**Primer for Conversion of OpenDSS format files to ePHASORsim input files**

A software process is developed to convert quasi-static time series (QSTS) based distribution models (from OpenDSS) to a real time dynamic phasor simulator format (ePHASORsim). In this document, the software process to convert one distribution models format to another is described, which would be helpful for researchers who intend to perform similar conversions. The converter could be utilized directly by Real-time simulator users who intend to perform software-in-loop or hardware-in-loop tests on large distribution test feeders for a range of use cases including testing functions of Advance Distribution Management Systems (ADMS) against a simulated distribution system. In the future, the developers intend to release the conversion tool as open source to enable use by others. This work was supported by the U.S. Department of Energy, Office of Electricity Delivery & Energy Reliability, as part of the Grid Modernization Laboratory Consortium award for the Advanced Distribution Management System (ADMS) Testbed Development.

**Introduction:**

***Opal RT’s ePHASORsim,*** a real time transient stability simulator was chosen as the testbed to deploy a large distribution feeder for implementing the various advanced distribution management system (ADMS) functionalities. “ePHASORsim” has the capability to simulate up to 20,000 buses on 1 CPU core with a time-step of 10 milliseconds in real-time. The real time simulator (RTS) calculates the root mean square (RMS) and angle values of voltage, current at every location/feeder node.

The following built-in positive sequence and 3-phase library available in excel spreadsheets are:

• Synchronous machine

• Excitation system, turbine-governor

• Controllable voltage source

• Controllable current source

• Transformer: OLTC, mutual impedance, unbalanced tap

• Line: PI section, sequence impedance, phase impedance

• Load: Constant power, constant current, constant impedance, ZIP

• Shunt devices

• Breaker, switches, and events

***Opendss*** contains a repository of open access distribution feeder model examples. Amongst them, as an example “Feeder J1”, located in the northeastern US, was selected as the distribution network to perform various control tests on the testbed. It is a 12-kV feeder with 1.7 MW of customer-owned PV systems. The substation serves a peak load of approximately 6 MW, which comprises of approximately 1300 residential, commercial, and light industrial customers. The feeder consists of load tap changer (LTC), multiple feeder regulators and switched capacitor banks to provide voltage regulation [1].

**Opendss “.dss” Files:**

The following individual files are available in large feeder “opendss” files that could be utilized to build the input data spreadsheets to be interfaced with the “ePHASORsim” simulator.

***Note:*** Typically for large feeders all the information is provided in separate “opendss” files as would be discussed below. In case all the feeder information is provided in one “.dss” file, it is recommended to prepare individual files as discussed below.

1. **Line codes:**

A file usually named as **“LineCodes.dss”** contains the overhead (OH), underground (UG) cable and service line configuration information that would be utilized to prepare the 3-phase line spreadsheet. Overhead and underground line configuration types are typically represented in symmetrical (positive, negative and zero sequence) impedance format and the service/cable line configuration types are typically represented in NxN matrix format, where “N” is the number of phases (typically 3). The information of units is utilized to get the ohms/units line impedance information of different configuration types that are extracted from the “LineCodes.dss” file. The units could be km, kft etc.

Using the programming language **Python**, all the line code information data is parsed into a dictionary format. Later on this dictionary is utilized to extract the line configuration information for individual OH, UG and service lines.

An excerpt of the OH and service line codes taken from “LineCodes.dss” is shown below in Figure 1:

|  |
| --- |
| (a)    (b) |
| Figure 1. Excerpt from “LineCodes.dss” for (a) Overhead line codes and (b) Services line codes. |

For example, it can be seen from the above excerpt that:

1. A line code named “OH-3x\_477AACN” for an OH line configuration has the properties of positive sequence impedance as follows: resistance, r1=0.12241009 Ω/km, inductive reactance, x1=0.3949091 Ω/km, capacitive reactance, c1=11.1973 nf/km and zero sequence impedance as follows: r0=0.25708354 Ω/km, inductive reactance, x0=1.1976615 Ω/km, capacitive reactance, c0=4.8372 nf/km for a base frequency of 60 Hz.
2. A line code named “1/0A” for a service line configuration has the properties of

Resistance matrix: rmatrix= Ω/km

Inductive reactance matrix, xmatrix= Ω/km

Capacitive reactance matrix, cmatrix = nf/km for a base frequency of 60 Hz.

***Note:*** In case some “LineCodes” (e.g. for service lines) exist in another file, it is recommended to copy and paste them in the main “LineCodes” file.

1. **OH, UG and Service Lines:**

Files named as **“Lines.dss”** and **“Services.dss”** contains the line information between any two buses. The line information is utilized to prepare the 3-phase line spreadsheet by making use of the line configuration dictionary that was discussed earlier.

An excerpt of the OH information from “Lines.dss” is shown below in Figure 2:

|  |
| --- |
|  |
| Figure 2. Excerpt from “Lines.dss” |

For example, the overhead line ID “OH\_B4784” between buses “bus1=B4787.1.2.3” and “bus2=B4784.1.2.3” of length = 32.308799 m, calls the linecode information “linecode=OH-3X\_477AAC\_4/0AAACN” particularly for the ohms/units information. The line is a 3phase bus connection.

An excerpt of the service line information from “Services.dss” is shown below in Figure 3:

|  |
| --- |
|  |
| Figure 3. Excerpt from “Services.dss” |

For example, the service line ID “B4859-1-A\_Cust3” between buses “bus1=X\_B4859-A.1” and “bus2= X\_B4859\_Cust3-A.1” of length = 0.1 kft, calls the linecode information “linecode=1/0A” specifically for the ohms/units information. The line is a 1-phase bus connection connected between A phases.

1. **Switches:**

Switches are used in distribution systems as disconnectors and provide flexibility in reconfiguration of networks. Switches/circuit breaker information could be available in **“Services.dss”** and **“Substation.dss”.**

An excerpt of switch information from “Services.dss” is shown below in Figure 4:

|  |
| --- |
|  |
| Figure 4. Excerpt from “Services.dss” |

For example, the switch line ID “Site4\_PV” between buses, bus1=B51854\_sec and bus2=B51854\_sec2 is shown in the above excerpt.

1. **Loads:**

Loads are either in 3 phases or single phase format in the distribution systems. Load information is available in **“Loads.dss”**.

An excerpt of load information from “Loads.dss” is shown below in Figure 5:

|  |
| --- |
| …contd. |
| Figure 5. Excerpt from “Loads.dss” |

For example, a load with an ID “B4891-1-C\_Cust3” is connected at bus= X\_B4891-1-Cust3-C.3 as shown in the above excerpt. The single-phase load is connected at phase C and has values of active power=1.720312 kW and reactive power=1.0857987 kVAr at a nominal voltage=0.24 kV.

1. **Transformers:**

The transformer information is available in **“Transformers.dss”** as service transformers at the customer end and could be available in **“Substation.dss”** as the step-down transformer to feed the distribution network.

An excerpt of transformer information from “Transformers.dss” is shown below in Figure 6:

|  |
| --- |
| ……..contd. |
| Figure 6. Excerpt from “Transformers.dss” |

For example, a transformer with an ID “B56446-1A” has its winding 1 connected to bus B56446.1 at 7.2 kV and winding 2 connected to bus X\_B56446-A.1 at 0.24 kV. The transformer is rated at 15 kVA. The resistance is 0.0104 pu and reactance is 0.015 pu.

***Note:*** Upon building the “Transformer 3-phase” excel file spreadsheet, modify the first 2 lines to accommodate windings 0 and 1 information which are the windings 1 and 2 information respectively.

***Important Note:*** OLTC and step voltage regulators are defined as transformers with taps in ePHASORSIM input files. In case regulators exist in a separate file it is recommended to copy them into the transformer files. It just makes life easier.

1. **Capacitors:**

Capacitors could be either connected as 3 phase banks or as individual single phase capacitors. The capacitors information is available in **“Capacitors.dss”**. There is an option to either switch them on/off.

An excerpt of capacitor information from “Capacitors.dss” is shown below in Figure 7:

|  |
| --- |
|  |
| Figure 7. Excerpt from “Capacitors.dss” |

For example, a capacitor with an ID “B4909-1” is connected to bus B4909 at 12.47 kV. The total kVAr for one step is 900kVAr.

1. **Distributed Generation/ PV:**

Usually information of distributed generators and PV are available in files called **“Distributedgen.dss”** and **“ExistingPV.dss”**. The information of bus IDs at which the “current injectors” exists, are extracted from the “Distributedgen.dss” and “ExistingPV.dss” files.

An excerpt of existing PV information from “ExistingPV.dss” and distributed generation from “Distributedgen.dss” is shown below in Figure 8:

|  |
| --- |
| (a)    (b) |
| Figure 8. Excerpt from (a) “ExistingPV.dss” and (b) “Distributedgen.dss”. |

For example, a PV system with an ID: 3P\_ExistingSite1 is connected to bus X\_5865228330A at a nominal voltage of 0.416 kV and rated at 314 kVA. A distributed generation with an ID: DG\_1 is connected to bus ls\_bus.1.2.3 at a nominal voltage of 12.47 KV and rated at 913.24 kW.

1. **Source/Substation:**

Information of source/substation bus and voltage is extracted from **“Substation.dss”.** The information could be typically extracted from the substation transformer.

An excerpt of source information from “Substation.dss” is shown below in Figure 9:

|  |
| --- |
|  |
| Figure 9. Excerpt from “Substation.dss” |

From the substation transformer information it can be seen that the source bus is named S at a nominal voltage of 68.8 kV.

