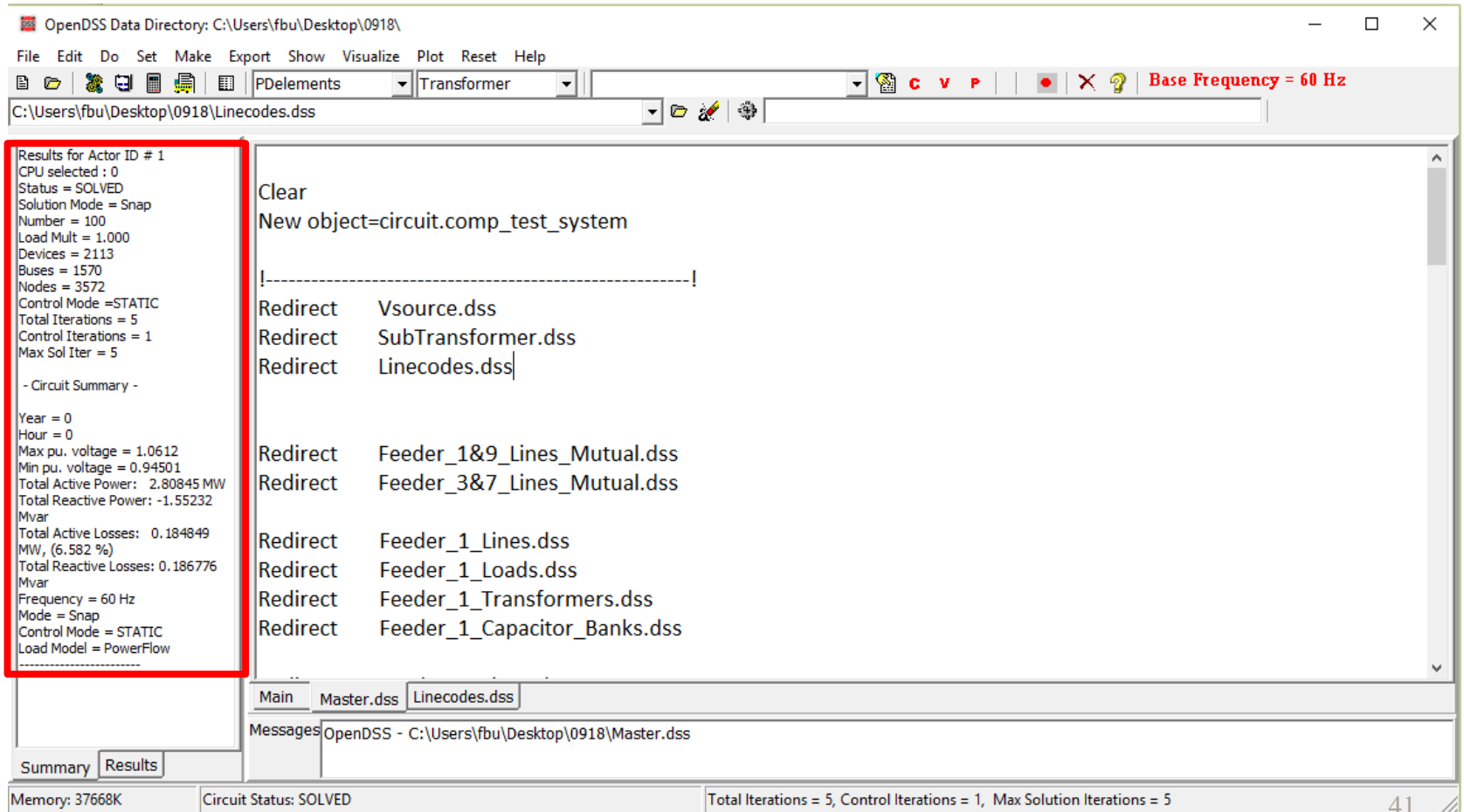
User Interface

- [6] Script Windows: allow us to directly edit various *.dss files.



User Interface

- [7] Summary Window: provides us the brief summary of solutions.



User Interface

- [8] Help button: brings up the OpenDSS Command and Element Property Reference which gives a tree-view guide to the various script commands in OpenDSS.

The screenshot displays the OpenDSS software interface. The title bar indicates the directory path: "OpenDSS Data Directory: C:\Users\fbu\Desktop\0918\". The menu bar includes "File", "Edit", "Do", "Set", "Make", "Export", "Show", "Visualize", "Plot", "Reset", and "Help". The toolbar contains various icons, including a help icon (a question mark) which is highlighted with a red box. The status bar on the right shows "Base Frequency = 60 Hz".

The main window shows a script editor with the following content:

```
Clear
New object=circuit.comp_test_system

|-----|
Redirect  Vsource.dss
Redirect  SubTransformer.dss
Redirect  Linecodes.dss

Redirect  Feeder_1&9_Lines_Mutual.dss
Redirect  Feeder_3&7_Lines_Mutual.dss

Redirect  Feeder_1_Lines.dss
Redirect  Feeder_1_Loads.dss
Redirect  Feeder_1_Transformers.dss
Redirect  Feeder_1_Capacitor_Banks.dss
```

The left sidebar displays "Results for Actor ID # 1" with the following summary:

```
Results for Actor ID # 1
CPU selected : 0
Status = SOLVED
Solution Mode = Snap
Number = 100
Load Mult = 1.000
Devices = 2113
Buses = 1570
Nodes = 3572
Control Mode =STATIC
Total Iterations = 5
Control Iterations = 1
Max Sol Iter = 5

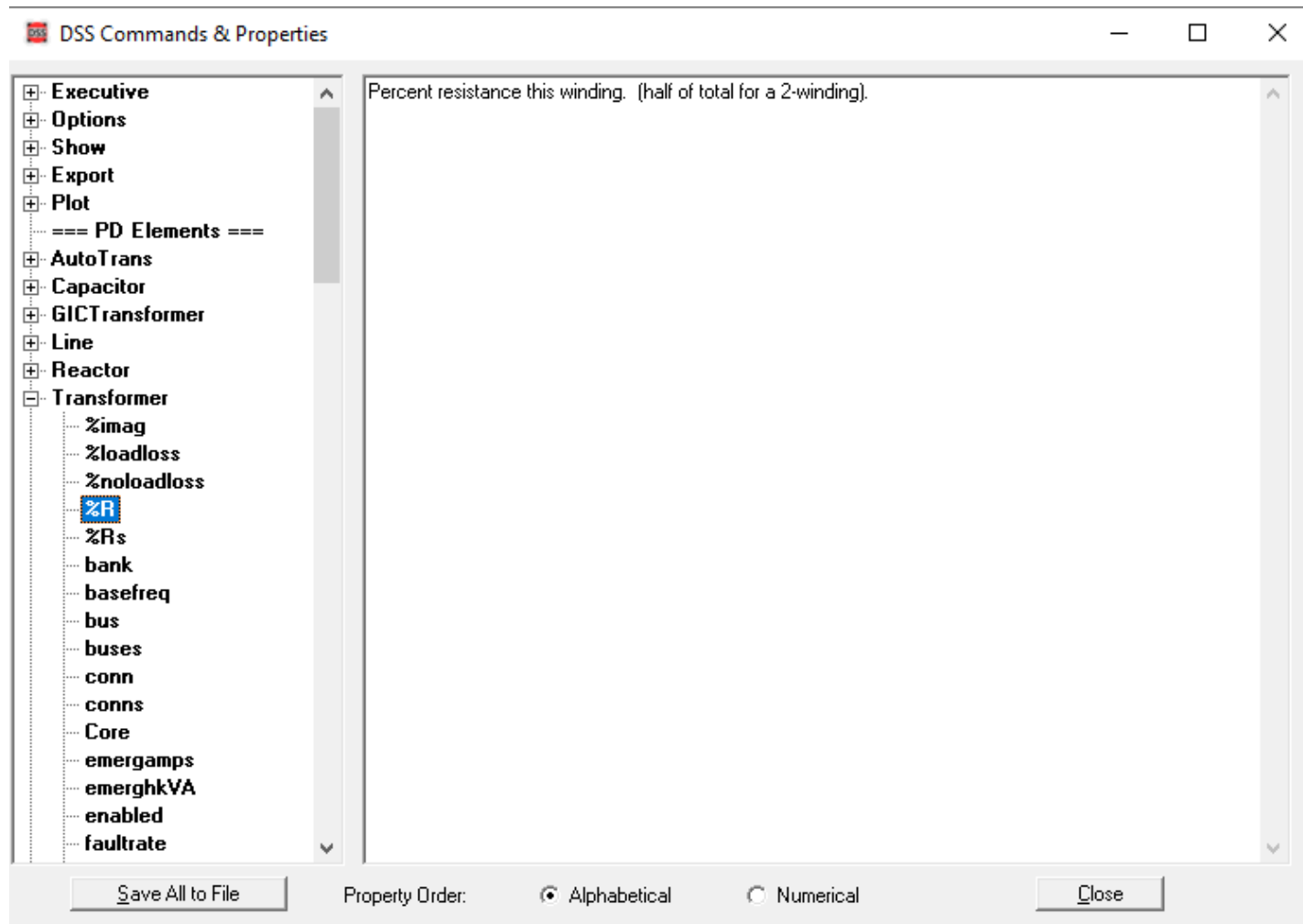
- Circuit Summary -

Year = 0
Hour = 0
Max pu. voltage = 1.0612
Min pu. voltage = 0.94501
Total Active Power: 2.80845 MW
Total Reactive Power: -1.55232
Mvar
Total Active Losses: 0.184849
MW, (6.582 %)
Total Reactive Losses: 0.186776
Mvar
Frequency = 60 Hz
Mode = Snap
Control Mode = STATIC
Load Model = PowerFlow
```

The bottom status bar shows "Memory: 37668K", "Circuit Status: SOLVED", and "Total Iterations = 5, Control Iterations = 1, Max Solution Iterations = 5". The page number "42" is visible in the bottom right corner.

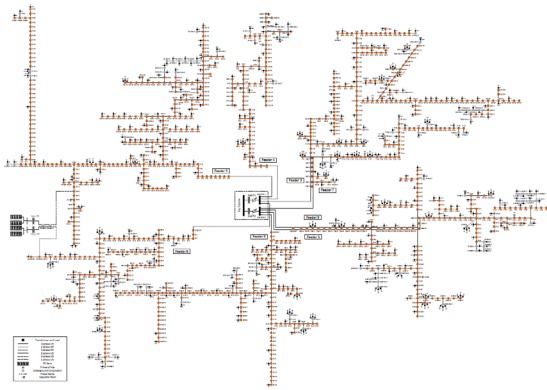
User Interface

- [8] Help button

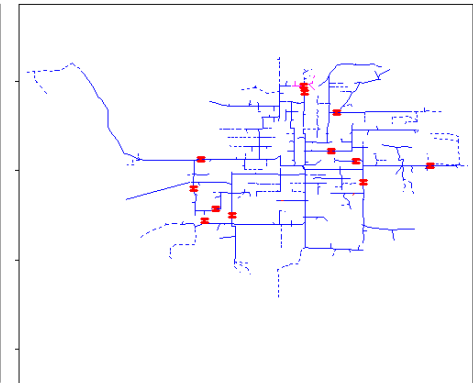
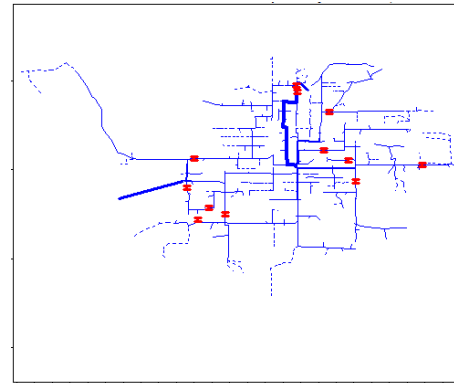


Workflow

4 steps to develop a project:



Define the circuit



Perform analysis

Name

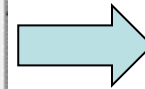
- Buscoords.dss
- Feeder_1_Capacitor_Banks.dss
- Feeder_1_Lines.dss
- Feeder_1_Loads.dss
- Feeder_1_Transformers.dss
- Linecodes.dss
- Master.dss
- SubTransformer.dss
- Vsource.dss

Set up the circuit options



Set	Make	Export	Show	Vis
Base Frequency=60				
Base Frequency=50				
Mode=				
Number=				
Bus=				
Load Model=				
Load Multiplier=				

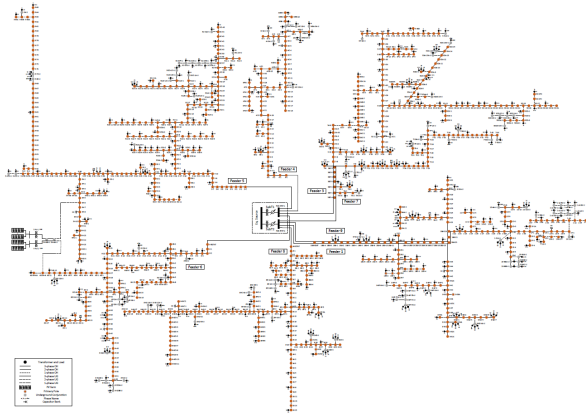
Solve



Set VoltageBases =
CalcVoltageBases
solve
BusCoords Bus
Show

- Do Selected **Ctrl+D**
- Save This Window
- Close Window
- Change to this Directory
- Open Selected File
- Edit Selected File
- Change Font...










Define A Circuit – Overall Concept



Define the circuit



Name

	Buscoords.dss
	Feeder_1_Capacitor_Banks.dss
	Feeder_1_Lines.dss
	Feeder_1_Loads.dss
	Feeder_1_Transformers.dss
	Linecodes.dss
	Master.dss
	SubTransformer.dss
	Vsource.dss

- Overall circuit model;
- Bus and terminal models;
- Power delivery elements;
- Power conversion elements;
- Putting it all together.

Define A Circuit – Overall Concept

Overall Circuit Model

- The OpenDSS consists of a model of the *electrical power distribution system* in the rms steady state, overlaid with a *communication network* that interconnects controls on power delivery elements and on power conversion elements.
- Compared with the communication network, the electric power distribution system is more commonly used for simulating real systems.

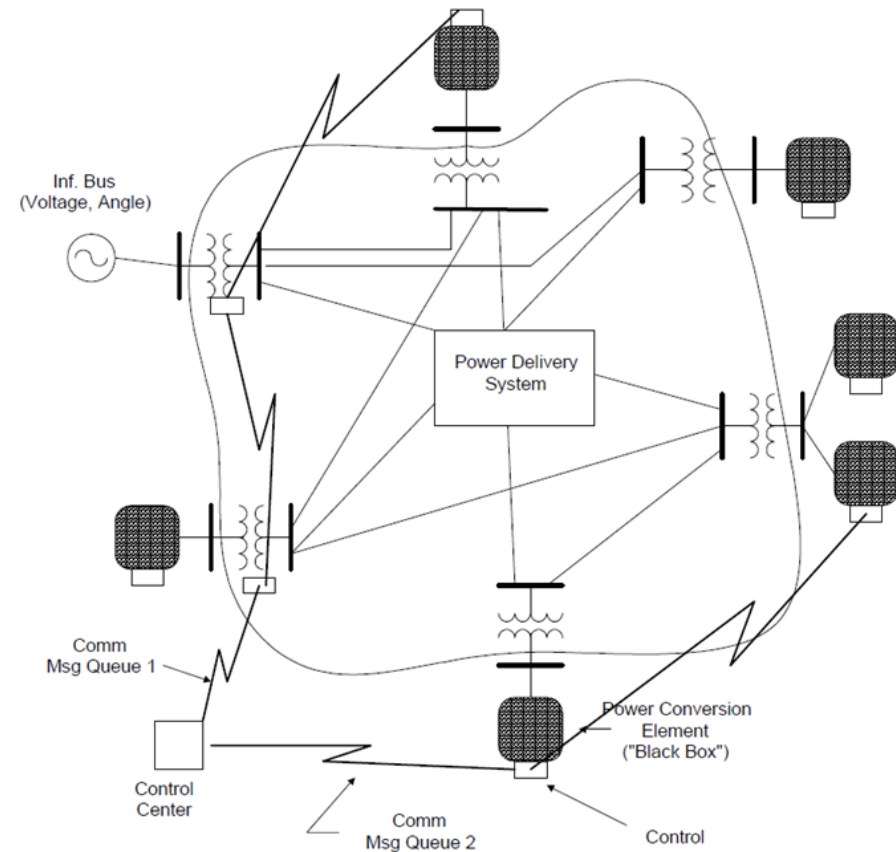
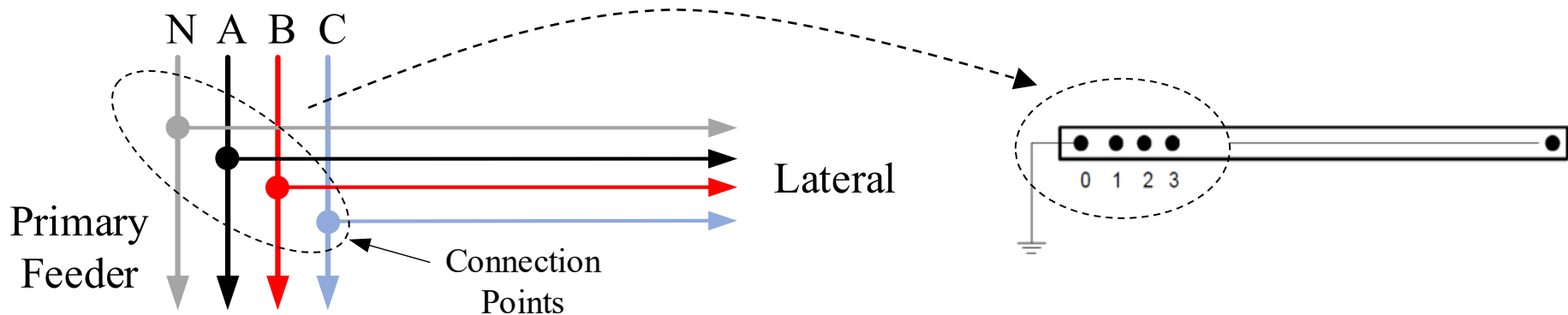


Figure 11: Electrical Circuit with Communications Network

Define A Circuit – Overall Concept

Bus Definition

- A **bus** is a circuit element having multiple ($[1..N]$) **nodes**. Buses are the connection point for all other circuit elements. In other words, bus is a connecting place with 1 or more nodes for connecting individual phases and other conductors.



- Node 0 of each bus is implicitly connected to the **voltage reference** (i.e., the node's voltage is always 0 and is never explicitly included in the Y matrix).
- In many other power system analysis programs, “bus” and “node” are nearly synonymous, but they are distinctively **different** in OpenDSS. Bus is the container of Node objects. That is to say, a Bus has Nodes.

47

Define A Circuit – Overall Concept

Bus Definition

- Bus naming:
Bus name can include *letters* and *numbers*, and can contain some *symbols*. It is better to avoid containing blanks, tabs, or other “white space”.

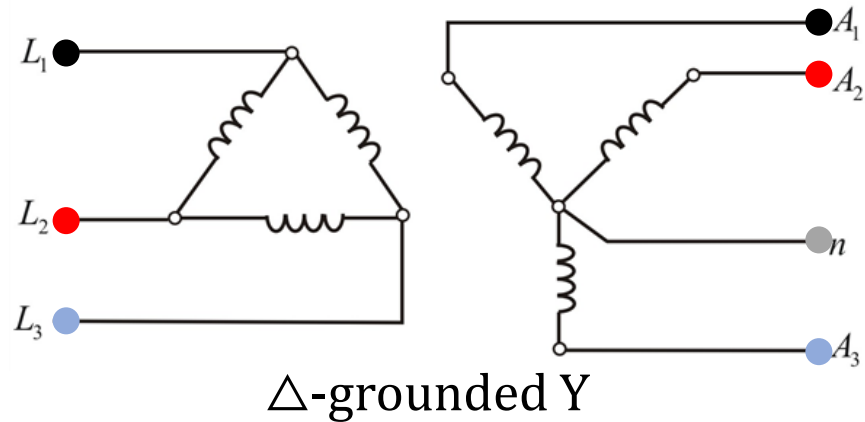
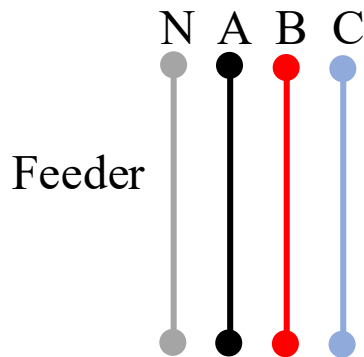
Example1:

```
//----- Primary Main Feeder -----//
New Line.FDR1_MF_1 Phases=3 Bus1=1&9-21X.1.2.3 Bus2=1-22.1.2.3 LineCode=1/0ACSR_#2ACSR7/1_3ph_OH Length=0.0553 units=kft
New Line.FDR1_MF_2 Phases=3 Bus1=1-22.1.2.3 Bus2=1-23.1.2.3 LineCode=1/0ACSR_#2ACSR7/1_3ph_OH Length=0.1382 units=kft
New Line.FDR1_MF_3 Phases=3 Bus1=1-23.1.2.3 Bus2=1-24.1.2.3 LineCode=1/0ACSR_#2ACSR7/1_3ph_OH Length=0.0829 units=kft
New Line.FDR1_MF_4 Phases=3 Bus1=1-24.1.2.3 Bus2=1-25.1.2.3 LineCode=1/0ACSR_#2ACSR7/1_3ph_OH Length=0.1105 units=kft
New Line.FDR1_MF_5 Phases=3 Bus1=1-25.1.2.3 Bus2=1-26.1.2.3 LineCode=1/0ACSR_#2ACSR7/1_3ph_OH Length=0.1105 units=kft
New Line.FDR1_MF_6 Phases=3 Bus1=1-26.1.2.3 Bus2=1-27.1.2.3 LineCode=1/0ACSR_#2ACSR7/1_3ph_OH Length=0.1105 units=kft
New Line.FDR1_MF_7 Phases=3 Bus1=1-27.1.2.3 Bus2=1-28.1.2.3 LineCode=1/0ACSR_#2ACSR7/1_3ph_OH Length=0.1105 units=kft
New Line.FDR1_MF_8 Phases=3 Bus1=1-28.1.2.3 Bus2=1-29.1.2.3 LineCode=1/0ACSR_#2ACSR7/1_3ph_OH Length=0.0829 units=kft
New Line.FDR1_MF_9 Phases=3 Bus1=1-29.1.2.3 Bus2=1-30.1.2.3 LineCode=1/0ACSR_#2ACSR7/1_3ph_OH Length=0.1658 units=kft
New Line.FDR1_MF_10 Phases=3 Bus1=1-30.1.2.3 Bus2=1-30X.1.2.3 LineCode=1/0ACSR_#2ACSR7/1_3ph_OH Length=0.1382 units=kft
New Line.FDR1_MF_11 Phases=3 Bus1=1-30X.1.2.3 Bus2=1-31.1.2.3 LineCode=1/0ACSR_#2ACSR7/1_3ph_OH Length=0.1105 units=kft
New Line.FDR1_MF_12 Phases=3 Bus1=1-31.1.2.3 Bus2=1-32.1.2.3 LineCode=#6Cu_#6Cu_3ph_OH Length=0.0276 units=kft
New Line.FDR1_MF_13 Phases=3 Bus1=1-32.1.2.3 Bus2=1-33.1.2.3 LineCode=#6Cu_#6Cu_3ph_OH Length=0.1382 units=kft
New Line.FDR1_MF_14 Phases=3 Bus1=1-31.1.2.3 Bus2=1-31A-U.1.2.3 LineCode=1/0ACSR_#2ACSR7/1_3ph_UG Length=0.4582 units=kft
```


Define A Circuit – Overall Concept

Terminal Definition

- Each electrical element in the power system has *one or more* terminals. Each terminal has *one or more* conductors.



- Each conductor conceptually contains a disconnect switch and a TCC (fuse) curve. However, in general, we define *external* fuses or switches, instead of using the built-in fuses or switches. The conductors are numbered [1, 2, 3,...].

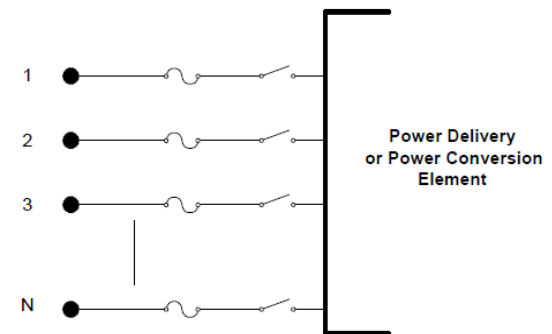


Figure 13: Terminal Definition

Define A Circuit – Overall Concept

Terminal Definition

- If the terminal is connected to an n-phase device, the first n conductors are orderly assumed to correspond to the phases. The remaining conductors are frequently neutrals or other non-power conductors.

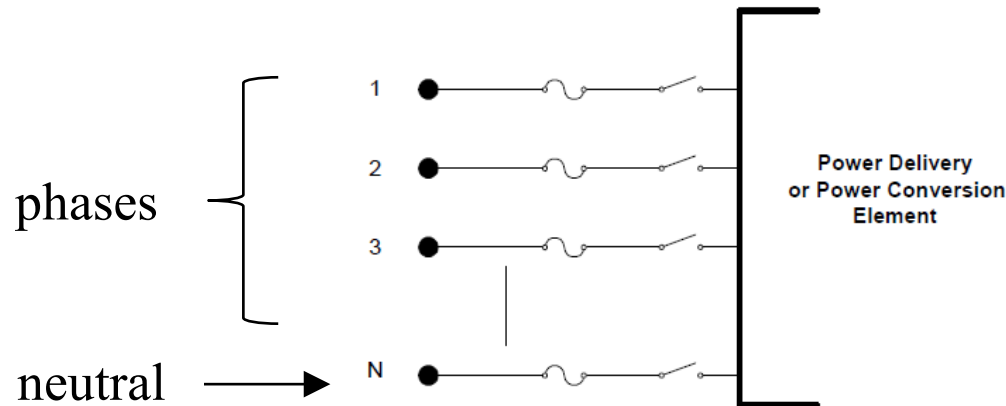


Figure 13: Terminal Definition

Note that Phase 1, 2, 3 do not necessarily correspond to Phase A, B, C, respectively.

Define A Circuit – Overall Concept

Power delivery elements

- Power delivery elements usually consist of *two or more* multiphase terminals.
- Their basic function is to *transport* energy from one point to another.
- In the power systems, the most common power delivery elements are *lines* and *transformers*.

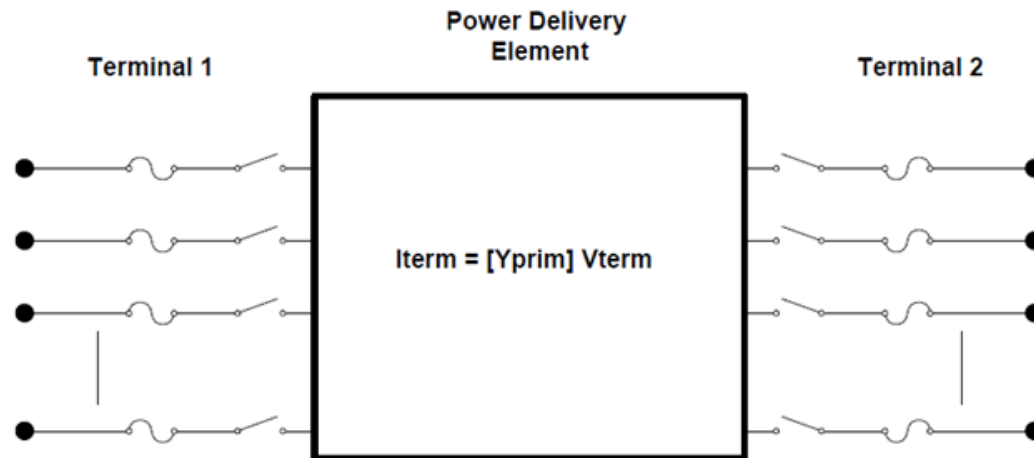


Figure 14: Power Delivery Element Definition

Define A Circuit – Overall Concept

Power conversion elements

- Power conversion elements *convert* power from electrical form to some other forms, or vice-versa.
- Most of the power conversion elements have only *one* connection to the power system and, therefore, only *one* multiphase terminal.
- In the power systems, the most common power delivery elements are load and generator.

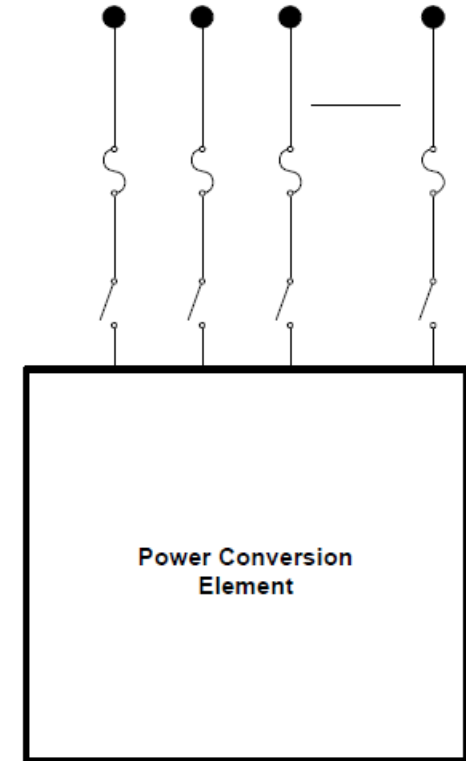


Figure 15: Power Conversion Element Definition

Define A Circuit – DSS Command

DSS command contains three sections:

- DSS command language syntax;
- DSS command reference;
- Option reference.

```
//----- Primary Main Feeder -----//  
New Line.FDR1_MF_1 Phases=3 Bus1=1&9-21X.1.2.3 Bus2=1-22.1.2.3 LineCode=1/OACSR_#2ACSR7/1_3ph_OH Length=0.0553 units=kft  
New Line.FDR1_MF_2 Phases=3 Bus1=1-22.1.2.3 Bus2=1-23.1.2.3 LineCode=1/OACSR_#2ACSR7/1_3ph_OH Length=0.1382 units=kft  
New Line.FDR1_MF_3 Phases=3 Bus1=1-23.1.2.3 Bus2=1-24.1.2.3 LineCode=1/OACSR_#2ACSR7/1_3ph_OH Length=0.0829 units=kft  
New Line.FDR1_MF_4 Phases=3 Bus1=1-24.1.2.3 Bus2=1-25.1.2.3 LineCode=1/OACSR_#2ACSR7/1_3ph_OH Length=0.1105 units=kft  
New Line.FDR1_MF_5 Phases=3 Bus1=1-25.1.2.3 Bus2=1-26.1.2.3 LineCode=1/OACSR_#2ACSR7/1_3ph_OH Length=0.1105 units=kft  
New Line.FDR1_MF_6 Phases=3 Bus1=1-26.1.2.3 Bus2=1-27.1.2.3 LineCode=1/OACSR_#2ACSR7/1_3ph_OH Length=0.1105 units=kft  
New Line.FDR1_MF_7 Phases=3 Bus1=1-27.1.2.3 Bus2=1-28.1.2.3 LineCode=1/OACSR_#2ACSR7/1_3ph_OH Length=0.1105 units=kft  
New Line.FDR1_MF_8 Phases=3 Bus1=1-28.1.2.3 Bus2=1-29.1.2.3 LineCode=1/OACSR_#2ACSR7/1_3ph_OH Length=0.0829 units=kft  
New Line.FDR1_MF_9 Phases=3 Bus1=1-29.1.2.3 Bus2=1-30.1.2.3 LineCode=1/OACSR_#2ACSR7/1_3ph_OH Length=0.1658 units=kft  
New Line.FDR1_MF_10 Phases=3 Bus1=1-30.1.2.3 Bus2=1-30X.1.2.3 LineCode=1/OACSR_#2ACSR7/1_3ph_OH Length=0.1382 units=kft  
New Line.FDR1_MF_11 Phases=3 Bus1=1-30X.1.2.3 Bus2=1-31.1.2.3 LineCode=1/OACSR_#2ACSR7/1_3ph_OH Length=0.1105 units=kft  
New Line.FDR1_MF_12 Phases=3 Bus1=1-31.1.2.3 Bus2=1-32.1.2.3 LineCode=#6Cu_#6Cu_3ph_OH Length=0.0276 units=kft  
New Line.FDR1_MF_13 Phases=3 Bus1=1-32.1.2.3 Bus2=1-33.1.2.3 LineCode=#6Cu_#6Cu_3ph_OH Length=0.1382 units=kft  
New Line.FDR1_MF_14 Phases=3 Bus1=1-31.1.2.3 Bus2=1-31A-U.1.2.3 LineCode=1/OACSR_#2ACSR7/1_3ph_UG Length=0.4582 units=kft
```

DSS commands specify the rules of writing .dss files

Define A Circuit – DSS Command

Command syntax

- The command language is of the following form:

Command parm1, parm2 parm3 parm 4

- Parameters (parm1, etc) may be separated by commas (,) or white spaces (blank, tab).
- Parameters may be *positional* or *named* (tagged). If named, an "=" sign is expected.

