# OpenDSS Training Workshop

## Basics and Scripting

Andres Ovalle, PhD
EPRI Knoxville, TN

August 22, 2022

ELECTRIFICATION & SUSTAINABLE ENERGY STRATEGY
TECHNOLOGY INNOVATION
GENERATION & LOW-CARBON RESOURCES
INTEGRATED GRID & ENERGY SYSTEMS
TRANSMISSION & DISTRIBUTION INFRASTRUCTURE
NUCLEAR
EPRI

# Instructor

## Andres Ovalle

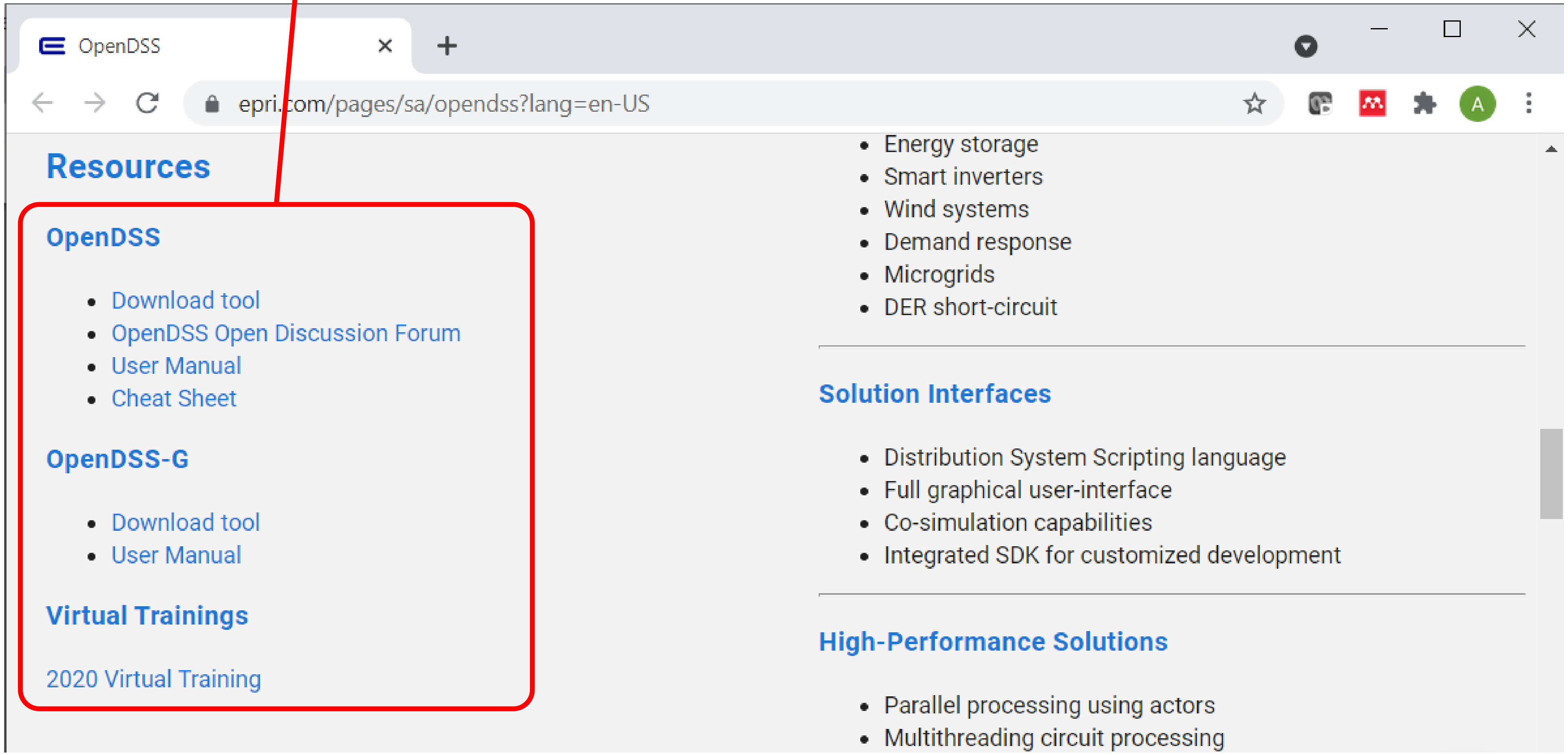
Andres Ovalle is an Engineer/Scientist III with the Power System Studies team at the Electric Power Research Institute (EPRI). His current research activities focus on modeling of power systems, system protection in distribution and transmission, and impacts geomagnetic disturbance related harmonics on power systems. Mr. Ovalle joined EPRI in 2018. Prior to joining EPRI, Mr. Ovalle was with the French National Railways Company (SNCF) and the Grenoble Electrical Engineering Laboratory (G2E-lab) for approximately 2 years where he worked as a postdoctoral research engineer in the use of energy storage for the support of electrified railways. Mr. Ovalle received the B.S.E.E. degree from the Universidad de Los Andes, Bogota, Colombia, in 2011, the M.E.E. degree from the Universidad de Los Andes in 2013, and the Ph.D. in Electrical Engineering from the Université de Grenoble Alpes, Grenoble, France, in 2016.