A high-level description of how the OpenDSS files are utilized to create the excel spreadsheets for ePHASORSIM is shown in Figure 10.

ephasorsim_1

Figure 10. Open source model files conversion to real-time simulator interfaced files.

**Preparing Input Data Spreadsheets for the “ePHASORsim” Simulator [2]**

1. A template of Excel file is available in ePHASORsim block set page in MATLAB’s help (Figure 11 below).

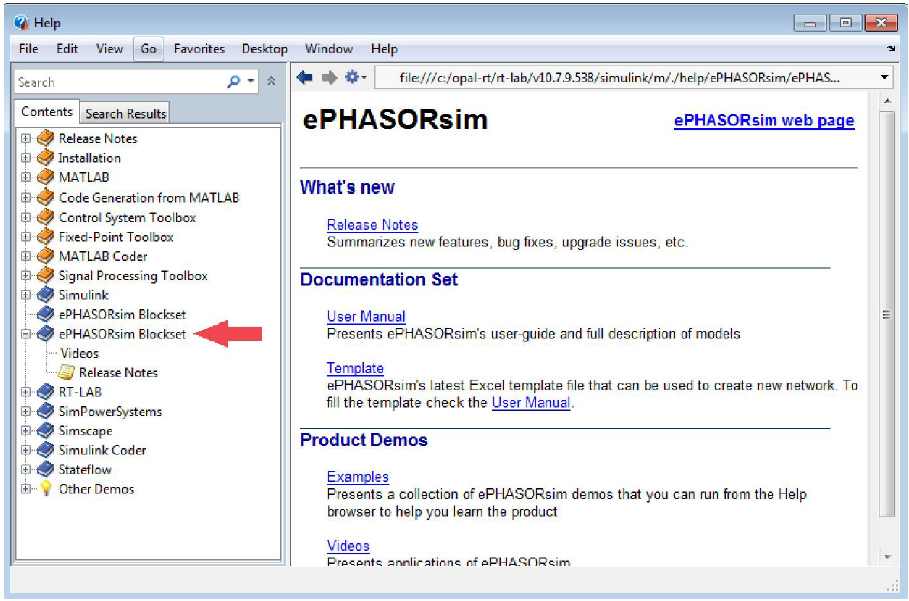


Figure 11: ePHASORsim blockset in MATLAB’s help

1. For creating an excel file, refer to ““Pins”” in their user guide.

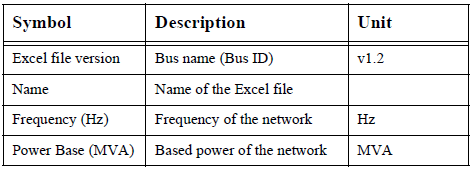
**Excel input file**

An Excel template is provided in ePHASORsim block set page in Matlab’s “Help”. This file can be saved as both XLS and XLSX types. In the Excel workbook, all the components, their required parameters and initial values are defined. A template Excel data file is included in the package. The file includes pages with specific names that must not be modified or changed:

1. **General**

Contains the basic information of the power system and could be simply copied from other example files.

Table 1: General parameters



An excerpt of the “General” sheet in an example main excel file is shown below.

|  |
| --- |
|  |
| Figure 12. Excerpt from “General” sheet in the main excel file. |

1. **Pins**

Outgoing information from the Simulink/RT-Lab blocks and incoming information to the Simulink/RT-Lab blocks is represented in the “Pins” sheet. There are two types of incoming and outgoing (I/O) pins:

• Outgoing: To define measurement probes or status monitoring in the power system (e.g. voltage of a bus, or in-service status of a shunt device).

• Incoming: To send operational commands to the simulator (e.g. applying faults).

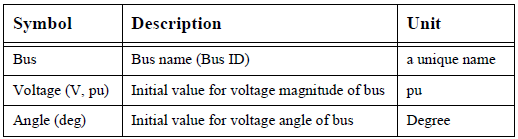
An appendix is provided at the end with the primer with the incoming/outgoing pins for positive sequence components and three-phase components. An excerpt from the “Pins” sheet in an example main excel file is shown below.

|  |
| --- |
|  |
| Figure 13. Excerpt from “Pins” sheet in an example main excel file. |

1. **Bus**

The “Bus” sheet contains all the bus ID information along with voltages and angle information for the initialization of the simulation run.

Table 2: Bus parameters

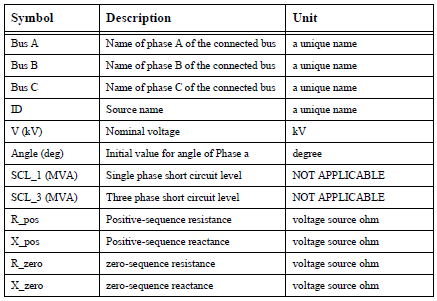


|  |
| --- |
|  |
| Figure 14. Excerpt from “Bus” sheet in the main excel file. |

1. **Vsource 3-phase**

The information of substation bus and voltage with its series impedance is provided in the “Vsource 3-phase” sheet.

Table 3: Vsource 3-phase parameters



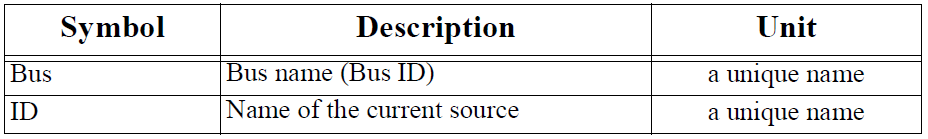
An excerpt from the “Vsource 3-phase” sheet in an example main excel file is shown below.

|  |
| --- |
|  |
| Figure 15. Excerpt from “Vsource 3-phase” sheet in an example main excel file.. |

1. **Current Injector**

The buses along with IDs of the current sources such as distributed generators (DGs) and PVs are described in the “Current Injector” sheet.

Table 4: Current injector parameters

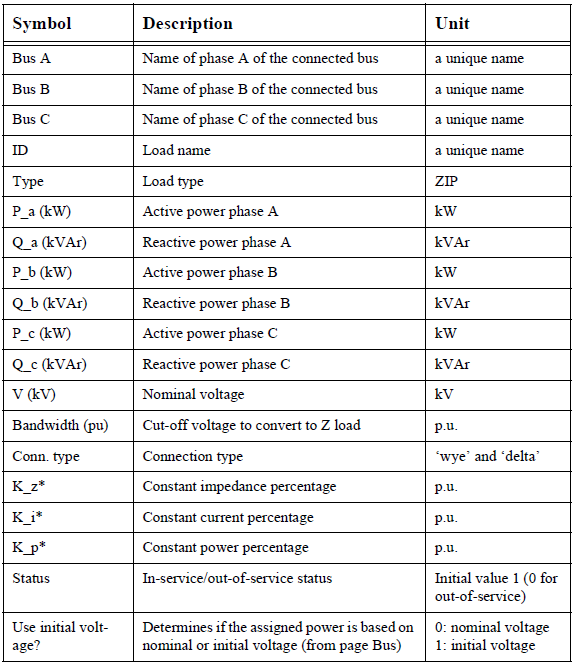


|  |
| --- |
|  |
| Figure 16. Excerpt from “Current Injector” sheet in an example main excel file. |

1. **Load 3-phase**

The 3-phase ZIP load is a combination of constant impedance, constant current, and constant power loads. The user needs to define the portion of each part out of 1 or 100.

Table 5: ZIP load parameters

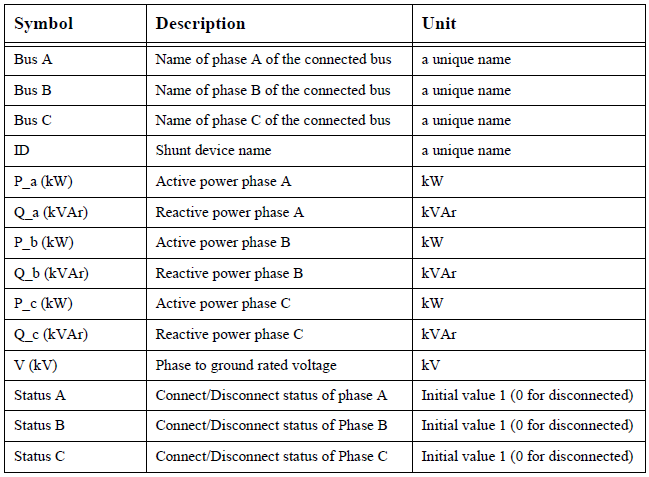


|  |
| --- |
|  |
| Figure 17. Excerpt from “Load 3-phase” sheet in an example main excel file. |

1. **Shunt 3-phase**

The “Shunt 3-phase” sheet is utilized to model shunt devices such as capacitor banks or reactors.