Example:

```
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.

Define A Circuit – DSS Command

Command reference

Currently, there are **98** commands in OpenDSS. Newer builds of the DSS may have additional properties and commands. You can 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
```

Define A Circuit – DSS Command

Command reference

/ ... */ Block Comments*

This block comment symbol is used to comment out whole sections (whole lines of script). 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
```


Define A Circuit – DSS Command

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 control actions to proper control element "DoPendingAction" handlers.

InitSnap

For step control of solution process: Initialize 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.

...

57

Define A Circuit – DSS Command

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.

Define A Circuit – DSS Command

Command reference

AddBusMarker

Add a marker to a bus in a circuit plot. Markers must be added before issuing the Plot command. The effect is persistent until circuit is cleared or the 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
...
```

For the color, you can use any of the standard color names or RGB numbers.

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.

Define A Circuit – DSS Command

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.

BusCoords

It defines x,y coordinates for buses. It is executed after executing Solve or MakeBusList command, so that bus lists are defined. You can read coordinates from a CSV file with records of the following form:

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

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

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

Define A Circuit – DSS Command

Command reference

CalcVoltageBases

It estimates the voltage base for each bus based on the array of voltage bases defined with a "**SetVoltagebases=...**" command. To calculate voltage bases, first, OpenDSS 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*.

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.

Define A Circuit – DSS Command

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

Clears 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.

Define A Circuit – DSS Command

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 example, for a contingency case.

Edit [Object] [Edit String]

Edits the specified object. When editing, the *object Class* and *Name fields* are required and you 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.

Define A Circuit – DSS Command

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 exporting 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:

...

Define A Circuit – DSS Command

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 parentheses.

...

Define A Circuit – DSS Command

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.

...

Define A Circuit – DSS Command

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.

...

Define A Circuit – DSS Command

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.

...

Define A Circuit – DSS Command

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.

Define A Circuit – DSS Command

Command reference

Get [Opt1] [opt2] etc.

Basically, it is the opposite of the SET command. It 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
```

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

Define A Circuit – DSS Command

Command reference

LatLongCoords

It defines x,y coordinates for buses using Latitude and Longitude values (decimal numbers). Similar to `BusCoords` command, it is executed after executing `Solve` command or `MakeBusList` command, so that bus lists are defined. You can read coordinates from a CSV file with records of the following 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 in the Result string, in kW, kvar.

Define A Circuit – DSS Command

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]

It adds an element which is described on the remainder of the line to the active circuit. The first parameter (Object=...) is required for the New command.

- 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 needs only to include definitions for property values that are different than the defaults.

...

Define A Circuit – DSS Command

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.

...

Define A Circuit – DSS Command

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 following 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`

...

Define A Circuit – DSS Command

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. 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. Therefore, you should issue the **Like=nnnn** property first. Example:

```
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.

Define A Circuit – DSS Command

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
```

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.

Define A Circuit – DSS Command

Command reference

Open [Object] [Term] [Cond]

It 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. 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
```

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

Define A Circuit – DSS Command

Command reference

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.

Plot (options ...)

Plot is a rather complex command that displays a variety of results in a variety of manners on graphs.

- You can 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):

...

Define A Circuit – DSS Command

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 should be defined. By default the thickness of the circuit lines is drawn proportional to power. The **Monitor** plot plots one or more channels from a Monitor element. 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 **Zones** plot draws the energymeter zones. **Autoadd** plot shows autoadded elements on the circuit plot. **General** plot 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. **Loadshape**, **Tshape** and **Priceshape** plots show the time-varying load, temperature and price curves, respectively. **Profile** plot expects voltage curves.

...

Define A Circuit – DSS Command

Command reference

Plot (options ...)

...

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

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

Dots = {Y | N} It specifies turning 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

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.

...

Define A Circuit – DSS Command

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). Specifies which monitor channels to 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) Specifies whether to show substations or not.

...

Define A Circuit – DSS Command

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

...

Define A Circuit – DSS Command

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.

- For Loadshapes, they may be plotted from the control panel of the user interface.
- For Power and Losses, they are specified in kW. C1 is used for default color. C2, C3 are used for gradients, tricolor plots. Scale is determined automatically of Max = 0 or not specified. Some examples:

...

Define A Circuit – DSS Command

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 lphlinestyle=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
```

Define A Circuit – DSS Command

Command reference

Powers

Returns the powers (complex) going into each conductors of ALL terminals of the active circuit element. The powers are 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 going into all terminals of the active circuit element. The sequence currents are returned as comma-separated magnitude only values. Order of returned values: 0, 1, 2 (for each terminal).

Define A Circuit – DSS Command

Command reference

SeqPowers

Returns the sequence powers going into all terminals of the active circuit element. The sequence powers are 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. The sequence voltages are 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. It works like the Edit command except that you don't specify object type and name.

Define A Circuit – DSS Command

Command reference

SetkVBase [bus=...] [kvll=..]

It explicitly sets the base voltage for a *specified* bus.

- For SetkVBase command, bus must be previously defined. And also, it will override the definitions determined by “Set Voltagebases” .
- Since we already have the “Set Voltagebases” command, why do we need the SetkVBase command? The reason is as follows: When there are only a few voltage bases in the circuit, and they are very distinct, the “Set Voltagebases” command can work well nearly every time, since the voltage base for each bus is set to the nearest voltage base specified in the voltage base array. However, problems arise if there are two voltage bases that are close together, such as 12.47 kV and 13.2 kV. Therefore, we can use the SetkVBase commands to remove this ambiguity.
- 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 examples are equivalent:

```
setkvbase Bus=B9654 kVLL=13.2
```

```
setkvbase B9654 13.2
```

```
setkvbase B9654 kvln=7.62
```

Define A Circuit – DSS Command

Command reference

Show <Quantity>

The Show command generally writes a text file report of the specified quantity for the most recent solution, and opens a viewer to display the file.

- Note that the Show command is updated frequently, often making the tutorial or document out of date. You can see the Help on the latest version.
- It 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 going 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.

...

Define A Circuit – DSS Command

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] It shows the power flow in various units. Default(*) is sequence power in kVA.

...

Define A Circuit – DSS Command

Command reference

Show <Quantity>

...

Voltages [LL |LN*] [Seq* | Nodes | Elements]. It 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

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

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

Define A Circuit – DSS Command

Command reference

Solve [see set command options ...]

It executes the solution mode specified by the Set Mode command. It may execute a *single* solution or *hundreds* of solutions.

Summary

Returns a power flow summary of the most recent solution.

Totals

Total of all EnergyMeter objects in the circuit. Reports register.

Variable

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

If the active element is a Pcelement, *Variable* returns the value of the specified state variable of that element. It applies only to PCElements that contain state variables. 91

Define A Circuit – DSS Command

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).
It shows the currents, voltages, or powers for selected element on a *drawing* in phasor quantities.

For example:

```
Visualize what=currents element=Line.632633
```

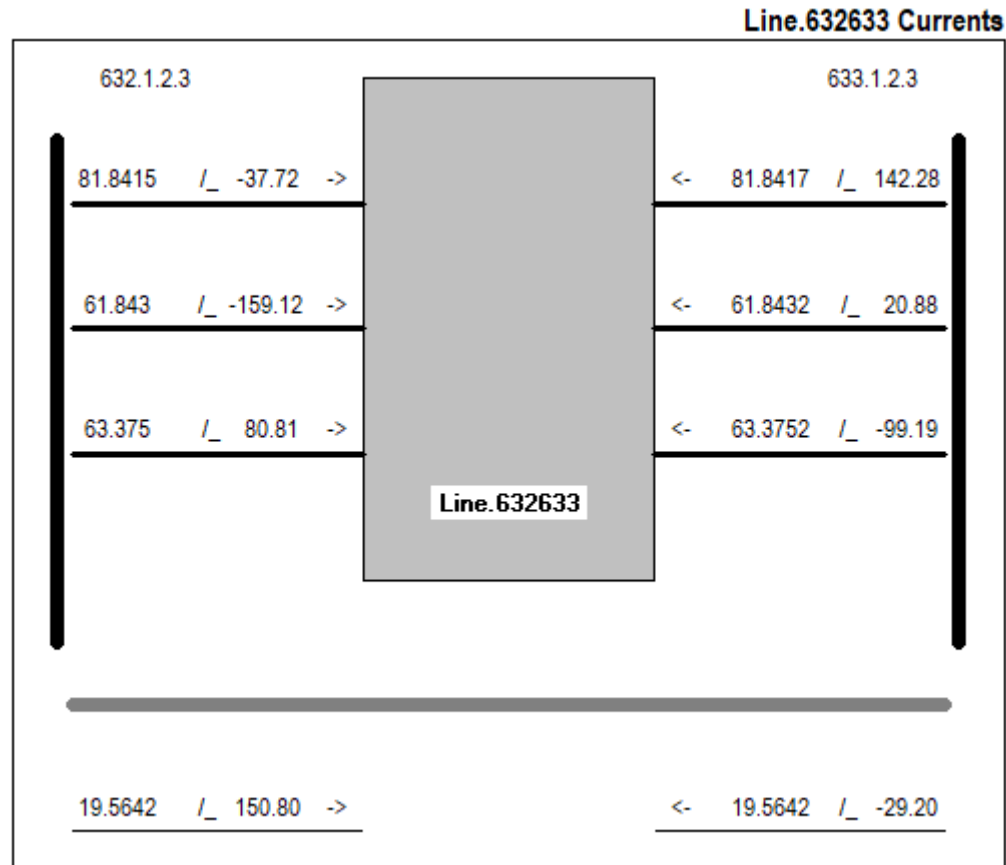
...

Define A Circuit – DSS Command

Command reference

Visualize

...



Define A Circuit – DSS Command

Command reference

Voltages

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

Ysc

Returns full short-circuit admittance, Y_{sc} , matrix for the ACTIVE BUS in comma-separated complex number form $G + jB$.

Zsc

Returns full Short-circuit impedance, Z_{sc} , matrix for the ACTIVE BUS in comma-separated complex number form.

Define A Circuit – DSS Command

Command reference

Zsc10

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

Define A Circuit – DSS Command

Option reference

A circuit can be executed in the following ways: snapshot power flow, time-series power flow, harmonics and faultstudy, etc. Therefore, the different modes and associated parameters should be defined before solving the circuit. DSS options are designed to define these solution modes and parameters.

- DSS options can be set using the Set command. For example:

```
Set mode=snapshot
```

```
Solve
```

```
Set mode=harmonics
```

```
Solve
```

- Alternatively, the examples above can be accomplished with just 2 lines, using the solve command:

```
Solve mode=snapshot
```

```
Solve mode=harmonics
```

The Solve command first executes the Set and then executes a solution. This allows for more concise syntax for some cases.

Define A Circuit – DSS Command

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.

97

Define A Circuit – DSS Command

Option reference

Algorithm =

{Normal | Newton} It specifies the solution algorithm. 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.

Define A Circuit – DSS Command

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)
```

Define A Circuit – DSS Command

Option reference

Basefrequency =

Default = 60. Set the fundamental frequency for harmonic solution and the default base frequency for all impedance quantities. Side effect: This setting 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: It sets the prefix for output files.

100

Define A Circuit – DSS Command

Option reference

CapMarkerCode =

It sets a numeric marker code for capacitors. Default is 37. See MarkerCode option.

CapMarkerSize =

It sets the size of Capacitor 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.

Define A Circuit – DSS Command

Option reference

CktModel =

{Multiphase | Positive} Default = Multiphase. Designates whether circuit model is to be interpreted as a normal multi-phase model or a positive-sequence only model. If Positive sequence, all power quantities are multiplied 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.

...

Define A Circuit – DSS Command

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.

...

Define A Circuit – DSS Command

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.

Define A Circuit – DSS Command

Option reference

DefaultBaseFrequency=

It sets Default Base Frequency, in Hz. The default value when first installed is 60 Hz. It needs 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.

Define A Circuit – DSS Command

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.

Define A Circuit – DSS Command

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.

Element =

Sets the active DSS element by name. You can use the complete object specification (class.name) or just the name. If full name is specified, class becomes the active class, also. See also the Select command.

Emergymaxpu =

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

Define A Circuit – DSS Command

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.

Define A Circuit – DSS Command

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)

Define A Circuit – DSS Command

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.

Define A Circuit – DSS Command

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".)

Define A Circuit – DSS Command

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.

Define A Circuit – DSS Command

Option reference

Markercode =

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

0	.	10	◻	20	^	30	▼	40	◄
1	+	11	◻	21	^	31	▼	41	◄
2	+	12	◻	22	∨	32	▼	42	◄
3	+	13	·	23	∨	33	▽	43	◄
4	*	14	+	24	●	34	▼	44	◄
5	×	15	◆	25	×	35	△	45	◄
6	×	16	○	26	●	36	▲	46	◄
7	·	17	○	27	○	37	⊥	47	◄
8	■	18	⊠	28	·	38	±		
9	■	19	◇	29	▼	39	⊕		

Define A Circuit – DSS Command

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. If MarkCapacitors=yes, it marks Capacitor objects 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. If MarkRegulators=yes, it marks RegControl object with a symbol. See RegMarkerCode. The bus coordinate for the controlled transformer winding must exist.

114

Define A Circuit – DSS Command

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.

Define A Circuit – DSS Command

Option reference

Mode=

It specifies the solution mode for the active circuit. Mode can be one of the following:

Mode=Snap:

It solves 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:

It performs a series of solutions following the daily load curves. The Stepsize defaults to 3600 sec (1 hr). You can 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).

...

Define A Circuit – DSS Command

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:

The DSS follows the duty cycle curves with the specified time increment. It performs the solution for the number of times specified by the Number parameter.

Mode=Dynamics:

DSS sets the solution mode for a dynamics solution. The dynamic solution must be preceded by a successful power flow solution, so that the machines can be initialized. DSS 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.

...

117

Define A Circuit – DSS Command

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.

Define A Circuit – DSS Command

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

...

Define A Circuit – DSS Command

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

...

Define A Circuit – DSS Command

Option reference

Mode=LD2

...

load at the selected hour by the L-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

...

Define A Circuit – DSS Command

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.

122

Define A Circuit – DSS Command

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.

Define A Circuit – DSS Command

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.

Define A Circuit – DSS Command

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.).

Define A Circuit – DSS Command

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.

Define A Circuit – DSS Command

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.

...

Define A Circuit – DSS Command

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)

128

Define A Circuit – DSS Command

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.

Define A Circuit – DSS Command

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.

Define A Circuit – DSS Command

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.

Define A Circuit – DSS Command

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.

Define A Circuit – DSS Command

Option reference

Type=

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

Ueregs =

Which EnergyMeter register(s) to use for UE (Unserved Energy) 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 (Energy Exceeding Normal)). 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.

Define A Circuit – DSS Command

Option reference

Voltagebases=

It defines legal bus voltage bases for this circuit.

- You can enter an array of the legal voltage bases, in phase-to-phase voltages, for example:

```
set voltagebases=[12.47, 34.5]
```

- 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 default voltage bases are as follows:

```
.208, .480, 12.47, 24.9, 34.5, 115.0, 230.0
```

- 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 report.

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.

Define A Circuit – DSS Command

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.

Define A Circuit – General Object

- Line Code;
- WireData;
- Line Geometry;
- Load Shape;
- Growth Shape;
- TCC_Curve.

Define A Circuit – General Object

General objects

These are objects *common* to all circuits in the OpenDSS. Why they are called general objects? Because any circuit can reference the data contained in these objects.

Linecode objects

LineCode objects contain *impedance* characteristics for lines and cables. You can define a line by its LineCode and its length. LineCode objects are usually defined in a separate file.

- The impedance characteristics of a line are described by its *series impedance* matrix and *nodal capacitance* matrix.
- Note that the impedances of lines may be *specified directly* and you do not need to use a line code, although the linecode will be more convenient most of the time. There may be hundreds of lines, but only *a few* different kinds of line constructions.

137

Define A Circuit – General Object

Line code

The LineCode properties, in order, are: (not case sensitive)

Nphases = Number of phases. Default = 3.
R1 = Positive-Sequence resistance, ohms per unit length.
X1 = Positive-Sequence reactance, ohms per unit length.
R0 = Zero-Sequence resistance, ohms per unit length.
X0 = Zero-Sequence reactance, ohms per unit length.
C1 = Positive-Sequence capacitance, nanofarads per unit length
C0 = Zero-Sequence capacitance, nanofarads per unit length
Units = {mi | km | kft | m | ft | in | cm} Length units. If units is not specified, it is assumed that the units correspond to the length being used in the Line models.
Rmatrix = Series resistance matrix, in ohms per unit length.
Xmatrix = Series reactance matrix, in ohms per unit length.
Cmatrix = Shunt nodal capacitance matrix, in nanofarads per unit length.
BaseFreq = Base Frequency at which the impedance values are specified. Default = 60.0 Hz.
Normamps = Normal ampacity, amps.
Emergamps = Emergency ampacity, amps.
Faultrate = Number of faults per year per unit length. This is the default for this general line construction.
...

138

Define A Circuit – General Object

Line code

```
...  
Pctperm = Percent of the fault that become permanent (requiring  
a line crew to repair and a sustained interruption).  
Kron = Y/N. Default=N. Perform Kron reduction on the impedance  
matrix after it is formed, reducing order by 1. Do this only on  
initial definition after matrices are defined. Ignored for  
symmetrical components.  
Rg = Carson earth return resistance per unit length used to  
compute impedance values at base frequency. See description  
above. For making better adjustments of line impedance values  
for frequency for harmonics studies. Default= 0.01805 ohms per  
1000 ft at 60 Hz. If you do not wish to adjust the earth return  
impedance for frequency, set both Rg and Xg to zero. Generally  
avoid Kron reduction if you will be solving at frequencies other  
than the base frequency and wish to adjust the earth return  
impedance.  
...
```

Define A Circuit – General Object

Line code

...

Xg = Carson earth return reactance per unit length used to compute impedance values at base frequency. See description above. For making better adjustments of line impedance values for frequency for harmonics studies. Default= 0.155081 ohms per 1000 ft at 60 Hz. If you do not wish to adjust the earth return impedance for frequency, set both Rg and Xg to zero. Generally avoid Kron reduction if you will be solving at frequencies other than the base frequency and wish to adjust the earth return impedance.

Rho = Earth resistivity used to compute earth correction factor. Default=100 meter ohms.

Like = Name of an existing LineCode object to build this like.

Define A Circuit – General Object

WireData

WireData defines the *raw conductor data* which is used to compute the impedance for a specific line geometry. The parameters are:

Rdc = dc Resistance, in ohms per unit length (see Runits). Defaults to Rac if not specified.

Rac = Resistance at 60 Hz per unit length. Defaults to Rdc if not specified.

Runits = Length units for resistance, {mi|kft|km|m|Ft|in|cm } Default=none.

GMRac = GMR (Geometrical Mean Radius) at 60 Hz. Defaults to .7788*radius, if not specified.

GMRunits = Units for GMR, {mi|kft|km|m|Ft|in|cm } Default=none.

Radius = Outside radius of conductor. Defaults to GMR/0.7788, if not specified.

Normamps = Normal ampacity, amperes. Defaults to Emergency amps/1.5 if not specified.

Emergamps = Emergency ampacity, amperes. Defaults to 1.5 * Normal Amps if not specified.

Diam = Diameter; Alternative method for entering radius.

Like = Make like another object of this class.

Define A Circuit – General Object

LineGeometry

LineGeometry is used to define the *positions* of the conductors. The properties are as follows:

Nconds = Number of conductors in this geometry. Default is 3. It triggers memory allocations. It should be defined first!

Nphases = Number of phases. Default =3; All other conductors are considered neutrals and might be reduced out.

Cond = Set this to number of the conductor you wish to define. Default is 1.

Wire = Code from WireData. MUST BE PREVIOUSLY DEFINED. no default.

X = x coordinate.

H = Height of conductor.

Units = Units for x and h: {mi|kft|km|m|Ft|in|cm } Initial default is "ft", but defaults to last unit defined.

Normamps = Normal ampacity, amperes for the line. Defaults to first conductor if not specified.

Emergamps = Emergency ampacity, amperes. Defaults to first conductor if not specified.

Reduce = { Yes | No} Default = no. Reduce to Nphases (Kron Reduction). Reduce out neutrals.

Like = Make like another object, e.g.:

New Capacitor.C2 like=c1 ...

142

Define A Circuit – General Object

LineGeometry – Example of Defining A Line

Define the wire data:

```
New Wiredata.ACSR336 GMR=0.0255000 DIAM=0.7410000 RAC=0.3060000
~ NormAmps=530.0000
~ Runits=mi radunits=in gmrunits=ft
New Wiredata.ACSR1/0 GMR=0.0044600 DIAM=0.3980000 RAC=1.120000
~ NormAmps=230.0000
~ Runits=mi radunits=in gmrunits=ft
```

Define the Geometry data:

```
New Linegeometry.HC2_336_1neut_0Mess nconds=4 nphases=3
~ cond=1 Wire=acsr336 x=-1.2909 h=13.716 units=m
~ cond=2 Wire=acsr336 x=-0.502 h=13.716 !units=m
~ cond=3 Wire=acsr336 x=0.5737 h=13.716 !units=m
~ cond=4 Wire= ACSR1/0 x=0 h=14.648 ! units=m ! neutral
```

Define a 300-ft line section:

```
New Line.Line1 Bus1=xxx Bus2=yyy
~ Geometry= HC2_336_1neut_0Mess
~ Length=300 units=ft
```

- RAC = Resistance at 60 Hz per unit length.
- Runits = Length units for resistance.
- Radunits = Units for outside radius.
- GMRunits = Units for GMR.

Define A Circuit – General Object

Loadshape

Loadshape object is very important for all types of *sequential* power flow solutions. Performing loadshape is a very powerful capability of OpenDSS.

- To represent the variations of the load over some time period, a LoadShape class is designed in OpenDSS. A loadshape consists of a series of *multipliers*, typically ranging from 0.0 to 1.0 which are applied to the base kW values of the load.
- Load shapes are generally *fixed interval*, but can also be variable interval. For the latter, both the time and the multiplier must be specified.
- All Loadshapes, whether they are daily, yearly, or some arbitrary duty cycle, are maintained in this loadshape class.
- The Loadshape arrays may be entered directly in command line, or the load shapes may be stored in files, from which the shapes are loaded.

Define A Circuit – General Object

Loadshape

The properties for LoadShape objects are:

Npts = Number of points to expect when defining the curve.

Interval = time interval of the data, in Hr. Default=1.0. If the load shape has non-uniformly spaced points, specify the interval as 0.0.

mInterval = Specify Interval in minutes.

sInterval = Specify Interval in seconds.

Mult = Array of multiplier values. Looking for Npts values.

Hour = Array of hour values corresponding to the multipliers.

Qmult = Array of multiplier values. Same property rules as the Mult property. If specified, the multiplier is applied to the Load (or Generator) kvar value. If omitted, the value of Mult is applied to the kvar value.

Mean = Mean of the multiplier array.

Stddev = Standard Deviation (see Mean, above).

Define A Circuit – General Object

Loadshape

The LoadShape object can also get load data from a *file*.

Csvfile = Name of a CSV file containing load shape data, one interval to a line.

Sngfile = Name of a binary file of single-precision floating point values containing the load shape data.

Dblfile = Name of a binary file of double-precision floating point values containing the load shape data.

Action= {Normalize | DblSave | SngSave} After defining load curve data, setting action=normalize will modify the multipliers so that the peak is 1.0. The mean and std deviation are recomputed. Setting action=DblSave or SngSave will cause the present mult and qmult values to be written to either a packed file of double or single, respectively.

UseActual = {Yes | No* | True | False*} If true, signals to Load, Generator, or other objects to use the value of the multiplier as the actual kW, kvar value rather than a multiplier.

...

Define A Circuit – General Object

Loadshape

...

Pmax, Qmax= If you define the LoadShape object with UseActual=Yes, when you define any of the Duty, Daily, or Yearly properties of a load or generator, the kW property is redefined to the kW and kvar values at the time of the peak kW in the loadshape. This will be the value used for the initial Snapshot solution. If you define more than one loadshape object, the last one overrides any previous definition, as with all OpenDSS properties. You can query the Pmax and Qmax properties of the Loadshape object to see what was computed.

Pbase = Base P value for normalization. Default is zero, meaning the peak will be used.

Like = Name of an existing loadshape object to base this one on.

Define A Circuit – General Object

Growthshape

A GrowthShape object is similar to a Loadshape object.

- However, it is intended to represent the growth in load *year-by-year*, and the way the curve is specified is entirely different. You must enter the growth for the *first* year. Thereafter, only the years where there is a change must be entered. Otherwise it is assumed the growth stays the same.
- Growth rate is specified by specifying the multiplier for the previous year's load. Thus, if the load grows 2.5% in 1999, the multiplier for that year will be specified as 1.025.

The parameters are:

Npts = Number of points to expect when defining the curve.

Year = Array of year values corresponding to the multiplier values. Enter only those years in which the multiplier changes.

Mult= Array of Multiplier values corresponding to the year values. Enter multiplier by which the load will grow in this year.

...

Define A Circuit – General Object

Growthshape

...

Csvfile = Name of a csv file containing one (year, mult) point per line. Separate the year from the multiplier by a comma.

Sngfile = Name of a file of single-precision numbers containing (year, mult) points packed.

Dblfile = Name of a file of single-precision numbers containing (year, mult) points packed.

Like = Name of an existing GrowthShape object to base this one on.

Define A Circuit – General Object

TCC_Curve

A TCC_Curve object is defined similarly to Loadshape object in that they all are defined by curves consisting of arrays of points.

- TCC_Curve is intended to model time-current characteristics for overcurrent relays, it is also used for other relay types requiring time curves. Both the *time array* and the *current array* must be entered.

The properties are:

Npts = Number of points to expect when defining the curve.

C_Array = Array of current (or voltage or whatever) values corresponding to time values in T_Array (see T_Array).

T_Array = Array of time values in sec. Typical array syntax:

```
t_array = (1, 2, 3, 4, ...)
```

You may also substitute a file designation: t_array = (file=filename). The specified file has one value per line.

Like = Name of an existing GrowthShape object to base this one on.

Define A Circuit – Source & Fault Objects

- Vsource Object;
- Isource Object;
- Fault Object.

Define A Circuit – Source & Fault Objects

V_{source}

- Voltage source is a special *power conversion* element. It is special because voltage sources are used to initialize the power flow solution with all other injection sources set to zero.
- A V_{source} object is a *two-terminal, multi-phase* Thevenin equivalent. That is, it is a voltage source behind an impedance. The V_{source} properties are specified as it would commonly be for a power system source equivalent: Line-line voltage (kV) and short circuit MVA.
- The most common way to use a voltage source object is with the first terminal connected to the bus of interest, with the second terminal connected to ground (voltage reference).

Define A Circuit – Source & Fault Objects

Vsource

The properties are:

Bus1 = Name of bus to which the source's first terminal is connected. You should remember to specify the node order if the terminals are connected in some unusual manner. Side effect: The processing of this property results in the setting of the Bus2 property, so that all conductors in terminal 2 are connected to ground.

Bus2 = Name of bus to which the source's second terminal is connected. If omitted, the second terminal is connected to ground (node 0) at the bus designated by the Bus1 property.

Basekv = base or rated Line-to-line kV.

Pu = Actual per unit at which the source is operating. It is assumed that it is balanced for all phases.

Angle = Base angle, in degrees, of the first phase.

Frequency = frequency of the source.

Phases = Number of phases. Default = 3.0.

Mvasc3 = 3-phase short circuit MVA= kV_{Base}^2/Z_{sc}

Mvasc1 = 1-phase short circuit MVA.

...

Define A Circuit – Source & Fault Objects

Vsource

The properties are, in order:

```
...  
x1r1 = Ratio of X1/R1. Default = 4.0.  
x0r0 = Ratio of X0/R0. Default = 3.0.  
Isc3 = Alternate method of defining the source impedance. 3-phase  
short circuit current, amps. Default is 10000.  
Isc1 = Alternate method of defining the source impedance.  
single-phase short circuit current, amps. Default is 10500.  
R1 = Alternate method of defining the source impedance.  
Positive-sequence resistance, in ohms. Default is 1.65.  
X1 = Alternate method of defining the source impedance.  
Positive-sequence reactance, in ohms. Default is 6.6.  
R0 = Alternate method of defining the source impedance.  
Zero-sequence resistance, in ohms. Default is 1.9.  
X0 = Alternate method of defining the source impedance.  
Zero-sequence reactance, in ohms. Default is 5.7.  
ScanType = {pos*| zero | none} Maintain specified symmetrical  
component sequence to assume for Harmonic mode solution. Default  
is positive sequence.  
...
```

Define A Circuit – Source & Fault Objects

Vsource

The properties are, in order:

...

Sequence = {pos*| neg | zero} Set the phase angle relationships for the specified symmetrical component sequence for solution modes other than Harmonics. Default is positive sequence.