# Installation and Startup

# OpenDSS in EPRI.com

Plenty of resources to start with at EPRI.com

https://www.epri.com/pages/sa/opendss?lang=en-US



## Resources

### OpenDSS

- Download tool
- OpenDSS Open Discussion Forum
- User Manual
- Cheat Sheet

### OpenDSS-G

- Download tool
- User Manual

### Virtual Trainings

2020 Virtual Training

- Energy storage
- Smart inverters
- Wind systems
- Demand response
- Microgrids
- DER short-circuit

### Solution Interfaces

- Distribution System Scripting language
- Full graphical user-interface
- Co-simulation capabilities
- Integrated SDK for customized development

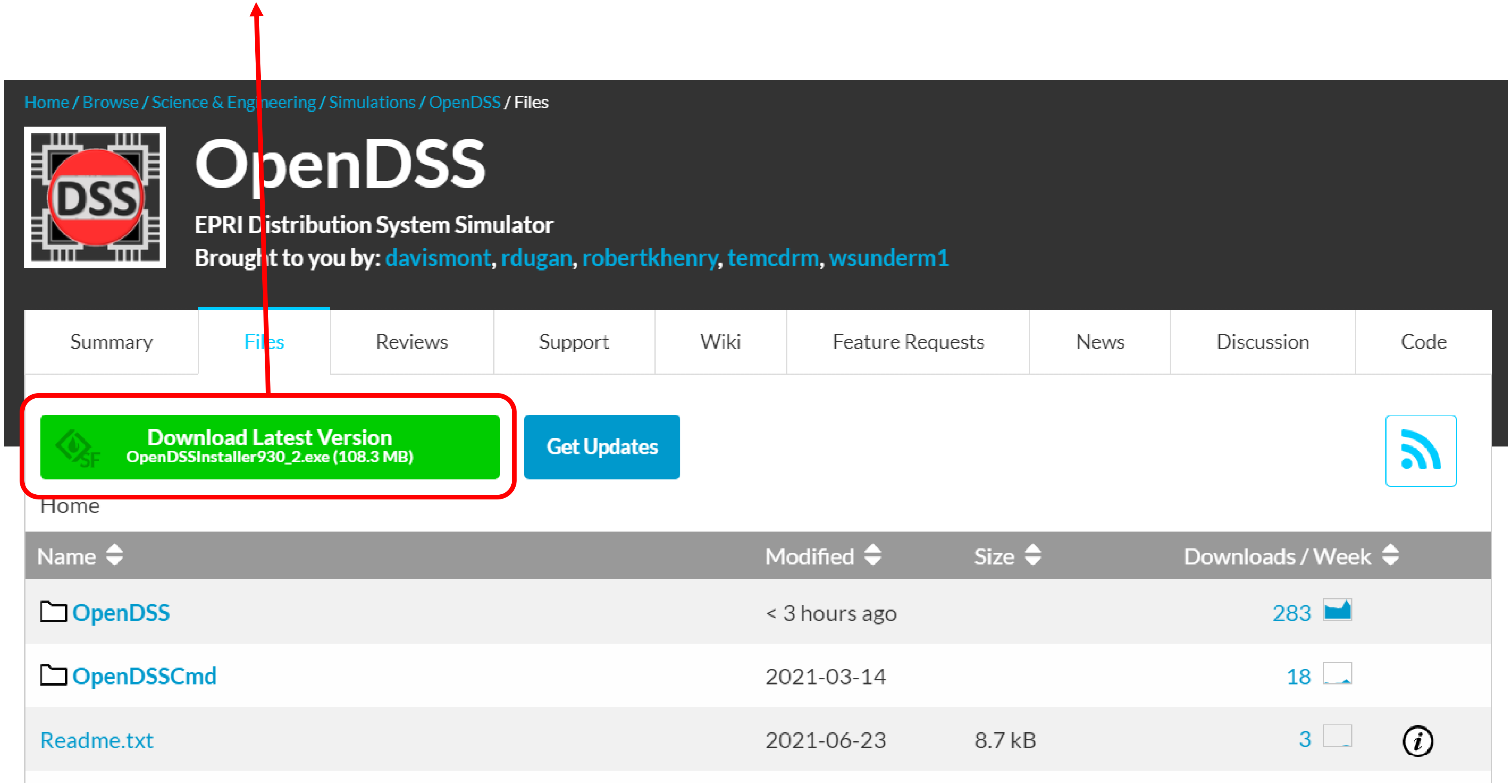
### High-Performance Solutions

- Parallel processing using actors
- Multithreading circuit processing

# OpenDSS in SourceForge.net

You can get the latest installer of OpenDSS from

https://sourceforge.net/projects/electricdss/files/



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

OpenDSS

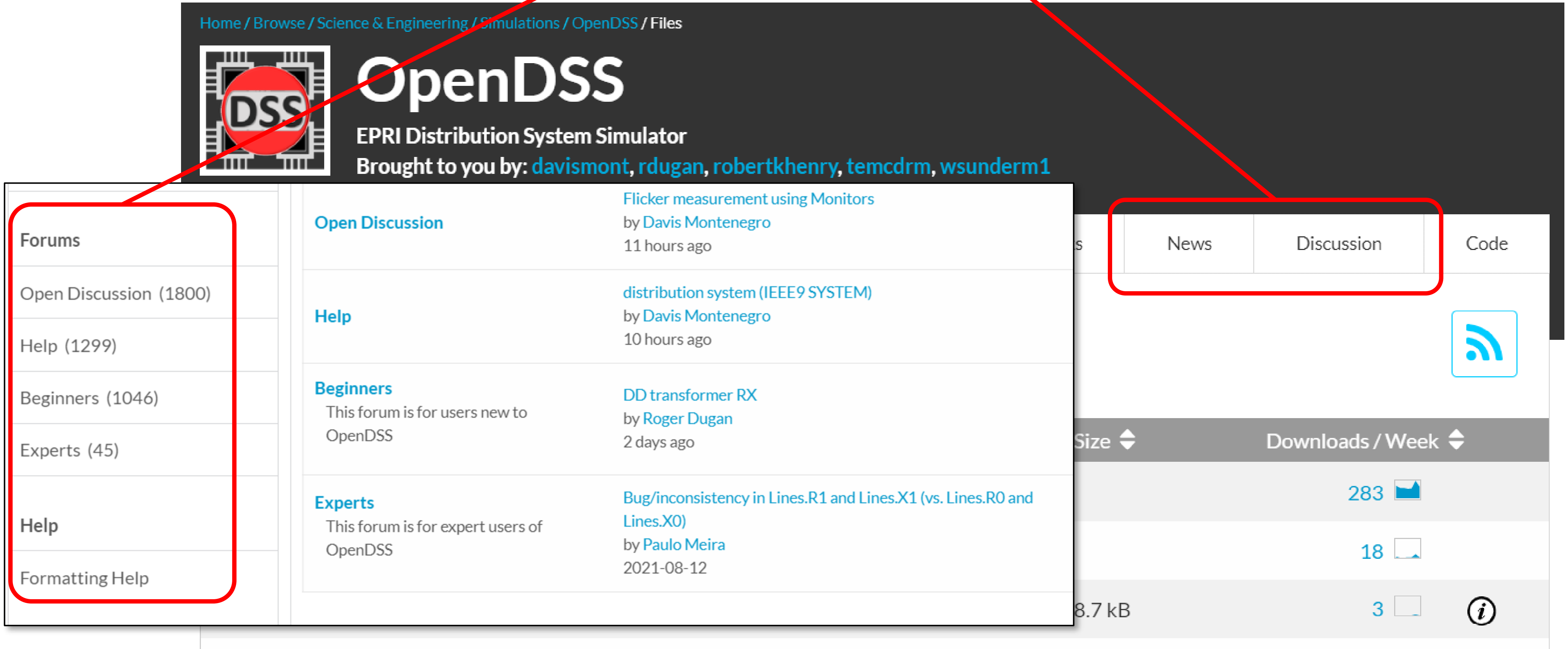
EPRI Distribution System Simulator

Brought to you by: davismont, rdugan, robertkhenry, temcdrm, wsunderm1