Table 6: Shunt devices parameters

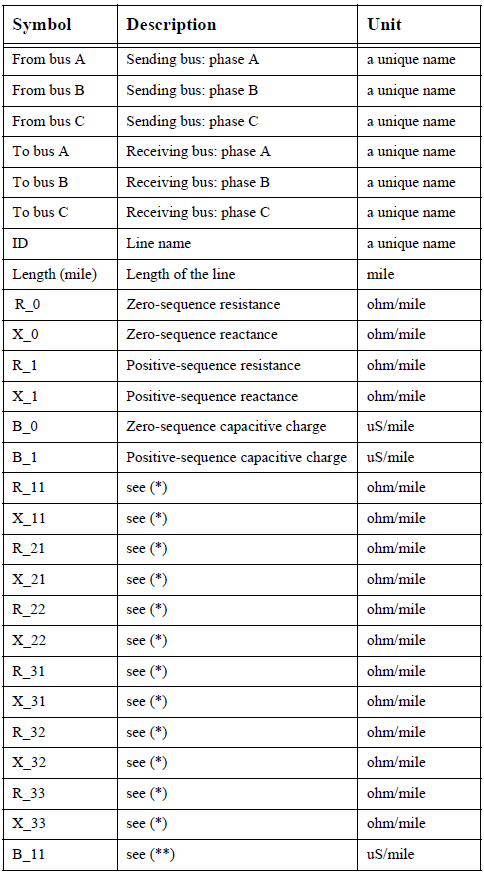


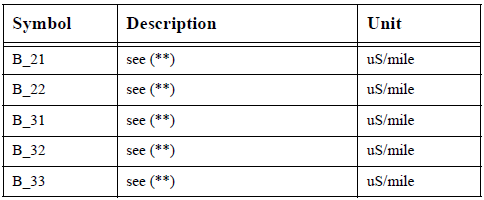
|  |
| --- |
|  |
| Figure 18. Excerpt from “Shunt 3-phase” sheet in an example main excel file. |

1. **Line 3-phase**

3-phase pi-section model line information is provided in the “Line 3-phase” sheet.

Table 7: Line 3-phase parameters



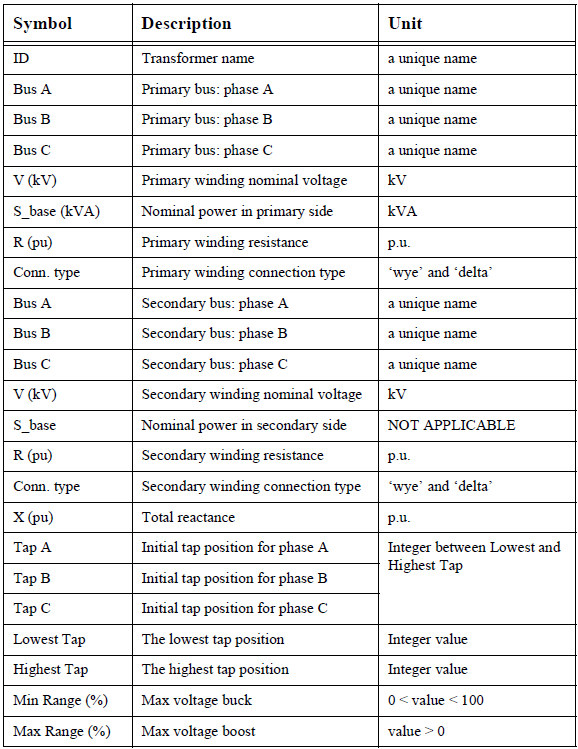


|  |
| --- |
| (b)  …….contd. |
| Figure 19. Excerpt from “Line 3-phase” sheet in an example main excel file for (a) Lines with parameters in positive and zero sequence format (b) Line with parameters in NxN format. |

1. **Transformer 3-phase**

Transformer Information in the feeder is provided in the “Transformer 3-phase” sheet.

Table 8: Transformer 3-phase parameters

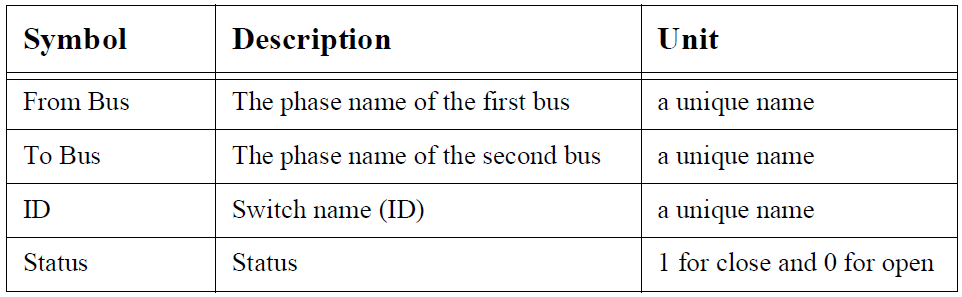


|  |
| --- |
| …..contd. |
| Figure 20. Excerpt from “Transformer 3-phase” sheet in an example main excel file. |

1. **Switch**

Switches such as breakers and disconnectors information is described in the “Switch” sheet.

Table 9: Switch parameters

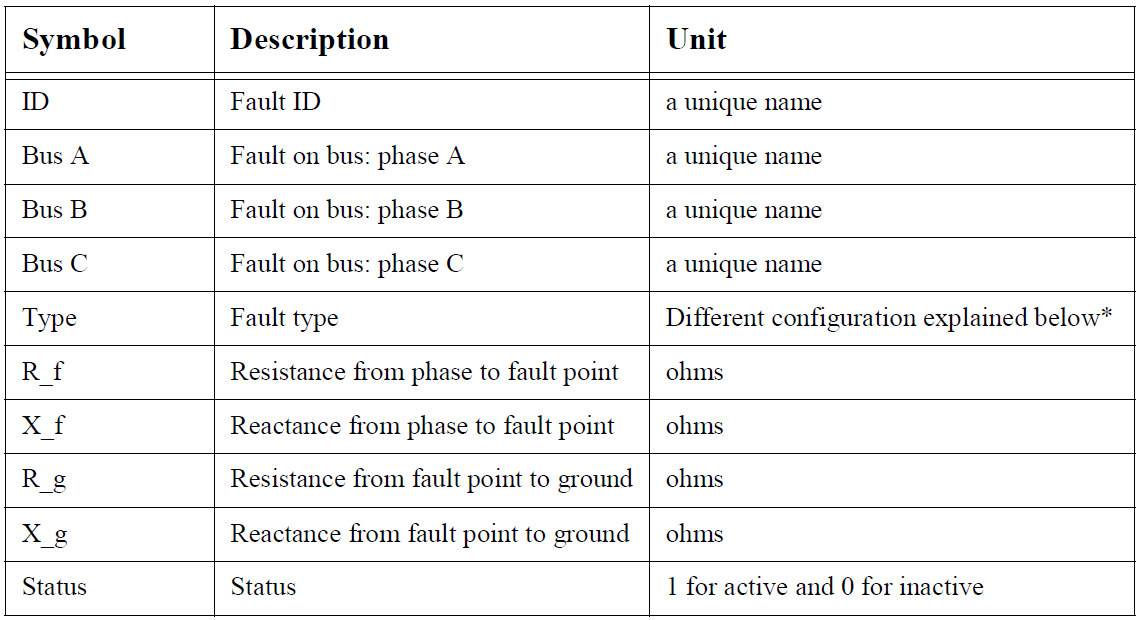


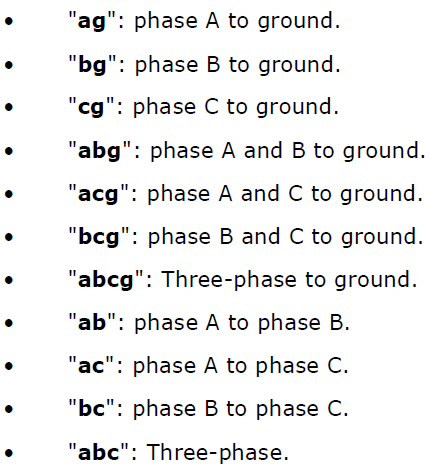
|  |
| --- |
|  |
| Figure 21. Excerpt from “Switch” sheet in an example main excel file. |

1. **Bus Faults 3-phase**

The “Bus Faults 3-phase” sheet is created in order to simulate different type of balanced and unbalanced faults in three-phase systems information.

Table 10: Buses and its phase to ground parameters information





|  |
| --- |
|  |
| Figure 22. Excerpt from “Bus Faults 3-phase” sheet in an example main excel file. |

**Data Parsing Conversion Process:**

**Python 2.7** was utilized to perform the files conversion or data parsing into the “ePHASORsim” excel spreadsheets from the “.dss” files. In order to perform the data parsing process, the feeder data was extracted from the “.dss” files using the regular expressions available in Python. Regular expressions are a sequence of characters that define a search pattern, mainly for use in pattern matching with strings, or string matching, i.e. "find and replace"-like operations [3], [4]. A summary flowchart of data parsing process from the OpenDSS based “.dss” files to input data spreadsheets to be interfaced with the ePHASORSIM simulator is shown in Figure 22. The packages/modules that were used along with the python software were:

* **Re**: Re or regular expression is used to perform matching operations. Both patterns and strings to be searched can be Unicode strings as well as 8-bit strings. It is the most important module used to perform pattern search as the “.dss” files are in text format and it is delimited by “=” symbols.
* **Numpy**: NumPy is the fundamental package for scientific computing with Python. They were used mainly for performing arithmetic computations for the feeder model data whenever required. NumPy’s main object is the homogeneous multidimensional array. It is a table of elements (usually numbers), all of the same type, indexed by a tuple of positive integers. In NumPy dimensions are called axes. The number of axes is rank.
* **Pandas**: Pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. This module was required for creating tables using the DataFrame object. It is used for creating the data frames to be written into the excel spreadsheets.
* **Itertool**: The chain () object is used to chain together the list of lists/tuples/iterables.
* **Collections:** In collections module counter is a dict subclass for counting hashable objects. A Counter is a container that keeps track of how many times equivalent values are added. It can be used to implement the same algorithms for which bag or multiset data structures are commonly used in other languages. It is used for finding the unique items in the list.