Spectrum = Name of harmonic spectrum for this source. Default is "defaultvsource", which is defined when the DSS starts.

Z1 = Positive-sequence impedance, in ohms, as a 2-element array representing a complex number.

Z2 = Negative-sequence impedance, in ohms, as a 2-element array representing a complex number.

Z0 = Zero-sequence impedance, in ohms, as a 2-element array representing a complex number.

BaseFreq = Base Frequency for impedance specifications. Default is 60 Hz.

Like = Name of an existing Vsource object on which to base this one.

Define A Circuit – Source & Fault Objects

Isource

Isource is a current source, and it is a one-terminal current source object that can be connected to any bus.

- Its most common use is likely to be used to represent harmonic sources and to be used in frequency response scans of circuit models. You can perform positive- or zero-sequence scans.
- You can generally attach as many Isource objects to a bus as you want. An Isource is assumed to be ideal and its Yprim matrix is zero.

The properties are:

phases = Number of phases. Defaults to 3. For 3 or less, phase shift defaults to 120 degrees.

bus1 = Name of bus to which the source is connected.

bus1=busname

bus1=busname.1.2.3

Amps = Magnitude of current source, of each phase, in Amps.

angle = Phase angle in degrees of first phase: e.g., Angle=10.3.

Phase shift between phases defaults to 120 degrees when number of phases ≤ 3

...

Define A Circuit – Source & Fault Objects

Isource

...

frequency = Source frequency. Defaults to circuit fundamental frequency.

Scantype = {pos*| zero | none} Maintain specified sequence for harmonic solution. Default is positive sequence. Otherwise, angle between phases rotates with harmonic.

Sequence = {pos*| neg | zero} Set the phase angles for the specified symmetrical component sequence for solution modes other than Harmonics. Default is positive sequence.

Spectrum = Harmonic spectrum assumed for this source. Default is "default".

Inherited properties:

basefreq = Base Frequency for ratings.

enabled = {Yes|No or True|False} Indicates whether this element is enabled.

like = Make like another object.

Define A Circuit – Source & Fault Objects

Fault Object

A Fault object is nothing more than a *resistor* network that can be configured in a variety of ways.

- It is a *two-terminal* device in which the second terminal defaults to ground. This is often what is desired when simulating a fault.
- However, the OpenDSS Fault object can be configured to represent *any* type of fault. For example, it can be connected between transmission overbuild and distribution underbuild to simulate the transmission falling onto the distribution circuit.
- A Fault object is a standard *Power Delivery component*. You can have as many Fault objects on the circuit as you wish .

The properties are:

phases = Number of Phases. Default is 1.

Bus1 = Name of bus to which the first terminal is connected.

Bus2 = Name of bus to which the second terminal is connected. Defaults to all phases connected to first bus, node 0, if not specified.

R = Resistance, of each phase, in ohms. Default is 0.0001.

...

Define A Circuit – Source & Fault Objects

Fault Object

The properties are:

...

Gmatrix = Use this to specify a nodal conductance (G) matrix to represent some arbitrary resistance network.

MinAmps = Minimum amps that can sustain a temporary fault. Default is 5.

ONtime = The time (sec) at which the fault is established for time varying simulations. Default is 0.0 (on at the beginning of the simulation).

pctperm = Percent of failures that become permanent. (not used)

Temporary = {Yes | No} Default is No. Designates whether the fault is temporary. For time-varying simulations, the fault will be removed if the current through the fault drops below the MINAMPS criteria.

%stddev = Percent standard deviation in resistance to assume for Monte Carlo fault (MF) solution mode for GAUSSIAN distribution. Default is 0 (no variation from mean).

Define A Circuit – Power Delivery Element

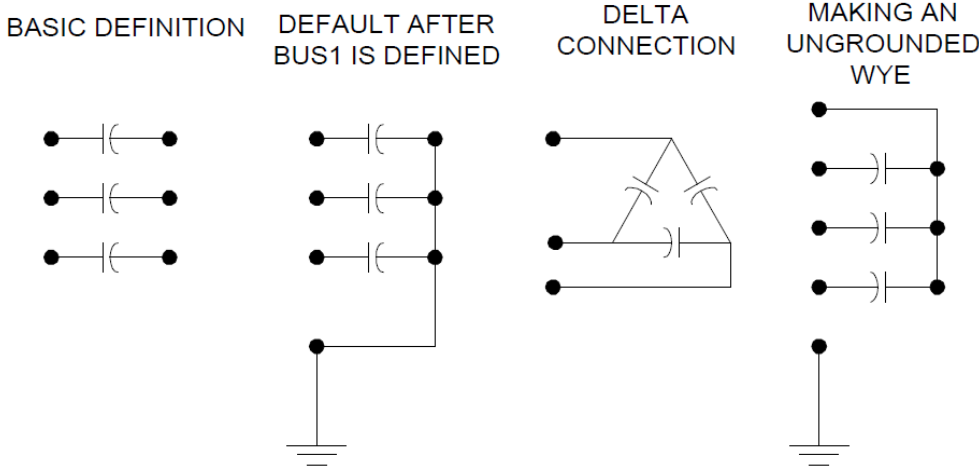
- Capacitor Object;
- Line Object;
- Reactor Object;
- Transformer Object.

Define A Circuit – Power Delivery Element

Capacitor Object

The capacitor model is basically implemented as a *two-terminal* power delivery element.

- For a capacitor object, if you don't specify a connection for the second bus, it will default to the 0 node (ground reference). That is, it defaults to a grounded wye (star) shunt capacitor bank.
- If you specify the connection to be "delta“, then the second terminal is eliminated.



Define A Circuit – Power Delivery Element

Capacitor Object

The properties are:

Bus1 = Name of the bus to which the first terminal is connected.

Bus2 = Name of the bus to which the second terminal is connected.

Phases = Number of phases. Default is 3.

Kvar = Rated kvar at rated kV, total of all phases. Each phase is assumed equal. It is the most common three ways to define a power capacitor.

Kv = Rated kV of the capacitor (not necessarily same as bus rating).

Conn = Connection of bank. One of {wye | ln} for wye connected banks or {delta | ll} for delta (line-line) connected banks. Default is wye (or straight-through for series capacitor).

Cmatrix = Alternate method of defining a capacitor bank.

Cuf = Alternate method of defining a capacitor bank.

R = Array of series resistance in each phase (line), ohms. Default is 0.0

XL = Array of series inductive reactance(s) in each phase (line) for filter, ohms at base frequency. Use this OR "h" property to define filter. Default is 0.0.

...

Define A Circuit – Power Delivery Element

Capacitor Object

The properties are:

...

Harm = ARRAY of harmonics to which each step is tuned. Zero is interpreted as meaning zero reactance (no filter). Default is zero.

Numsteps = Number of steps in this capacitor bank. Default = 1. Forces reallocation of the capacitance, reactor, and states array.

states = ARRAY of integers {1|0} used for representing the state of each step (on|off). Defaults to 1 when reallocated (on). Capcontrol will modify this array as it turns steps on or off.

Normamps = Normal current rating. Automatically computed if kvar is specified. Otherwise, you need to specify if you wish to use it.

Emergamps = Overload rating. Defaults to 135% of Normamps.

Faultrate = Annual failure rate. Failure events per year. Default is 0.0005.

Pctperm = Percent of faults that are permanent. Default is 100.0.

Basefreq = Base frequency, Hz. Default is 60.0

Like = Name of another Capacitor object on which to base this one.

163

Define A Circuit – Power Delivery Element

Line Object

The Line element is used to model most *multi-phase, two-port* lines or cables. It is a “Pi” model with shunt capacitance.

- Line element is described by its *impedance*. Impedances may be specified by symmetrical component values or by matrix values. Alternatively, you can simply refer to an existing *LineCode* object, or you can specify an existing *Geometry* object and the line impedances will be computed.
- For the units property, you can declare any length measurement in whatever units you want. Internally, everything is converted to meters. Just be sure to declare the units. Otherwise, they are assumed to be compatible with other data.
- The default Line object is a 1000-ft overhead line with 336 MCM ACSR conductor on a 8-ft crossarm.

Define A Circuit – Power Delivery Element

Line Object

The properties, are:

Bus1 = Name of bus for terminal 1.

Bus2 = Name of bus for terminal 2.

Linecode = Name of an existing LineCode object containing impedance definitions.

Length = Length multiplier to be applied to the impedance data.

Phases = No. of phases. Default = 3.

R1 = positive-sequence resistance, ohms per unit length.

X1 = positive-sequence reactance, ohms per unit length.

R0 = zero-sequence resistance, ohms per unit length.

X0 = zero-sequence reactance, ohms per unit length.

C1 = positive-sequence capacitance, nanofarads per unit length.
Setting any of R1, R0, X1, X0, C1, C0 forces the program to use the symmetrical component line definition.

C0 = zero-sequence capacitance, nanofarads per unit length.

B1 = Alternate way to enter C1, microS per unit length.

B0 = Alternate way to enter C0, microS per unit length.

...

Define A Circuit – Power Delivery Element

Line Object

The properties, are:

...

Normamps = Normal ampacity, amps.

Emergamps = Emergency ampacity, amps. Usually the one-hour rating.

Faultrate = Number of faults per year per unit length. This is the default for this general line construction.

Pctperm = Percent of the faults that become permanent (requiring a line crew to repair and a sustained interruption).

Repair = Hours to repair.

BaseFreq = Base Frequency at which the impedance values are specified. Default = 60.0 Hz.

Rmatrix = Series resistance matrix, ohms per unit length. See Command Language for syntax. Lower triangle form is acceptable.

Xmatrix = Series reactance matrix, ohms per unit length.

Cmatrix = Shunt nodal capacitance matrix, nanofarads per unit length.

Switch = {y/n | T/F} Default= no/false. Designates this line as a switch for graphics and algorithmic purposes.

...

Define A Circuit – Power Delivery Element

Line Object

The properties, are:

...

Rg = Carson earth return resistance per unit length used to compute impedance values at base frequency.

Xg = Carson earth return reactance per unit length used to compute impedance values at base frequency.

Rho = Earth resistivity used to compute earth correction factor. Overrides Line geometry definition if specified. Default=100 meter ohms.

Geometry = Geometry code for LineGeometry Object. Supercedes any previous definition of line impedance. Line constants are computed for each frequency change or rho change.

EarthModel = One of {Carson | FullCarson | Deri}. Default is the global value established with the Set EarthModel option. See the Options Help on EarthModel option. This is used to override the global value for this line. This option applies only when the "geometry" property is used.

Units = Length Units = {none | mi | kft | km | m | Ft | in | cm } Default is None - assumes length units match impedance units.

Like = Name of an existing Line object to build this like.

Define A Circuit – Power Delivery Element

Reactor Object

The Reactor object is implemented with basically the same philosophy as a Capacitor (and a Fault object).

- It is a *constant impedance* element that can be configured into a variety of connections. Like the Capacitor, the second terminal defaults to a Wye connection to ground, if not specified. In that case, it is flagged as a Shunt element.
- By default, the Reactor has *no coupling* between the phases. Shunt reactors would typically be defined by kV and kvar properties, similar to a capacitor. Series reactors without mutual coupling would be defined by the R and X properties.
- Note that if the connection is specified as “delta”, only the first terminal matters; there is no second terminal.
- Mutual coupling between phases can be achieved by specifying Rmatrix and Xmatrix properties.

Define A Circuit – Power Delivery Element

Reactor Object

The properties, are:

phases = Number of phases.

bus1 = Name of the bus for terminal 1.

bus2 = Name of bus for terminal 2. Defaults to all phases connected to first bus, node 0. (Shunt Wye Connection).

kv = For 2, 3-phase, specify the phase-phase kV. Otherwise specify actual coil rating.

kvar = Total kvar, of all phases. Evenly divided among phases. It only determines X. You should specify R separately.

conn = {wye | delta |LN |LL} Default is wye, which is equivalent to LN. If Delta, then there is only one terminal.

Parallel = {Yes | No} Default=No. Indicates whether Rmatrix and Xmatrix are to be considered to be in parallel. This makes a significant difference in harmonic studies. Default is series. For other models, specify R and Rp.

R = Resistance (in series with reactance), of each phase, in ohms.

Rmatrix = Resistance matrix, lower triangle, ohms at base frequency. Order of the matrix is the number of phases. Mutually exclusive to specifying parameters by kvar or R.

Rp = Resistance in parallel with R and X (the entire branch). Assumed infinite if not specified.

...

Define A Circuit – Power Delivery Element

Reactor Object

The properties, are:

...

X = Reactance, of each phase, in ohms at base frequency.

Xmatrix = Reactance matrix, lower triangle, ohms at base frequency. Order of the matrix is the number of phases. Mutually exclusive to specifying parameters by kvar or X.

Z = Alternative way of defining R and X properties. Enter a 2-element array representing $R + jX$ in ohms.

Z1 = Positive-sequence impedance, ohms, as a 2-element array representing a complex number.

Z2 = Negative-sequence impedance, ohms, as a 2-element array representing a complex number.

Z0 = Zero-sequence impedance, ohms, as a 2-element array representing a complex number.

Properties inherited from the circuit element class:

normamps = Normal rated current.

emergamps = Maximum current.

repair = Hours to repair.

faultrate = No. of failures per year.

Pctperm = Percent of failures that become permanent.

...

Define A Circuit – Power Delivery Element

Reactor Object

Properties inherited from the circuit element class:

...

basefreq = Base Frequency for ratings.

enabled = {Yes|No or True|False} Indicates whether this element is enabled.

like = Make like another object.

Define A Circuit – Power Delivery Element

Transformer Object

The Transformer object is implemented as a *multi-terminal* (two or more) power delivery element.

- A transformer consists of *two or more Windings*, connected in a variety of ways (with a default Wye-Delta connection). You can specify the parameters of one winding at a time. Alternatively, you can use arrays to set all the winding values at once. For the first way, you can use the "wdg=..." parameter to select a winding for editing.
- Transformers have *one or more phases*. The number of conductors per terminal is always one more than the number of phases. For wye- or star-connected windings, the extra conductor is the neutral conductor. For delta-connected windings, the extra conductor is open internally (you normally leave this connected to node 0).

Define A Circuit – Power Delivery Element

Transformer Object

The properties, are:

Phases = Number of phases. Default is 3.

Windings = Number of windings. Default is 2.

Note: For defining the winding values *one winding at a time*, use the following parameters. If you define one winding at a time, you should always start the winding definition with "wdg = ...".

Wdg = An integer representing the winding which will become the active winding for subsequent data.

Bus = The name of bus to which the winding is connected.

Conn = Connection of this winding. One of {wye | ln} for wye connected banks or {delta | ll} for delta (line-line) connected banks. Default is wye.

Kv = Rated voltage of this winding, in kV. For transformers designated 2- or 3-phase, enter phase-to-phase kV. For all other designations, enter actual winding kV rating. Two-phase transformers are assumed to be employed in a 3-phase system. Default is 12.47 kV.

...

Define A Circuit – Power Delivery Element

Transformer Object

The properties, are:

```
...
Kva = Base kVA rating of this winding.
Tap = Per unit tap on which this winding is set.
%R = Percent resistance of this winding on the rated kVA base.
      (Reactance is between two windings and is specified separately --
      see below.)
rneut = Neutral resistance to ground in ohms for this winding.
      Ignored if delta winding. For open ungrounded neutral, set to a
      negative number. Default is -1 (capable of being ungrounded).
xneut = Neutral reactance in ohms for this winding. Ignored if
      delta winding. Assumed to be in series with neutral resistance.
      Default is 0.
```

Note: Alternatively, you can use the following properties to set the winding values using *arrays* (setting of `wdg= ...` is ignored). The names of these properties are simply the plural form of the property name above.

```
Buses = Array of bus definitions for windings [1, 2. ...].
Conns = Array of winding connections for windings [1, 2. ...].
KVs = Array of kV ratings for windings [1,2,...].
```

```
...
```

174

Define A Circuit – Power Delivery Element

Transformer Object

The properties, are:

...

KVAs = Array of base kVA ratings for windings [1,2,...].

Taps = Array of per unit taps for windings [1,2,...].

%Rs = Array of percent resistances for windings [1, 2. ...]

In addition, you can use the following properties to define the *reactances* of the transformer. For 2- and 3-winding transformers, you may use the conventional XHL, XLT, and XHT (or X12, X23, X13) parameters. You may also put the values in an array (xscarray), which is required for higher phase order transformers. There are always $n*(n-1)/2$ different short circuit reactances, where n is the number of windings. Always use the kVA base of the first winding for entering impedances. Impedance values are entered in percent.

XHL (or X12) = Percent reactance high-to-low (winding 1 to winding 2).

XLT (or X23) = Percent reactance low-to-tertiary (winding 2 to winding 3).

XHT (or X13) = Percent reactance high-to-tertiary (winding 1 to winding 3).

...

175

Define A Circuit – Power Delivery Element

Transformer Object

The properties, are:

```
...
General transformer rating data:
Thermal = Thermal time constant, hrs. Default is 2.
n = Thermal exponent, n, from IEEE/ANSI C57. Default is 0.8.
m = Thermal exponent, m, from IEEE/ANSI C57. Default is 0.8.
flrise = Full-load temperature rise, degrees centigrade. Default
is 65.
hsrise = Hot-spot temperature rise, degrees centigrade. Default
is 15.
%Loadloss = Percent Losses at rated load. Causes the %r values
to be set for windings 1 and 2.
%No-loadloss = Percent No load losses at nominal voltage. Default
is 0. Causes a resistive branch to be added in parallel with the
magnetizing inductance.
%imag = Percent magnetizing current. Default is 0. An inductance
is used to represent the magnetizing current. This is embedded
within the transformer model as the primitive Y matrix is being
computed.
...
```


Define A Circuit – Power Delivery Element

Transformer Object

The properties, are:

...

Ppm_Antifloat = Parts per million for anti floating reactance to be connected from each terminal to ground. Default is 1. That is, the diagonal of the primitive Y matrix is increased by a factor of 1.000001.

NormHKVA = Normal maximum kVA rating for H winding (1). Usually 100 - 110% of maximum nameplate rating.

EmergHKVA = Emergency maximum kVA rating for H winding (1). Usually 140-150% of maximum nameplate rating. This is the amount of loading that will cause 1% loss of life in one day.

Faultrate = Failure rate for transformer. Defaults to 0.007 per year. All are considered permanent.

Basefreq = Base frequency, Hz. Default is 60.0

Like = Name of another Transformer object on which to base this one.

Sub = Yes/No. Designates whether this transformer is to be treated as a substation. Default is No.

Define A Circuit – Power Conversion Element

- Load Object;
- Generator Object;
- Storage Object.

Define A Circuit – Power Conversion Element

Load Object

A Load is a complicated Power Conversion element that is at the heart of many analysis.

- It is basically defined by its nominal kW and PF or its kW and kvar. Then it may be modified by a number of *multipliers*, including the global circuit load multiplier, yearly load shape, daily load shape, and a duty-cycle load shape.
- The default is for the load to be a *current injection source*. Thus, its primitive Y matrix contains only the impedance that might exist from the neutral of a wye-connected load to ground.
- Loads are assumed *balanced* for the number of specified phases. If you would like unbalanced loads, you can define separate single-phase loads.

Define A Circuit – Power Conversion Element

Load Object

There are three normal ways to specify the base load:

1. kW, PF
2. kW, kvar
3. kVA, PF

The properties are:

Bus1 = Name of bus to which the load is connected. You should include node definitions if the terminal conductors are connected abnormally.

Phases = No. of phases of this load.

Kv = Base voltage for load. For 2- or 3-phase loads, it is specified in phase-to-phase kV. For all other loads, specify the actual kV across the load branch. If wye (star) connected, then specify phase-to-neutral (L-N) kV. If delta or phase-to-phase connected, specify the phase-to-phase (L-L) kV.

...

Define A Circuit – Power Conversion Element

Load Object

The properties are:

...

Kw = nominal active power, in kW, for the load. Total of all phases. See kVA.

Pf = nominal Power Factor for load. Negative PF is leading. Specify either PF or kvar (see below). If both are specified, the last one specified takes precedence.

Model = An integer defining how the load will vary with voltage. The load models currently implemented are:

1: Constant P and constant Q (Default): Commonly used for power flow studies

2: Constant Z (or constant impedance)

3: Constant P and quadratic Q

4: Exponential

5: Constant I (or constant current magnitude). Sometimes used for rectifier load

6: Constant P and fixed Q (at the nominal value)

7: Constant P and quadratic Q (i.e., fixed reactance)

8: ZIP (see ZIPV)

...

Define A Circuit – Power Conversion Element

Load Object

The properties are:

...

Yearly = Name of Yearly load shape.

Daily = Name of Daily load shape.

Duty = Name of Duty cycle load shape. Defaults to Daily load shape if not defined.

Growth = Name of Growth Shape. Growth factor defaults to the circuit's default growth rate if not defined. (see Set %Growth command)

Conn = {wye | y | LN} for Wye (Line-Neutral) connection; {delta | LL} for Delta (Line-Line) connection. Default = wye.

Kvar = Base kvar. If this is specified, it supercedes PF. (see PF)

Rneut = Neutral resistance, ohms. If entered as negative, non-zero number, neutral is assumed open, or ungrounded. Ignored for delta or line-line connected loads.

Xneut = Neutral reactance, ohms. Ignored for delta or line-line connected loads. Assumed to be in series with Rneut value.

Status = {fixed| variable}. Default is variable. If fixed, then the load is not modified by multipliers; it is fixed at its defined base value.

...

Define A Circuit – Power Conversion Element

Load Object

The properties are:

...

Class = An integer number segregating the load according to a particular class.

Vminpu = Default = 0.95. Minimum per unit voltage for which the MODEL is assumed to apply. Below this value, the load model reverts to a constant impedance model.

Vmaxpu = Default = 1.05. Maximum per unit voltage for which the MODEL is assumed to apply. Above this value, the load model reverts to a constant impedance model.

VminNorm = Minimum per unit voltage for load EEN (Energy Exceeding Normal) evaluations, Normal limit.

VminEmerg = Minimum per unit voltage for load UE (Unserviced Energy) evaluations, Emergency limit. Default = 0, which defaults to system "vminemerg" property (see Set Command under Executive).

XfkVA = Default = 0.0. Rated kVA of service transformer for allocating loads based on connected kVA at a bus.

AllocationFactor = Default = 0.5. Allocation factor for allocating loads based on connected kVA at a bus.

...

183

Define A Circuit – Power Conversion Element

Load Object

The properties are:

...

kVA = Definition of the Base load in kVA, total of all phases. This is intended to be used in combination with the power factor (PF) to determine the actual load. Legal ways to define base load (kW and kvar):

kW, PF

kW, kvar

kVA, PF

XFKVA * Allocationfactor, PF

kWh/(kWhdays*24) * Cfactor, PF

%mean = Percent mean value for load to use for monte carlo studies if no loadshape is assigned to this load. Default is 50.

%stddev = Percent Std deviation value for load to use for monte carlo studies if no loadshape is assigned to this load. Default is 10.

CVRwatts = Exponential parameter that defines the relationship between voltage (V) and active power (P) based on the following: $P/P0 = (V/V0)^{CVRwatts}$. P0 is the nominal power of the load at the base voltage V0.

...

Define A Circuit – Power Conversion Element

Load Object

The properties are:

...

CVRvars = Exponential parameter that defines the relationship between voltage (V0) and reactive power (Q0) based on the following: $Q/Q0 = (V/V0)^{CVRvars}$. Q0 is the nominal power of the load at the base voltage V0.

kWh = kWh billed for this period. Default is 0. See help on kVA and Cfactor and kWhDays.

kWhDays = Length of kWh billing period in days (24 hr days). Default is 30. Average demand is computed using this value.

CFactor = Factor relating average kW to peak kW. Default is 4.0. See kWh and kWhdays. See kVA.

CVRCurve = Default is NONE. Curve describing both watt and var factors as a function of time.

NumCust = Number of customers, this load. Default is 1.

spectrum = Name of harmonic current spectrum for this load. Default is "defaultload", which is defined when the DSS starts.

...

Define A Circuit – Power Conversion Element

Load Object

The properties are:

...

ZIPV = Array of 7 coefficients:

1. First 3 are ZIP weighting factors for active power (should sum to 1)
2. Next 3 are ZIP weighting factors for reactive power (should sum to 1)
3. Last 1 is cut-off voltage in p.u. of base kV; load is 0 below this cut-off, no defaults; all coefficients must be specified if using model=8.

%SeriesRL = Percent of load that is series R-L for Harmonic studies. Default is 50. Remainder is assumed to be parallel R and L. This has a significant impact on the amount of damping observed in Harmonics solutions.

Basefreq = Base frequency for which this load is defined. Default is 60.0.

Like = Name of another Load object on which to base this one.

Define A Circuit – Power Conversion Element

Generator Object

A Generator is a Power Conversion element similar to a Load object. Its rating is basically defined by its nominal kW and PF or its kW and kvar. It can also be modified by a number of *multipliers*, including the global circuit load multiplier, yearly load shape, daily load shape, and a dutycycle load shape.

- For power flow studies, the generator is essentially a *negative load* that can be dispatched.
- If the dispatch value (DispValue property) is 0, the generator always follows the appropriate *dispatch curve*, which is simply a Loadshape object.
- If you want to model a generator that is fully on whenever it is dispatched on, simply designate "Status=Fixed".
- Generator powers are assumed *balanced* over the number of phases. If you would like unbalanced generators, enter separate single-phase generators.

Define A Circuit – Power Conversion Element

Generator Object

Generator power models for power flow simulations are:

1. Constant P, Q (* dispatch curve, if appropriate).
 2. Constant Z (For simple, approximate solution)
 3. Constant P, |V|, somewhat like a standard power flow with voltage magnitudes and angles as the variables instead of P and Q.
 4. Constant P, fixed Q. P follows dispatch; Q is always the same.
 5. Constant P, fixed reactance. P follows dispatch, Q is computed as if it were a fixed reactance.
 6. User-written model
 7. Current-limited constant P, Q model (like some inverters).
- Most of the time you will use #1 for planning studies, assuming you want to specify a specific power. All generator models can follow Loadshapes. Some follow only the P component while the Type 1 can follow both a P and Q characteristic.

The properties, in numerical order, are:

bus1 = Name of bus to which the generator is connected.

Phases = No. of phases of this generator.

...

Define A Circuit – Power Conversion Element

Generator Object

The properties, in numerical order, are:

...

Kv = Base voltage for generator. For 2- or 3-phase generators, specify phase-to-phase kV. For all other generators, specify the actual kV across the generator branch. If wye (star) connected, specify the phase-to-neutral (L-N) kV. If delta or phase-to-phase connected, specify the phase-to-phase (L-L) kV.

Kw = nominal kW for generator. Total of all phases.

Pf = nominal Power Factor for generator. Negative PF is leading (absorbing vars). Specify either PF or kvar (see below). If both are specified, the last one specified takes precedence.

Model= An integer defining how the generator will vary with voltage. Presently defined models are:

- 1: Generator injects a constant kW at specified power factor.
- 2: Generator is modeled as a constant admittance.
- 3: Constant kW, constant kV. Somewhat like a conventional transmission power flow P-V generator.
- 4: Constant kW, Fixed Q (Q never varies)

...

Define A Circuit – Power Conversion Element

Generator Object

The properties, in numerical order, are:

...

5: Const kW, Fixed Q (as a constant reactance)

6: Compute load injection from User-written Model. (see usage of Xd, Xdp)

7: Constant kW, kvar, but current is limited when voltage is below Vminpu

Yearly = Name of Yearly load shape.

Daily = Name of Daily load shape.

Duty = name of Duty cycle load shape. Defaults to Daily load shape if not defined.

Dispvalue = Dispatch value. If = 0.0 then Generator follows dispatch curves. If > 0 then Generator is ON only when the global load multiplier exceeds this value. Then the generator follows dispatch curves (see also Status)

Conn = {wye | y | LN} for Wye (Line-Neutral) connection; {delta | LL} for Delta (Line-Line) connection. Default = wye.

Kvar = Base kvar. If this is specified, will supercede PF. (see PF)

...

Define A Circuit – Power Conversion Element

Generator Object

The properties, in numerical order, are:

...

Rneut = Neutral resistance, in ohms. If entered as negative, non-zero number, neutral is assumed open, or ungrounded. Ignored for delta or line-line connected generators. Default is 0.

Xneut = Neutral reactance, in ohms. Ignored for delta or line-line connected generators. Assumed to be in series with Rneut value.

Status = {fixed | variable}. If Fixed, then dispatch multipliers do not apply. The generator is always at full power when it is ON. Default is Variable (follows curves).

Class = An integer number segregating the generator according to a particular class.

Maxkvar = Maximum kvar limit for Model = 3. Defaults to twice the specified load kvar. Always reset this if you change PF or kvar properties.

Minkvar = Minimum kvar limit for Model = 3. Enter a negative number if generator can absorb vars. Defaults to negative of Maxkvar. Always reset this if you change PF or kvar properties.

...

Define A Circuit – Power Conversion Element

Generator Object

The properties, in numerical order, are:

...

Pvfactor = Convergence deceleration factor for P-V generator model (Model=3). Default is 0.1. If the circuit converges easily, you may want to use a higher number such as 1.0. Use a lower number if solution diverges. Use Debugtrace=yes to create a file that will trace the convergence of a generator model.

Debugtrace = {Yes | No } Default is no. Turn this on to capture the progress of the generator model for each iteration. Creates a separate file for each generator named "GEN_name.CSV".

Vminpu = Default = 0.95. Minimum per unit voltage for which the Model is assumed to apply. Below this value, the generator model reverts to a constant impedance model. For Model 7, this is used to determine the upper current limit. For example, if Vminpu is 0.90 then the current limit is $(1/0.90) = 111\%$.

Vmaxpu = Default = 1.05. Maximum per unit voltage for which the Model is assumed to apply. Above this value, the generator model reverts to a constant impedance model.

...

Define A Circuit – Power Conversion Element

Generator Object

The properties, in numerical order, are:

...

ForceON = {Yes | No} Forces generator ON despite requirements of other dispatch modes. Stays ON until this property is set to NO, or an internal algorithm cancels the forced ON state.

kVA = kVA rating of electrical machine. Defaults to 1.2* kW, if not specified. Applied to machine or inverter definition for Dynamics mode solutions.

MVA = MVA rating of electrical machine. Alternative to using kVA=.

Xd = Per unit synchronous reactance of machine. Presently used only for Thevinen impedance for power flow calcs of user models (model=6). Typically use a value from 0.4 to 1.0. Default is 1.0

Xdp = Per unit transient reactance of the machine. Used for Dynamics mode and Fault studies. Default is 0.27. For user models, this value is used for the Thevinen/Norton impedance for Dynamics Mode.

H = Per unit mass constant of the machine. MW-sec/MVA. Default is 1.0.

D = Damping constant. Usual range is 0 to 4. Default is 1.0. Adjust to get damping.

...

Define A Circuit – Power Conversion Element

Generator Object

The properties, in numerical order, are:

...

UserModel = Name of DLL containing user-written model, which computes the terminal currents for Dynamics studies, overriding the default model. Set to "none" to negate previous setting.