| Name | Modified | Size | Downloads / Week |
|---|---|---|---|
| OpenDSS | < 3 hours ago | | 283 |
| OpenDSSCmd | 2021-03-14 | | 18 |
| Readme.txt | 2021-06-23 | 8.7 kB | 3 |

# OpenDSS in SourceForge.net

The *News* and forums in the *Discussion* tab are one of the best resources to get your questions answered



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

OpenDSS

EPRI Distribution System Simulator

Brought to you by: davismont, rdugan, robertkhenry, temcdrm, wsunderm1

**Forums**

- Open Discussion (1800)
- Help (1299)
- Beginners (1046)
- Experts (45)

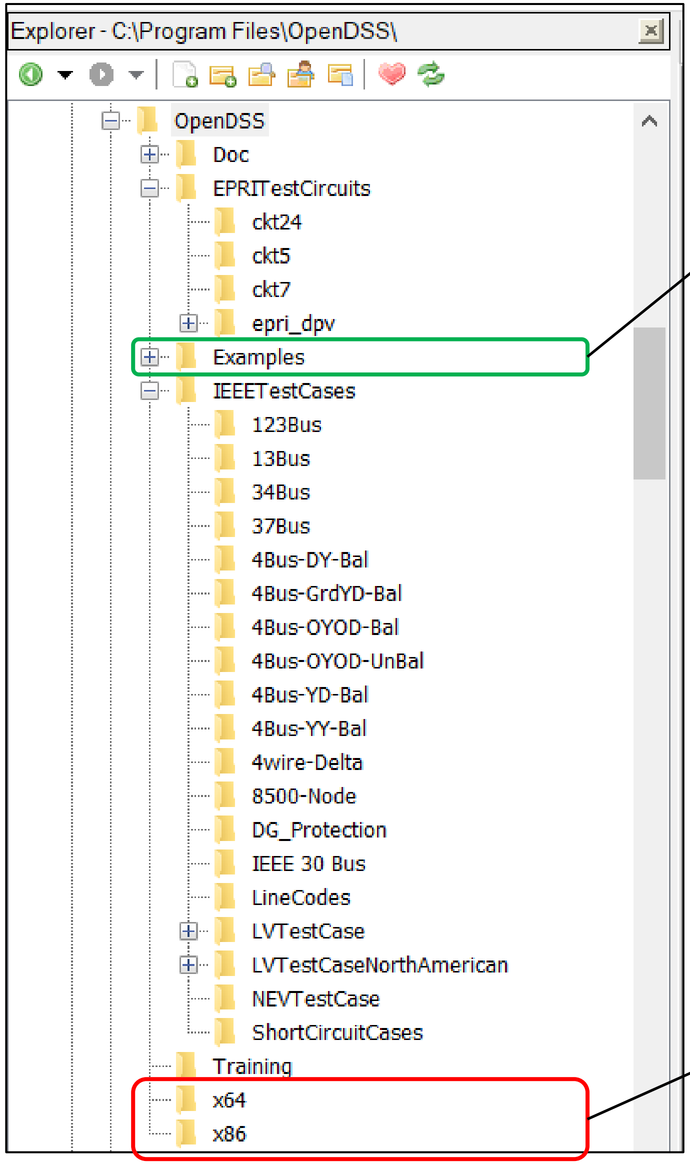
**Help**

- Formatting Help

| | |
|---|---|
| **Open Discussion** | Flicker measurement using Monitors<br>by Davis Montenegro<br>11 hours ago |
| **Help** | distribution system (IEEE9 SYSTEM)<br>by Davis Montenegro<br>10 hours ago |
| **Beginners**<br>This forum is for users new to OpenDSS | DD transformer RX<br>by Roger Dugan<br>2 days ago |
| **Experts**<br>This forum is for expert users of OpenDSS | Bug/inconsistency in Lines.R1 and Lines.X1 (vs. Lines.R0 and Lines.X0)<br>by Paulo Meira<br>2021-08-12 |