***Note:*** Opendss could use strings that are case insensitive, but the “ePHASORsim” excel spreadsheets strings are case sensitive, therefore in order to be consistent all the terms are converter into lower cases during the data parsing process.

|  |
| --- |
| ephasorsim_flowchart |
| Figure 23: Flowchart of data parsing process from “.dss” files to input data spreadsheets to be interfaced with the “ePHASORsim” simulator. |

In order to perform excel spreadsheet (both .xlsx and .csv) file reading and writing, the pandas library is utilized to perform data structures and data analysis for the Python programming language. Specifically to obtain the bus details the power flow analysis is performed in “opendss” and a “.csv” file is created to get the results. The “.csv” files contain the bus details along with the voltage magnitude and angles. The bus IDs information read using the “Pandas” library is utilized to create the “Bus Faults 3-phase” sheet while the bus IDs, magnitude and angles information is utilized to create the “Bus” sheet.

**Preparing ePHASORsim \*.mdl file**

1. Change the root of MATLAB to the directory of the folder (C:\...\test).
2. Go to Simulink Library.
3. Select RT-LAB/Phasor.
4. Drag Solver into a new model page.
5. Save this model in the folder used before (e.g. test).
6. Open the mask and do the followings:

* In ‘Network data’ tab (Figure below):
* Select Excel as ‘Input File Format’.
* Insert the name of your Excel file in ‘Excel File’.

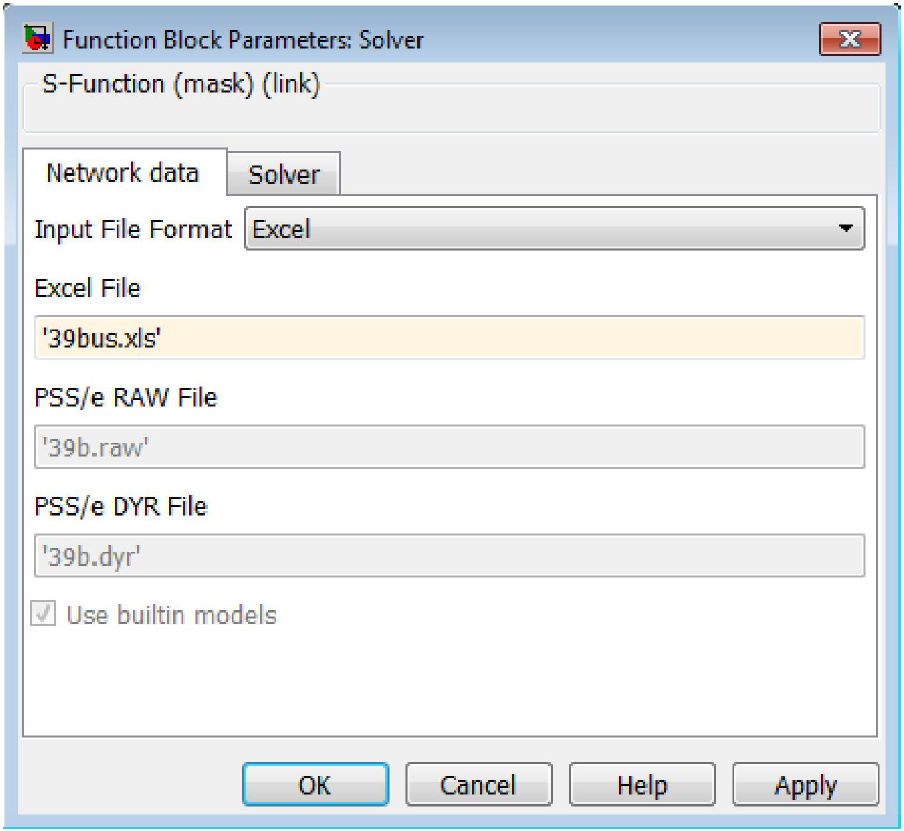


Figure 24: Network data tab in the mask

* In ‘Solver’ tab (Figure below):
* Select a specific time step (The default value is 10 ms).
* Enter the number of partitions (The default value is 1).
* Check ‘Export admittance matrix’ to have admittance matrix in the \*.csv format.

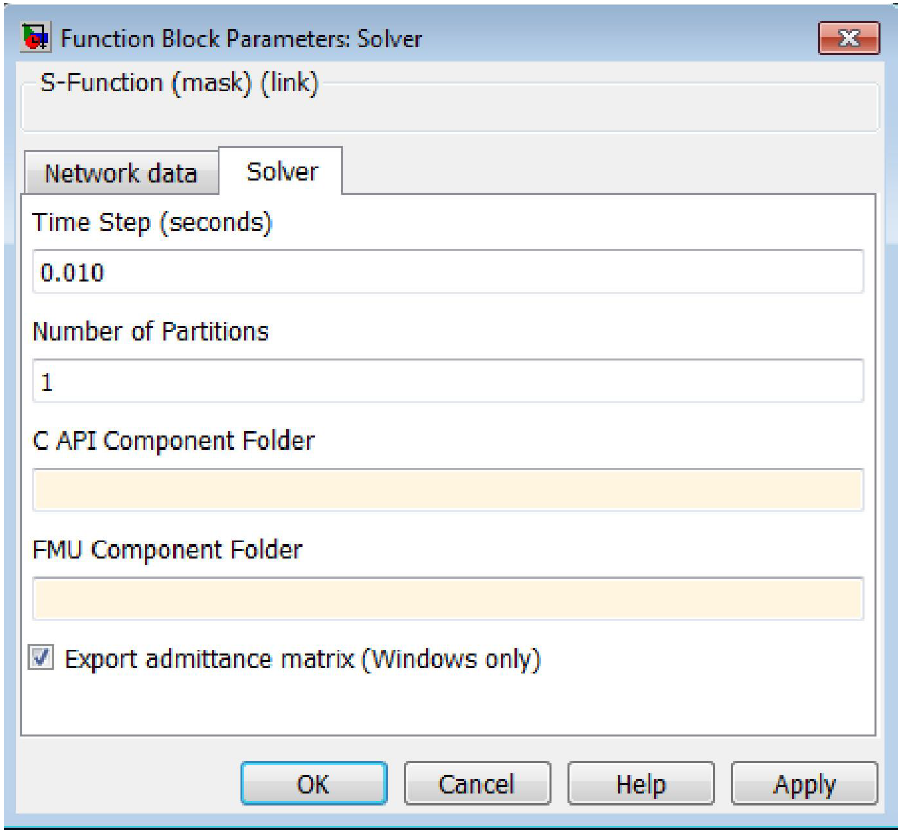


Figure 25: Solver tab in the mask.

1. Press OK or Apply. If the \*.xls file is valid for ePHASORsim, a file with the same name as the input Excel file and with the extension of opal will be generated automatically in the current folder.
2. In order to have access to this user manual, press Help.
3. Modify the remaining parts of the model as you prefer, add scopes or displays where appropriate. Ensure the simulation is set to use a fixed time step (Configuration/Simulation parameters), and that the time-step is consistent with that specified in the ‘Solver’.
4. Run the model to verify a reasonable value appears on the scopes. Save your model.

The screenshots of the main model along with its subsystems in Matlab/Simulink environment for an example case is shown below. The blocks utilized are available in the RT Lab toolbox of Matlab.

|  |
| --- |
|  |
| Figure 26: Main model in RT\_lab in Matlab/Simulink environment. |

|  |
| --- |
|  |
| Figure 27: Sm\_master subsystem |

The “Sm\_master” subsystem contains the Solver.

|  |
| --- |
|  |
| Figure 28: Sc\_console subsystem |

|  |
| --- |
|  |
| Figure 29: Sc\_console-> scopes subsystem |

**Creation of excel spreadsheets from the Python Codes:**

A workbook is created from the python codes using the DataFrame module which cannot be used directly as the input files for ePHASORsim, at this point. Large portions from the python created excel spreadsheets can be used to create new excel spreadsheets to be interfaced with ePHASORsim. Make sure to use the valued cell portions to copy-paste from the python created spreadsheets to the new excel spreadsheets. Any non-empty cells if copied would transfer ‘NaNs’.

All value cells from sheets can be used as it is when created using the python codes. The **Pins** and **Vresults** sheet can be modiefied

**Pins**:

There are a number of ways in which the pin rows can be created. The python codes create rows than can be used for copy-paste to the new excel workbook. In the new workbook, the first row defines whether the row is an input or an output list. More details on the ePHASORsim components that can be used to read from and write into the ePHASORsim block can be found in the ePHASORsim appendix.