UserData = String (in quotes or parentheses) that gets passed to user-written model for defining the data required for that model.

ShaftModel = Name of user-written DLL containing a Shaft model, which models the prime mover and determines the power on the shaft for Dynamics studies. Models additional mass elements other than the single-mass model in the DSS default model. Set to "none" to negate previous setting.

ShaftData = String that gets passed to user-written shaft dynamic model for defining the data for that model.

spectrum = Name of harmonic voltage or current spectrum for this generator. Voltage behind "Xd" for machine. Current injection for inverter.

Basefreq= Base frequency for which this generator is defined. Default is 60.0.

Like= Name of another Generator object on which to base this one.

Define A Circuit – Power Conversion Element

Storage Object

- Storage object is a recently added model into OpenDSS, there is no description about it now. If you want to know more about storage object, you can search in the OpenDSS forum or contact Roger Dugan.
- However, the properties of storage object are summarized into a table, you can refer to the OpenDSS manual.

Define A Circuit – Control Element

- CapControl Object;
- RegControl Object.

Define A Circuit – Control Element

Control Object

- In OpenDSS, control elements are modeled *separately* from the power-carrying elements, which is one of the distinctive capabilities of OpenDSS. This design provides significant *flexibility*.
- Control elements typically have a *Sample* function that samples the voltage and current at the terminal that the control is monitoring.

CapControl Object

The capacitor control monitors the *voltage* and *current* at a terminal of a PDelement or a PCElement, and sends *switching messages* to a Capacitor object.

Define A Circuit – Control Element

CapControl Object

The properties are:

Element = Specifies the full object name of the circuit element, typically a line or transformer, to which the capacitor control's PT and/or CT are connected. There is no default; must be specified.

Capacitor = Specifies the name of Capacitor element which the CapControl controls. No Default; Must be specified. Do not specify the full object name. Example: Capacitor=cap1

CTPhase = Number of the phase being monitored for CURRENT control or one of {AVG | MAX | MIN} for all phases. Default=1. If delta or L-L connection, enter the first or the two phases being monitored [1-2, 2-3, 3-1]. It must be less than the number of phases. Does not apply to kvar control which uses all phases by default.

CTratio = Ratio of the CT from line amps to control ampere setting, for current and kvar control types.

...

Define A Circuit – Control Element

CapControl Object

The properties are:

...

DeadTime = Specifies the dead time after capacitor is turned OFF before it can be turned back ON. Default is 300 sec.

Delay = Specifies time delay, in seconds, from when the control is armed before it sends out the switching command to turn ON. The control may reset before the action actually occurs. This is used to determine which capacity control will act first. Default is 15. You may specify any floating point number to achieve a model of whatever condition is necessary.

DelayOFF = Specifies the time delay, in seconds, for control to turn OFF when present state is ON. Default is 15.

EventLog = {Yes/True* | No/False} Default is YES for CapControl. Log control actions to Eventlog.

OFFsetting = Value at which the control arms to switch the capacitor OFF. For Time control, it is OK to have Off time the next day (< On time)

...

Define A Circuit – Control Element

CapControl Object

The properties are:

...

ONsetting = Value at which the control arms to switch the capacitor ON (or ratchet up a step).

Type = Specifies the control type. Be careful to specify the ONsetting and OFFsetting appropriately with the type of control. Type of Control: (1) **Current**: Line Amps / Ctratio; (2) **Voltage**: Line-Neutral (or Line-Line for delta) Volts / Ptratio; (3) **kvar**: Total kvar, all phases (3-phase for pos seq model). This is directional; (4) **PF**: Power Factor, Total power in monitored terminal. Negative for Leading; (5) **Time**: Hrs from Midnight as a floating point number (decimal). 7:30am would be entered as 7.5.

PTPhase = Number of the phase being monitored for VOLTAGE control or one of {AVG | MAX | MIN} for all phases. Default=1. If delta or L-L connection, enter the first or the two phases being monitored [1-2, 2-3, 3-1]. Must be less than the number of phases. Does not apply to kvar control which uses all phases by default.

...

200

Define A Circuit – Control Element

CapControl Object

The properties are:

...

PTratio = Ratio of the PT that converts the monitored voltage to the control voltage. Default is 60. If the capacitor is Wye, the 1st phase line-to-neutral voltage is monitored. Else, the line-to-line voltage (1st - 2nd phase) is monitored.

terminal = Number of the terminal of the circuit element to which the CapControl is connected. 1 or 2, typically. Default is 1.

Vbus = Name of bus to use for voltage override function. Default is the bus at monitored terminal. Sometimes it is useful to monitor a bus in another location to emulate various DMS control algorithms.

Vmax = Maximum voltage, in volts. If the voltage across the capacitor divided by the PTRATIO is greater than this voltage, the capacitor will switch OFF regardless of other control settings. Default is 126 (goes with a PT ratio of 60 for 12.47 kV system).

...

Define A Circuit – Control Element

CapControl Object

The properties are:

...

Vmin = Minimum voltage, in volts. If the voltage across the capacitor divided by the PTRATIO is less than this voltage, the capacitor will switch ON regardless of other control settings. Default is 115 (goes with a PT ratio of 60 for 12.47 kV system).

VoltOverride = {Yes | No} Default is No. Switch to indicate whether VOLTAGE OVERRIDE is to be considered. Vmax and Vmin must be set to reasonable values if this property is Yes.

Define A Circuit – Control Element

RegControl Object

This control is designed to emulate a standard utility voltage regulator or LTC control.

- RegControl is attached to a particular winding of a transformer as the winding is to be monitored. It generally also adjusts the taps in that winding, but could also be directed to control the taps in another winding.
- This control has *line drop compensator modeling* by setting the R, X, CTprim, and PTratio properties.
- This control can also monitor the voltage at a *remote bus* to emulate various Smart Grid devices. This is a useful function for performing volt/var optimization.

A simple example of a regulator in a 12.47 kV system:

```
New RegControl.Reg1 Transformer=T1 Winding=2 Vreg=122 band=3  
ptratio=60
```

Define A Circuit – Control Element

RegControl Object

With a line-drop compensator, the definition might look like

```
New RegControl.Reg1 Transformer=T1 Winding=2 Vreg=122 band=3  
~ ptratio=60 CTprim=300 R=2 X=0
```

Controlling a bus at the end of the feeder to 118 V:

```
New RegControl.Reg1 Transformer=T1 Winding=2 Vreg=118 band=2  
bus=MyEndBus
```

The properties are:

transformer = Name of Transformer element to which the RegControl is connected. Do not specify the full object name; "Transformer" is assumed for the object class. Example:

```
Transformer=Xfmr1
```

winding = Number of the winding of the transformer element that the RegControl is monitoring. 1 or 2, typically.

vreg = Voltage regulator setting, in VOLTS, for the winding being controlled. Multiplying this value by the ptratio should yield the voltage across the WINDING of the controlled transformer. Default is 120.0

...

Define A Circuit – Control Element

RegControl Object

The properties are:

...

band = Bandwidth in VOLTS for the controlled bus (see help for ptratio property). Default is 3.0

delay = Time delay, in seconds, from when the voltage goes out of band to when the tap changing begins. This is used to determine which regulator control will act first. Default is 15. You may specify any floating point number to achieve a model of whatever condition is necessary.

ptratio = Ratio of the PT that converts the controlled winding voltage to the regulator voltage. Default is 60. If the winding is Wye, the line-to-neutral voltage is used. Else, the line-to-line voltage is used.

CTprim = It is the rating, in Amperes, of the primary CT rating for converting the line amps to control amps. The typical default secondary ampere rating is 0.2 Amps (check with manufacturer specs).

R = R setting on the line drop compensator in the regulator, expressed in VOLTS.

...

205

Define A Circuit – Control Element

RegControl Object

The properties are:

...

X = X setting on the line drop compensator in the regulator, expressed in VOLTS.

PTphase = For multi-phase transformers, the number of the phase being monitored or one of { MAX | MIN} for all phases. Default=1. Must be less than or equal to the number of phases. Ignored for regulated bus.

bus = Name of a bus (busname.nodename) in the system to use as the controlled bus instead of the bus to which the transformer winding is connected or the R and X line drop compensator settings.

debugtrace = {Yes | No* } Default is no. Turn this on to capture the progress of the regulator model for each control iteration. Creates a separate file for each RegControl named "REG_name.CSV".

EventLog = {Yes/True* | No/False} Default is YES for regulator control. Log control actions to Eventlog.

...

Define A Circuit – Control Element

RegControl Object

The properties are:

...

inversetime = {Yes | No* } Default is no. The time delay is adjusted inversely proportional to the amount the voltage is outside the band down to 10%.

maxtapchange = Maximum allowable tap change per control iteration in STATIC control mode. Default is 16.

Set this to 1 to better approximate actual control action.

Set this to 0 to fix the tap in the current position.

revband = Bandwidth for operating in the reverse direction.

revDelay = Time Delay in seconds (s) for executing the reversing action once the threshold for reversing has been exceeded. Default is 60 s.

reversible = {Yes |No*} Indicates whether or not the regulator can be switched to regulate in the reverse direction. Default is No. Typically applies only to line regulators and not to LTC on a substation transformer.

revNeutral = {Yes | No*} Default is no. Set this to Yes if you want the regulator to go to neutral in the reverse direction.

...

207

Define A Circuit – Control Element

RegControl Object

The properties are:

...

revR = R line drop compensator setting for reverse direction.

revThreshold = kW reverse power threshold for reversing the direction of the regulator. Default is 100.0 kw.

revvreg = Voltage setting in volts for operation in the reverse direction.

revX = X line drop compensator setting for reverse direction.

tapdelay = Delay in sec between tap changes. Default is 2. This is how long it takes between changes after the first change.

TapNum = An integer number indicating the tap position that the controlled transformer winding tap position is currently at, or is being set to.

vlimit = Voltage Limit for bus to which regulated winding is connected (e.g. first customer). Default is 0.0. Set to a value greater than zero to activate this function.

Define A Circuit – Meter Element

- EnergyMeter Object
 - Registers
 - Meter zones
 - Zones on Meshed Networks
 - Sampling
 - EEN and UE Definitions
 - Properties
- Monitor Object.

Define A Circuit – Meter Element

EnergyMeter Object

- An EnergyMeter object is an intelligent meter connected to a terminal of a circuit element. It simulates the behavior of an actual energy meter. However, it has more capability because it can access values at other places in the circuit rather than simply at the location at which it is installed. It measures not only power and energy values at its location, but losses and overload values within a defined region of the circuit.
- The operation of the object is simple. It has several registers that accumulate certain values. At the beginning of a study, the registers are reset to zero. At the end of each subsequent solution, the meter is instructed to take a sample. Energy values are then integrated using the interval of time that has passed since the previous solution.

Define A Circuit – Meter Element

EnergyMeter Object

-- Registers

- There are two types of registers:
 1. Energy Accumulators (for energy values)
 2. Maximum power values ("drag hand" registers).
- The energy registers may use trapezoidal integration (system option), which allows for somewhat arbitrary time step sizes between solutions with less integration error. This is important for using load duration curves approximated with straight lines, for example.
- The present definitions of the registers are, for example for a 22 kV system:

```
Hour, "kWh", "kvarh", "MaxkW", "MaxkVA", "ZonekWh", "Zonekvarh",  
"ZoneMaxkW", "ZoneMaxkVA", "OverloadkWhNormal",  
"OverloadkWhEmerg"
```
- Registers are frequently added for various purposes. You can view the present meters simply by solving and taking a sample. Then execute a Show Meters command.

211

Define A Circuit – Meter Element

EnergyMeter Object

-- Meter Zones

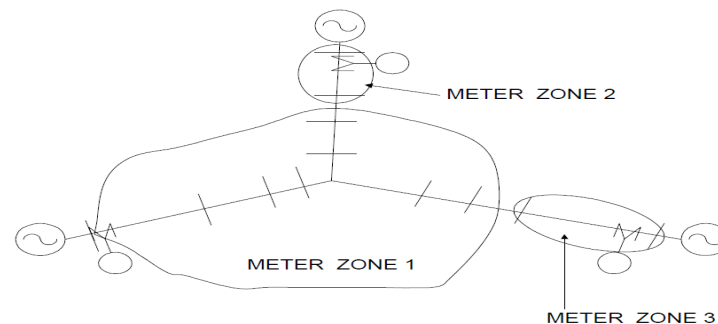
- The EnergyMeter object uses the concept of a zone. This is an area of the circuit for which the meter is responsible. It can compute energies, losses, etc., for any power delivery object and Load object in its zone (Generator objects have their own intrinsic meters).
- A zone is a collection of circuit elements "downline" from the meter. This concept is nominally applicable to radial circuits, but also has some applicability to meshed circuits.
- Zones are automatically updated after a change in the circuit unless the ZONELOCK option (Set command) is set to true (Yes). Then zones remain fixed after initial determination.

Define A Circuit – Meter Element

EnergyMeter Object

-- Zones on Meshed Networks

- While the concept of zones nominally applies to radial circuits, judicious placement of energy meters can make the concept useful for meshed networks as well. Keep in mind that there can be many EnergyMeter objects defined in the circuit. Their placement does not necessarily have to represent reality; they are for the reporting of power and energy quantities throughout the system.
- The automatic algorithm for determining zones will determine zones consistently for meshed networks, although the zones themselves may not be radial. If there are several meters on the network that could be monitoring the same zone, the first one defined will have access to all the elements except the ones containing the other meters. The others will have only one element in their zone, as in the Figure below.



Define A Circuit – Meter Element

EnergyMeter Object

-- Sampling

The sampling algorithms are as follows:

- **Local Energy and Power Values:** Simply compute the power into the terminal on which the meter is installed and integrate using the interval between the present solution and the previous solution. This operation uses the voltage and current computed from the present solution.
- **Losses in Zone:** Accumulate the kW losses in each power delivery element in the zone.
- **Load in Zone:** While sampling the losses in each power delivery element, accumulate the power in all loads connected to the downline bus(es) of the element.
- **Overload Energy in Zone:** For each power delivery element in the zone, compute the amount of power exceeding the rating of the element compared to both normal and emergency ratings.
- **EEN and UE in Zone:** For each load in the zone marked as exceeding normal or unserved, compute the present power. Integrate to get energies.

* EEN: Energy Exceeding Normal, UE: Unserved Energy

214

Define A Circuit – Meter Element

EnergyMeter Object

-- EEN and UE Definitions

- EEN (Energy Exceeding Normal) refers to load energy considered unserved because the current or voltage exceeds Normal ratings. UE (Unserved Energy) refers to load energy considered unserved because the power (actually the current) exceeds Emergency, or maximum, ratings.
- In radial systems (default), a load is marked as unserved with respect to either normal or emergency ratings if either:
 - The voltage at the load bus is below minimum ratings; the current in any power delivery element supplying the load exceeds the current ratings.
 - Either the entire load or just the portion above rating at the bus that is considered unserved is counted as unserved, depending on whether the Excess option or the Total option have been specified.

Define A Circuit – Meter Element

EnergyMeter Object

-- Properties

Element = Name of an existing circuit element to which the monitor is to be connected. Note that there may be more than one circuit element with the same name (not wise, but it is allowed). The monitor will be placed at the first one found in the list.

Terminal = No. of the terminal to which the monitor will be connected, normally the source end.

Action = Optional action to execute.

Clear = reset all registers to zero

Save = Saves (appends) the present register values to a file. File name is MTR_metername.CSV, where metername is the name of the energy meter.

Take = Takes a sample at the present solution.

Option = Options: Enter a string ARRAY of any combination of the following. Options processed left-to-right:

(E)xcess : (default) UE/EEN is estimate of only energy exceeding capacity

(T)otal : UE/EEN is total energy after capacity exceeded.

(R)adial : (default) Treats zone as a radial circuit

...

Define A Circuit – Meter Element

EnergyMeter Object

-- Properties

...

(M)esh : Treats zone as meshed network (not radial).

(C)ombined: (default) Load UE or EEN are computed from both overload and undervoltage criteria.

(V)oltage: Load UE/EEN are computed only from the undervoltage criteria.

Example: option=(E, R, C)

In a meshed network, the overload registers represent the total of the power delivery element overloads and the load UE/EEN registers will contain only those loads that are "unserved", which are those with low voltages. In a radial circuit, the overload registers record the max overload (absolute magnitude, not percent) in the zone. Loads become unserved either with low voltage or if any line in their path to the source is overloaded.

KWNorm = Upper limit on kW load in the zone, Normal configuration. Default is 0.0 (ignored). If specified, overrides limits on individual lines for overload EEN. KW above this limit for the entire zone is considered EEN.

...

217

Define A Circuit – Meter Element

EnergyMeter Object

-- Properties

...

KWEmerg = Upper limit on kW load in the zone, Emergency configuration. Default is 0.0 (ignored). If specified, overrides limits on individual lines for overload UE. KW above this limit for the entire zone is considered UE.

Peakcurrent = ARRAY of current magnitudes representing the peak currents measured at this location for the load allocation function (for loads defined with xfkva=). Default is (400, 400, 400). Enter one current for each phase.

Zonelist = ARRAY of full element names for this meter's zone. Default is for meter to find it's own zone. If specified, DSS uses this list instead. It can access the names in a singlecolumn text file. Examples:

```
Zonelist =[line.L1, transformer.T1, Line.L3]
```

```
Zonelist =(file=branchlist.txt)
```

LocalOnly = {Yes | No} Default is NO. If Yes, meter considers only the monitored element for EEN and UE calcs. Uses whole zone for losses.

...

Define A Circuit – Meter Element

EnergyMeter Object

-- Properties

...

Mask = Mask for adding registers whenever all meters are totalized. Array of floating point numbers representing the multiplier to be used for summing each register from this meter. Default = (1, 1, 1, 1, ...). You only have to enter as many as are changed (positional). Useful when two meters monitor same energy, etc.

Losses = {Yes | No} Default is YES. Compute Zone losses. If NO, then no losses at all are computed.

LineLosses = {Yes | No} Default is YES. Compute Line losses. If NO, then none of the line losses are computed.

XfmrLosses = {Yes | No} Default is YES. Compute Transformer losses. If NO, transformers are ignored in loss calculations.

SeqLosses = {Yes | No} Default is YES. Compute Sequence losses in lines and segregate by line mode losses and zero mode losses.

3PhaseLosses = {Yes | No} Default is YES. Compute Line losses and segregate by 3-phase and other (1- and 2-phase) line losses.

...

Define A Circuit – Meter Element

EnergyMeter Object

-- Properties

...

VbaseLosses = {Yes | No} Default is YES. Compute losses and segregate by voltage base. If NO, then voltage-based tabulation is not reported. Make sure the voltage bases of the buses are assigned BEFORE defining the EnergyMeter to ensure that it will automatically pick up the voltage bases. Or, issue the CalcVoltageBases command after defining the EnergyMeter. Each voltage base has four(4) loss registers: total, line, load, and no-load, respectively. There are sufficient registers to report losses in five (5) different voltage levels.

BaseFreq = Base frequency for ratings.

Enabled = {Yes|No or True|False} Indicates whether the element is enabled.

Like= Name of another EnergyMeter object on which to base this one.

Define A Circuit – Monitor Element

Monitor Object

- A monitor is a circuit element that is connected to a *terminal* of another circuit element. It takes a sample when instructed, recording the *time* and the complex *values* of voltage and current, or power, at all phases.
- In essence, it works like a real power monitor. The data in the file can be converted to CSV form. You may accomplish this by either the **Show Monitor** command or the **Export Monitor** command.
- Monitors may be connected to both power delivery elements and power conversion elements.

The parameters are:

Element = Name of an existing circuit element to which the monitor is to be connected. Note that there may be more than one circuit element with the same name (not wise, but it is allowed). The monitor will be placed at the first one found in the list.

Define A Circuit – Monitor Element

Monitor Object

The parameters are:

...

Terminal = No. of the terminal to which the monitor will be connected.

Mode = An integer bitmask code to describe what it is that the monitor will save. Monitors can save two basic types of quantities: 1) Voltage and current; 2) Power. The Mode codes are defined as follows:

0: Standard mode - V and I, each phase, complex

1: Power each phase, complex (kw and kvars)

2: Transformer taps (connect Monitor to a transformer winding)

3: State variables (connect Montor to a PCElement)

+16: Sequence components: V012, I012

+32: Magnitude only

+64: Positive sequence only or Average of phases, if not 3 phases

For example, Mode=33 (33=32+1) will save the magnitude of the power (kVA) only in each phase. Mode=112 (112=0+16+32+64) saves Positive sequence voltages and currents, magnitudes only.

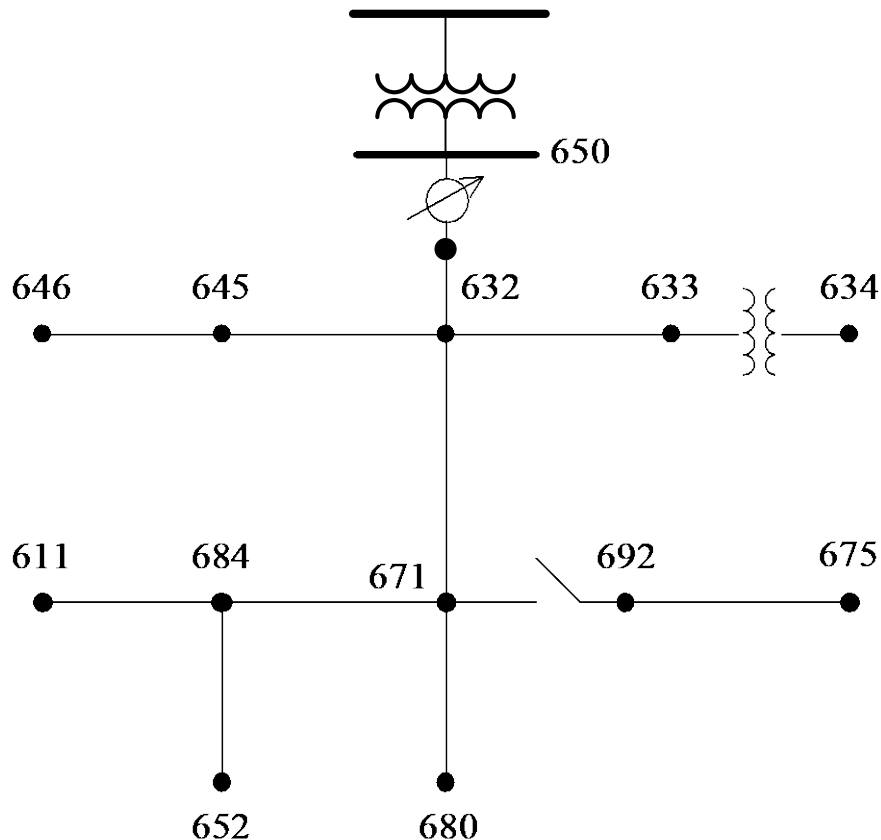
Action = {clear | save} Forces clearing of the monitor's buffer, or saving to disk.

Part II: Three Typical Cases

- Static Snapshot Case
- Time-series Static Case
- Fault and Dynamic Case

Case 1: IEEE 13 Node Test Feeder

Circuit



- 1 voltage source
- 15 buses
- 2 transformers
- 1 voltage regulator
- 1 switch
- 10 lines
- 8 spot loads
- 1 capacitor bank
- 1 1-phase capacitor
- 1 bus coordinate file

Case 1: IEEE 13 Node Test Feeder

Given Parameters

Transformer Data:

	kVA	kV-high	kV-low	R - %	X - %
Substation:	5,000	115 - D	4.16 Gr. Y	1	8
XFM -1	500	4.16 – Gr.W	0.48 – Gr.W	1.1	2

Spot Load Data:

Node	Load Model	Ph-1		Ph-2		Ph-3	
		kW	kVAr	kW	kVAr	kW	kVAr
634	Y-PQ	160	110	120	90	120	90
645	Y-PQ	0	0	170	125	0	0
646	D-Z	0	0	230	132	0	0
652	Y-Z	128	86	0	0	0	0
671	D-PQ	385	220	385	220	385	220
675	Y-PQ	485	190	68	60	290	212
692	D-I	0	0	0	0	170	151
611	Y-I	0	0	0	0	170	80
	TOTAL	1158	606	973	627	1135	753

Regulator Data:

Regulator ID:	1	2	3
Connection:	1-Ph,LG	1-Ph,LG	1-Ph,LG
Monitoring Phase:	A	B	C
Bandwidth:	2.0 volts	2.0 volts	2.0 volts
PT Ratio:	20	20	20
Primary CT Rating:	700	700	700
Compensator Settings:	Ph-A	Ph-B	Ph-C
R - Setting:	3	3	3
X - Setting:	9	9	9
Voltage Level:	122	122	122

Capacitor Data:

Node	Ph		
	Ph-A	Ph-B	Ph-C
	kVAr	kVAr	kVAr
675	200	200	200
611			100
Total	200	200	300

225

Case 1: IEEE 13 Node Test Feeder

Given Parameters

Overhead Line Configuration Data:

Config.	Phasing	Phase	Neutral	Spacing
		ACSR	ACSR	ID
601	B A C N	556,500 26/7	4/0 6/1	500
602	C A B N	4/0 6/1	4/0 6/1	500
603	C B N	1/0	1/0	505
604	A C N	1/0	1/0	505
605	C N	1/0	1/0	510

Underground Line Configuration Data:

Config.	Phasing	Cable	Neutral	Space ID
606	A B C N	250,000 AA, CN	None	515
607	A N	1/0 AA, TS	1/0 Cu	520

Line Segment Data:

Node A	Node B	Length(ft.)	Config.	Phasing
632	645	500	603	B,C
632	633	500	602	A,B,C
633	634	0	XFM-1	A,B,C
645	646	300	603	B,C
RG60	632	2000	601	A,B,C
684	652	800	607	A
632	671	2000	601	A,B,C
671	684	300	604	A,C
671	680	1000	601	A,B,C
671	692	0	Switch	A,B,C
684	611	300	605	C
692	675	500	606	A,B,C

Source Data:

Base kV.	pu	Phasing	Angle	MVAsc3	MVAsc1
606	1.0001	A B C N	30	20,000	21,000

Case 1: IEEE 13 Node Test Feeder

Given Parameters

Configuration 601:

Z (R +jX) in ohms per mile
0.3465 1.0179 0.1560 0.5017 0.1580 0.4236
0.3375 1.0478 0.1535 0.3849
0.3414 1.0348

Configuration 602:

Z (R +jX) in ohms per mile
0.7526 1.1814 0.1580 0.4236 0.1560 0.5017
0.7475 1.1983 0.1535 0.3849
0.7436 1.2112

Configuration 603:

Z (R +jX) in ohms per mile
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
1.3294 1.3471 0.2066 0.4591
1.3238 1.3569

Configuration 604:

Z (R +jX) in ohms per mile
1.3238 1.3569 0.0000 0.0000 0.2066 0.4591
0.0000 0.0000 0.0000 0.0000
1.3294 1.3471

Configuration 605:

Z (R +jX) in ohms per mile
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000
1.3292 1.3475

Configuration 606:

Z (R +jX) in ohms per mile
0.7982 0.4463 0.3192 0.0328 0.2849 -0.0143
0.7891 0.4041 0.3192 0.0328
0.7982 0.4463
B in micro Siemens per mile
96.8897 0.0000 0.0000
96.8897 0.0000
96.8897

Configuration 607:

Z (R +jX) in ohms per mile
1.3425 0.5124 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000
0.0000 0.0000
B in micro Siemens per mile
88.9912 0.0000 0.0000
0.0000 0.0000
0.0000

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model -- Vsource

Source Data:

Base kV.	pu	Phasing	Angle	MVAsc3	MVAsc1
606	1.0001	A B C N	30	20,000	21,000

OpenDSS Code:

```
// define a new circuit
Clear
Set DefaultBaseFrequency=60
new circuit.IEEE13Nodeckt
~ basekv=115 pu=1.0001 phases=3 bus1=SourceBus
~ Angle=30
~ MVAsc3=20000 MVASC1=21000
```

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model -- Vsource

OpenDSS Code:

```
// define a new circuit  
Clear  
Set DefaultBaseFrequency=60  
new circuit.IEEE13Nodeckt  
~ basekv=115 pu=1.0001 phases=3 bus1=SourceBus  
~ Angle=30  
~ MVAsc3=20000 MVASC1=21000
```

Explanation:

// or !

“//” or “!” indicate that this statement is a comment line.

Clear

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

229

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model -- Vsource

OpenDSS Code:

```
// define a new circuit  
Clear  
Set DefaultBaseFrequency=60  
new circuit.IEEE13Nodeckt  
~ basekv=115 pu=1.0001 phases=3 bus1=SourceBus  
~ Angle=30  
~ MVAsc3=20000 MVASC1=21000
```

Explanation:

Set DefaultBaseFrequency

It sets Default Base Frequency, in Hz. The default value when first installed is 60 Hz. {50|60}, Hz

New circuit.IEEE13Nodeckt

It creates a new circuit object named IEEE13Nodeckt.

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model -- Vsource

OpenDSS Code:

```
// define a new circuit
Clear
Set DefaultBaseFrequency=60
new circuit.IEEE13Nodeckt
~ basekv=115 pu=1.0001 phases=3 bus1=SourceBus
~ Angle=30
~ MVAAsc3=20000 MVASC1=21000
```

Explanation:

~

It is a More Command. The More command continues editing the active object. The same as M or More.

basekv (Vsource)

Sets the nominal Line-to-line kV of Vsource.

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model -- Vsource

OpenDSS Code:

```
// define a new circuit
Clear
Set DefaultBaseFrequency=60
new circuit.IEEE13Nodeckt
~ basekv=115 pu=1.0001 phases=3 bus1=SourceBus
~ Angle=30
~ MVAAsc3=20000 MVASC1=21000
```

Explanation:

pu

Actual per unit at which the source is operating. It is assumed balanced for all phases. In general, the voltage range is generally $\pm 5\%$ of the nominal voltage. That means the pu is in the range of 0.95~1.05.

phases

Number of phases. Default = 3.0. {1|2|3}.

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model -- Vsource

OpenDSS Code:

```
// define a new circuit  
Clear  
Set DefaultBaseFrequency=60  
new circuit.IEEE13Nodeckt  
~ basekv=115 pu=1.0001 phases=3 bus1=SourceBus  
~ Angle=30  
~ MVAsc3=20000 MVASC1=21000
```

Explanation:

bus1

Name of bus to which the source's first terminal is connected.

Angle

Base angle, in degrees, of the first phase.

MVAsc3

3-phase short circuit MVA.

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model -- Vsource

OpenDSS Code:

```
// define a new circuit
Clear
Set DefaultBaseFrequency=60
new circuit.IEEE13Nodeckt
~ basekv=115 pu=1.0001 phases=3 bus1=SourceBus
~ Angle=30
~ MVAsc3=20000 MVAsc1=21000
```

Explanation:

MVAsc1

1-phase short circuit MVA.

* MVAsc3 and MVAsc1 are used for determining the positive- and zero-sequence impedances, i.e., $Z1=R1+jX1$ and $Z0=R0+jX0$. In general, they are obtained from system data. Default $x1/r1$ is 4.0 and Default $x0/r0$ is 3.0.