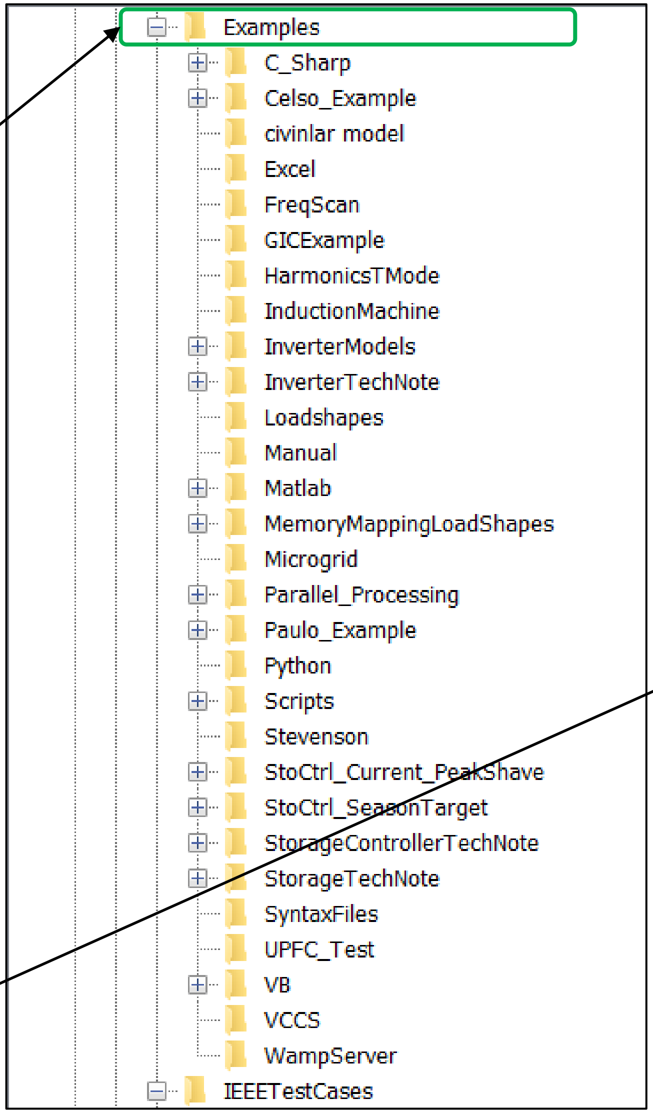
| Size | Downloads / Week |
|---|---|
| | 283 |
| | 18 |
| 8.7 kB | 3 |

# OpenDSS Files in Installation folder



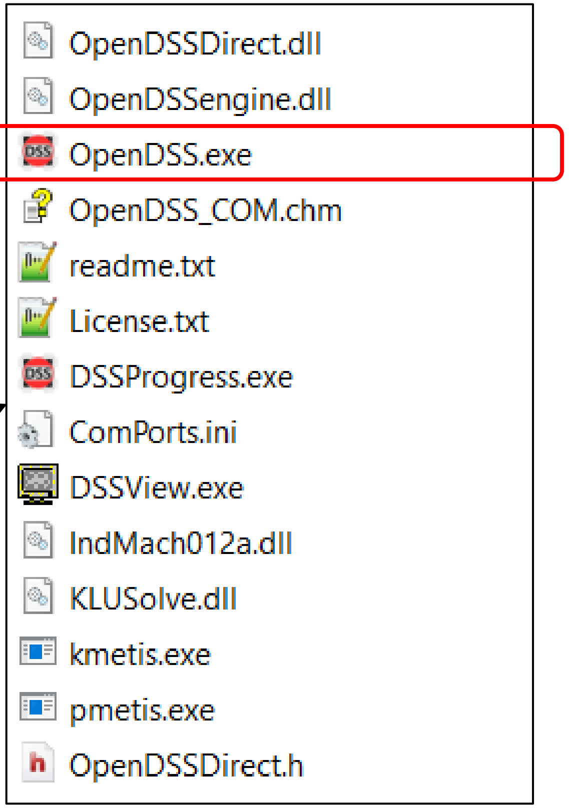
Explorer - C:\Program Files\OpenDSS\

- OpenDSS
  - Doc
  - EPRITestCircuits
    - ckt24
    - ckt5
    - ckt7
    - epri_dpv
  - Examples
  - IEEETestCases
    - 123Bus
    - 13Bus
    - 34Bus
    - 37Bus
    - 4Bus-DY-Bal
    - 4Bus-GrdYD-Bal
    - 4Bus-OYOD-Bal
    - 4Bus-OYOD-UnBal
    - 4Bus-YD-Bal
    - 4Bus-YY-Bal
    - 4wire-Delta
    - 8500-Node
    - DG_Protection
    - IEEE 30 Bus
    - LineCodes
    - LVTestCase
    - LVTestCaseNorthAmerican
    - NEVTestCase
    - ShortCircuitCases
  - Training
  - x64
  - x86



- Examples
  - C_Sharp
  - Celso_Example
  - civinlar model
  - Excel
  - FreqScan
  - GICExample
  - HarmonicsTMode
  - InductionMachine
  - InverterModels
  - InverterTechNote
  - Loadshapes
  - Manual
  - Matlab
  - MemoryMappingLoadShapes
  - Microgrid
  - Parallel_Processing
  - Paulo_Example
  - Python
  - Scripts
  - Stevenson
  - StoCtrl_Current_PeakShave
  - StoCtrl_SeasonTarget
  - StorageControllerTechNote
  - StorageTechNote
  - SyntaxFiles
  - UPFC_Test
  - VB
  - VCCS
  - WampServer
- IEEETestCases