**Row 2:** To read feeder node magnitudes in the order. Phase–[A, B, C, A, B, C…….].

[s\_a/Vmag, s\_b/Vmag, s\_c/Vmag, ls\_bus\_a/Vmag, ls\_bus\_b/Vmag, ls\_bus\_c/Vmag, feederhead\_a/Vmag, feederhead\_b/Vmag, feederhead\_c/Vmag, b12541\_a/Vmag, b13552\_a/Vmag, …etc]

**Row 3:** To read feeder node angles in the order, Phase–[A, B, C, A, B, C…….].

[s\_a/Vang, s\_b/Vang, s\_c/Vang, ls\_bus\_a/Vang, ls\_bus\_b/Vang, ls\_bus\_c/Vang, feederhead\_a/Vang, feederhead\_b/Vang, feederhead\_c/Vang, b12541\_a/Vang, b13552\_a/Vang,…etc]

**Row 4:** To read feeder lines with current magnitude fields in the order–[A, B, C, A, B, C…….].

[line\_oh\_b13552/Imag\_0\_a, line\_oh\_b13552/Imag\_0\_b, line\_oh\_b13552/Imag\_0\_c, line\_oh\_b41869/Imag\_0\_a, line\_oh\_b41869/Imag\_0\_b, line\_oh\_b41869/Imag\_0\_c, line\_oh\_b34986/Imag\_0\_a, line\_oh\_b34986/Imag\_0\_b, line\_oh\_b34986/Imag\_0\_c, line\_oh\_g43133/Imag\_0\_a, line\_oh\_g43133/Imag\_0\_b, line\_oh\_g43133/Imag\_0\_c,… etc.]

**Row 5:** To read feeder lines with current angle fields in the order–[A, B, C, A, B, C…….].

[line\_oh\_b13552/Iang\_0\_a, line\_oh\_b13552/Iang\_0\_b, line\_oh\_b13552/Iang\_0\_c, line\_oh\_b41869/Iang\_0\_a, line\_oh\_b41869/Iang\_0\_b, line\_oh\_b41869/Iang\_0\_c, line\_oh\_b34986/Iang\_0\_a, line\_oh\_b34986/Iang\_0\_b, line\_oh\_b34986/Iang\_0\_c, line\_oh\_g43133/Iang\_0\_a, line\_oh\_g43133/Iang\_0\_b, line\_oh\_g43133/Iang\_0\_c,…etc.]

**Row 6:** To read feeder lines with current angle fields in the order–[A, B, C, A, B, C…….].

[line\_oh\_b13552/Iang\_0\_a, line\_oh\_b13552/Iang\_0\_b, line\_oh\_b13552/Iang\_0\_c, line\_oh\_b41869/Iang\_0\_a, line\_oh\_b41869/Iang\_0\_b, line\_oh\_b41869/Iang\_0\_c, line\_oh\_b34986/Iang\_0\_a, line\_oh\_b34986/Iang\_0\_b, line\_oh\_b34986/Iang\_0\_c, line\_oh\_g43133/Iang\_0\_a, line\_oh\_g43133/Iang\_0\_b, line\_oh\_g43133/Iang\_0\_c,…etc.]

**Row 7:** To Write loads active and reactive powers.

[load\_x\_b13552\_cust2-a/P\_a, load\_x\_b13552\_cust2-a/Q\_a, load\_x\_b13552\_cust1-a/P\_a, load\_x\_b13552\_cust1-a/Q\_a, load\_x\_b34986\_cust1-c/P\_c, load\_x\_b34986\_cust1-c/Q\_c, …etc]

**Row 8:** To write transformer with Taps.

[vr\_b18862\_b18865reg/tap\_b, vr\_b18862\_b18865reg/tap\_c, vr\_b4877\_b4870reg/tap\_a, vr\_b4877\_b4870reg/tap\_b, vr\_b4877\_b4870reg/tap\_c, vr\_b19007\_b19007reg/tap\_a, vr\_b19007\_b19007reg/tap\_b, vr\_b19007\_b19007reg/tap\_c]

**Row 9:** Feeder nodes with magnitude fields in the order. Phase –[A, A, A….. B, B, B…….C, C, C].

[s\_a/Vmag, ls\_bus\_a/Vmag, feederhead\_a/Vmag, b12541\_a/Vmag, b13552\_a/Vmag, b41868\_a/Vmag, b41869\_a/Vmag, b4609\_a/Vmag, …etc]

**Row 10:** Feeder nodes with angle fields in the order. Phase –[A, A, A….. B, B, B…….C, C, C].

[s\_a/Vang, ls\_bus\_a/Vang, feederhead\_a/Vang, b12541\_a/Vang, b13552\_a/Vang, b41868\_a/Vang, b41869\_a/Vang, b4609\_a/Vang,…etc]

***Important Note:*** The values obtained from ePHASORsim workspace using the outputs Row 9 and Row 10 can be used to validate the OpenDSS results that are

**Vresults:**

Python codes are used to create the actual values from OpenDSS as rows of list values. This would be used later on for comparison with the ePHASORsim results.

***Important Note:*** To create the Vresults spreadsheets, OpenDSS must be run first and the results spreadsheet must be creates using: **Export->Voltages->Buses**. Make sure the create results excel spreadsheet has the same name for reading in the python conversion tool.

**Row 2:** Feeder node magnitude values (pu) in the order. Phase–[A, B, C, A, B, C…….].

**Row 3:** Feeder node angle values (degrees) in the order. Phase–[A, B, C, A, B, C…….].

**Row 4:** Feeder node base values (Nominal) in the order. Phase–[A, B, C, A, B, C…….].

**Row 5:** Feeder node magnitude values in the order. Phase–[A, A, A, …, B, B, B……., C, C, C,…]. Order based on distance from substation.

**Row 6:** Feeder node angle values in the order. Phase–[A, A, A, …, B, B, B……., C, C, C,…]. Order based on distance from substation.

**Row 7:** Feeder node base values (Nominal) in the order. Phase–[A, A, A, …, B, B, B……., C, C, C,…]. Order based on distance from substation.

**Opendss Appendix [3]**

**Object = EXECUTIVE**

**Property Description**

// Comment. Command line is ignored.

? Inquiry for property value. Result is put into GlobalReault and can be seen in the Result Window. Specify the full property name. Example: ? Line.Line1.R1 Note you can set this property merely by saying:  
Line.line1.r1=.058

\_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.

~ Continuation of editing on the active object. An abbreviation. Example New Line.Line1 Bus1=aaa bus2=bbb  
~ R1=.058  
~ X1=.1121

About Display "About Box". (Result string set to Version string.)

AddMarker Add a marker to the active plot. Example:   
  
AddMarker Bus=busname code=nn color=$00FF0000 size=3

AlignFile Alignfile [file=]filename. Aligns DSS script files in columns for easier reading.

Buscoords Define x,y coordinates for buses. Execute after Solve command performed so that bus lists are defined. Reads coordinates from a CSV file with records of the form: busname, x, y.  
Example: BusCoords [file=]xxxx.csv

Calcvoltagebases Calculates voltagebase for buses based on voltage bases defined with Set voltagebases=... command.

CD Change default directory to specified directory CD dirname

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

Classes List of intrinsic DSS Classes. Returns comma-separated list in Result variable.

Clear Clear all circuits currently in memory.

Close Opposite of the Open command.

CloseDI Close all DI files ... useful at end of yearly solution where DI files are left open. (Reset and Set Year=nnn will also close the DI files)

Comparecases [Case1=]casename [case2=]casename [register=](register number) [meter=]{Totals\* | SystemMeter | metername}.   
Compares yearly simulations of two specified cases with respect to the quantity in the designated register from the designated meter file. Defaults: Register=9 meter=Totals. Example: Comparecases base pvgens 10

Compile Reads the designated file name containing DSS commands and processes them as if they were entered directly into the command line. The file is said to be "compiled." Similar to "redirect" except changes the default directory to the path of the specified file.  
  
Syntax:  
Compile filename

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.

Disable Disables a circuit element or entire class. Example: Disable load.loadxxx Disable generator.\* (Disables all generators) The item remains defined, but is not included in the solution.

Distribute kw=nn how={Proportional | Uniform |Random | Skip} skip=nn PF=nn file=filename MW=nn  
Distributes generators on the system in the manner specified by "how".  
kW = total generation to be distributed (default=1000)   
how= process name as indicated (default=proportional to load)  
skip = no. of buses to skip for "How=Skip" (default=1)  
PF = power factor for new generators (default=1.0)  
file = name of file to save (default=distgenerators.txt)  
MW = alternate way to specify kW (default = 1)

Enable Enables a circuit element or entire class. Example: Enable load.loadxxx  
Enable generator.\* (enables all generators)

Estimate Execute state estimator on present circuit given present sensor values.

Export Export various solution values to CSV (or XML) files for import into other programs. Creates a new file except for Energymeter and Generator objects, for which the results for each device of this class are APPENDED to the CSV File. You may export to a specific file by specifying the file name as the LAST parameter on the line. Otherwise, the default file names shown below are used. 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 elements.  
  
Syntax for Implemented Exports:  
  
Export Voltages [Filename] (EXP\_VOLTAGES.CSV)  
Export SeqVoltages [Filename] (EXP\_SEQVOLTAGES.CSV)  
Export Currents [Filename] (EXP\_CURRENTS.CSV)  
Export Estimation [Filename] (EXP\_ESTIMATION.CSV)  
Export Capacity [Filename] (EXP\_CAPACITY.CSV)  
Export Overloads [Filename] EXP\_OVERLOADS.CSV)  
Export Unserved [UEonly] [Filename] EXP\_UNSERVED.CSV)  
Export SeqCurrents [Filename] (EXP\_SEQCURRENTS.CSV)  
Export Powers [MVA] [Filename](EXP\_POWERS.CSV)  
Export P\_Byphase [MVA] [Filename](EXP\_P\_BYPHASE.CSV)  
Export SeqPowers [MVA] [Filename](EXP\_SEQPOWERS.CSV)  
Export Faultstudy [Filename] (EXP\_FAULTS.CSV)  
Export Generators [Filename | /m ] (EXP\_GENMETERS.CSV)  
Export Loads [Filename] (EXP\_LOADS.CSV)  
Export Meters [Filename |/m ] (EXP\_METERS.CSV)  
Export Monitors monitorname (file name is assigned)   
Export Yprims [Filename] (EXP\_YPRIMS.CSV) (all YPrim matrices)  
Export Y [Filename] (EXP\_Y.CSV) (system Y matrix)  
Export seqZ [Filename] (EXP\_SEQZ.CSV) (equiv sequence Z1, Z0 to bus)  
Export CDPSM [Filename] (CDPSM.XML) (IEC 61968-13, CDPSM format)  
Export Buscoords [Filename] [EXP\_BUSCOORDS.CSV]  
  
May be abreviated Export V, Export C, etc. Default is "V". If Set ShowExport=Yes, the output file will be automatically displayed in the default editor.