234

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model -- Transformer

Transformer Data:

	kVA	kV-high	kV-low	R - %	X - %
Substation:	5,000	115 - D	4.16 Gr. Y	1	8
XFM -1	500	4.16 - Gr.W	0.48 - Gr.W	1.1	2

OpenDSS Code:

```
// Substation transformer definitions
```

```
New Transformer.Sub Phases=3 Windings=2 XHL=8
```

```
~ wdg=1 bus=SourceBus conn=delta kv=115 kva=5000 %r=0.5
```

```
~ wdg=2 bus=650 conn=wye kv=4.16 kva=5000 %r=0.5
```

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model -- Transformer

OpenDSS Code:

```
// Substation transformer definitions
New Transformer.Sub Phases=3 Windings=2 XHL=8
~ wdg=1 bus=SourceBus conn=delta kv=115 kva=5000 %r=0.5
~ wdg=2 bus=650          conn=wye kv=4.16 kva=5000 %r=0.5
```

Explanation:

New Transformer.Sub

Defines a transformer object named Sub.

Phases

Number of phases. Default is 3. {1|3}

Windings

Number of windings. Default is 2. {2|3|4}

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model -- Transformer

OpenDSS Code:

```
// Substation transformer definitions
```

```
New Transformer.Sub Phases=3 Windings=2 XHL=8
```

```
~ wdg=1 bus=SourceBus conn=delta kv=115 kva=5000 %r=0.5
```

```
~ wdg=2 bus=650          conn=wye kv=4.16 kva=5000 %r=0.5
```

	kVA	kV-high	kV-low	R - %	X - %
Substation:	5,000	115 - D	4.16 Gr. Y	1	8

Explanation:

XHL (or X12)

Percent reactance high-to-low (winding1 to winding 2).

Wdg

It specifies an integer representing the winding which will become the active winding for subsequent data.

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model -- Transformer

OpenDSS Code:

```
// Substation transformer definitions
New Transformer.Sub Phases=3 Windings=2 XHL=8
~ wdg=1 bus=SourceBus conn=delta kv=115 kva=5000 %r=0.5
~ wdg=2 bus=650          conn=wye kv=4.16 kva=5000 %r=0.5
```

Explanation:

bus

It specifies the name of bus to which the winding is connected.

conn

It Specifies the connection of this winding. One of {wye | ln} for wye connected banks or {delta | ll} for delta (line-line) connected banks. Default is wye.

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model -- Transformer

OpenDSS Code:

```
// Substation transformer definitions
New Transformer.Sub Phases=3 Windings=2 XHL=8
~ wdg=1 bus=SourceBus conn=delta kv=115 kva=5000 %r=0.5
~ wdg=2 bus=650          conn=wye kv=4.16 kva=5000 %r=0.5
```

Explanation:

kv

It specifies the rated voltage of this winding, in kV. For transformers designated 2- or 3-phase, enter phase-to-phase kV. For all other designations, enter actual winding kV rating. Default is 12.47 kV.

kva

It specifies the base kVA rating of this winding.

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model -- Transformer

OpenDSS Code:

```
// Substation transformer definitions
New Transformer.Sub Phases=3 Windings=2 XHL=8
~ wdg=1 bus=SourceBus conn=delta kv=115 kva=5000 %r=0.5
~ wdg=2 bus=650          conn=wye kv=4.16 kva=5000 %r=0.5
```

Explanation:

%r

It specifies the percent resistance of this winding on the rated kVA base. (It equals half of total for a 2-winding transformer).

* Transformer parameters are obtained from transformer nameplate.

	kVA	kV-high	kV-low	R - %	X - %
Substation:	5,000	115 - D	4.16 Gr. Y	1 * 0.5	8

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model -- Regulator

Voltage Regulator Data:

Regulator ID:	1	2	3
Connection:	1-Ph,LG	1-Ph,LG	1-Ph,LG
Monitoring Phase:	A	B	C
Bandwidth:	2.0 volts	2.0 volts	2.0 volts
PT Ratio:	20	20	20
Primary CT Rating:	700	700	700
Compensator Settings:	Ph-A	Ph-B	Ph-C
R - Setting:	3	3	3
X - Setting:	9	9	9
Voltage Level:	122	122	122

OpenDSS Code:

```
// Voltage Regulator Definitions
New Transformer.Reg1 phases=1 bank=reg1 XHL=0.01 kVAs=[1666 1666]
~ Buses=[650.1 RG60.1] kVs=[2.4 2.4]
new regcontrol.Reg1 transformer=Reg1 winding=2 vreg=122 band=2 ptratio=20 ctprim=700 R=3 X=9

New Transformer.Reg2 phases=1 bank=reg1 XHL=0.01 kVAs=[1666 1666]
~ Buses=[650.2 RG60.2] kVs=[2.4 2.4]
new regcontrol.Reg2 transformer=Reg2 winding=2 vreg=122 band=2 ptratio=20 ctprim=700 R=3 X=9

New Transformer.Reg3 phases=1 bank=reg1 XHL=0.01 kVAs=[1666 1666]
~ Buses=[650.3 RG60.3] kVs=[2.4 2.4]
new regcontrol.Reg3 transformer=Reg3 winding=2 vreg=122 band=2 ptratio=20 ctprim=700 R=3 X=9
```

241

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model -- Regulator

OpenDSS Code:

```
// Voltage regulator and control definitions
New Transformer.Reg1 phases=1 bank=reg1 XHL=0.01 kVAs=[1666 1666]
~ Buses=[650.1 RG60.1] kVs=[2.4 2.4]
new regcontrol.Reg1 transformer=Reg1 winding=2 vreg=122 band=2 ptratio=20 ctprim=700 R=3 X=9
```

Explanation:

Note that voltage regulator is defined using the class of transformer. Different with the transformer, additional RegControl Object should be defined for each regulator.

New Transformer.Reg1

Defines a regulator named Reg1.

phases

Specifies Number of phases.

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model -- Regulator

OpenDSS Code:

```
// Voltage regulator and control definitions
New Transformer.Reg1 phases=1 bank=reg1 XHL=0.01 kVAs=[1666 1666]
~ Buses=[650.1 RG60.1] kVs=[2.4 2.4]
new regcontrol.Reg1 transformer=Reg1 winding=2 vreg=122 band=2 ptratio=20 ctprim=700 R=3 X=9
```

Explanation:

bank

Specifies the name of the bank this regulator is part of.

XHL (or X12)

Specifies the percent reactance high-to-low (winding 1 to winding 2).

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model -- Regulator

OpenDSS Code:

```
// Voltage regulator and control definitions
New Transformer.Reg1 phases=1 bank=reg1 XHL=0.01 kVAs=[1666 1666]
~ Buses=[650.1 RG60.1] kVs=[2.4 2.4]
new regcontrol.Reg1 transformer=Reg1 winding=2 vreg=122 band=2 ptratio=20 ctprim=700 R=3 X=9
```

Explanation:

kVAs=[1666 1666]

Specifies an array of base kVA ratings for windings [1, 2]. The kVA rating of winding 1 is 1666kVA, the kVA rating of winding 2 is 1666 kVA.

* Note that compared with the aforementioned way of defining the properties of transformer windings, i.e., specifying the properties for each winding, individually, this is another way of defining the winding properties, i.e., using property arrays.

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model -- Regulator

OpenDSS Code:

```
// Voltage regulator and control definitions
New Transformer.Reg1 phases=1 bank=reg1 XHL=0.01 kVAs=[1666 1666]
~ Buses=[650.1 RG60.1] kVs=[2.4 2.4]
new regcontrol.Reg1 transformer=Reg1 winding=2 vreg=122 band=2 ptratio=20 ctpri=700 R=3 X=9
```

Explanation:

Buses=[650.1 RG60.1]

Specifies an array of the names of buses to which the windings [1, 2] are connected. Winding 1 is connected to Bus 650, winding 2 is connected to Bus RG60.

kVs=[2.4 2.4]

Specifies an array of kV ratings for windings [1,2]. The kV rating of winding 1 is 2.4 kV, the kV rating of winding 2 is 2.4 kV.

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model -- Regulator

OpenDSS Code:

```
// Voltage regulator and control definitions
New Transformer.Reg1 phases=1 bank=reg1 XHL=0.01 kVAs=[1666 1666]
~ Buses=[650.1 RG60.1] kVs=[2.4 2.4]
new regcontrol.Reg1 transformer=Reg1 winding=2 vreg=122 band=2 ptratio=20 ctprim=700 R=3 X=9
```

Explanation:

new regcontrol.Reg1

Defines a regulator control object named Reg1.

transformer

Specifies the name of transformer to which the RegControl is connected.

winding

Specifies the number of transformer winding that the RegControl is monitoring. 1 or 2, typically.

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model -- Regulator

OpenDSS Code:

```
// Voltage regulator and control definitions
New Transformer.Reg1 phases=1 bank=reg1 XHL=0.01 kVAs=[1666 1666]
~ Buses=[650.1 RG60.1] kVs=[2.4 2.4]
new regcontrol.Reg1 transformer=Reg1 winding=2 vreg=122 band=2 ptratio=20 ctprim=700 R=3 X=9
```

Explanation:

vreg

Specifies the voltage regulator setting, in volts, for the winding being controlled.

band

Specifies the bandwidth, in volts, for the controlled bus. Default is 3.0.

ptratio

Specifies the ratio of the PT that converts the controlled winding voltage to the regulator voltage. Default is 60. If the winding is Wye, the line-to-neutral voltage is used. Else, the line-to-line voltage is used.

247

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model -- Regulator

OpenDSS Code:

```
// Voltage regulator and control definitions
New Transformer.Reg1 phases=1 bank=reg1 XHL=0.01 kVAs=[1666 1666]
~ Buses=[650.1 RG60.1] kVs=[2.4 2.4]
new regcontrol.Reg1 transformer=Reg1 winding=2 vreg=122 band=2 ptratio=20 ctprim=700 R=3 X=9
```

Explanation:

ctprim

Specifies the rating, in Amperes, of the primary CT rating for converting the line amps to control amps. The typical default secondary ampere rating is 0.2 Amps.

R

Specifies the R setting on the line drop compensator in the regulator, in VOLTS.

X

Specifies the X setting on the line drop compensator in the regulator, in VOLTS.

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model – Line Configuration

Line Configuration Data:

Configuration 606:

```
      Z (R +jX) in ohms per mile
0.7982  0.4463   0.3192  0.0328   0.2849 -0.0143
          0.7891  0.4041   0.3192  0.0328
                    0.7982  0.4463

      B in micro Siemens per mile
      96.8897   0.0000   0.0000
          96.8897   0.0000
                96.8897
```

Configuration 607:

```
      Z (R +jX) in ohms per mile
1.3425  0.5124   0.0000  0.0000   0.0000  0.0000
          0.0000  0.0000   0.0000  0.0000
                    0.0000  0.0000

      B in micro Siemens per mile
      88.9912   0.0000   0.0000
          0.0000   0.0000
                0.0000
```

OpenDSS Code:

```
// Linecode definitions
New Linecode.mtx606 nphases=3 Units=mi
~ Rmatrix=[0.7982 |0.3192 0.7891 |0.2849 0.3192 0.7982 ]
~ Xmatrix=[0.4463 |0.0328 0.4041 |-0.0143 0.0328 0.4463 ]
~ Cmatrix=[257.01 |0 257.01 |0 0 257.01 ]
~ units=mi

New linecode.mtx607 nphases=1 BaseFreq=60
~ rmatrix = (1.3425 )
~ xmatrix = (0.5124 )
~ cmatrix = [236.06]
~ units=mi
```

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model – Line configuration

OpenDSS Code:

```
// Linecode definitions
New Linecode.mtx606 nphases=3 BaseFreq=60
~ Rmatrix=[0.7982 |0.3192 0.7891 |0.2849 0.3192 0.7982 ]
~ Xmatrix=[0.4463 |0.0328 0.4041 |-0.0143 0.0328 0.4463 ]
~ Cmatrix=[257.01 |0 257.01 |0 0 257.01 ]
~ units=mi
```

Explanation:

New linecode.mtx606

Defines a linecode object named mtx606.

nphases

Specifies the number of phases. Default = 3.

BaseFreq

Specifies the Base Frequency at which the impedance values are specified. {50 | 60},
Hz, Default = 60.0 Hz.

250

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model – Line configuration

OpenDSS Code:

```
// Linecode definitions
New Linecode.mtx606 nphases=3 BaseFreq=60
~ Rmatrix=[0.7982 |0.3192 0.7891 |0.2849 0.3192 0.7982 ]
~ Xmatrix=[0.4463 |0.0328 0.4041 |-0.0143 0.0328 0.4463 ]
~ Cmatrix=[257.01 |0 257.01 |0 0 257.01 ]
~ units=mi
```

Explanation:

Rmatrix

Defines the series resistance matrix, in ohms per unit length. The order of matrices expected is the number of phases. The matrices may be entered in lower triangle form or full matrix. The result is always symmetrical.

The "|" separates rows.

Xmatrix

Similar with Rmatrix, Xmatrix is a series reactance matrix, in ohms per unit length.

251

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model – Line configuration

OpenDSS Code:

```
// Linecode definitions
New Linecode.mtx606 nphases=3 BaseFreq=60
~ Rmatrix=[0.7982 |0.3192 0.7891 |0.2849 0.3192 0.7982 ]
~ Xmatrix=[0.4463 |0.0328 0.4041 |-0.0143 0.0328 0.4463 ]
~ Cmatrix=[257.01 |0 257.01 |0 0 257.01 ]
~ units=mi
```

Explanation:

Cmatrix

Defines the shunt nodal capacitance matrix, in nanofarads per unit length. The order of matrices expected is the number of phases. The matrices may be entered in lower triangle form or full matrix. The result is always symmetrical.

The "|" separates rows.

Units

Specifies the length units of linecode. If not specified, it is assumed that the units correspond to the length being used in the Line models. {mi|kft|km|m|Ft|in|cm}. 252

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model – Line

Line Segment Data:

Node A	Node B	Length(ft.)	Config.	Phasing
632	645	500	603	B,C
632	633	500	602	A,B,C
645	646	300	603	B,C
RG60	632	2000	601	A,B,C
684	652	800	607	A
632	671	2000	601	A,B,C
671	684	300	604	A,C
671	680	1000	601	A,B,C
684	611	300	605	C
692	675	500	606	A,B,C

OpenDSS Code:

```
// Line definitions
New Line.632645 Phases=2 Bus1=632.3.2 Bus2=645.3.2 LineCode=mtx603 Length=500 units=ft
New Line.632633 Phases=3 Bus1=632.1.2.3 Bus2=633.1.2.3 LineCode=mtx602 Length=500 units=ft
New Line.645646 Phases=2 Bus1=645.3.2 Bus2=646.3.2 LineCode=mtx603 Length=300 units=ft
New Line.RG60632 Phases=3 Bus1=RG60.1.2.3 Bus2=632.1.2.3 LineCode=mtx601 Length=2000 units=ft
New Line.684652 Phases=1 Bus1=684.1 Bus2=652.1 LineCode=mtx607 Length=800 units=ft
New Line.632671 Phases=3 Bus1=632.1.2.3 Bus2=671.1.2.3 LineCode=mtx601 Length=2000 units=ft
New Line.671684 Phases=2 Bus1=671.1.3 Bus2=684.1.3 LineCode=mtx604 Length=300 units=ft
New Line.671680 Phases=3 Bus1=671.1.2.3 Bus2=680.1.2.3 LineCode=mtx601 Length=1000 units=ft
New Line.684611 Phases=1 Bus1=684.3 Bus2=611.3 LineCode=mtx605 Length=300 units=ft
New Line.692675 Phases=3 Bus1=692.1.2.3 Bus2=675.1.2.3 LineCode=mtx606 Length=500 units=ft
```

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model – Line

OpenDSS Code:

```
// Line definitions  
New Line.632645 Phases=2 Bus1=632.3.2 Bus2=645.3.2 LineCode=mtx603 Length=500 units=ft
```

Explanation:

New Line.650632

Defines a line object named 650632.

Phases

Specifies the number of phases. Default = 3.

Bus1

Specifies the name of bus for terminal 1.

Bus2

Specifies the name of bus for terminal 2.

LineCode

Specifies the name of an existing LineCode object containing impedance definitions.

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model – Line

OpenDSS Code:

```
// Line definitions
```

```
New Line.632645 Phases=2 Bus1=632.3.2 Bus2=645.3.2 LineCode=mtx603 Length=500 units=ft
```

Explanation:

Length

Specifies a length multiplier to be applied to the impedance data.

Units

Specifies the length Units = {none | mi | kft | km | m | Ft | in | cm}.

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model – Load

Spot Load Data:

Node	Load	Ph-1	Ph-1	Ph-2	Ph-2	Ph-3	Ph-3
	Model	kW	kVAr	kW	kVAr	kW	kVAr
634	Y-PQ	160	110	120	90	120	90
645	Y-PQ	0	0	170	125	0	0
646	D-Z	0	0	230	132	0	0
652	Y-Z	128	86	0	0	0	0
671	D-PQ	385	220	385	220	385	220
675	Y-PQ	485	190	68	60	290	212
692	D-I	0	0	0	0	170	151
611	Y-I	0	0	0	0	170	80

OpenDSS Code:

// Load definitions

```
New Load.634a Bus1=634.1 Phases=1 Conn=Wye Model=1 kV=0.277 kW=160 kvar=110
New Load.634b Bus1=634.2 Phases=1 Conn=Wye Model=1 kV=0.277 kW=120 kvar=90
New Load.634c Bus1=634.3 Phases=1 Conn=Wye Model=1 kV=0.277 kW=120 kvar=90
New Load.645 Bus1=645.2 Phases=1 Conn=Wye Model=1 kV=2.4 kW=170 kvar=125
New Load.646 Bus1=646.2.3 Phases=1 Conn=Delta Model=2 kV=4.16 kW=230 kvar=132
New Load.652 Bus1=652.1 Phases=1 Conn=Wye Model=2 kV=2.4 kW=128 kvar=86
New Load.671 Bus1=671.1.2.3 Phases=3 Conn=Delta Model=1 kV=4.16 kW=1155 kvar=660
New Load.675a Bus1=675.1 Phases=1 Conn=Wye Model=1 kV=2.4 kW=485 kvar=190
New Load.675b Bus1=675.2 Phases=1 Conn=Wye Model=1 kV=2.4 kW=68 kvar=60
New Load.675c Bus1=675.3 Phases=1 Conn=Wye Model=1 kV=2.4 kW=290 kvar=212
New Load.692 Bus1=692.3.1 Phases=1 Conn=Delta Model=5 kV=4.16 kW=170 kvar=151
New Load.611 Bus1=611.3 Phases=1 Conn=Wye Model=5 kV=2.4 kW=170 kvar=80
```


Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model – Load

OpenDSS Code:

```
// Load definitions  
New Load.634a Bus1=634.1 Phases=1 Conn=Wye Model=1 kV=0.277 kW=160 kvar=110
```

Explanation:

New Load.634a

Defines a load object named 634a.

Bus1

Specifies the name of bus to which the load is connected.

Phases

Specifies the number of phases of this load.

Conn

{wye | y | LN} for Wye (Line-Neutral) connection; {delta | LL} for Delta (Line-Line) connection. Default = wye.

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model – Load

OpenDSS Code:

```
// Load definitions  
New Load.634a Bus1=634.1 Phases=1 Conn=Wye Model=1 kV=0.277 kW=160 kvar=110
```

Explanation:

Model

An integer defining how the load will vary with voltage. The load models currently implemented are:

- 1: Constant P and constant Q (Default): Commonly used for power flow studies,
- 2: Constant Z (or constant impedance),
- 3: Constant P and quadratic Q ($Q/Q_0 = (V/V_0)^2$),
- 4: Exponential: $P/P_0 = (V/V_0)^{CVRwatts}$ and $Q/Q_0 = (V/V_0)^{CVRvars}$
- 5: Constant I (or constant current magnitude),
- 6: Constant P and fixed Q (at the nominal value)
- 7: Constant P and quadratic Q (i.e., fixed reactance),
- 8: ZIP
- ...

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model – Load

OpenDSS Code:

```
// Load definitions  
New Load.634a Bus1=634.1 Phases=1 Conn=Wye Model=1 kV=0.277 kW=160 kvar=110
```

Explanation:

Model

...

"Constant" power value (either P or Q) may be modified by loadshape multipliers.

"Fixed" power values are always the same.

- ZIP model:
$$P = P_0 \left[a_1 \left(\frac{V}{V_0} \right)^2 + a_2 \left(\frac{V}{V_0} \right) + a_3 \right]$$
$$Q = Q_0 \left[a_4 \left(\frac{V}{V_0} \right)^2 + a_5 \left(\frac{V}{V_0} \right) + a_6 \right]$$

kV

Specifies the Base voltage for load. For 2- or 3-phase loads, it is specified in phase-to-phase kV.

For all other loads, it is specified in the actual kV across the load branch. If the load is wye (star) connected, then specify phase-to-neutral (L-N) kV. If the load is delta or phase-to-phase connected, specify the phase-to-phase (L-L) kV.

259

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model – Load

OpenDSS Code:

```
// Load definitions  
New Load.634a Bus1=634.1 Phases=1 Conn=Wye Model=1 kV=0.277 kW=160 kvar=110
```

Explanation:

kW

Specifies the nominal active power, in kW, for the load. Total of all phases.

kVar

Specifies the nominal reactive power, in kVar, for the load. Total of all phases.

*** If there is no specification, the kW and kVar are divided evenly in each phase. If the load is unbalanced, the load should be defined for each phase.**

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model – Capacitor

Capacitor Data:

Node	Ph-A	Ph-B	Ph-C
	kVAr	kVAr	kVAr
675	200	200	200
611			100

OpenDSS Code:

```
// Capacitor definitions
```

```
New Capacitor.Cap1 Bus1=675 phases=3 kVAR=600 kV=4.16
```

```
New Capacitor.Cap2 Bus1=611.3 phases=1 kVAR=100 kV=2.4
```

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model – Capacitor

OpenDSS Code:

```
// Capacitor definitions  
New Capacitor.Cap1 Bus1=675 phases=3 kVAR=600 kV=4.16  
New Capacitor.Cap2 Bus1=611.3 phases=1 kVAR=100 kV=2.4
```

Explanation:

New Capacitor.Cap1

Define a capacitor object named Cap1.

Bus1

Specifies the name of the first bus to which the capacitor is connected.

phases

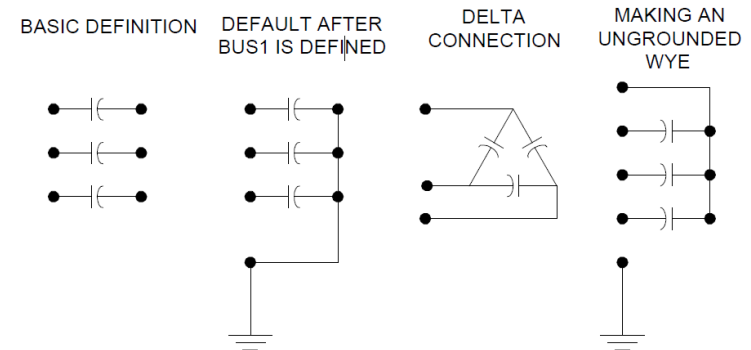
Specifies the number of phases.

kVAR

Specifies the rated kvar at rated kV, total of all phases. Each phase is assumed equal.

kV

For 2- or 3-phase, enter line-to-line rated voltage. For 1-phase, enter line-to-ground rated voltage. (For Delta connection this is always line-to-line rated voltage). 262



Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model – Switch

Switch Data:

Node A	Node B	Length(ft.)	Config.	Phasing
671	692	0	Switch	A,B,C

OpenDSS Code:

```
// Switch definitions  
New Line.671692 Phases=3 Bus1=671 Bus2=692 Switch=y r1=1e-  
4 r0=1e-4 x1=0.000 x0=0.000 c1=0.000 c0=0.000
```

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model – Switch

OpenDSS Code:

```
// Switch definitions
```

```
New Line.671692 Phases=3 Bus1=671 Bus2=692 Switch=y r1=1e-4 r0=1e-4 x1=0.000 x0=0.000 c1=0.000 c0=0.000
```

Explanation:

New Line.671692

Defines a line object named 671692.

Phases

Specifies the number of phases.

Bus1

Specifies the name of bus for terminal 1.

Bus2

Specifies the name of bus for terminal 2.

Switch

Specifies whether this line is a switch or not. {y/n | T/F}. Default= no/false.

y or T: Designates this line as a switch; n or F: Does not designate this line as a switch.

*** In OpenDss, a switch object is defined as a special line object.**

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model – Switch

OpenDSS Code:

```
// Switch definitions
```

```
New Line.671692 Phases=3 Bus1=671 Bus2=692 Switch=y r1=1e-4 r0=1e-4 x1=0.000 x0=0.000 c1=0.000 c0=0.000
```

Explanation:

r1

Specifies the positive-sequence resistance, in ohms per unit length.

r0

Specifies the zero-sequence resistance, in ohms per unit length.

x1

Specifies the positive-sequence reactance, in ohms per unit length.

x0

Specifies the zero-sequence reactance, in ohms per unit length.

c1

Specifies the positive-sequence capacitance, in nanofarads per unit length.

c0

Specifies the zero-sequence capacitance, in nanofarads per unit length.

265

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model – Solve

OpenDSS Code:

```
// Solve
Set Voltagebases=[115, 4.16, .48]
calcv
Set mode=snap
Set algorithm=normal
Solve
BusCoords IEEE13Node_BusXY.csv
```

Explanation:

Set Voltagebases

Defines the bus voltage bases for the defined circuit. You can enter an array of the legal voltage bases, in phase-to-phase voltages.

- The voltage level can be selected from the following values: .208, .480, 12.47, 24.9, 34.5, 115.0, 230.0

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model – Solve

OpenDSS Code:

```
// Solve
Set Voltagebases=[115, 4.16, .48]
calcv
Set mode=snap
Set algorithm=normal
Solve
BusCoords IEEE13Node_BusXY.csv
```

Explanation:

calcv

Also called CalcVoltageBases.

Estimates the voltage base for each bus based on the array of voltage bases defined with a "Set Voltagebases=..." command. The voltage base for each bus is then set to the nearest voltage base specified in the voltage base array.

Set mode=snap

Specifies the solution mode for the active circuit.

Snap = Solve a single snapshot power flow for the present conditions.

267

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model – Solve

OpenDSS Code:

```
// Solve
Set Voltagebases=[115, 4.16, .48]
calcv
Set mode=snap
Set algorithm=normal
Solve
BusCoords IEEE13Node_BusXY.csv
```

Bus Name	X	Y
SourceBus	200	400
650	200	350
RG60	200	300
646	0	250
645	100	250
632	200	250
633	350	250
634	400	250
611	0	100
684	100	100
671	200	100
692	250	100
675	400	100
652	100	0
680	200	0

Explanation:

Set algorithm=normal

Specifies the solution algorithm type. {Normal | Newton}

Solve

Solves the circuit. Default mode is snapshot.

Buscoords

Defines x,y coordinates for buses via reading coordinates from a CSV file.

It executes after the Solve command is executed, so that bus lists are defined.

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model – Show

OpenDSS Code:

```
// Show  
Show Voltages LN Nodes  
Show Currents Elem  
Show Powers kVA Elem  
Show Losses  
Show Taps
```

Explanation:

The Show command 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.

Show Voltages LN Nodes

Shows the line-to-ground voltages by node. (Report)

Show Currents Elem

Shows currents of circuit elements. (Report)

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model – Show

OpenDSS Code:

```
// Show  
Show Voltages LN Nodes  
Show Currents Elem  
Show Powers kVA Elem  
Show Losses  
Show Taps
```

Explanation:

Show Powers kVA Elem

Shows powers of circuit elements in kVA. (Report)

Show Losses

Shows losses of power delivery elements. (Report)

Show Taps

Shows the controlled transformer's tap positions. (Report)

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model – Visualization

OpenDSS Code:

```
// Visualization  
Visualize what=currents element=Line.632633  
Visualize what=voltages element=Line.632633
```

Explanation:

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

What=currents

Specifies that currents of circuit elements are visualized, in amps.

element=Line.632633

Specifies which element are visualized.

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model – Plot

OpenDSS Code:

```
// Plot
Set markCapacitors=yes CapMarkersize=3
Set markRegulators=yes RegMarkersize=5
AddBusMarker Bus=632 code=34 color=Red size=3
Plot Circuit Power Max=2000 dots=y labels=n C1=Blue 1ph=3
Plot circuit Losses Max=0.02 dots=y labels=n subs=n C1=Blue
ClearBusMarkers
```

Explanation:

Set markCapacitors=yes

Marks Capacitor object locations with a symbol. {YES/TRUE | NO/FALSE}
Default is NO.

Set CapMarkersize=3

Specifies the size of Capacitor marker on the plot. Default is 3.

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model – Plot

OpenDSS Code:

```
// Plot
Set markCapacitors=yes CapMarkersize=3
Set markRegulators=yes RegMarkersize=5
AddBusMarker Bus=632 code=34 color=Red size=3
Plot Circuit Power Max=2000 dots=y labels=n C1=Blue 1ph=3
Plot circuit Losses Max=0.02 dots=y labels=n subs=n C1=Blue
ClearBusMarkers
```

Explanation:

AddBusMarker

Adds a marker to a bus in a circuit plot before issuing the Plot command.

Bus=632

Specifies the bus to which the marker is added.

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model – Plot

OpenDSS Code:

```
// Plot
Set markCapacitors=yes CapMarkersize=3
Set markRegulators=yes RegMarkersize=5
AddBusMarker Bus=632 code=34 color=Red size=3
Plot Circuit Power Max=2000 dots=y labels=n C1=Blue 1ph=3
Plot circuit Losses Max=0.02 dots=y labels=n subs=n C1=Blue
ClearBusMarkers
```

Explanation:

Code=34

Specifies the Number code for bus marker on circuit plots. Marker codes are:

0	.	10	•	20	^	30	▼	40	◄
1	•	11	◻	21	^	31	▼	41	◄◄
2	+	12	◻	22	∨	32	▽	42	◄△
3	+	13	•	23	∨	33	▽	43	◄◄
4	*	14	•	24	●	34	▼	44	◄◄
5	×	15	◆	25	×	35	△	45	◄◄
6	×	16	◊	26	•	36	▲	46	◄◄
7	•	17	◊	27	◊	37	⊥	47	◄◄
8	■	18	■	28	•	38	±		
9	■	19	◊	29	▼	39	⊕		

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model – Plot

OpenDSS Code:

```
// Plot
Set markCapacitors=yes CapMarkersize=3
Set markRegulators=yes RegMarkersize=5
AddBusMarker Bus=632 code=34 color=Red size=3
Plot Circuit Power Max=2000 dots=y labels=n C1=Blue 1ph=3
Plot circuit Losses Max=0.02 dots=y labels=n subs=n C1=Blue
ClearBusMarkers
```

Explanation:

color=red

Specifies the color of the bus marker on circuit plots.

size=3

Specifies the size of the bus marker on circuit plots.

plot

Displays a variety of results in a variety of manners on graphs.