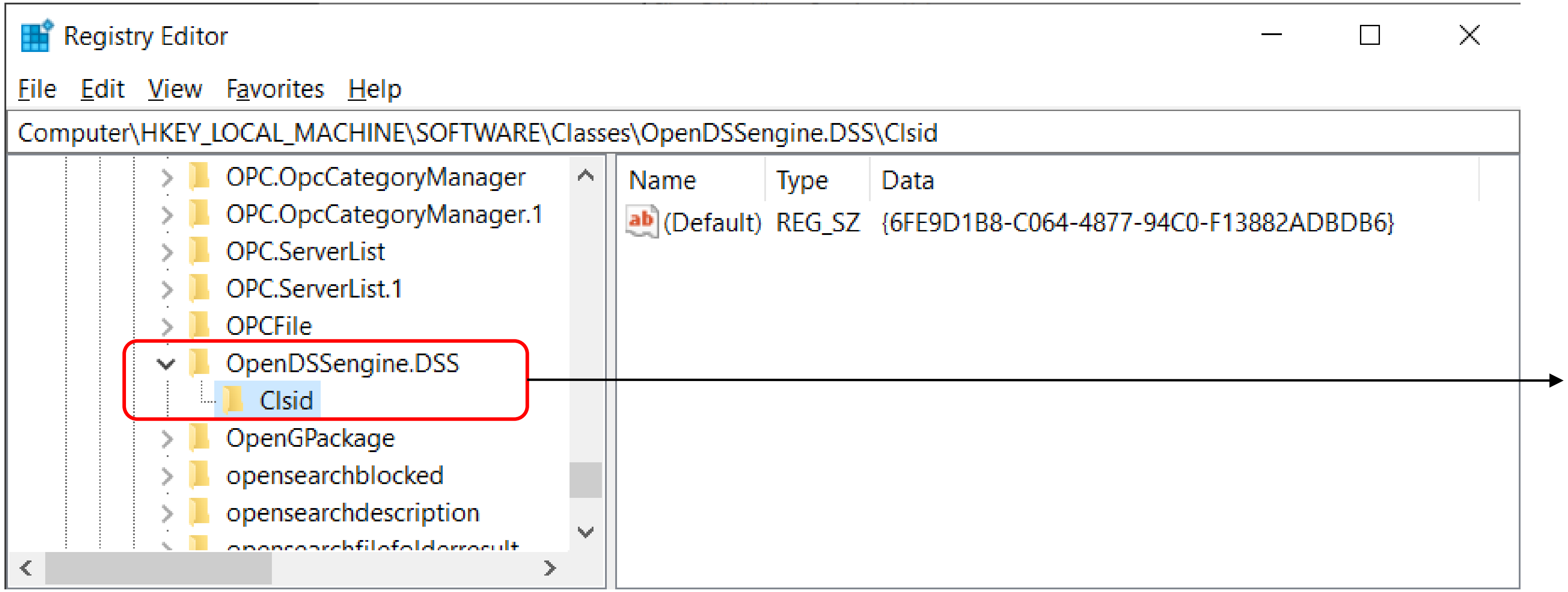
Main OpenDSS program files



- OpenDSSDirect.dll
- OpenDSSengine.dll
- OpenDSS.exe
- OpenDSS_COM.chm
- readme.txt
- License.txt
- DSSProgress.exe
- ComPorts.ini
- DSSView.exe
- IndMach012a.dll
- KLUSolve.dll
- kmetis.exe
- pmetis.exe
- OpenDSSDirect.h

# COM Server Registration

The installer automatically registers the OpenDSS COM server in the Windows Registry Entry



Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Classes\OpenDSSengine.DSS\Clsid

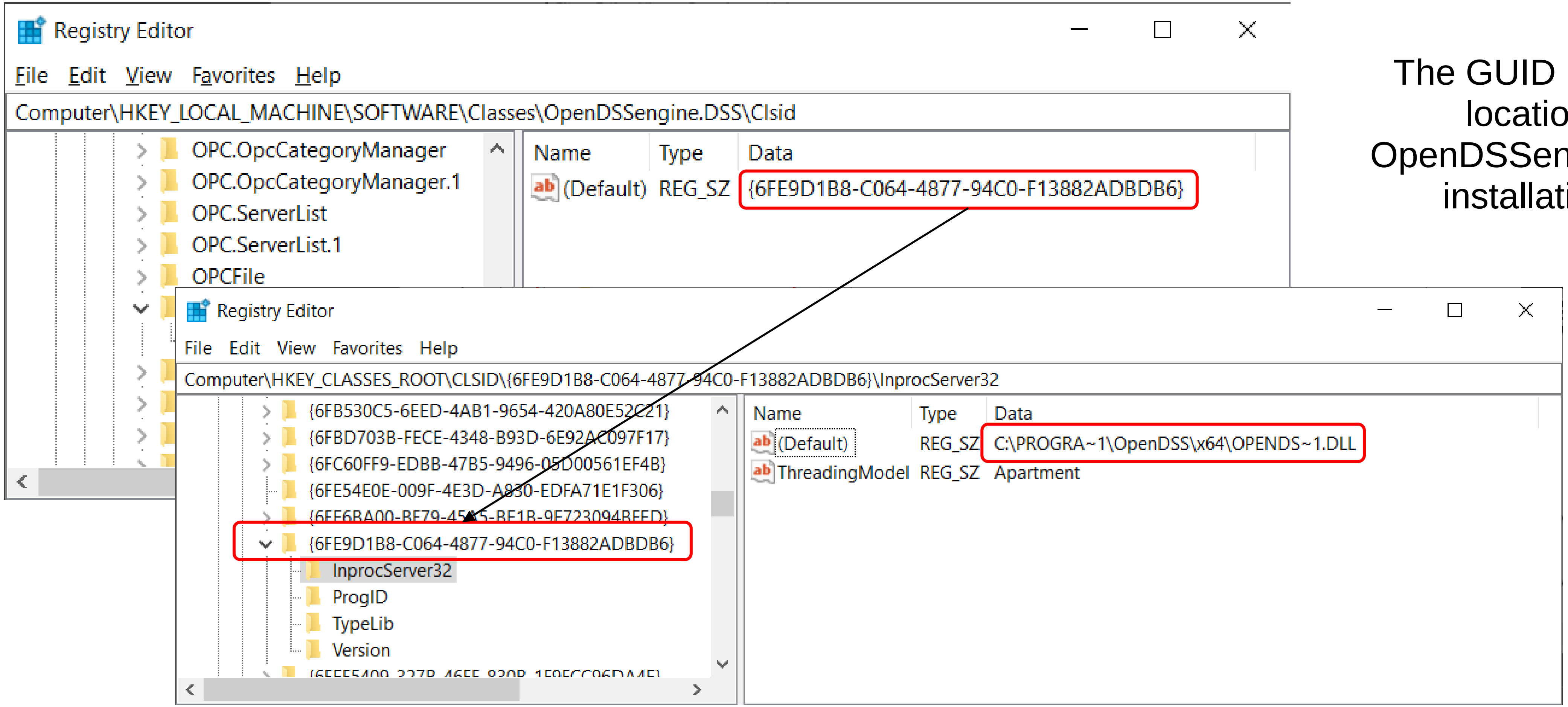
| Name | Type | Data |
|---|---|---|
| (Default) | REG_SZ | {6FE9D1B8-C064-4877-94C0-F13882ADBDB6} |