Fileedit Edit specified file in default text file editor (see Set Editor= option). Fileedit EXP\_METERS.CSV (brings up the meters export file) "FileEdit" may be abbreviated to a unique character string.

Formedit FormEdit [class.object]. Brings up form editor on active DSS object. Get Returns DSS property values set using the Set command. Result is returne in Result property of the Text interface.   
  
VBA Example:  
  
DSSText.Command = "Get mode"  
Answer = DSSText.Result  
  
Multiple properties may be requested on one get. The results are appended and the individual values separated by commas.  
  
See help on Set command for property names.

Help Gives this display.

Init This command forces re-initialization of the solution for the next Solve command. To minimize iterations, most solutions start with the previous solution unless there has been a circuit change. However, if the previous solution is bad, it may be necessary to re-initialize. In most cases, re-initialization results in a zero-load power flow solution with only the series power delivery elements considered.

MakePosSeq Attempts to convert present circuit model to a positive sequence equivalent. It is recommended to Save the circuit after this and edit the saved version to correct possible misinterpretations.

New Create a new object within the DSS. Object becomes the active object  
Example: New Line.line1 ...

Open Opens the specified terminal and conductor of the specified circuit element. If the conductor is not specified, all phase conductors of the terminal are opened.

Examples:  
Open line.line1 2   
(opens all phases of terminal 2)  
  
Open line.line1 2 3  
(opens the 3rd conductor of terminal 2)

Panel Displays main control panel window.

Plot Plots results in a variety of manners. Implemented options (in order):   
Type = {Circuit | Monitor | Daisy | Zones | AutoAdd | General (bus data) }  
Quantity = {Voltage | Current | Power | Losses | Capacity | (Value Index for General, AutoAdd, or Circuit[w/ file]) }  
Max = {0 | value corresponding to max scale or line thickness}  
Dots = {Y | N}  
Labels = {Y | N}  
Object = [metername for Zone plot | Monitor name | File Name for General bus data or Circuit branch data]  
ShowLoops = {Y | N} (default=N)  
R3 = pu value for tri-color plot max range [.85] (Color C3)  
R2 = pu value for tri-color plot mid range [.50] (Color C2)  
C1, C2, C3 = {RGB color number}   
Channels=(array of channel numbers for monitor plot)  
Bases=(array of base values for each channel for monitor plot). Default is 1.0 for each. Set Base= after defining channels.  
Subs={Y | N} (default=N) (show substations)  
Thickness=max thickness allowed for lines in circuit plots (default=7)  
Buslist=[Array of Bus Names | File=filename ] (for Daisy plot)  
  
Power and Losses in kW. C1 used for default color. C2, C3 used for gradients, tri-color plots. Scale determined automatically of Max = 0 or not specified. Example:  
  
Plot daisy power 5000 dots=N !! Generators by default  
Plot daisy power 5000 dots=N Buslist=[file=MyBusList.txt]  
Plot circuit quantity=7 Max=.010 dots=Y Object=branchdata.csv  
Plot General Quantity=2 Object=valuefile.csv

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

puvoltages Just like the Voltages command, except the voltages are in per unit if the kVbase at the bus is defined.

Quit Shuts down DSS unless this is the DLL version. Then it does nothing; DLL parent is responsible for shutting down the DLL.

Redirect Reads the designated file name containing DSS commands and processes them as if they were entered directly into the command line. Similar to "Compile", but leaves current directory where it was when Redirect command is invoked.Can temporarily change to subdirectories if nested Redirect commands require.  
ex: redirect filename

Reduce {All | MeterName} Default is "All". Reduce the circuit according to reduction options. See "Set ReduceOptions" and "Set Keeplist" options.Energymeter objects actually perform the reduction. "All" causes all meters to reduce their zones.

Sample Force all monitors and meters to take a sample now.

Save {Save [class=]{Meters | Circuit | Voltages | (classname)} [file=]filename [dir=]directory   
 Default class = Meters, which saves the present values in both monitors and energy meters in the active circuit. "Save Circuit" saves the present enabled circuit elements to the specified subdirectory in standard DSS form with a Master.txt file and separate files for each class of data. If Dir= not specified a unique name based on the circuit name is created automatically. If Dir= is specified, any existing files are overwritten.   
"Save Voltages" saves the present solution in a simple CSV format in a file called DSS\_SavedVoltages. Used for VDIFF command.  
Any class can be saved to a file. If no filename specified, the classname is used.

Select Selects an element and makes it the active element. You can also specify the active terminal (default = 1).  
Syntax:  
Select [element=]elementname [terminal=]terminalnumber   
Example:  
Select Line.Line1   
~ R1=.1  
(continue editing)  
Select Line.Line1 2   
Voltages (returns voltages at terminal 2 in Result)

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

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

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

Set Used to set various DSS solution modes and options. You may also set the options with the Solve command. See "Options" for help.

SetkVBase Command to explicitly set the base voltage for a bus. Bus must be previously defined. Parameters in order are:  
Bus = {bus name}  
kVLL = (line-to-line base kV)  
kVLN = (line-to-neutral base kV)  
kV base is normally given in line-to-line kV (phase-phase). However, it may also be specified by line-to-neutral kV.  
The following exampes are equivalent:

setkvbase Bus=B9654 kVLL=13.2  
setkvbase B9654 13.2  
setkvbase B9654 kvln=7.62

Show Writes selected results to a text file and brings up the editor (see Set Editor=....) with the file for you to browse.  
Valid Options (\*=default):  
Show Buses  
Show Currents [[residual=]yes|no\*] [Seq\* | Elements]  
Show COnvergence (convergence report)  
Show CONTrolQueue   
Show ELements [Classname] (shows names of all elements in circuit or all elements of a class)  
Show Faults (after Fault Study)  
Show Generators  
Show Losses  
Show MEters  
Show Monitor Monitorname  
Show PAnel (control panel)  
Show Powers [MVA|kVA\*] [Seq\* | Elements]  
Show Voltages [LL |LN\*] [Seq\* | Nodes | Elements]  
Show Zone EnergyMeterName [Treeview]  
Show AutoAdded (see AutoAdd solution mode)  
Show Taps (regulated transformers)  
Show Overloads (overloaded PD elements)  
Show Unserved [UEonly] (unserved loads)  
Show EVentlog  
Show VAriables  
Show Isolated  
Show Ratings (ratings of PD Elements)  
Show LOOps  
Show YPrim (shows Yprim for active ckt element)  
Show Y (shows system Y)  
Show BUSFlow busname [MVA|kVA\*] [Seq\* | Elements]  
Show LIneConstants [frequency] [none|mi|km|kft|m|me|ft|in|cm] [rho]  
  
Default is "show voltages LN Seq".

Solve Perform the solution of the present solution mode. You can set any option that you can set with the Set command (see Set). The Solve command is virtually synonymous with the Set command except that a solution is performed after the options are processed.

Summary Displays a power flow summary of the most recent solution.

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

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

Vdiff Displays the difference between the present solution and the last on saved using the SAVE VOLTAGES command.

Voltages Returns the voltages for the ACTIVE BUS in the Result string. For setting the active Bus, use the Select command or the Set Bus= option. Returned as magnitude and angle quantities, comma separated, one set per conductor of the terminal.

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

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

Zsc10 Returns symmetrical component impedances, Z1, Z0 for the ACTIVE BUS in comma-separated R+jX form.

ZscRefresh Refreshes Zsc matrix for the ACTIVE BUS.

**Object = CAPACITOR**

**Property Description**

(1) bus1 Name of first bus. Examples: bus1=busname bus1=busname.1.2.3

(2) bus2 Name of 2nd bus. Defaults to all phases connected to first bus, node 0. (Shunt Wye Connection) Not necessary to specify for delta (LL) connection

(3) phases Number of phases.

(4) kvar Total kvar, if one step, or ARRAY of kvar ratings for each step. Evenly divided among phases. See rules for NUMSTEPS.

(5) kv For 2, 3-phase, kV phase-phase. Otherwise specify actual can rating.

(6) conn ={wye | delta |LN |LL} Default is wye, which is equivalent to LN

(7) cmatrix Nodal cap. matrix, lower triangle, microfarads, of the following form:  
cmatrix="c11 | -c21 c22 | -c31 -c32 c33" All steps are assumed the same if this property is used.

(8) states ARRAY of integers {1|0} states 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.

(9) basefreq Base Frequency for ratings.

(10) enabled {Yes|No or True|False} Indicates whether this element is enabled.

**Object = LINE**

**Property Description**

(1) bus1 Name of bus to which first terminal is connected. Example: bus1=busname (assumes all terminals connected in normal phase order) bus1=busname.3.1.2.0 (specify terminal to node connections explicitly)

(2) bus2 Name of bus to which 2nd terminal is connected.