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model – Plot

OpenDSS Code:

```
// Plot
Set markCapacitors=yes CapMarkersize=3
Set markRegulators=yes RegMarkersize=5
AddBusMarker Bus=632 code=34 color=Red size=3
Plot Circuit Power Max=2000 dots=y labels=n C1=Blue 1ph=3
Plot circuit Losses Max=0.02 dots=y labels=n subs=n C1=Blue
ClearBusMarkers
```

Explanation:

Circuit

Creates displays on which the thicknesses of LINE are varied according to: Power, Current, Voltage or Losses.

Max=2000

Specifies the value corresponding to max scale or line thickness on the circuit plots. Power and Losses in kW.

dots=y

Specifies whether to show buses as dots or not. {y | n}

276

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model – Plot

OpenDSS Code:

```
// Plot
Set markCapacitors=yes CapMarkersize=3
Set markRegulators=yes RegMarkersize=5
AddBusMarker Bus=632 code=34 color=Red size=3
Plot Circuit Power Max=2000 dots=y labels=n C1=Blue 1ph=3
Plot circuit Losses Max=0.02 dots=y labels=n subs=n C1=Blue
ClearBusMarkers
```

Explanation:

labels=n

Specifies whether to show bus names or not. {y | n}

C1=Blue

Specifies the unicolor on circuit plots. Colors may be specified by their RGB color number or standard color names.

1ph=3

Specifies the Line style for drawing 1-phase lines. It is a number in the range of [1..7]. Default is 1 (solid). Use 3 for dotted; 2 for dashed.

277

Case 1: IEEE 13 Node Test Feeder

Build OpenDSS Model – Plot

OpenDSS Code:

```
// Plot
Set markCapacitors=yes CapMarkersize=3
Set markRegulators=yes RegMarkersize=5
AddBusMarker Bus=632 code=34 color=Red size=3
Plot Circuit Power Max=2000 dots=y labels=n C1=Blue 1ph=3
Plot circuit Losses Max=0.02 dots=y labels=n subs=n C1=Blue
ClearBusMarkers
```

Explanation:

subs=n

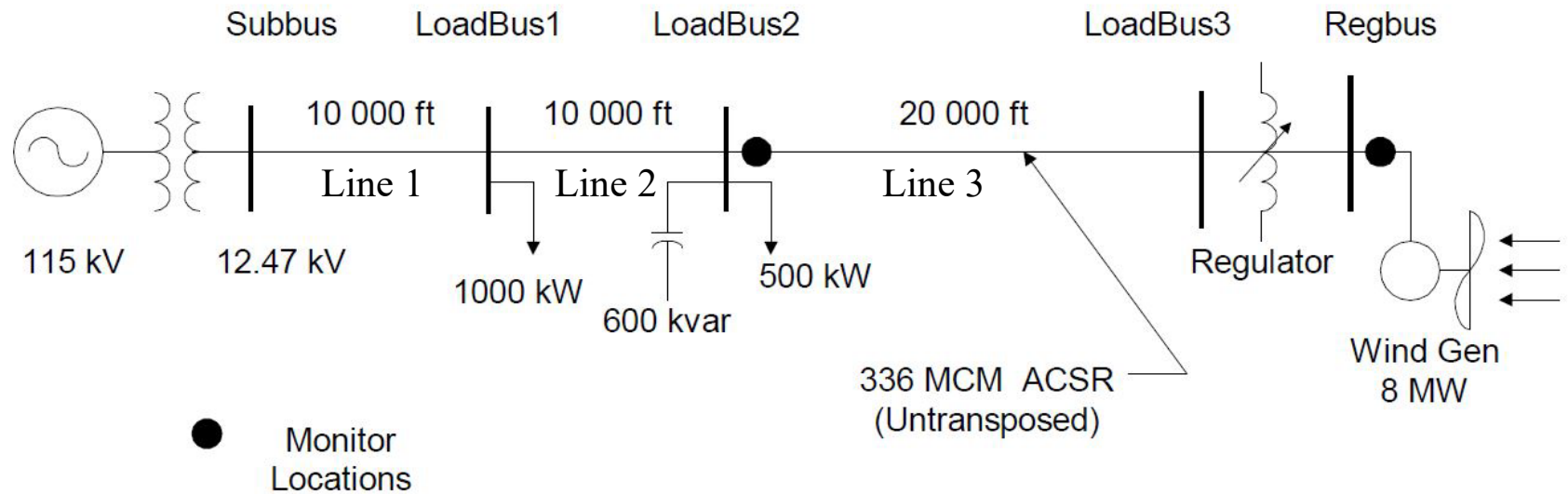
Specifies whether to display a marker at each transformer declared to be a substation. At least one bus coordinate must be defined for the transformer. {Yes | No*}

ClearBusMarkers

Clears the present definitions of bus markers.

Case II: Time-series Power Flow

Circuit



- 1 voltage source
- 1 substation transformer
- 3 line segments
- 2 time-series spot loads
- 1 capacitor with control
- 1 regulator with control
- 1 wind generator
- 4 monitors

* Time range : 1 day, Time resolution: 1 second (86,400 snapshots),

Case II: Time-series Power Flow

Build OpenDSS Model – Circuit, Vsource

OpenDSS Code:

```
clear  
New object=circuit.ExampleCircuit  
~ basekv=115 1.00 0.0 60.0 3 20000 21000 4.0 3.0
```

Explanation:

Clear

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

New object=circuit.ExampleCircuit

Defines a new circuit named ExampleCircuit.

~

More Command. The More command continues editing the active object. The same as M or More.

280

Case II: Time-series Power Flow

Build OpenDSS Model – Circuit, Vsource

OpenDSS Code:

```
clear  
New object=circuit.ExampleCircuit  
~ basekv=115 1.00 0.0 60.0 3 20000 21000 4.0 3.0
```

Explanation:

basekv=115 1.00 0.0 60.0 3 20000 21000 4.0 3.0

There are *two* ways of setting property values of DSS circuit elements.

1. Simply specify the setting using complete element and property name, “=”, and a value. For example,

```
New Line.line1.R1=.05 X1=.12 R0=.1 X0=.4
```

2. Positional property rule. For example, we can use

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

to set the R1, X1, R0, X0 properties of Line.line1 in sequence, which is equivalent to the code using 1st way. In this case, the 2nd way is adopted. 281

Case II: Time-series Power Flow

Build OpenDSS Model – Circuit, Vsource

OpenDSS Code:

```
clear  
New object=circuit.ExampleCircuit  
~ basekv=115 1.00 0.0 60.0 3 20000 21000 4.0 3.0
```

Explanation:

basekv=115 1.00 0.0 60.0 3 20000 21000 4.0 3.0

The above code equals: basekv=115 pu=1.00 Angle=0.0 Frequency=60.0
Phases=3 Mvasc3=20000 Mvasc1=21000 x1r1=4.0 x0r0=3.0

basekv: Specifies the voltage Base or rated Line-to-line kV of Vsource.

pu: Specifies the actual per unit at which the source is operating. It is assumed balanced for all phases.

Angle: Specifies the Base angle, in degrees, of the first phase.

Frequency: Specifies the frequency of the source.

...

Case II: Time-series Power Flow

Build OpenDSS Model – Circuit, Vsource

OpenDSS Code:

```
clear  
New object=circuit.ExampleCircuit  
~ basekv=115 1.00 0.0 60.0 3 20000 21000 4.0 3.0
```

Explanation:

basekv=115 1.00 0.0 60.0 3 20000 21000 4.0 3.0

This above code equals: basekv=115 pu=1.00 Angle=0.0 Frequency=60.0

Phases=3 Mvasc3=20000 Mvasc1=21000 x1r1=4.0 x0r0=3.0

phases: Specifies the number of phases. Default = 3.0.

MVAsc3: Specifies the 3-phase short circuit MVA.

MVAsc1: Specifies the 1-phase short circuit MVA.

x1r1: Specifies the ratio of X1/R1. Default = 4.0.

x0r0: Specifies the ratio of X0/R0. Default = 3.0.

Case II: Time-series Power Flow

Build OpenDSS Model – Substation transformer

OpenDSS Code:

```
// Define Substation transformer  
New transformer.subxfrm phases=3 windings=2 buses=(SourceBus subbus)  
~ conns='delta wye' kvs=(115 12.47) kvas=(20000 20000) XHL=7
```

Explanation:

New transformer.subxfrm

Defines a transformer object named subxfrm.

phases

Specifies the number of phases. Default is 3.

windings

Specifies the total number of windings. Default is 2.

Case II: Time-series Power Flow

Build OpenDSS Model – Substation transformer

OpenDSS Code:

```
// Define Substation transformer  
New transformer.subxfm phases=3 windings=2 buses=(SourceBus subbus)  
~ conns='delta wye' kvs=(115 12.47) kvas=(20000 20000) XHL=7
```

Explanation:

buses=(SourceBus subbus)

The above command equals: bus1=SourceBus bus2=subbus

bus1: Specifies the name of bus for terminal 1.

bus2: Specifies the name of bus for terminal 2.

conns='delta wye'

Specify that the connection of winding1 is 'delta', the connection of winding2 is 'wye'.

Case II: Time-series Power Flow

Build OpenDSS Model – Substation transformer

OpenDSS Code:

```
// Define Substation transformer  
New transformer.subxfm phases=3 windings=2 buses=(SourceBus subbus)  
~ conns='delta wye' kvs=(115 12.47) kvas=(20000 20000) XHL=7
```

Explanation:

kvs=(115 12.47)

Specify that the rated voltage of winding 1 is 115 kV. The rated voltage of winding 2 is 12.47kV.

kvas=(20000 20000)

Specify that the base kVA rating of winding1 is 20000 kVA. The base kVA rating of winding2 is 20000kVA.

XHL

Specify the percent reactance high-to-low (winding 1 to winding 2).

Case II: Time-series Power Flow

Build OpenDSS Model – Linecode

OpenDSS Code:

```
// Define a linecode for the lines - unbalanced 336 MCM ACSR connection
New linecode.336matrix nphases=3
~ rmatrix=(0.0868455 | 0.0298305 0.0887966 | 0.0288883 0.0298305 0.0868455)
~ xmatrix=(0.2025449 | 0.0847210 0.1961452 | 0.0719161 0.0847210 0.2025449)
~ cmatrix=(2.74 | -0.70 2.96 | -0.34 -0.71 2.74)
~ Normamps = 400 Emergamps=600
```

Explanation:

New linecode.336matrix

Defines a linecode object named 336matrix.

nphases

Specifies the number of phases. Default = 3.

Case II: Time-series Power Flow

Build OpenDSS Model – Linecode

OpenDSS Code:

```
// Define a linecode for the lines - unbalanced 336 MCM ACSR connection
New linecode.336matrix nphases=3
~ rmatrix=(0.0868455 | 0.0298305 0.0887966 | 0.0288883 0.0298305 0.0868455)
~ xmatrix=(0.2025449 | 0.0847210 0.1961452 | 0.0719161 0.0847210 0.2025449)
~ cmatrix=(2.74 | -0.70 2.96 | -0.34 -0.71 2.74)
~ Normamps = 400 Emergamps=600
```

Explanation:

rmatrix

Series resistance matrix, in ohms per unit length. The order of matrices expected is the number of phases. The matrices may be entered in lower triangle form or full matrix. The result is always symmetrical.

The "|" separates rows.

xmatrix

Series reactance matrix, in ohms per unit length. Similar with rmatrix.

Case II: Time-series Power Flow

Build OpenDSS Model – Linecode

OpenDSS Code:

```
// Define a linecode for the lines - unbalanced 336 MCM ACSR connection
New linecode.336matrix nphases=3
~ rmatrix=(0.0868455 | 0.0298305 0.0887966 | 0.0288883 0.0298305 0.0868455)
~ xmatrix=(0.2025449 | 0.0847210 0.1961452 | 0.0719161 0.0847210 0.2025449)
~ cmatrix=(2.74 | -0.70 2.96 | -0.34 -0.71 2.74)
~ Normamps = 400 Emergamps=600
```

Explanation:

cmatrix

Shunt nodal capacitance matrix, in nanofarads per unit length. The order of matrices expected is the number of phases. The matrices may be entered in lower triangle form or full matrix. The result is always symmetrical. The "|" separates rows.

Case II: Time-series Power Flow

Build OpenDSS Model – Linecode

OpenDSS Code:

```
// Define a linecode for the lines - unbalanced 336 MCM ACSR connection
New linecode.336matrix nphases=3
~ rmatrix=(0.0868455 | 0.0298305 0.0887966 | 0.0288883 0.0298305 0.0868455)
~ xmatrix=(0.2025449 | 0.0847210 0.1961452 | 0.0719161 0.0847210 0.2025449)
~ cmatrix=(2.74 | -0.70 2.96 | -0.34 -0.71 2.74)
~ Normamps = 400 Emergamps=600
```

Explanation:

Normamps

Specifies the normal ampacity, in amperes.

Emergamps

Specifies the emergency ampacity, in amperes.

Case II: Time-series Power Flow

Build OpenDSS Model – Line

OpenDSS Code:

```
// Define the lines  
New line.line1 bus1=subbus bus2=loadbus1 linecode=336matrix length=10 units=kft  
New line.line2 loadbus1 loadbus2 336matrix 10  
New line.line3 Loadbus2 loadbus3 336matrix 20
```

Explanation:

New line.11

Defines a line object named 11.

Bus1

Specifies the name of bus for terminal 1.

Bus2

Specifies the name of bus for terminal 2.

Case II: Time-series Power Flow

Build OpenDSS Model – Line

OpenDSS Code:

```
// Define the lines
New line.line1 bus1=subbus bus2=loadbus1 linecode=336matrix length=10 units=kft
New line.line2 loadbus1 loadbus2 336matrix 10
New line.line3 Loadbus2 loadbus3 336matrix 20
```

Explanation:

LineCode

Specifies the name of an existing LineCode object containing impedance definitions.

Length

Specifies a length multiplier to be applied to the impedance data.

units

Specifies the length Units.

{none | mi | kft | km | m | Ft | in | cm }

Case II: Time-series Power Flow

Build OpenDSS Model – Loadshape

OpenDSS Code:

```
// Define some load shapes for the loads and wind
New loadshape.day 8 3.0
~ mult=(.3 .36 .48 .62 .87 .95 .94 .60)
New loadshape.wind 2400 {1 24 /}
~ mult=(file=zavwind.csv) action=normalize
```

Explanation:

A Loadshape object is very important for all types of **sequential** power flow solutions. It consists of a series of **multipliers**, typically ranging from 0.0 to 1.0 which are applied to the base kW values of the load, to represent variation of the load over some time period.

New loadshape.day 8 3.0

Defines a loadshape object named day.

The above code equals: New loadshape.day Npts=8 Interval=3.0.

Case II: Time-series Power Flow

Build OpenDSS Model – Loadshape

OpenDSS Code:

```
// Define some load shapes for the loads and wind
New loadshape.day 8 3.0
~ mult=(.3 .36 .48 .62 .87 .95 .94 .60)
New loadshape.wind 2400 {1 24 /}
~ mult=(file=zavwind.csv) action=normalize
```

Explanation:

New loadshape.day 8 3.0

New loadshape.day Npts=8 Interval=3.0.

Npts: Specifies the number of points to expect when defining the curve.

Interval: Specifies the time interval of the data, in Hr. Default=1.0. If the load shape has non-uniformly spaced points, specify the interval as 0.0.

Note:

1. In defaults: the loadshape is constant during each 3-hr interval. $8*3=24$ hrs.
2. For the loadshape named wind, $2400*1/24=100$ hrs. Considering the simulation time range is 24 hours (specified later), the remaining 76 points are clipped.

Case II: Time-series Power Flow

Build OpenDSS Model – Loadshape

OpenDSS Code:

```
// Define some load shapes for the loads and wind
New loadshape.day 8 3.0
~ mult=(.3 .36 .48 .62 .87 .95 .94 .60)
New loadshape.wind 2400 {1 24 /}
~ mult=(file=zavwind.csv) action=normalize
```

Explanation:

Mult

Specifies an array of multiplier values. It looks for Npts values. Omitted values are assumed to be zero, and **extra values are ignored**.

You may also use the syntax:

```
Mult=[file=myfile.txt],
Mult=[file=myfile.csv]
Mult=[dblfile=myfile.dbl],
Mult=[sngfile=myfile.sng]
```

Case II: Time-series Power Flow

Build OpenDSS Model – Loadshape

OpenDSS Code:

```
// Define some load shapes for the loads and wind
New loadshape.day 8 3.0
~ mult=(.3 .36 .48 .62 .87 .95 .94 .60)
New loadshape.wind 2400 {1 24 /}
~ mult=(file=zavwind.csv) action=normalize
```

Explanation:

action

After defining the load curve data, setting **action=normalize** will modify the multipliers so that the peak is 1.0. The mean and std deviation are recomputed.

Case II: Time-series Power Flow

Build OpenDSS Model – Load

OpenDSS Code:

```
// Define the loads
New load.load1 bus1=loadbus1 phases=3 kv=12.47 kw=1000.0 pf=0.88 model=1
~ class=1 duty=day
New load.load2 bus1=loadbus2 phases=3 kv=12.47 kw=500.0 pf=0.88 model=1
~ class=1 duty=day conn=delta
```

Explanation:

New Load.load1

Defines a load object named load1.

bus1

Specifies the name of bus to which the load is connected.

phases

Specifies the number of phases of this load.

Case II: Time-series Power Flow

Build OpenDSS Model – Load

OpenDSS Code:

```
// Define the loads
New load.load1 bus1=loadbus1 phases=3 kv=12.47 kw=1000.0 pf=0.88 model=1
~ class=1 duty=day
New load.load2 bus1=loadbus2 phases=3 kv=12.47 kw=500.0 pf=0.88 model=1
~ class=1 duty=day conn=delta
```

Explanation:

kv

Specifies the Base voltage for load. For 2- or 3-phase loads, it is specified in phase-to-phase kV.

For all other loads, it is specified in the actual kV across the load branch. If the load is wye (star) connected, then specify phase-to-neutral (L-N) kV. If the load is delta or phase-to-phase connected, specify the phase-to-phase (L-L) kV.

kw

Specifies the nominal active power, in kW, for the load. Total of all phases.²⁹⁸

Case II: Time-series Power Flow

Build OpenDSS Model – Load

OpenDSS Code:

```
// Define the loads
New load.load1 bus1=loadbus1 phases=3 kv=12.47 kw=1000.0 pf=0.88 model=1
~ class=1 duty=day
New load.load2 bus1=loadbus2 phases=3 kv=12.47 kw=500.0 pf=0.88 model=1
~ class=1 duty=day conn=delta
```

Explanation:

pf

Specifies the nominal power factor for load. Negative PF is leading. Specify either PF or kvar. If both are specified, the last one specified takes precedence.

Model

An integer defining how the load will vary with voltage. The load models currently implemented are:

1: Constant P and constant Q (Default): Commonly used for power flow studies,

...

299

Case II: Time-series Power Flow

Build OpenDSS Model – Load

OpenDSS Code:

```
// Define the loads
New load.load1 bus1=loadbus1 phases=3 kv=12.47 kw=1000.0 pf=0.88 model=1
~ class=1 duty=day
New load.load2 bus1=loadbus2 phases=3 kv=12.47 kw=500.0 pf=0.88 model=1
~ class=1 duty=day conn=delta
```

Explanation:

Model

...

2: Constant Z (or constant impedance),

3: Constant P and quadratic Q ($Q/Q_0=(V/V_0)^2$),

4: Exponential: $P/P_0=(V/V_0)^{CVRwatts}$ and $Q/Q_0=(V/V_0)^{CVRvars}$

5: Constant I (or constant current magnitude),

6: Constant P and fixed Q (at the nominal value)

7: Constant P and quadratic Q (i.e., fixed reactance),

...

300

Case II: Time-series Power Flow

Build OpenDSS Model – Load

OpenDSS Code:

```
// Define the loads
New load.load1 bus1=loadbus1 phases=3 kv=12.47 kw=1000.0 pf=0.88 model=1
~ class=1 duty=day
New load.load2 bus1=loadbus2 phases=3 kv=12.47 kw=500.0 pf=0.88 model=1
~ class=1 duty=day conn=delta
```

Explanation:

Model

...

8: ZIP

"Constant" power value (either P or Q) may be modified by loadshape multipliers. "Fixed" power values are always the same.

- ZIP model:

$$P = P_0 \left[a_1 \left(\frac{V}{V_0} \right)^2 + a_2 \left(\frac{V}{V_0} \right) + a_3 \right]$$

$$Q = Q_0 \left[a_4 \left(\frac{V}{V_0} \right)^2 + a_5 \left(\frac{V}{V_0} \right) + a_6 \right]$$

Case II: Time-series Power Flow

Build OpenDSS Model – Load

OpenDSS Code:

```
// Define the loads
New load.load1 bus1=loadbus1 phases=3 kv=12.47 kw=1000.0 pf=0.88 model=1
~ class=1 duty=day
New load.load2 bus1=loadbus2 phases=3 kv=12.47 kw=500.0 pf=0.88 model=1
~ class=1 duty=day conn=delta
```

Explanation:

class

An integer number segregating the load according to a particular class.

duty

Specifies the name of duty cycle load shape. Defaults to Daily load shape if not defined.

conn

Specifies the winding connection. {wye | y | LN} for Wye (Line-Neutral) connection; {delta | LL} for Delta (Line-Line) connection. Default = wye. 302

Case II: Time-series Power Flow

Build OpenDSS Model – Capacitor

OpenDSS Code:

```
// Capacitor with control  
New capacitor.Cap1 bus1=loadbus2 phases=3 kvar=600 kv=12.47  
New capcontrol.Cap1Ctrl element=line.line3 terminal=1 capacitor=Cap1  
~ type=current ctratio=1 ONsetting=60 OFFsetting=55 delay=2
```

Explanation:

New Capacitor.Cap1

Defines a capacitor object named Cap1.

Bus1

Specifies the name of the first bus to which the capacitor is connected.

phases

Specifies the number of phases.

kvar

Specifies the rated kvar at rated kV, total of all phases. Each phase is assumed equal.

303

Case II: Time-series Power Flow

Build OpenDSS Model – Capacitor

OpenDSS Code:

```
// Capacitor with control  
New capacitor.Cap1 bus1=loadbus2 phases=3 kvar=600 kv=12.47  
New capcontrol.Cap1Ctrl element=line.line3 terminal=1 capacitor=Cap1  
~ type=current ctratio=1 ONsetting=60 OFFsetting=55 delay=2
```

Explanation:

kv

Specifies the kV rating of capacitor. For Phases=2 or Phases=3, enter line-to-line (phase-to-phase) rated voltage. For all other numbers of phases, enter actual can rating. (For Delta connection this is always line-to-line rated voltage).

New capcontrol.Cap1Ctrl

Defines a capacitor control object named Cap1Ctrl.

* The capacitor control monitors the voltage and current at a terminal of a PDelement or a PCElement and sends switching messages to a Capacitor object.

Case II: Time-series Power Flow

Build OpenDSS Model – Capacitor

OpenDSS Code:

```
// Capacitor with control  
New capacitor.Cap1 bus1=loadbus2 phases=3 kvar=600 kv=12.47  
New capcontrol.Cap1Ctrl element=line.line3 terminal=1 capacitor=Cap1  
~ type=current ctratio=1 ONsetting=60 OFFsetting=55 delay=2
```

Explanation:

element

Specifies the full object name of the circuit element, typically a line or transformer, to which the capacitor control's PT and/or CT are connected. There is no default; must be specified.

terminal

Specifies the number of terminal of the circuit element to which the CapControl is connected. 1 or 2, typically. Default is 1.

Case II: Time-series Power Flow

Build OpenDSS Model – Capacitor

OpenDSS Code:

```
// Capacitor with control  
New capacitor.Cap1 bus1=loadbus2 phases=3 kvar=600 kv=12.47  
New capcontrol.Cap1Ctrl element=line.line3 terminal=1 capacitor=Cap1  
~ type=current ctratio=1 ONsetting=60 OFFsetting=55 delay=2
```

Explanation:

capacitor

Specifies the name of Capacitor which the CapControl controls. No Default; Must be specified.

type

Specifies the Control type. {Current | voltage | kvar | PF | time }.

ctratio

Specifies the ratio of the CT from line amps to control ampere setting for current and kvar control types.

Case II: Time-series Power Flow

Build OpenDSS Model – Capacitor

OpenDSS Code:

```
// Capacitor with control  
New capacitor.Cap1 bus1=loadbus2 phases=3 kvar=600 kv=12.47  
New capcontrol.Cap1Ctrl element=line.line3 terminal=1 capacitor=Cap1  
~ type=current ctratio=1 ONsetting=60 OFFsetting=55 delay=2
```

Explanation:

onsetting

Specifies the value at which the control arms to switch the capacitor ON. For different types of control, the value is calculated by:

- Current: Line Amps / CTratio
- Voltage: Line-Neutral (or Line-Line for delta) Volts / PTratio
- kvar: Total kvar, all phases. This is directional.
- PF: Power Factor, Total power in monitored terminal.
- Time: Hrs from Midnight as a floating point number (decimal). 7:30am would be entered as 7.5.

Case II: Time-series Power Flow

Build OpenDSS Model – Capacitor

OpenDSS Code:

```
// Capacitor with control  
New capacitor.Cap1 bus1=loadbus2 phases=3 kvar=600 kv=12.47  
New capcontrol.Cap1Ctrl element=line.line3 terminal=1 capacitor=Cap1  
~ type=current ctratio=1 ONsetting=60 OFFsetting=55 delay=2
```

Explanation:

offsetting

Specifies the value at which the control arms to switch the capacitor OFF.

For different types of control, the value is calculated by:

- Current: Line Amps / CTratio
- Voltage: Line-Neutral (or Line-Line for delta) Volts / PTratio
- kvar: Total kvar, all phases. This is directional.
- PF: Power Factor, Total power in monitored terminal.
- Time: Hrs from Midnight as a floating point number (decimal). 7:30am would be entered as 7.5.

Case II: Time-series Power Flow

Build OpenDSS Model – Capacitor

OpenDSS Code:

```
// Capacitor with control  
New capacitor.Cap1 bus1=loadbus2 phases=3 kvar=600 kv=12.47  
New capcontrol.Cap1Ctrl element=line.line3 terminal=1 capacitor=Cap1  
~ type=current ctratio=1 ONsetting=60 OFFsetting=55 delay=2
```

Explanation:

delay

Specifies the time delay, in seconds, from when the control is armed before it sends out the switching command to turn the capacitor ON.

Case II: Time-series Power Flow

Build OpenDSS Model – Regulator

OpenDSS Code:

```
// Regulator for DG
New transformer.reg1 phases=3 windings=2
~ buses=(loadbus3 regbus) conns='wye wye' kvs=(12.47 12.47) kvas=(8000 8000) XHL=1
// Regulator Control definitions
New regcontrol.subxfrmCtrl transformer=subxfrm winding=2 vreg=125
~ band=3 ptratio=60 delay=10
New regcontrol.reg1Ctrl transformer=reg1 winding=2 vreg=122 band=3
~ ptratio=60 delay=15
```

Explanation:

New Transformer.reg1

Defines a transformer named reg1.

phases

Specifies the number of phases.

windings

Specifies the number of windings.

Case II: Time-series Power Flow

Build OpenDSS Model – Regulator

OpenDSS Code:

```
// Regulator for DG
New transformer.reg1 phases=3 windings=2
~ buses=(loadbus3 regbus) conns='wye wye' kvs=(12.47 12.47) kvas=(8000 8000) XHL=1
// Regulator Control definitions
New regcontrol.subxfrmCtrl transformer=subxfrm winding=2 vreg=125
~ band=3 ptratio=60 delay=10
New regcontrol.reg1Ctrl transformer=reg1 winding=2 vreg=122 band=3
~ ptratio=60 delay=15
```

Explanation:

buses=(loadbus3 regbus)

An array specifying the names of buses to which the windings are connected. Winding1 of transformer is connected to loadbus3, winding2 is connected to regbus.

conns='wye wye'

Specifies an array of connections for winding1 and winding2.

kvs=(12.47 12.47)

Specifies an array of kV ratings for winding1 and winding2.

311

Case II: Time-series Power Flow

Build OpenDSS Model – Regulator

OpenDSS Code:

```
// Regulator for DG
New transformer.reg1 phases=3 windings=2
~ buses=(loadbus3 regbus) conns='wye wye' kvs=(12.47 12.47) kvas=(8000 8000) XHL=1
// Regulator Control definitions
New regcontrol.subxfrmCtrl transformer=subxfrm winding=2 vreg=125
~ band=3 ptratio=60 delay=10
New regcontrol.reg1Ctrl transformer=reg1 winding=2 vreg=122 band=3
~ ptratio=60 delay=15
```

Explanation:

kvas=(8000 8000)

Specifies an array of base kVA rating for winding1 and winding2.

XHL (or X12)

Specifies the percent reactance high-to-low (winding 1 to winding 2).

Case II: Time-series Power Flow

Build OpenDSS Model – Regulator

OpenDSS Code:

```
// Regulator for DG
New transformer.reg1 phases=3 windings=2
~ buses=(loadbus3 regbus) conns='wye wye' kvs=(12.47 12.47) kvas=(8000 8000) XHL=1
// Regulator Control definitions
New regcontrol.subxfmCtrl transformer=subxfm winding=2 vreg=125
~ band=3 ptratio=60 delay=10
New regcontrol.reg1Ctrl transformer=reg1 winding=2 vreg=122 band=3
~ ptratio=60 delay=15
```

Explanation:

New regcontrol.subxfmCtrl

Defines a regulator control object named subxfmCtrl.

transformer

Specifies the name of transformer to which the RegControl is connected.

winding

Specifies the number of the winding of the transformer that the RegControl is monitoring. 1 or 2, typically.

313

Case II: Time-series Power Flow

Build OpenDSS Model – Regulator

OpenDSS Code:

```
// Regulator for DG
New transformer.reg1 phases=3 windings=2
~ buses=(loadbus3 regbus) conns='wye wye' kvs=(12.47 12.47) kvas=(8000 8000) XHL=1
// Regulator Control definitions
New regcontrol.subxfrmCtrl transformer=subxfrm winding=2 vreg=125
~ band=3 ptratio=60 delay=10
New regcontrol.reg1Ctrl transformer=reg1 winding=2 vreg=122 band=3
~ ptratio=60 delay=15
```

Explanation:

vreg

Specifies the voltage regulator setting, in VOLTS, for the winding being controlled.

band

Specifies the bandwidth, in VOLTS, for the controlled bus. Default is 3.0

Case II: Time-series Power Flow

Build OpenDSS Model – Regulator

OpenDSS Code:

```
// Regulator for DG
New transformer.reg1 phases=3 windings=2
~ buses=(loadbus3 regbus) conns='wye wye' kvs=(12.47 12.47) kvas=(8000 8000) XHL=1
// Regulator Control definitions
New regcontrol.subxfrmCtrl transformer=subxfrm winding=2 vreg=125
~ band=3 ptratio=60 delay=10
New regcontrol.reg1Ctrl transformer=reg1 winding=2 vreg=122 band=3
~ ptratio=60 delay=15
```

Explanation:

ptratio

Specifies the ratio of the PT that converts the controlled winding voltage to the regulator voltage. Default is 60. If the winding is Wye, the line-to-neutral voltage is used. Else, the line-to-line voltage is used.

delay

Specifies the time delay, in seconds, from when the voltage goes out of band to when the tap changing begins. This is used to determine which regulator control will act first. Default is 15.