The COM Server shows up as OpenDSSEngine.DSS in the **Windows Registry**

The OpenDSS COM server is now available to any COM client/program on the computer

# COM Server Registration

The installer automatically registers the OpenDSS COM server in the Windows Registry Entry



Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Classes\OpenDSSengine.DSS\Clsid

- OPC.OpcCategoryManager
- OPC.OpcCategoryManager.1
- OPC.ServerList
- OPC.ServerList.1
- OPCFile

| Name | Type | Data |
|---|---|---|
| (Default) | REG_SZ | {6FE9D1B8-C064-4877-94C0-F13882ADBDB6} |

Computer\HKEY_CLASSES_ROOT\CLSID\{6FE9D1B8-C064-4877-94C0-F13882ADBDB6}\InprocServer32

- {6FB530C5-6EED-4AB1-9654-420A80E52C21}
- {6FBD703B-FECE-4348-B93D-6E92AC097F17}
- {6FC60FF9-EDBB-47B5-9496-05D00561EF4B}
- {6FE54E0E-009F-4E3D-A830-EDFA71E1F306}
- {6FE6BA00-BE79-45[illegible]5-BE1B-9F723094BFED}
- {6FE9D1B8-C064-4877-94C0-F13882ADBDB6}
  - InprocServer32
  - ProgID
  - TypeLib
  - Version
- {6FEF5409-327B-46EE-930B-1F9FCC96DA4E}

| Name | Type | Data |
|---|---|---|
| (Default) | REG_SZ | C:\PROGRA~1\OpenDSS\x64\OPENDS~1.DLL |
| ThreadingModel | REG_SZ | Apartment |

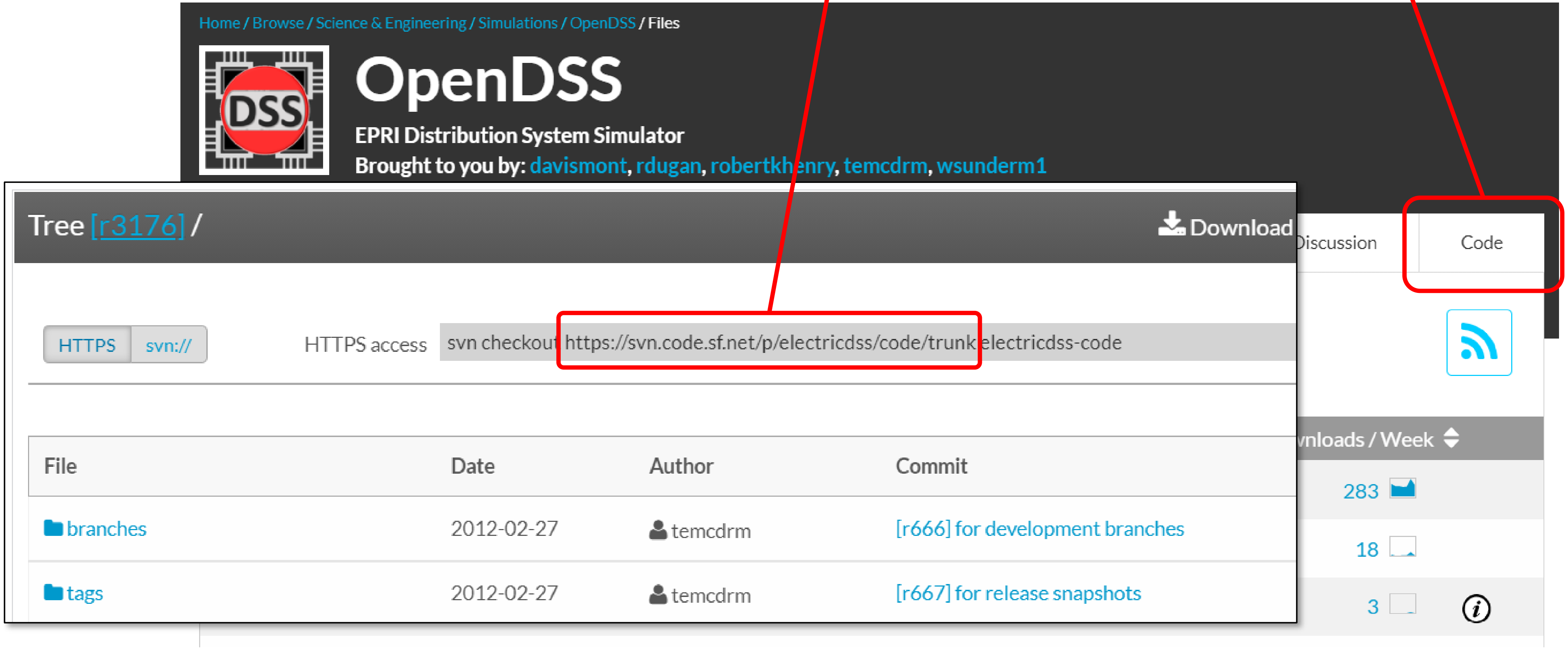
The GUID points to the location of the OpenDSSengine.dll in the installation folder