(3) linecode Name of linecode object describing line impedances. If you use a line code, you do not need to specify the impedances here. The line code must have been PREVIOUSLY defined. The values specified last will prevail over those specified earlier (left-to-right sequence of properties). You can subsequently change the number of phases if symmetrical component quantities are specified. If no line code or impedance data are specified, the line object defaults to 336 MCM ACSR on 4 ft spacing.

(4) length Length of line. Default is 1.0. If units do not match the impedance data, specify "units" property.

(5) phases Number of phases, this line.

(6) r1 Positive-sequence Resistance, ohms per unit length. Setting any of R1, R0, X1, X0, C1, C0 forces the program to use the symmetrical component line definition. See also Rmatrix.

(7) x1 Positive-sequence Reactance, ohms per unit length. Setting any of R1, R0, X1, X0, C1, C0 forces the program to use the symmetrical component line definition. See also Xmatrix

(8) r0 Zero-sequence Resistance, ohms per unit length.

(9) x0 Zero-sequence Reactance, ohms per unit length.

(10) c1 Positive-sequence capacitance, nf per unit length. Setting any of R1, R0, X1, X0, C1, C0 forces the program to use the symmetrical component line definition. See also Cmatrix.

(11) c0 Zero-sequence capacitance, nf per unit length.

(12) rmatrix Resistance matrix, lower triangle, ohms per unit length. Order of the matrix is the number of phases. May be used to specify the impedance of any line configuration. Using any of Rmatrix, Xmatrix, Cmatrix forces program to use the matrix values for line impedance definition. For balanced line models, you may use the standard symmetrical component data definition instead.

(13) xmatrix Reactance matrix, lower triangle, ohms per unit length. Order of the matrix is the number of phases. May be used to specify the impedance of any line configuration. Using any of Rmatrix, Xmatrix, Cmatrix forces program to use the matrix values for line impedance definition. For balanced line models, you may use the standard symmetrical component data definition instead.

(14) cmatrix Nodal Capacitance matrix, lower triangle, nf per unit length.Order of the matrix is the number of phases. May be used to specify the shunt capacitance of any line configuration. Using any of Rmatrix, Xmatrix, Cmatrix forces program to use the matrix values for line impedance definition. For balanced line models, you may use the standard symmetrical component data definition instead.

(15) Switch {y/n | T/F} Default= no/false. Designates this line as a switch for graphics and algorithmic purposes. SIDE EFFECT: Sets R1=0.001 X1=0.0. You must reset if you want something different.

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

(17) basefreq Base Frequency for ratings.

(18) enabled {Yes|No or True|False} Indicates whether this element is enabled.

**Object = LINECODE**

**Property Description**

(1) nphases Number of phases in the line this line code data represents. Setting this property reinitializes the line code. Impedance matrix is reset for default symmetrical component.

(2) r1 Positive-sequence Resistance, ohms per unit length. See also Rmatrix.

(3) x1 Positive-sequence Reactance, ohms per unit length. See also Xmatrix

(4) r0 Zero-sequence Resistance, ohms per unit length.

(5) x0 Zero-sequence Reactance, ohms per unit length.

(6) c1 Positive-sequence capacitance, nf per unit length. See also Cmatrix.

(7) c0 Zero-sequence capacitance, nf per unit length.

(8) units One of (ohms per ...) {none|mi|km|kft|m|me|ft|in|cm}. Default is none; assumes units agree with length unitsgiven in Line object

(9) rmatrix Resistance matrix, lower triangle, ohms per unit length. Order of the matrix is the number of phases. May be used to specify the impedance of any line configuration. For balanced line models, you may use the standard symmetrical component data definition instead.

(10) xmatrix Reactance matrix, lower triangle, ohms per unit length. Order of the matrix is the number of phases. May be used to specify the impedance of any line configuration. For balanced line models, you may use the standard symmetrical component data definition instead.

(11) cmatrix Nodal Capacitance matrix, lower triangle, nf per unit length.Order of the matrix is the number of phases. May be used to specify the shunt capacitance of any line configuration. For balanced line models, you may use the standard symmetrical component data definition instead.

(12) baseFreq Frequency at which impedances are specified.

**Object = LOAD**

**Property Description**

(1) phases Number of Phases, this load. Load is evenly divided among phases.

(2) bus1 Bus to which the load is connected. May include specific node specification.

(3) kV Nominal rated (1.0 per unit) voltage, kV, for load. For 2- and 3-phase loads, specify phase-phase kV. Otherwise, specify actual kV across each branch of the load. If wye (star), specify phase-neutral kV. If delta or phase-phase connected, specify phase-phase kV.

(4) kW Total base kW for the load. Normally, you would enter the maximum kW for the load for the first year and allow it to be adjusted by the load shapes, growth shapes, and global load multiplier. Legal ways to define base load: kW, PF  
kW, kvar kVA, PF XFKVA \* Allocation factor, PF kWh/(kWhdays\*24) \* Cfactor, PF

(5) pf Load power factor. Enter negative for leading powerfactor (when kW and kvar have opposite signs.)

(6) model Integer code for the model to use for load variation with voltage. Valid values are:  
1:Standard constant P+jQ load. (Default) 2:Constant impedance load. 3:Const P, Quadratic Q (like a motor). 4:Nominal Linear P, Quadratic Q (feeder mix). Use this with CVRfactor. 5:Constant Current Magnitude 6:Const P, Fixed Q 7:Const P, Fixed Impedance Q For Types 6 and 7, only the P is modified by load multipliers.

(7) conn ={wye or LN | delta or LL}. Default is wye.

(8) kvar Specify the base kvar for specifying load as kW & kvar. Assumes kW has been already defined. Alternative to specifying the power factor. Side effect: the power factor and kVA is altered to agree.

(9) status ={Variable | Fixed | Exempt}. Default is variable. If Fixed, no load multipliers apply; however, growth multipliers do apply. All multipliers apply to Variable loads. Exempt loads are not modified by the global load multiplier, such as in load duration curves, etc. Daily multipliers do apply, so this is a good way to represent industrial load that stays the same for the period study.

(10) class An arbitrary integer number representing the class of load so that load values may be segregated by load value. Default is 1; not used internally.

(11) 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.

(12) 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.

(13) kVA Specify base Load in kVA (and power factor) Legal ways to define base load: kW, PF kW, kvar kVA, PF XFKVA \* Allocationfactor, PF kWh/(kWhdays\*24) \* Cfactor, PF

(14) basefreq Base Frequency for ratings.

(15) enabled {Yes|No or True|False} Indicates whether this element is enabled.

**Object = TRANSFORMER**

**Property Description**

(1) phases Number of phases this transformer. Default is 3.

(2) windings Number of windings, this transformers. (Also is the number of terminals) Default is 2.

(3) wdg Set this = to the number of the winding you wish to define. Then set the values for this winding. Repeat for each winding. Alternatively, use the array collections (buses, kvas, etc.) to define the windings. Note: reactances are BETWEEN pairs of windings; they are not the property of a single winding.

(4) bus Bus connection spec for this winding.

(5) conn Connection of this winding. Default is "wye" with the neutral solidly grounded.

(6) kV For 2-or 3-phase, enter phase-phase kV rating. Otherwise, kV rating of the actual winding

(7) kVA Base kVA rating of the winding. Side effect: forces change of max normal and emerg kva ratings.If 2-winding transformer, forces other winding to same value. When winding 1 is defined, all other windings are defaulted to the same rating and the first two winding resistances are defaulted to the %loadloss value.

(8) tap Per unit tap that this winding is on.

(9) %R Percent resistance this winding. (half of total for a 2-winding).

(10) buses Use this to specify all the bus connections at once using an array. Example: New Transformer.T1 buses="Hibus, lowbus"

(11) conns Use this to specify all the Winding connections at once using an array. Example: New Transformer.T1 buses="Hibus, lowbus" ~ conns=(delta, wye)

(12) kVs Use this to specify the kV ratings of all windings at once using an array. Example: New Transformer.T1 buses="Hibus, low bus" ~ conns=(delta, wye) ~ kvs=(115, 12.47)  
See kV= property for voltage rules.

(13) kVAs Use this to specify the kVA ratings of all windings at once using an array.

(14) taps Use this to specify the p.u. tap of all windings at once using an array.

(15) Xhl Use this to specify the percent reactance, H-L (winding 1 to winding 2). Use for 2- or 3-winding transformers. On the kva base of winding 1.

(16) %loadloss Percent load loss at full load. The %R of the High and Low windings (1 and 2) are adjusted to agree at rated kVA loading.

(17) %noloadloss Percent no load losses at rated excitatation voltage. Default is 0. Converts to a resistance in parallel with the magnetizing impedance in each winding.

(18) normhkVA Normal maximum kVA rating of H winding (winding 1). Usually 100% - 110% of maximum nameplate rating, depending on load shape. Defaults to 110% of kVA rating of Winding 1.

(19) sub ={Yes|No} Designates whether this transformer is to be considered a substation. Default is No.

(20) MaxTap Max per unit tap for the active winding. Default is 1.10

(21) MinTap Min per unit tap for the active winding. Default is 0.90

(22) NumTaps Total number of taps between min and max tap. Default is 32.

(23) subname Substation Name. Optional. Default is null. If specified, printed on plots

(24) %Rs Use this property to specify all the winding %resistances using an array. Example: New Transformer.T1 buses="Hibus, lowbus" ~ %Rs=(0.2 0.3)

(25) normamps Normal rated current.