315

Case II: Time-series Power Flow

Build OpenDSS Model – Wind generator

OpenDSS Code:

```
// Define a wind generator of 8MW  
New generator.gen1 bus1=regbus kV=12.47 kW=8000 pf=1 conn=delta  
~ duty=wind Model=1
```

Explanation:

A Generator is a Power Conversion element similar to a Load object. Its rating is basically defined by its nominal kW and PF or its kW and kvar. Then it may be modified by a number of multipliers, including the global circuit load multiplier, yearly load shape, daily load shape, and a dutycycle load shape.

New generator.gen1

Defines a generator object named gen1.

bus1

Specifies the name of bus to which the generator is connected.

Case II: Time-series Power Flow

Build OpenDSS Model – Wind generator

OpenDSS Code:

```
// Define a wind generator of 8MW  
New generator.gen1 bus1=regbus kV=12.47 kW=8000 pf=1 conn=delta  
~ duty=wind Model=1
```

Explanation:

kV

Specifies the Base voltage for generator. For 2- or 3-phase generators, it is specified in phase-to-phase kV.

For all other generators, it is specified in the actual kV across the generator branch. If the generator is wye (star) connected, specify the phase-to-neutral (L-N) kV. If the generator is delta or phase-to-phase connected, specify the phase-to-phase (L-L) kV.

kW

Specifies the nominal kW for generator. Total of all phases.

pf

Specifies the nominal power factor for generator.

Case II: Time-series Power Flow

Build OpenDSS Model – Wind generator

OpenDSS Code:

```
// Define a wind generator of 8MW  
New generator.gen1 bus1=regbus kV=12.47 kW=8000 pf=1 conn=delta  
~ duty=wind Model=1
```

Explanation:

conn

Specifies the winding connection of the generator. {wye | y | LN} for Wye (Line-Neutral) connection; {delta | LL} for Delta (Line-Line) connection. Default = wye.

duty

Specifies the name of Duty cycle load shape. Defaults to Daily load shape if not defined.

Model

An integer defining how the generator will vary with voltage. Presently defined models are:

...

Case II: Time-series Power Flow

Build OpenDSS Model – Wind generator

OpenDSS Code:

```
// Define a wind generator of 8MW  
New generator.gen1 bus1=regbus kV=12.47 kW=8000 pf=1 conn=delta  
~ duty=wind Model=1
```

Explanation:

Model

...

- 1: Generator injects a constant kW at specified power factor.
- 2: Generator is modeled as a constant admittance.
- 3: Const kW, constant kV. Somewhat like a conventional transmission power flow P-V generator.
- 4: Const kW, Fixed Q (Q never varies).
- 5: Const kW, Fixed Q(as a constant reactance).
- 6: Compute load injection from User-written Model.
- 7: Constant kW, kvar, but current is limited when voltage is below V_{minpu} .

Case II: Time-series Power Flow

Build OpenDSS Model – Monitor

OpenDSS Code:

```
// Define some monitors so we can see what's happening
New Monitor.gen1 element=generator.gen1 terminal=1 mode=1
New Monitor.loadbus2 load.load2 1 mode=0
New Monitor.line3 line.line3 1 mode=48
New Energymeter.em1 line.line1
```

Explanation:

A monitor is a circuit element that is connected to a terminal of another circuit element. It takes a sample when instructed, recording the time and the complex values of voltage and current, or power, at all phases.

New Monitor.gen1

Defines a monitor object named gen1.

element

Specify the name of an existing circuit element to which the monitor is connected.

Case II: Time-series Power Flow

Build OpenDSS Model – Monitor

OpenDSS Code:

```
// Define some monitors so we can see what's happening
New Monitor.gen1 element=generator.gen1 terminal=1 mode=1
New Monitor.loadbus2 load.load2 1 mode=0
New Monitor.line3 line.line3 1 mode=48
New Energymeter.em1 line.line1
```

Explanation:

terminal

Specifies the number of the terminal to which the monitor is connected.

mode

An integer bitmask code to describe what the monitor will save. Monitors can save two basic types of quantities: 1) Voltage and current; 2) Power. The Mode codes are defined as follows:

0: Standard mode - V and I, each phase, complex

1: Power each phase, complex (kw and kvars)

...

321

Case II: Time-series Power Flow

Build OpenDSS Model – Monitor

OpenDSS Code:

```
// Define some monitors so we can see what's happening
New Monitor.gen1 element=generator.gen1 terminal=1 mode=1
New Monitor.loadbus2 load.load2 1 mode=0
New Monitor.line3 line.line3 1 mode=48
New Energymeter.em1 line.line1
```

Explanation:

mode

...

2: Transformer taps (connect Monitor to a transformer winding)

3: State variables (connect Monitor to a Power conversion Element)

4: Flicker level and severity index (Pst) for voltages.

5: Solution variables (Iterations, etc). Normally, these would be actual phasor quantities from solution.

6: Capacitor Switching

...

Case II: Time-series Power Flow

Build OpenDSS Model – Monitor

OpenDSS Code:

```
// Define some monitors so we can see what's happening
New Monitor.gen1 element=generator.gen1 terminal=1 mode=1
New Monitor.loadbus2 load.load2 1 mode=0
New Monitor.line3 line.line3 1 mode=48
New Energymeter.em1 line.line1
```

Explanation:

mode

...

+16: Sequence components: V012, I012

+32: Magnitude only

+64: Positive Sequence only or Average of phases, if not 3 phases.

For example, Mode=33 (33=1+32) will save the magnitude of the power (kVA) only in each phase. Mode=112 (112=0+16+32+64) saves Positive sequence voltages and currents, magnitudes only.

Case II: Time-series Power Flow

Build OpenDSS Model – Energymeter

OpenDSS Code:

```
// Define some monitors so we can see what's happening
New Monitor.gen1 element=generator.gen1 terminal=1 mode=1
New Monitor.loadbus2 load.load2 1 mode=0
New Monitor.line3 line.line3 1 mode=48
New Energymeter.em1 line.line1
```

Explanation:

* An EnergyMeter object is an intelligent meter connected to a terminal of a circuit element. It simulates the behavior of an actual energy meter.

New Energymeter.em1

Defines an energymeter object named em1.

line.line1

The above code is the same as: Element=line.line1

It designates the name of an existing circuit element to which the Energymeter is connected.

Case II: Time-series Power Flow

Build OpenDSS Model – Solve

OpenDSS Code:

```
Set voltagebases=(115 12.47 .48)
Calc voltagebases
Set controlmode=time
Set mode=dutycycle number=86400 hour=0 stepsize=1 sec=0
Solve
```

Explanation:

Set voltagebases=(115 12.47 .48)

Defines legal bus voltage bases for this circuit. Enter an array of the legal voltage bases, in phase-to-phase voltages. In this case, three voltage bases are defined, 115kV, 12.47kV and 0.48kV.

- The voltage level can be selected from the following values: .208, .480, 12.47, 24.9, 34.5, 115.0, 230.0

Case II: Time-series Power Flow

Build OpenDSS Model – Solve

OpenDSS Code:

```
Set voltagebases=(115 12.47 .48)
Calcvoltagebases
Set controlmode=time
Set mode=dutycycle number=86400 hour=0 stepsize=1 sec=0
Solve
```

Explanation:

Calcvoltagebases

Estimates the voltage base for each bus based on the array of voltage bases defined with a "Set Voltagebases=..." command. The voltage base for each bus is then set to the nearest voltage base specified in the voltage base array.

ControlMode

Specifies the Control mode for the solution. {OFF | STATIC |EVENT | TIME}, Default is "STATIC".

The detailed meanings are as follows:

...

326

Case II: Time-series Power Flow

Build OpenDSS Model – Solve

OpenDSS Code:

```
Set voltagebases=(115 12.47 .48)
Calcvoltagebases
Set controlmode=time
Set mode=dutycycle number=86400 hour=0 stepsize=1 sec=0
Solve
```

Explanation:

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.

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.

Case II: Time-series Power Flow

Build OpenDSS Model – Solve

OpenDSS Code:

```
Set voltagebases=(115 12.47 .48)
Calcvoltagebases
Set controlmode=time
Set mode=dutycycle number=86400 hour=0 stepsize=1 sec=0
Solve
```

Explanation:

Mode

Specifies the solution mode for the active circuit. Mode can be one of snap, daily, direct, dutycycle, dynamics, faultstudy, harmonics, or yearly.

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).

...

Case II: Time-series Power Flow

Build OpenDSS Model – Solve

OpenDSS Code:

```
Set voltagebases=(115 12.47 .48)
Calcvoltagebases
Set controlmode=time
Set mode=dutycycle number=86400 hour=0 stepsize=1 sec=0
Solve
```

Explanation:

Mode

...

Mode=Daily:

Do a series of solutions following the daily load curves. The Step size defaults to 3600 sec (1 hr).

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.

...

Case II: Time-series Power Flow

Build OpenDSS Model – Solve

OpenDSS Code:

```
Set voltagebases=(115 12.47 .48)
Calcvoltagebases
Set controlmode=time
Set mode=dutycycle number=86400 hour=0 stepsize=1 sec=0
Solve
```

Explanation:

Mode

...

Mode=Dutycycle:

The solver follows the duty cycle curves with the time increment specified.

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.

...

Case II: Time-series Power Flow

Build OpenDSS Model – Solve

OpenDSS Code:

```
Set voltagebases=(115 12.47 .48)
Calcvoltagebases
Set controlmode=time
Set mode=dutycycle number=86400 hour=0 stepsize=1 sec=0
Solve
```

Explanation:

Mode

...

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.

...

Case II: Time-series Power Flow

Build OpenDSS Model – Solve

OpenDSS Code:

```
Set voltagebases=(115 12.47 .48)
Calcvoltagebases
Set controlmode=time
Set mode=dutycycle number=86400 hour=0 stepsize=1 sec=0
Solve
```

Explanation:

Mode

...

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.

Case II: Time-series Power Flow

Build OpenDSS Model – Solve

OpenDSS Code:

```
Set voltagebases=(115 12.47 .48)
Calcvoltagebases
Set controlmode=time
Set mode=dutycycle number=86400 hour=0 stepsize=1 sec=0
Solve
```

Explanation:

number

Specifies the number of time steps or solutions to run.

hour

Sets the hour to be used for the start time of the solution of the active circuit.

stepsize

Sets the time step size for the solution of the active circuit. Default unit is sec.

sec

Sets the seconds from the hour for the start time for the solution of the active circuit.

Case II: Time-series Power Flow

Build OpenDSS Model – Solve

OpenDSS Code:

```
// Show some results  
Plot monitor, object=loadbus2  
Visualize element=Line.line1 what=powers  
Plot profile
```

Explanation:

plot

Displays a variety of results in a variety of manners on graphs.

monitor

Specifies that the type to be plotted is monitor. The Monitor plot plots one or more channels from a Monitor element.

object

Specifies what object to plot.

[metername for Zone plot | Monitor name | File Name for General bus data or Circuit branch data].

Case II: Time-series Power Flow

Build OpenDSS Model – Solve

OpenDSS Code:

```
// Show some results  
Plot monitor, object=loadbus2  
Visualize element=Line.line1 what=powers  
Plot profile
```

Explanation:

channel

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

visualize

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

Case II: Time-series Power Flow

Build OpenDSS Model – Solve

OpenDSS Code:

```
// Show some results  
Plot monitor, object=loadbus2  
Visualize element=Line.line1 what=powers  
Plot profile
```

Explanation:

element

Specifies the names of elements to be visualized. It should follow the following format:

element=full_element_name (i.e., class.name).

what

Specifies the what quantity to be visualized.

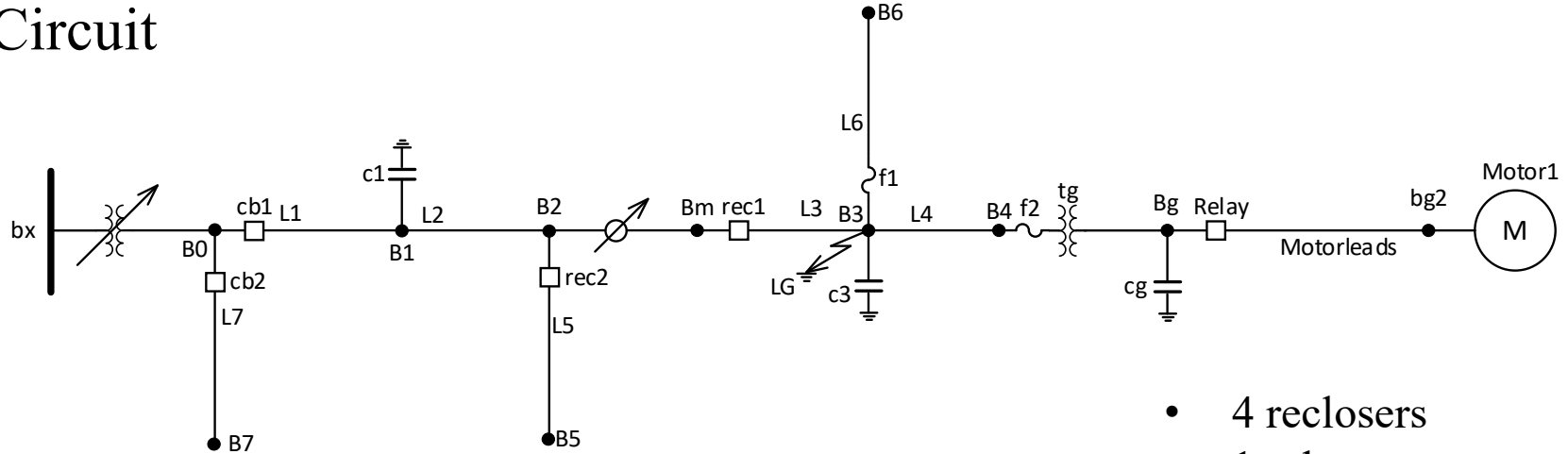
It can be the following item: currents, voltages, powers.

plot profile

Plot voltage profiles.

Case III: Fault and Dynamic

Circuit



- 12 buses
 - 1 voltage source
 - 2 transformers
 - 1 voltage regulator bank with control
 - 3 capacitor banks
 - 7 line segments
 - 4 reclosers
 - 1 relay
 - 2 fuses
 - 1 induction motor
 - 7 loads
- From 0 to 0.3 second, the circuit is in a normal status (no fault).
 - At $t=0.3$ second, a 0.21-second temporary SLG (Phase A) fault occurs at Bus **B3**.
 - Sensing the undervoltages and overcurrents, the Relay operates 0.1 s later (at $t=0.4$ second), isolating the induction motor.
 - Since the time duration of the fault is 0.21 s, the fault still exists. Therefore, the recloser rec1 disconnects line **L3** 0.11s later (at $t=0.51$ second). Also, almost at the same time, the temporary fault disappears.
 - After 0.5 seconds (at $t=1.01$ second), the recloser reconnects line **L3**.

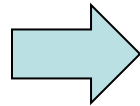
337

Case III: Fault and Dynamic

OpenDSS files

17 files

- Capacitor.dss
- Fault.dss
- Fuse.dss
- IndMach012.dss
- line.dss
- LineCode.dss
- Load.dss
- LoadShape.dss
- Master.DSS
- Monitor.dss
- Recloser.dss
- RegControl.dss
- Relay.dss
- Run.dss
- TCC_Curve.dss
- Transformer.dss
- Vsource.dss



- Define capacitor banks
- Define fault
- Define fuse
- Define induction motor
- Define line segment
- Define line configuration
- Define load
- Define loadshape
- Define Master file
- Define monitors
- Define reclosers
- Define regulator control
- Define relay
- Define a file to perform dynamic analysis
- Define operating curves for relay, recloser and fuses
- Define transformers
- Define voltage source

Case III: Fault and Dynamic

- Run.dss file

It is the main file which is defined to perform dynamic analysis.

Two steps to perform dynamic analysis:

1. Run static power flow to obtain the initialization for dynamic simulation
2. Perform the dynamic simulation

OpenDSS Code (Initialization):

```
Redirect "Master.DSS" !Compile  
New Monitor.Mtime Vsource.source mode=5 ! solution time monitor  
Set Maxcontroliter=20  
Set maxiterations=20  
solve
```

Explanation:

Redirect

Resets the current directory to Master.DSS file.

Case III: Fault and Dynamic

- Run.dss file

OpenDSS Code (Initialization):

```
Redirect "Master.DSS" !Compile  
New Monitor.Mtime Vsource.source mode=5 ! solution time monitor  
Set Maxcontroliter=20  
Set maxiterations=20  
solve
```

Explanation:

New Monitor.Mtime

Defines a monitor object named Mtime, which is used for collecting time information for plotting.

Vsource.source

It is equivalent with Element=Vsource.source

It specifies the Name of an existing circuit element to which the monitor is to be connected.

Case III: Fault and Dynamic

- Run.dss file

OpenDSS Code (Initialization):

```
Redirect "Master.DSS" !Compile
New Monitor.Mtime Vsource.source mode=5 ! solution time monitor
Set Maxcontroliter=20
Set maxiterations=20
solve
```

Explanation:

mode

Mode is an integer bitmask code to describe what the monitor will save. Monitors can save two basic types of quantities: 1) Voltage and current; 2) Power.

The Mode codes are defined as follows:

0: Standard mode - V and I, each phase, complex

1: Power each phase, complex (kw and kvars)

2: Transformer taps (connect Monitor to a transformer winding)

3: State variables (connect Monitor to a PCElement)

...

Case III: Fault and Dynamic

- Run.dss file

OpenDSS Code (Initialization):

```
Redirect "Master.DSS" !Compile
```

```
New Monitor.Mtime Vsource.source mode=5 ! solution time monitor
```

```
Set Maxcontroliter=20
```

```
Set maxiterations=20
```

```
solve
```

Explanation:

mode

...

4: Flicker level and severity index (Pst) for voltages.

5: Solution variables (Iterations, etc). Normally, these would be actual phasor quantities from solution.

6: Capacitor Switching

+16: Sequence components: V012, I012

+32: Magnitude only

+64: Positive Sequence only or Average of phases, if not 3 phases.

For example, Mode=33 (33=1+32) will save the magnitude of the power (kVA) only in each phase.

342

Case III: Fault and Dynamic

- Run.dss file

OpenDSS Code (Initialization):

```
Redirect "Master.DSS" !Compile
New Monitor.Mtime Vsource.source mode=5 ! solution time monitor
Set Maxcontroliter=20
Set maxiterations=20
solve
```

Explanation:

Maxcontroliter

Sets the maximum control iterations per solution. Default is 10.

maxiterations

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

Solve

Solves the circuit. Default mode is snapshot.

Case III: Fault and Dynamic

- Run.dss file

OpenDSS Code (Perform Dynamic Simulation):

```
Set mode=dynamics number=1 h=0.000166667  
solve  
Solve number= 10000  
get time  
get number
```

Explanation:

Mode

Specifies the solution mode for the active circuit. Mode can be one of snap, daily, direct, dutycycle, dynamics, faultstudy, harmonics, yearly.

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).

...

Case III: Fault and Dynamic

- Run.dss file

OpenDSS Code (Perform Dynamic Simulation):

```
Set mode=dynamics number=1 h=0.000166667
solve
Solve number= 10000
get time
get number
```

Explanation:

Mode

...

Mode=Daily:

Do a series of solutions following the daily load curves. The stepsize defaults to 3600 sec (1 hr).

Mode=Direct:

Solves 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.

...

Case III: Fault and Dynamic

- Run.dss file

OpenDSS Code (Perform Dynamic Simulation):

```
Set mode=dynamics number=1 h=0.000166667  
solve  
Solve number= 10000  
get time  
get number
```

Explanation:

Mode

...

Mode=Dutycycle:

Follows the duty cycle curves with the time increment specified.

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.

...

Case III: Fault and Dynamic

- Run.dss file

OpenDSS Code (Perform Dynamic Simulation):

```
Set mode=dynamics number=1 h=0.000166667  
solve  
Solve number= 10000  
get time  
get number
```

Explanation:

Mode

...

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.

...

Case III: Fault and Dynamic

- Run.dss file

OpenDSS Code (Perform Dynamic Simulation):

```
Set mode=dynamics number=1 h=0.000166667  
solve  
Solve number= 10000  
get time  
get number
```

Explanation:

Mode

...

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.

...

Case III: Fault and Dynamic

- Run.dss file

OpenDSS Code (Perform Dynamic Simulation):

```
Set mode=dynamics number=1 h=0.000166667  
solve  
Solve number= 10000  
get time  
get number
```

Explanation:

Mode

...

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.

number

Specifies the number of time steps in each solution.

Case III: Fault and Dynamic

- Run.dss file

OpenDSS Code (Perform Dynamic Simulation):

```
Set mode=dynamics number=1 h=0.000166667  
solve  
Solve number= 10000  
get time  
get number
```

Explanation:

h

Also called stepsize. It sets the time step size for the solution of the active circuit. Default unit is second. Normally, this is specified for dynamic solution.

Solve

Solves the circuit.

Solve number= 10000

Specifies the number of solutions in dynamic simulation to run.

Case III: Fault and Dynamic

- Run.dss file

OpenDSS Code (Perform Dynamic Simulation):

```
Set mode=dynamics number=1 h=0.000166667  
solve  
Solve number= 10000  
get time  
get number
```

Explanation:

get time

Obtains time values of solutions in dynamic simulation.

get number

Obtains number values of solutions in dynamic simulation.

Case III: Fault and Dynamic

- Run.dss file

OpenDSS Code (Plot Results):

Plot monitor object= cb1 channels=(1, 3, 5)

Plot monitor object= cb1 channels=(7, 9, 11)

Plot monitor object= rec1 channels=(1, 3, 5)

Plot monitor object= F1 channels=(1, 3, 5)

Explanation:

plot

Displays a variety of results in a variety of manners on graphs.

monitor

Specifies that the type to be plotted is monitor. The Monitor plot plots one or more channels from a Monitor element.

object

Specifies what object to plot.

[metername for Zone plot | Monitor name | File Name for General bus data or Circuit branch data].

Case III: Fault and Dynamic

- Run.dss file

OpenDSS Code (Plot Results):

Plot monitor object= cb1 channels=(1, 3, 5)

Plot monitor object= cb1 channels=(7, 9, 11)

Plot monitor object= rec1 channels=(1, 3, 5)

Plot monitor object= F1 channels=(1, 3, 5)

Explanation:

channel

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

Case III: Fault and Dynamic

- Master.dss file

OpenDSS Code:

```
Clear
New Circuit.NewModel
Redirect Vsource.dss
Redirect LineCode.dss
Redirect LoadShape.dss
Redirect TCC_Curve.dss
Redirect line.dss
Redirect Load.dss
Redirect Transformer.dss
Redirect RegControl.dss
Redirect Capacitor.dss
Redirect Fault.dss
Redirect IndMach012.dss
Redirect Relay.dss
Redirect Recloser.dss
Redirect Fuse.dss
Redirect Monitor.dss
Buscoords buscoords.dat

set voltagebases=(115 12.47 .48)
calcv
```

Case III: Fault and Dynamic

- Vsource.dss file

OpenDSS Code:

```
Edit "Vsource.source" bus1=Bx basekv=115 pu=1.0475 angle=0 frequency=60  
phases=3 MVAAsc3=20000 MVAAsc1=21000 x1r1=4 x0r0=3
```

Explanation:

Edit

Edits the object specified. In this case, the Vsource object named source is edited.

bus1

Specifies the bus to which Vsource is connected.

basekv

Specifies Base or rated Line-to-line kV of Vsource.

pu

Specifies the actual per unit at which the source is operating. It is assumed balanced for all phases.

Angle

Specifies Base angle, in degrees, of the first phase.

355

Case III: Fault and Dynamic

- Vsource.dss file

OpenDSS Code:

```
Edit "Vsource.source" bus1=Bx basekv=115 pu=1.0475 angle=0 frequency=60  
phases=3 MVAsc3=20000 MVAsc1=21000 x1r1=4 x0r0=3
```

Explanation:

frequency

Specifies the frequency of the source.

phases

Specifies the number of phases. Default = 3.0.

MVAsc3

Specifies the 3-phase short circuit MVA.

MVAsc1

Specifies the 1-phase short circuit MVA.

x1r1

Specifies the ratio of X1/R1. Default = 4.0.

x0r0

Specifies the ratio of X0/R0. Default = 3.0.

Case III: Fault and Dynamic

- Linecode.dss file

OpenDSS Code:

```
New "LineCode.3x_336aa_1/0asn" nphases=3 baseFreq=60 normamps=396 emergamps=495  
r1=0.05284 x1=0.12447 r0=0.14504 x0=0.41903 c1=3.30053 c0=1.43804 units=kft
```

...

Explanation:

LineCode.3x_336aa_1/0asn

Defines a linecode object named 3x_336aa_1/0asn.

nphases

Specifies the number of phases. Default = 3.

BaseFreq

Specifies the Base Frequency at which the impedance values are specified. Default = 60.0 Hz.

Normamps

Specifies the normal ampacity, in amps.

emergamps

Specifies the emergency ampacity, in amps.

357

Case III: Fault and Dynamic

- Linecode.dss file

OpenDSS Code:

```
New "LineCode.3x_336aa_1/0asn" nphases=3 baseFreq=60 normamps=396 emergamps=495  
r1=0.05284 x1=0.12447 r0=0.14504 x0=0.41903 c1=3.30053 c0=1.43804 units=kft
```

...

Explanation:

r1

Specifies the positive-sequence resistance, in ohms per unit length.

x1

Specifies the positive-sequence reactance, in ohms per unit length.

r0

Specifies the zero-sequence resistance, in ohms per unit length.

x0

Specifies the zero-sequence reactance, in ohms per unit length.

Case III: Fault and Dynamic

- Linecode.dss file

OpenDSS Code:

```
New "LineCode.3x_336aa_1/0asn" nphases=3 baseFreq=60 normamps=396 emergamps=495  
r1=0.05284 x1=0.12447 r0=0.14504 x0=0.41903 c1=3.30053 c0=1.43804 units=kft
```

...

Explanation:

c1

Specifies the positive-sequence capacitance, in nanofarads per unit length.

c0

Specifies the zero-sequence capacitance, in nanofarads per unit length.

units

Specifies the length units. If not specified, it is assumed that the units correspond to the length being used in the Line models.

{mi | km | kft | m | ft | in | cm}.

Case III: Fault and Dynamic

- Lodeshape.dss file

OpenDSS Code:

```
New "LoadShape.wind2400" npts=2501 interval=0.000277778 csvfile=WPWind2400.csv  
action=normalize
```

Explanation:

New "LoadShape.wind2400"

Defines a loadshape object named wind2400.

Npts

Specifies the number of points to expect when defining the curve.

interval

Specifies the time interval of the data, in Hr. Default=1.0. If the load shape has non-uniformly spaced points, specify the interval as 0.0.

csvfile

Specifies the name of a CSV file containing active power load shape data, one interval to a line.

Case III: Fault and Dynamic

- Lodeshape.dss file

OpenDSS Code:

```
New "LoadShape.wind2400" npts=2501 interval=0.000277778 csvfile=WPWind2400.csv  
action=normalize
```

Explanation:

action

{**Normalize** | **DblSave** | **SngSave**} After defining load curve data, setting **action=normalize** will modify the multipliers, so that the peak is 1.0. The mean and std deviation are recomputed.

mean

Specifies the mean of the multiplier array.

stddev

Specifies the standard deviation of the multiplier array.

Case III: Fault and Dynamic

- TCC_Curve.dss file

OpenDSS Code:

```
New "TCC_Curve.a" npts=15 C_array=(1.1, 1.3, 1.5, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 50, 100, )  
T_array=(7.1769, 2.2079, 1.234, 0.534, 0.2215, 0.134, 0.0965, 0.0769, 0.0653, 0.0578, 0.0528,  
0.0492, 0.0378, 0.0346, 0.0342, )
```

Explanation:

A TCC_Curve object is defined similarly to Loadshape and Growthshape objects in that they all are defined by curves consisting of arrays of points.

New "TCC_Curve.a"

Defines a TCC_curve object named a.

Npts

Number of points to expect when defining the curve.

Case III: Fault and Dynamic

- TCC_Curve.dss file

OpenDSS Code:

```
New "TCC_Curve.a" npts=15 C_array=(1.1, 1.3, 1.5, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 50, 100, )  
T_array=(7.1769, 2.2079, 1.234, 0.534, 0.2215, 0.134, 0.0965, 0.0769, 0.0653, 0.0578, 0.0528,  
0.0492, 0.0378, 0.0346, 0.0342, )
```

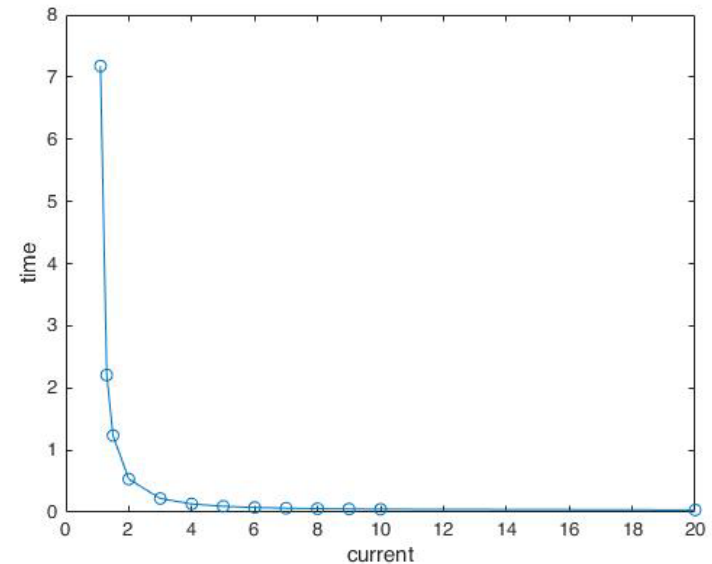
Explanation:

C_Array

Specifies an array of current (or voltage or whatever) values corresponding to time values in T_Array.

T_Array

Specifies an array of time values in seconds. Typical array syntax:
t_array = (1, 2, 3, 4, ...)



363

Case III: Fault and Dynamic

- Line.dss file

OpenDSS Code:

```
New "line.l1" bus1=B0 bus2=B1 linecode="Unbalanced 336 ACSR" length=10 units=kft
```

Explanation:

New Line.l1

Defines a line object named l1.

Phases

Specifies the number of phases. Default = 3.

Bus1

Specifies the name of bus for terminal 1.

Bus2

Specifies the name of bus for terminal 2.

Case III: Fault and Dynamic

- Line.dss file

OpenDSS Code:

```
New "line.l1" bus1=B0 bus2=B1 linecode="Unbalanced 336 ACSR" length=10 units=kft
```

Explanation:

LineCode

Specifies the name of an existing LineCode object containing impedance definitions.

Length

Specifies the length multiplier to be applied to the impedance data.

Units

Specifies the length units. {none | mi | kft | km | m | Ft | in | cm}.

Case III: Fault and Dynamic

- Load.dss file

OpenDSS Code:

```
New "Load.ldb0" bus1=B0 conn=wye phases=3 kW=0.1 pf=1 model=2 kV=12.47
```

Explanation:

New Load.ldb0

Defines a load object named ldb0.

bus1

Specifies the name of bus to which the load is connected.

conn

Specifies the connection of load. {wye | y | LN} for Wye (Line-Neutral) connection; {delta | LL} for Delta (Line-Line) connection. Default = wye.

phases

Specifies the number of phases of this load.

kW

Specifies the nominal active power, in kW, for the load. Total of all phases.

Case III: Fault and Dynamic

- Load.dss file

OpenDSS Code:

```
New "Load.ldb0" bus1=B0 conn=wye phases=3 kW=0.1 pf=1 model=2 kV=12.47
```

Explanation:

pf

Specifies the nominal power factor for load. Negative PF is leading. Specify either PF or kvar. If both are specified, the last one specified takes precedence.

Model

An integer defining how the load will vary with voltage. The load models currently implemented are:

- 1: Constant P and constant Q (Default): Commonly used for power flow studies,
- 2: Constant Z (or constant impedance),
- 3: Constant P and quadratic Q ($Q/Q_0 = (V/V_0)^2$),
- 4: Exponential: $P/P_0 = (V/V_0)^{CVRwatts}$ and $Q/Q_0 = (V/V_0)^{CVRvars}$
- 5: Constant I (or constant current magnitude),

...

367

Case III: Fault and Dynamic

- Load.dss file

OpenDSS Code:

```
New "Load.ldb0" bus1=B0 conn=wye phases=3 kW=0.1 pf=1 model=2 kV=12.47
```

Explanation:

Model

...

6: Constant P and fixed Q (at the nominal value)

7: Constant P and quadratic Q (i.e., fixed reactance),

8: ZIP

"Constant" power value (either P or Q) may be modified by loadshape multipliers.

"Fixed" power values are always the same.

- ZIP model:

$$P = P_0 \left[a_1 \left(\frac{V}{V_0} \right)^2 + a_2 \left(\frac{V}{V_0} \right) + a_3 \right]$$

$$Q = Q_0 \left[a_4 \left(\frac{V}{V_0} \right)^2 + a_5 \left(\frac{V}{V_0} \right) + a_6 \right]$$

Case III: Fault and Dynamic

- Load.dss file

OpenDSS Code:

```
New "Load.ldb0" bus1=B0 conn=wye phases=3 kW=0.1 pf=1 model=2 kV=12.47
```

Explanation:

kV

Specifies the Base voltage for load. For 2- or 3-phase loads, it is specified in phase-to-phase kV.

For all other loads, it is specified in the actual kV across the load branch. If the load is wye (star) connected, then specify phase-to-neutral (L-N). If the load is delta or phase-to-phase connected, specify the phase-to-phase (L-L) kV.

Case III: Fault and Dynamic

- Transformer.dss file

OpenDSS Code:

```
New "Transformer.sub" phases=3 windings=2 buses=(bx, b0, ) conns=(delta, wye, ) kVs=(115, 12.47, ) kVAs=(20000, 20000, ) taps=(1, 1, ) Xhl=7
```

Explanation:

New "Transformer.sub"

Defines a transformer object named sub.

phases

Specifies the number of phases. Default is 3.

windings

Specifies the number of windings. Default is 2.

buses=(bx, b0,)

Specifies an array of names of buses to which the windings are connected. The above code equals: bus1=bx bus2=b0

bus1: Name of bus for terminal 1.

bus2: Name of bus for terminal 2.

Case III: Fault and Dynamic

- Transformer.dss file

OpenDSS Code:

```
New "Transformer.sub" phases=3 windings=2 buses=(bx, b0, ) conns=(delta, wye, ) kVs=(115, 12.47, ) kVAs=(20000, 20000, ) taps=(1, 1, ) Xhl=7
```

Explanation:

conns=(delta, wye,)

Specifies an array of connections for windings. The connection of winding1 is 'delta'. The connection of winding2 is 'wye'.

kvs=(115, 12.47,)

Specifies an array of kV ratings for windings. The rated voltage of winding1 is 115 kV. The rated voltage of winding2 is 12.47kV.

kvas=(20000, 20000,)

Specifies an array of kVA ratings for windings. The base kVA rating of winding1 is 20000 kVA. The base kVA rating of winding2 is 20000kVA.

Case III: Fault and Dynamic

- Transformer.dss file

OpenDSS Code:

```
New "Transformer.sub" phases=3 windings=2 buses=(bx, b0, ) conns=(delta, wye, ) kVs=(115,  
12.47, ) kVAs=(20000, 20000, ) taps=(1, 1, ) Xhl=7
```

Explanation:

taps(1, 1,)

Specifies an array of tap positions for windings. The tap position of winding1 is 1. The tap position of winding2 is 1.

xhl

Specifies the percent reactance high-to-low (winding 1 to winding 2).

Case III: Fault and Dynamic

- RegControl.dss file

OpenDSS Code:

```
New "RegControl.reg1a" transformer=Reg1a winding=2 vreg=119.9926 band=3 ptratio=60  
delay=15 tapdelay=2 reversible=yes revvreg=119.9926 revband=3 CTprim=600 R=5 X=3  
revR=5 revX=3
```

Explanation:

New "RegControl.reg1a"

Defines a regulator control object named reg1a.

transformer

Specifies the name of transformer to which the RegControl is connected.

winding

Specifies the number of the winding of the transformer that the RegControl is monitoring. 1 or 2, typically.

vreg

Specifies the voltage regulator setting, in volts, for the winding being controlled.
Default is 120.0

Case III: Fault and Dynamic

- RegControl.dss file

OpenDSS Code:

```
New "RegControl.reg1a" transformer=Reg1a winding=2 vreg=119.9926 band=3 ptratio=60  
delay=15 tapdelay=2 reversible=yes revvreg=119.9926 revband=3 CTprim=600 R=5 X=3  
revR=5 revX=3
```

Explanation:

band

Specifies the bandwidth, in volts, for the controlled bus. Default is 3.0

ptratio

Specifies the ratio of the PT that converts the controlled winding voltage to the regulator voltage. Default is 60. If the winding is Wye, the line-to-neutral voltage is used. Else, the line-to-line voltage is used.

delay

Specifies the time delay, in seconds, from when the voltage goes out of band to when the tap changing begins. This is used to determine which regulator control will act first. Default is 15.

Case III: Fault and Dynamic

- RegControl.dss file

OpenDSS Code:

```
New "RegControl.reg1a" transformer=Reg1a winding=2 vreg=119.9926 band=3 ptratio=60  
delay=15 tapdelay=2 reversible=yes revvreg=119.9926 revband=3 CTprim=600 R=5 X=3  
revR=5 revX=3
```

Explanation:

tapdelay

Specifies the delay, in seconds, between tap changes. Default is 2. This is how long it takes between changes after the first change.

reversible

Indicates whether or not the regulator can be switched to regulate in the reverse direction. Default is No. {Yes |No*}

revvreg

Specifies the voltage setting, in volts, for operation in the reverse direction.

Case III: Fault and Dynamic

- RegControl.dss file

OpenDSS Code:

```
New "RegControl.reg1a" transformer=Reg1a winding=2 vreg=119.9926 band=3 ptratio=60  
delay=15 tapdelay=2 reversible=yes revvreg=119.9926 revband=3 CTprim=600 R=5 X=3  
revR=5 revX=3
```

Explanation:

revband

Specifies the bandwidth for operating in the reverse direction.

CTprim

Specifies the rating, in Amperes, of the primary CT rating for converting the line amps to control amps. The typical default secondary ampere rating is 0.2 Amps.

R

Specifies the R setting on the line drop compensator in the regulator, in volts.

Case III: Fault and Dynamic

- RegControl.dss file

OpenDSS Code:

```
New "RegControl.reg1a" transformer=Reg1a winding=2 vreg=119.9926 band=3 ptratio=60  
delay=15 tapdelay=2 reversible=yes revvreg=119.9926 revband=3 CTprim=600 R=5 X=3  
revR=5 revX=3
```

Explanation:

X

Specifies the X setting on the line drop compensator in the regulator, in volts.

revR

Specifies the R line drop compensator setting for reverse direction, in volts.

revX

Specifies the X line drop compensator setting for reverse direction, in volts.

Case III: Fault and Dynamic

- Capacitor.dss file

OpenDSS Code:

```
New "Capacitor.c1" conn=wye bus1=B1 phases=3 kvar=( 600) kv=12.47
```

Explanation:

New "Capacitor.c1"

Defines a capacitor object named c1.

Bus1

Specifies the name of bus to which the first bus is connected.

conn

Specifies the connection of bank. One of {wye | ln} for wye connected banks or {delta | ll} for delta (line-line) connected banks.

phases

Specifies the number of phases.

Case III: Fault and Dynamic

- Capacitor.dss file

OpenDSS Code:

```
New "Capacitor.c1" conn=wye bus1=B1 phases=3 kvar=( 600) kv=12.47
```

Explanation:

kVAR

Specifies the rated kvar at rated kV, total of all phases. Each phase is assumed equal.

kV

Specifies the kV rating of the capacitor bank. For Phases=2 or Phases=3, enter line-to-line (phase-to-phase) rated voltage. For all other numbers of phases, enter actual can rating. (For Delta connection this is always line-to-line rated voltage).

Case III: Fault and Dynamic

- Fault.dss file

OpenDSS Code:

```
New "Fault.thefault" bus1=B3.1 bus2=B3.0 phases=1 r=0.0001 ONtime=.3 temporary=yes
```

Explanation:

A fault object is nothing more than a resistor network with a variety of configurations.

New "Fault.thefault"

Defines a fault object named thefault.

bus1

Specifies the name of first bus the resistor is connected to.

bus2

Specifies the name of second bus the resistor is connected to.

phases

Specifies the number of phases. Default is 1.

Case III: Fault and Dynamic

- Fault.dss file

OpenDSS Code:

```
New "Fault.thefault" bus1=B3.1 bus2=B3.0 phases=1 r=0.0001 ONtime=.3 temporary=yes
```

Explanation:

r

Specifies the resistance of each phase, in ohms. Default is 0.0001.

ontime

Specifies the time, in seconds, at which the fault is established for time varying simulations.

Temporary

Designates whether the fault is temporary or not.
{Yes | No} Default is No.

Case III: Fault and Dynamic

- IndMach012.dss file

OpenDSS Code:

```
New "IndMach012.Motor1" bus1=Bg2 kW=1200 conn=delta kVA=1500.000 H=6  
duty=Wind2400 purs=0.048 puxs=0.075 purr=0.018 puxr=0.12 puxm=3.8 slip=0.02  
SlipOption=variableslip kv=0.48
```

Explanation:

INDMACH012 is an induction machine (asynchronous machine) model.

New "IndMach012.Motor1"

Defines an IndMach012 object named Motor1.

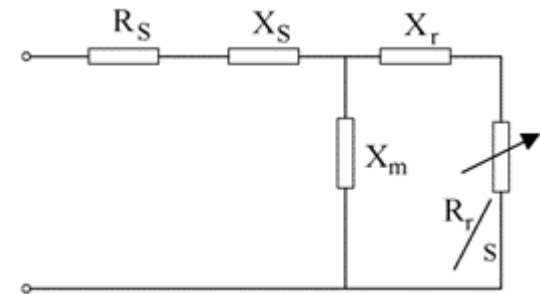
bus1

Specifies the name of bus to which the machine is connected.

kW

Specifies the shaft power, in kW, for the Induction Machine. A positive value denotes power for a load. Negative value denotes an induction generator. Total of all phases.

$$\begin{cases} \frac{dE'_d}{dt} = -\frac{1}{T'} [E'_d + (X - X')I_q] - (\omega - 1)E'_q \\ \frac{dE'_q}{dt} = -\frac{1}{T'} [E'_q + (X - X')I_d] + (\omega - 1)E'_d \\ \frac{d\omega}{dt} = -\frac{1}{2H} [(A\omega^2 + B\omega + C)T_0 - (E'_d I_d + E'_q I_q)] \\ I_d = \frac{1}{R_s^2 + X'^2} [R_s (U_d - E'_d) + X' (U_q - E'_q)] \\ I_q = \frac{1}{R_s^2 + X'^2} [R_s (U_q - E'_q) - X' (U_d - E'_d)] \end{cases}$$



⁵⁸²
<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1626370>

Case III: Fault and Dynamic

- IndMach012.dss file

OpenDSS Code:

```
New "IndMach012.Motor1" bus1=Bg2 kW=1200 conn=delta kVA=1500.000 H=6  
duty=Wind2400 purs=0.048 puxs=0.075 purr=0.018 puxr=0.12 puxm=3.8 slip=0.02  
SlipOption=variableslip kv=0.48
```

Explanation:

conn

Specifies the connection of stator: Delta or Wye. Default is Delta.

kVA

Specifies the rated kVA for the machine.

H

Specifies the per unit mass constant of the machine. In MW-sec/MVA.
Default is 1.0.

duty

Specifies the load shape to use for simulations.

Case III: Fault and Dynamic

- IndMach012.dss file

OpenDSS Code:

```
New "IndMach012.Motor1" bus1=Bg2 kW=1200 conn=delta kVA=1500.000 H=6  
duty=Wind2400 purs=0.048 puxs=0.075 purr=0.018 puxr=0.12 puxm=3.8 slip=0.02  
SlipOption=variableslip kv=0.48
```

Explanation:

purs

Specifies the per unit stator resistance. Default is 0.0053.

puxs

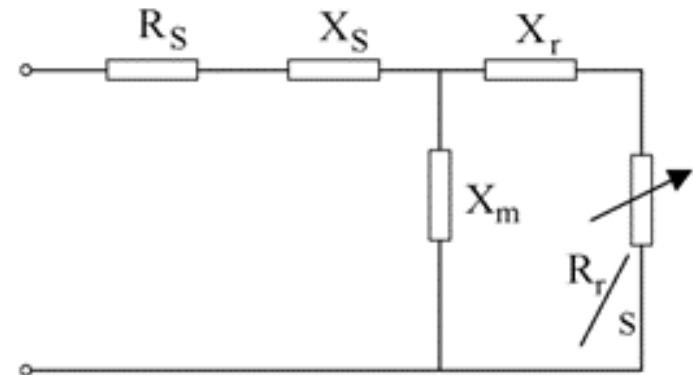
Specifies the per unit stator leakage reactance. Default is 0.106.

purr

Specifies the per unit rotor resistance. Default is 0.007.

puxr

Specifies the per unit rotor leakage reactance. Default is 0.12.



Case III: Fault and Dynamic

- IndMach012.dss file

OpenDSS Code:

```
New "IndMach012.Motor1" bus1=Bg2 kW=1200 conn=delta kVA=1500.000 H=6  
duty=Wind2400 purs=0.048 puxs=0.075 purr=0.018 puxr=0.12 puxm=3.8 slip=0.02  
SlipOption=variableslip kv=0.48
```

Explanation:

puxm

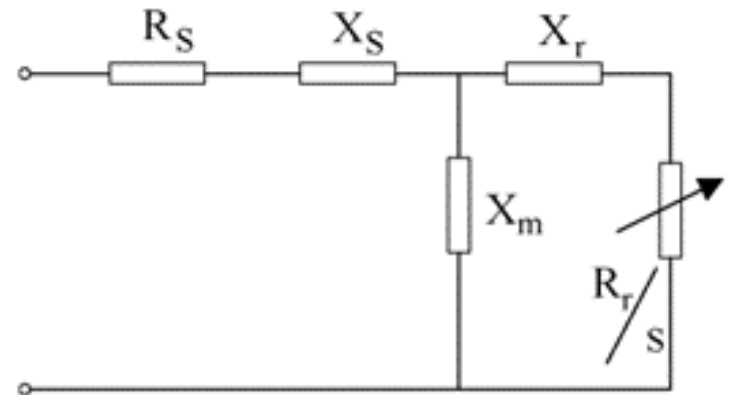
Specifies the per unit magnetizing reactance. Default is 4.0.

slip

Specifies the initial slip value. Default is 0.007.

SlipOption

Specifies the option for slip model. One of {fixedslip | variableslip* }. Variable slip is the default and slip will be computed to satisfy the power specification. If fixed slip, kW is computed to match the slip specification.



Case III: Fault and Dynamic

- IndMach012.dss file

OpenDSS Code:

```
New "IndMach012.Motor1" bus1=Bg2 kW=1200 conn=delta kVA=1500.000 H=6  
duty=Wind2400 purs=0.048 puxs=0.075 purr=0.018 puxr=0.12 puxm=3.8 slip=0.02  
SlipOption=variableslip kv=0.48
```

Explanation:

kv

Specifies the Base voltage for the induction machine. For 2- or 3-phase generators, specified in phase-to-phase kV. For all other generators, the actual kV across the generator branch. If wye (star) connected, specify the phase-to-neutral (L-N) kV. If delta or phase-to-phase connected, specify the phase-to-phase (L-L) kV.

Case III: Fault and Dynamic

- Relay.dss file

OpenDSS Code:

```
New "Relay.mfrov/uv" MonitoredObj=line.MotorLeads MonitoredTerm=1 type=voltage kvbase=0.48 Delay=0
New "Relay.mfr46" MonitoredObj=line.MotorLeads MonitoredTerm=1 type=46 46BaseAmps=1800
46%Pickup=20 46isqt=1 Delay=0.1
New "Relay.mfr47" MonitoredObj=line.MotorLeads MonitoredTerm=1 type=47 kvbase=0.48 47%Pickup=2
Delay=0.1
```

Explanation:

Relays are used to control distribution circuit breakers. The inverse time overcurrent characteristics is commonly used. An inverse time-current characteristic means that the relay will operate faster with increased current.

New "Relay.mfrov/uv"

Defines a relay object named mfrov/uv.

MonitoredObj

Specifies the full object name of the circuit element, typically a line, transformer, load, or generator, to which the relay's PT and/or CT are connected. This is the "monitored" element. There is no default; must be specified. 387

Case III: Fault and Dynamic

- Relay.dss file

OpenDSS Code:

```
New "Relay.mfrov/uv" MonitoredObj=line.MotorLeads MonitoredTerm=1 type=voltage kvbase=0.48 Delay=0
New "Relay.mfr46" MonitoredObj=line.MotorLeads MonitoredTerm=1 type=46 46BaseAmps=1800
46%Pickup=20 46isqt=1 Delay=0.1
New "Relay.mfr47" MonitoredObj=line.MotorLeads MonitoredTerm=1 type=47 kvbase=0.48 47%Pickup=2
Delay=0.1
```

Explanation:

MonitoredTerm

Specifies the number of the terminal of the circuit element to which the Relay is connected. 1 or 2, typically. Default is 1.

type

Specifies the type of the relay. Voltage: undervoltage and overvoltage protection.
46: negative sequence current protection.
47: negative sequence voltage protection.

Case III: Fault and Dynamic

- Relay.dss file

OpenDSS Code:

```
New "Relay.mfrov/uv" MonitoredObj=line.MotorLeads MonitoredTerm=1 type=voltage kvbase=0.48 Delay=0
New "Relay.mfr46" MonitoredObj=line.MotorLeads MonitoredTerm=1 type=46 46BaseAmps=1800
46%Pickup=20 46isqt=1 Delay=0.1
New "Relay.mfr47" MonitoredObj=line.MotorLeads MonitoredTerm=1 type=47 kvbase=0.48 47%Pickup=2
Delay=0.1
```

Explanation:

kvbase

Specifies the voltage base (kV) for the relay. Specify line-line for 3 phase devices); line-neutral for 1-phase devices.

delay

Specifies the trip time delay, in seconds, for definite time relays.

46BaseAmps

Specifies the Base current, in amps, for 46 relay (neg seq current).

Case III: Fault and Dynamic

- Relay.dss file

OpenDSS Code:

```
New "Relay.mfrov/uv" MonitoredObj=line.MotorLeads MonitoredTerm=1 type=voltage kvbase=0.48 Delay=0
New "Relay.mfr46" MonitoredObj=line.MotorLeads MonitoredTerm=1 type=46 46BaseAmps=1800
46%Pickup=20 46isqt=1 Delay=0.1
New "Relay.mfr47" MonitoredObj=line.MotorLeads MonitoredTerm=1 type=47 kvbase=0.48 47%Pickup=2
Delay=0.1
```

Explanation:

46%Pickup

Specifies the percent pickup current for 46 relay (neg seq current). Default is 20.0.

46isqt

Specifies the negative-sequence I-squared-t trip value for 46 relay. Default is 1 (trips in 1 sec for 1 per unit neg-seq current). Should be 1 to 99.

47%Pickup

Specifies the percent voltage pickup for 47 relay (Neg seq voltage). Default is 2.

Case III: Fault and Dynamic

- Recloser.dss file

OpenDSS Code:

```
New "Recloser.cb1" MonitoredObj=Line.L1 MonitoredTerm=1 NumFast=4 PhaseFast=Ext_Inv  
PhaseDelayed=Ext_Inv PhaseTrip=800 TDPHFast=1 TDPHDelayed=1 PhaseInst=2400  
GroundFast=Ext_Inv GroundDelayed=Ext_Inv GroundTrip=800 TDGrFast=1 TDGrDelayed=1  
GroundInst=1200 Shots=4 RecloseIntervals=(0.5, 2, 2, )
```

A recloser is a self-controlled device for automatically *interrupting* and *reclosing* an alternating-current circuit, with a predetermined *sequence* of opening and reclosing followed by resetting, hold closed, or lockout.



Case III: Fault and Dynamic

- Recloser.dss file

OpenDSS Code:

```
New "Recloser.cb1" MonitoredObj=Line.L1 MonitoredTerm=1 NumFast=4 PhaseFast=Ext_Inv  
PhaseDelayed=Ext_Inv PhaseTrip=800 TDPhFast=1 TDPhDelayed=1 PhaseInst=2400  
GroundFast=Ext_Inv GroundDelayed=Ext_Inv GroundTrip=800 TDGrFast=1 TDGrDelayed=1  
GroundInst=1200 Shots=4 RecloseIntervals=(0.5, 2, 2, )
```

Explanation:

New "Recloser.cb1"

Defines a recloser object named cb1.

MonitoredObj

Specifies the full object name of the circuit element, typically a line, transformer, load, or generator, to which the Recloser's PT and/or CT are connected. This is the "monitored" element. There is no default; must be specified.

MonitoredTerm

Specifies the number of the terminal of the circuit element to which the Recloser is connected. 1 or 2, typically. Default is 1.

Case III: Fault and Dynamic

- Recloser.dss file

OpenDSS Code:

```
New "Recloser.cb1" MonitoredObj=Line.L1 MonitoredTerm=1 NumFast=4 PhaseFast=Ext_Inv  
PhaseDelayed=Ext_Inv PhaseTrip=800 TDPhFast=1 TDPhDelayed=1 PhaseInst=2400  
GroundFast=Ext_Inv GroundDelayed=Ext_Inv GroundTrip=800 TDGrFast=1 TDGrDelayed=1  
GroundInst=1200 Shots=4 RecloseIntervals=(0.5, 2, 2, )
```

Explanation:

NumFast

Specifies the number of Fast (fuse saving) operations. Default is 1.

Fast and delayed operation: The trip operations of the recloser can be all fast, all delayed, or any combination of fast operations, followed by delayed operations up to a maximum total of four. Fast operations clear temporary faults before branch line fuses are damaged. Delayed operations allow time for fuses or other downline protective devices to clear to limit permanent faults to the smallest section of line.

phaseFast

Specifies the name of the TCC Curve object that determines the Phase Fast trip.

Case III: Fault and Dynamic

- Recloser.dss file

OpenDSS Code:

```
New "Recloser.cb1" MonitoredObj=Line.L1 MonitoredTerm=1 NumFast=4 PhaseFast=Ext_Inv  
PhaseDelayed=Ext_Inv PhaseTrip=800 TDPhFast=1 TDPhDelayed=1 PhaseInst=2400  
GroundFast=Ext_Inv GroundDelayed=Ext_Inv GroundTrip=800 TDGrFast=1 TDGrDelayed=1  
GroundInst=1200 Shots=4 RecloseIntervals=(0.5, 2, 2, )
```

Explanation:

PhaseDelayed

Specifies the name of the TCC Curve object that determines the Phase Delayed trip.

PhaseTrip

Specifies a Multiplier or actual phase amps for the phase TCC curve. Defaults to 1.0.

TDPhFast

Specifies a time dial for Phase Fast trip curve. It is a multiplier on time axis of specified curve. Default=1.0.

TDPhDelayed

Specifies a time dial for Phase Delayed trip curve. It is a multiplier on time axis of specified curve. Default=1.0.

394

Case III: Fault and Dynamic

- Recloser.dss file

OpenDSS Code:

```
New "Recloser.cb1" MonitoredObj=Line.L1 MonitoredTerm=1 NumFast=4 PhaseFast=Ext_Inv  
PhaseDelayed=Ext_Inv PhaseTrip=800 TDPhFast=1 TDPhDelayed=1 PhaseInst=2400  
GroundFast=Ext_Inv GroundDelayed=Ext_Inv GroundTrip=800 TDGrFast=1 TDGrDelayed=1  
GroundInst=1200 Shots=4 RecloseIntervals=(0.5, 2, 2, )
```

Explanation:

PhaseInst

Specifies the actual amps (Current relay) or kW (reverse power relay) for instantaneous phase trip which is assumed to happen in 0.01 sec + Delay Time. Default is 0.0, which signifies no instantaneous trip.

GroundFast

Specifies the name of the TCC Curve object that determines the Ground Fast trip.

GroundDelayed

Specifies the name of the TCC Curve object that determines the Ground Delayed trip.

Case III: Fault and Dynamic

- Recloser.dss file

OpenDSS Code:

```
New "Recloser.cb1" MonitoredObj=Line.L1 MonitoredTerm=1 NumFast=4 PhaseFast=Ext_Inv  
PhaseDelayed=Ext_Inv PhaseTrip=800 TDPhFast=1 TDPhDelayed=1 PhaseInst=2400  
GroundFast=Ext_Inv GroundDelayed=Ext_Inv GroundTrip=800 TDGrFast=1 TDGrDelayed=1  
GroundInst=1200 Shots=4 RecloseIntervals=(0.5, 2, 2, )
```

Explanation:

GroundTrip

Specifies a multiplier or actual ground amps (3I0) for the ground TCC curve. Defaults to 1.0.

TDGrFast

Specifies the time dial for Ground Fast trip curve. It is a multiplier on time axis of specified curve. Default=1.0.

TDGrDelayed

Specifies the time dial for Ground Delayed trip curve. It is a multiplier on time axis of specified curve. Default=1.0.

Case III: Fault and Dynamic

- Recloser.dss file

OpenDSS Code:

```
New "Recloser.cb1" MonitoredObj=Line.L1 MonitoredTerm=1 NumFast=4 PhaseFast=Ext_Inv  
PhaseDelayed=Ext_Inv PhaseTrip=800 TDPHFast=1 TDPHDelayed=1 PhaseInst=2400  
GroundFast=Ext_Inv GroundDelayed=Ext_Inv GroundTrip=800 TDGrFast=1 TDGrDelayed=1  
GroundInst=1200 Shots=4 RecloseIntervals=(0.5, 2, 2, )
```

Explanation:

GroundInst

Specifies the actual amps for instantaneous ground trip which is assumed to happen in 0.01 sec + Delay Time. Default is 0.0, which signifies no instantaneous trip.

shots

Specifies the total Number of fast and delayed shots to lockout. Default is 4. This is one more than the number of reclose intervals.

RecloseIntervals

Specifies an array of reclose intervals. Default for Recloser is (0.5, 2.0, 2.0) seconds.

397

Case III: Fault and Dynamic

- Fuse.dss file

OpenDSS Code:

```
New "Fuse.f1" MonitoredObj=Line.L6 MonitoredTerm=1 FuseCurve=Klink RatedCurrent=65
```

Explanation:

- Fuses have elements that melt if enough current flows through it for enough time.
- Industry standards specify two types of expulsion fuses.
 - K-type: fast fuse with speed ratio of 6-8
 - T-type: slower fuse with speed ratio of 10-13
- The speed ratio is the ratio of
 - The melting current at 0.1 second to
 - the melting current at X seconds, where X is 300 for fuse ratings below 100 amps and X is 600 for fuse ratings above 100 amps.

The current rating is the level of current the fuse can safely carry for an indefinite period of time.

Case III: Fault and Dynamic

- Fuse.dss file

OpenDSS Code:

```
New "Fuse.f1" MonitoredObj=Line.L6 MonitoredTerm=1 FuseCurve=Klink RatedCurrent=65
```

Explanation:

New "Fuse.f1"

Defines a fuse object named f1.

MonitoredObj

Specifies the full object name of the circuit element, typically a line, transformer, load, or generator, to which the Fuse is connected. This is the "monitored" element. There is no default; must be specified.

MonitoredTerm

Specifies the number of the terminal of the circuit element to which the Fuse is connected. 1 or 2, typically. Default is 1.

Case III: Fault and Dynamic

- Fuse.dss file

OpenDSS Code:

```
New "Fuse.f1" MonitoredObj=Line.L6 MonitoredTerm=1 FuseCurve=Klink RatedCurrent=65
```

Explanation:

Fusecurve

Specifies the name of the TCC Curve object that determines the fuse blowing.

RatedCurrent

Specifies a multiplier or actual phase amps for the phase TCC curve. Defaults to 1.0.

Case III: Fault and Dynamic

- Monitor.dss file

OpenDSS Code:

```
New "Monitor.cb1" element=Line.L1 terminal=1 mode=0
```

Explanation:

New "Monitor.cb1"

Defines a monitor object named cb1.

element

Specifies the name of an existing circuit element to which the monitor is to be connected.

terminal

Specifies the number of the terminal to which the monitor will be connected.

mode

An integer bitmask code to describe what the monitor will save. Monitors can save two basic types of quantities: 1) Voltage and current; 2) Power. The Mode codes are defined as follows:

0: Standard mode - V and I, each phase, complex

...

Case III: Fault and Dynamic

- Monitor.dss file

OpenDSS Code:

```
New "Monitor.cb1" element=Line.L1 terminal=1 mode=0
```

Explanation:

mode

...

1: Power each phase, complex (kw and kvars)

2: Transformer taps (connect Monitor to a transformer winding)

3: State variables (connect Monitor to a PCElement)

4: Flicker level and severity index (Pst) for voltages.

5: Solution variables (Iterations, etc). Normally, these would be actual phasor quantities from solution.

6: Capacitor Switching

+16: Sequence components: V012, I012

+32: Magnitude only

+64: Positive Sequence only or Average of phases, if not 3 phases.

For example, Mode=33 (33=1+32) will save the magnitude of the power (kVA) only in each phase. Mode=112 (0+16+32+64) saves Positive sequence voltages and currents, magnitudes only.

402

Thank you!