(26) basefreq Base Frequency for ratings.

(27) enabled {Yes|No or True|False} Indicates whether this element is enabled.

**Object = VSOURCE**

**Property Description**

(1) bus1 Name of bus to which the main terminal (1) is connected. bus1=busname bus1=busname.1.2.3

(2) basekv Base Source kV, usually phase-phase (L-L) unless you are making a positive-sequence model or 1-phase modelin which case, it will be phase-neutral (L-N) kV.

(3) pu Per unit of the base voltage that the source is actually operating at. "pu=1.05"

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

(5) frequency Source frequency. Defaults to system default base frequency.

(6) phases Number of phases. Defaults to 3.

(7) MVAsc3 MVA Short circuit, 3-phase fault. Default = 2000. Z1 is determined by squaring the base kv and dividing by this value. For single-phase source, this value is not used.

(8) MVAsc1 MVA Short Circuit, 1-phase fault. Default = 2100. The "single-phase impedance", Zs, is determined by squaring the base kV and dividing by this value. Then Z0 is determined by Z0 = 3Zs - 2Z1. For 1-phase sources, Zs is used directly. Use X0R0 to define X/R ratio for 1-phase source.

(9) x1r1 Positive-sequence X/R ratio. Default = 4.

(10) x0r0 Zero-sequence X/R ratio.Default = 3.

(11) Isc3 Alternate method of defining the source impedance. 3-phase short circuit current, amps. Default is 10000.

(12) Isc1 Alternate method of defining the source impedance. single-phase short circuit current, amps. Default is 10500.

(13) R1 Alternate method of defining the source impedance. Positive-sequence resistance, ohms. Default is 1.65.

(14) X1 Alternate method of defining the source impedance. Positive-sequence reactance, ohms. Default is 6.6.

(15) R0 Alternate method of defining the source impedance. Zero-sequence resistance, ohms. Default is 1.9.

(16) X0 Alternate method of defining the source impedance. Zero-sequence reactance, ohms. Default is 5.7.

(18) bus2 Name of bus to which 2nd terminal is connected. bus2=busname bus2=busname.1.2.3 Default is Bus1.0.0.0 (grounded wye connection)

(20) basefreq Base Frequency for ratings.

(21) enabled {Yes|No or True|False} Indicates whether this element is enabled.

**ePHASORsim Appendix [2]**

**I/O Pins for positive sequence components**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Pin’s Description** | **Pin Type** | **Value** | **Instruction** |
| 1 | Fault on bus: three-phase-to-ground | I/O | 1: Active  0: Inactive | busID/active3PGFault |
| 2 | Voltage magnitude of a bus (RMS) | O | p.u. | busID/Vmag |
| 3 | Voltage angle of a bus | O | Degree | busID/Vang |
| 4 | In-service/out-of-service of a load | I/O | 1: In-service  0: Out-of-service | loadID/status |
| 5 | Adjust load’s active and reactive power | I | W  VAr | loadID/P  loadID/Q |
| 6 | Current magnitude of a load | O | p.u. | loadID/Imag |
| 7 | Current angle of a load | O | Degree | loadID/Iang |
| 8 | Fault on line: three-phase-to-ground | I/O | 1: Active  0: Inactive | lineID/faulty |
| 9 | Distance for line fault from starting bus | I/O | 0 < value < 1 | lineID/fault\_distance\_factor |
| 10 | In-service/out-of-service of a line | I/O | 1: In-service  0: Out-of-service | lineID/status |
| 11 | Sending end current magnitude of a line (RMS) | O | p.u. | lineID/Imag0 |
| 12 | Sending end current angle of a line | O | Degree | lineID/Iang0 |
| 13 | Receiving end current magnitude of a  line (RMS) | O | p.u. | lineID/Imag1 |
| 14 | Receiving end current angle of a line | O | Degree | lineID/Iang1 |
| 15 | Changing the tap ratio of a transformer | I/O | Between [-100%,  100%] | transformerID/tap\_ratio |

I/O Pins for three-phase components

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Pin’s Description** | **Pin Type** | **Value** | **Instruction** |
| 1 | In-service/out-service of a line | I/O | 1: In-service  0: Out-of-service | lineID/status |
| 2 | Phase-based tap position | I/O | Integer between  [min\_tap,  max\_tap] | transformerID/tap\_a  transformerID/tap\_b  transformerID/tap\_c |
| 3 | In-service/out-of-service of a load | I/O | 1: In-service  0: Out-of-service | loadID/status |
| 4 | Switches and breakers (each phase) | I/O | 1: Close  0: Open | switchID/status |
| 5 | Adjust load’s active and reactive  power for phase A (load profile) | I/O | kW  kVAr | loadID/P\_a  loadID/Q\_a |
| 6 | Adjust load’s active and reactive  power for phase B (load profile) | I/O | kW  kVAr | loadID/P\_b  loadID/Q\_b |
| 7 | Adjust load’s active and reactive  power for phase C (load profile) | I/O | kW  kVAr | loadID/P\_c  loadID/Q\_c |
| 8 | Voltage magnitude of each phase of a bus | O | V (RMS) | Bus Phase ID/Vmag |
| 9 | Voltage angle of each phase of a bus | O | Degree | Bus Phase ID/Vang |
| 10 | Current magnitude of each phase of load, shunt device, voltage source | O | A (RMS) | itemID/Imag\_a  itemID/Imag\_b  itemID/Imag\_c |
| 11 | Current angle of each phase of load, shunt device, voltage source | O | Degree | itemID/Iang\_a  itemID/Iang\_b  itemID/Iang\_c |
| 12 | Current magnitude of each phase of line (sending end), transformer (primary side) | O | A (RMS) | itemID/Imag\_0\_a  itemID/Imag\_0\_b  itemID/Imag\_0\_c |
| 13 | Current magnitude of each phase of line (receiving end), transformer (secondary side) | O | A (RMS) | itemID/Imag\_1\_a  itemID/Imag\_1\_b  itemID/Imag\_1\_c |
| 14 | Current angle of each phase of line (sending end), transformer (primary side) | O | Degree | itemID/Iang\_0\_a  itemID/Iang\_0\_b  itemID/Iang\_0\_c |
| 15 | Current angle of each phase of line  (receiving end), transformer (secondary side) | O | Degree | itemID/Iang\_1\_a  itemID/Iang\_1\_b  itemID/Iang\_1\_c |
| 16 | Current magnitude of a switch | O | A (RMS) | switchID/Imag |
| 17 | Current angle of a switch | O | Degree | switchID/Iang |
| 18 | Three-phase bus fault status | I/O | 1: active  0: Inactive | faultID/active |
| 19 | Current magnitude of bus fault | O | A (RMS) | faultID/Imag\_a  faultID/Imag\_b  faultID/Imag\_c |
| 20 | Current angle of bus fault | O | Degree | faultID/Iang\_a  faultID/Iang\_b  faultID/Iang\_c |
| 21 | Connect/Disconnect status of shunt device | I/O | 1: Connected  0: Disconnected | shuntID/status\_a  shuntID/status\_b  shuntID/status\_c |
| 22 | Current injector magnitude | I | A or p.u. (RMS) | itemID/Imag |
| 23 | Current injector angle | I | Degree | itemID/Iang |
| 24 | Voltage source magnitude | I | V or p.u. (RMS) | itemID/Vmag |
| 25 | Voltage source angle | I | Degree | itemID/Vang |

**Important Notes When Trying to use different feeder models/Lessons learnt**

This conversion process is not perfect but needs some modifications. OpenDSS can run power flow and give results for many abnormal situations, but the same isn’t true when it comes to the converted input files in ePHASORSIM

1. Dataframes are created for writing into the excel spreadsheets. When writing the Dataframes from into the excel spreadsheets “NaNs” get written that are seen as empty spaces in the spreadsheets.
2. In case there is a line that does not have a load or PV connected. It must be removed or add “0s” for no-loads at the end of the feeder.
3. If buses have “\_”at the end of their ID’s they need to be removed, as the “\_” act as delimiters for phases. E.g. If a bus has a name in OpenDSS as “709099\_”, when transferred to ePHASORsim the ids in 3 phase representation would be “709099\_\_a”, “709099\_\_b”, and “709099\_\_c”. This would not work well in ePHASORSIM. Instead the ID’s in OpenDSS should be changed to “709099” to avoid any issues.
4. Make sure to only have KW, KVAr or Pf in the load file OpenDSS. Remove all unnecessary load shape, multiplication factor.
5. Make the substation impedance is a very low number .e.g. r0=1e-9, x0=1e-9, r1=1e-9, x1=1e-9. This makes the voltage at the source a perfect 1 pu.
6. Make sure the numtaps=32 in the OpenDSS file for the transformers, in case it is represented. If not mentioned the default is 32.
7. In OpenDSS, the constant PQ loads are transformed to constant Z loads when voltage levels cross the default levels of Vmin=0.95 and Vmax=1.05. This could be taken care by explicitly entering Vmin=0.8 and Vmax=1.2.

**References**

1. OpenDSS Program [Online], Available: <http://sourceforge.net/projects/electricdss>.
2. OPAL-RT (2014). ePHASORsim user guide. Retrieved from http://www.opal-rt.com/systems-ephasorsim/.
3. Python Software Foundation. Python Language Reference, version 2.7. Available at http://www.python.org.
4. Wes McKinney. Data Structures for Statistical Computing in Python, Proceedings of the 9th Python in Science Conference, 51-56 (2010).