# Getting the Source Code

# OpenDSS in SourceForge.net

In the *Code* tab you can get the URL of the main repository or download a snapshot of the source code

https://svn.code.sf.net/p/electricdss/code/trunk



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

OpenDSS

EPRI Distribution System Simulator

Brought to you by: davismont, rdugan, robertkhenry, temcdrm, wsunderm1

Tree [r3176] /

HTTPS access

```
svn checkout https://svn.code.sf.net/p/electricdss/code/trunk electricdss-code
```

| File | Date | Author | Commit |
|---|---|---|---|
| branches | 2012-02-27 | temcdrm | [r666] for development branches |
| tags | 2012-02-27 | temcdrm | [r667] for release snapshots |

| vnloads / Week |
|---|
| 283 |
| 18 |
| 3 |

# Accessing the Source Code Repository with TortoiseSVN

- Install a **TortoiseSVN** client from [https://tortoisesvn.net/downloads.html](https://tortoisesvn.net/downloads.html)

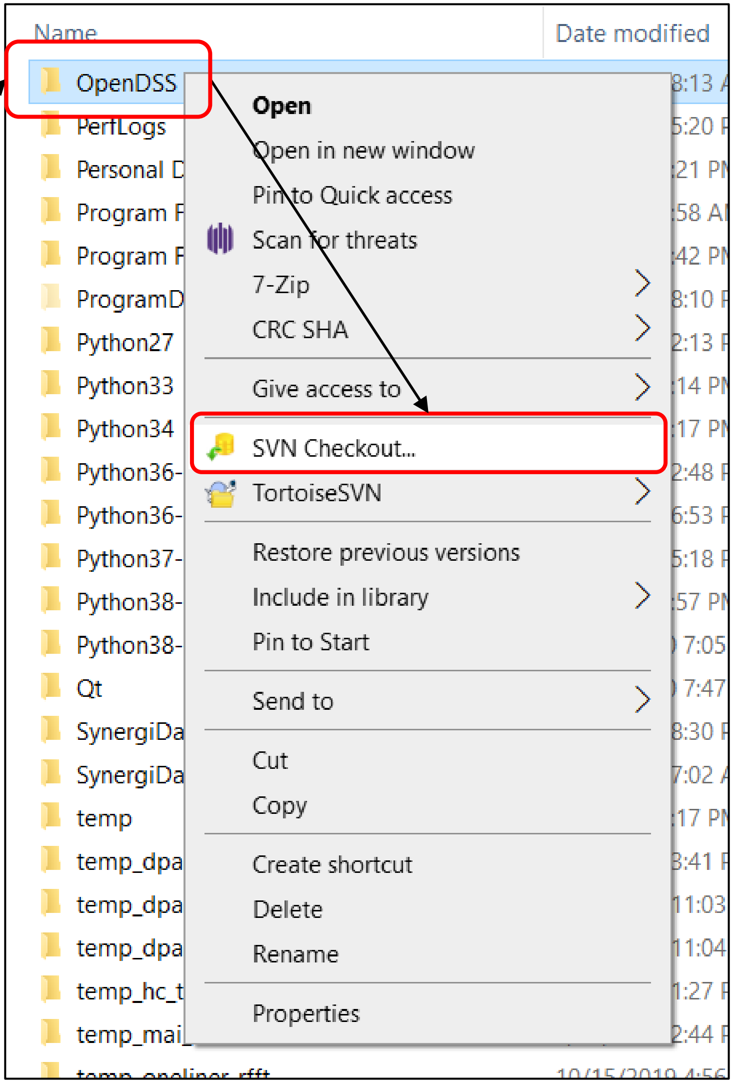
- Grab the OpenDSS files from SourceForge:

1. Create a clean directory such as `C:\OpenDSS`

2. Right-click on it and choose *SVN Checkout…* from the menu

3. The repository URL is [https://svn.code.sf.net/p/electricdss/code/trunk](https://svn.code.sf.net/p/electricdss/code/trunk)

Thereafter, to update a folder or file, right-click on the folder or file and select **SVN Update**



# Accessing the Source Code Repository with TortoiseSVN

- Install a **TortoiseSVN** client from https://tortoisesvn.net/downloads.html

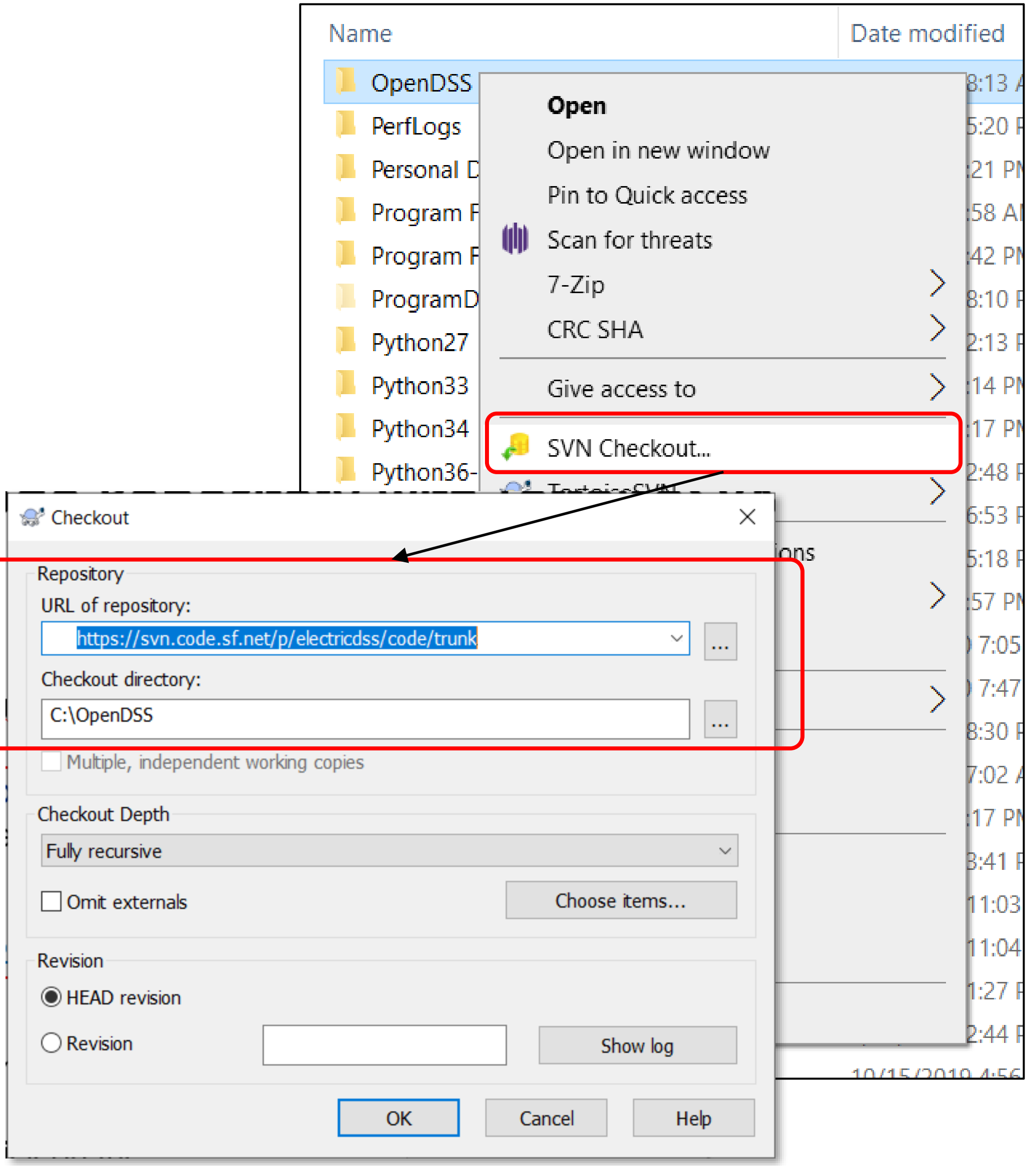
- Grab the OpenDSS files from SourceForge:

1. Create a clean directory such as `C:\OpenDSS`

2. Right-click on it and choose *SVN Checkout…* from the menu

3. The repository URL is https://svn.code.sf.net/p/electricdss/code/trunk

Thereafter, to update a folder or file, right-click on the folder or file and select **SVN Update**



Repository

URL of repository: https://svn.code.sf.net/p/electricdss/code/trunk

Checkout directory: C:\OpenDSS

□ Multiple, independent working copies

Checkout Depth: Fully recursive

□ Omit externals

Revision

● HEAD revision

○ Revision

# Accessing the Source Code Repository with TortoiseSVN

- Install a **TortoiseSVN** client from [https://tortoisesvn.net/downloads.html](https://tortoisesvn.net/downloads.html)

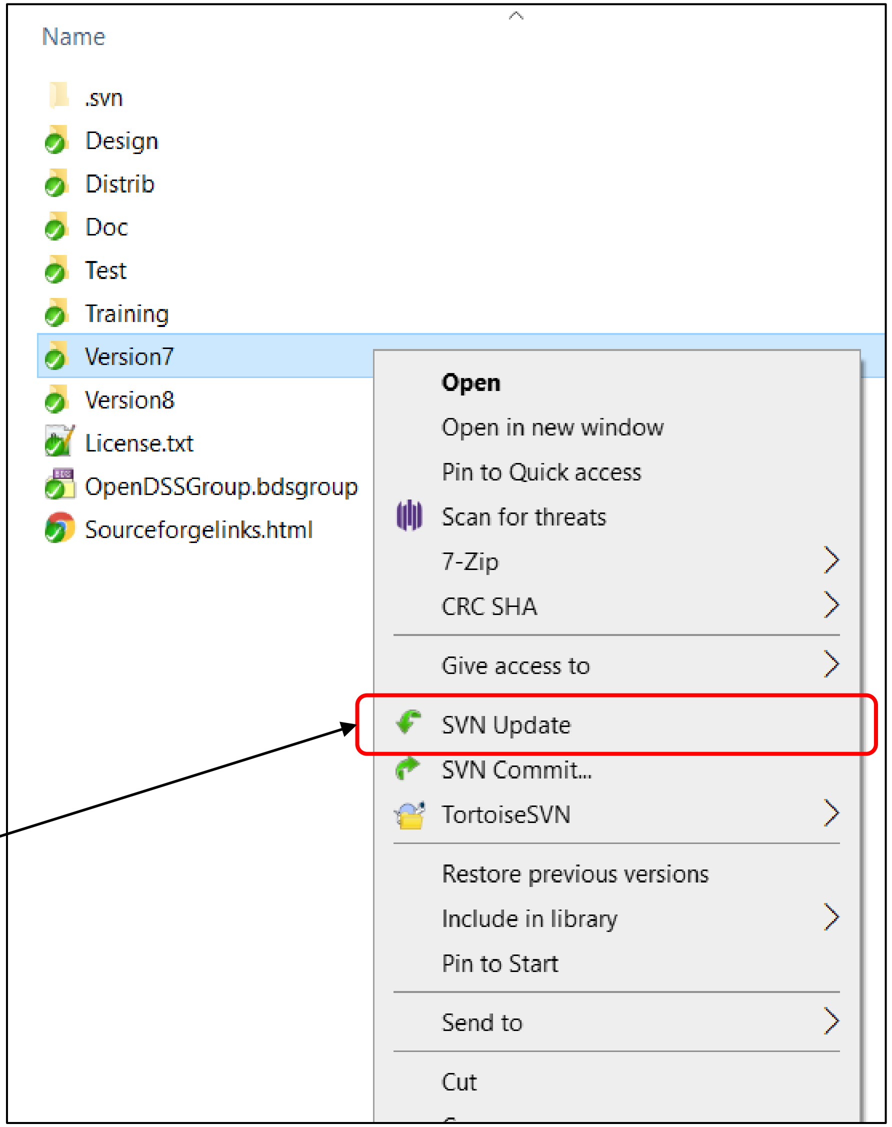
- Grab the OpenDSS files from SourceForge:

  1. Create a clean directory such as `C:\OpenDSS`

  2. Right-click on it and choose *SVN Checkout…* from the menu

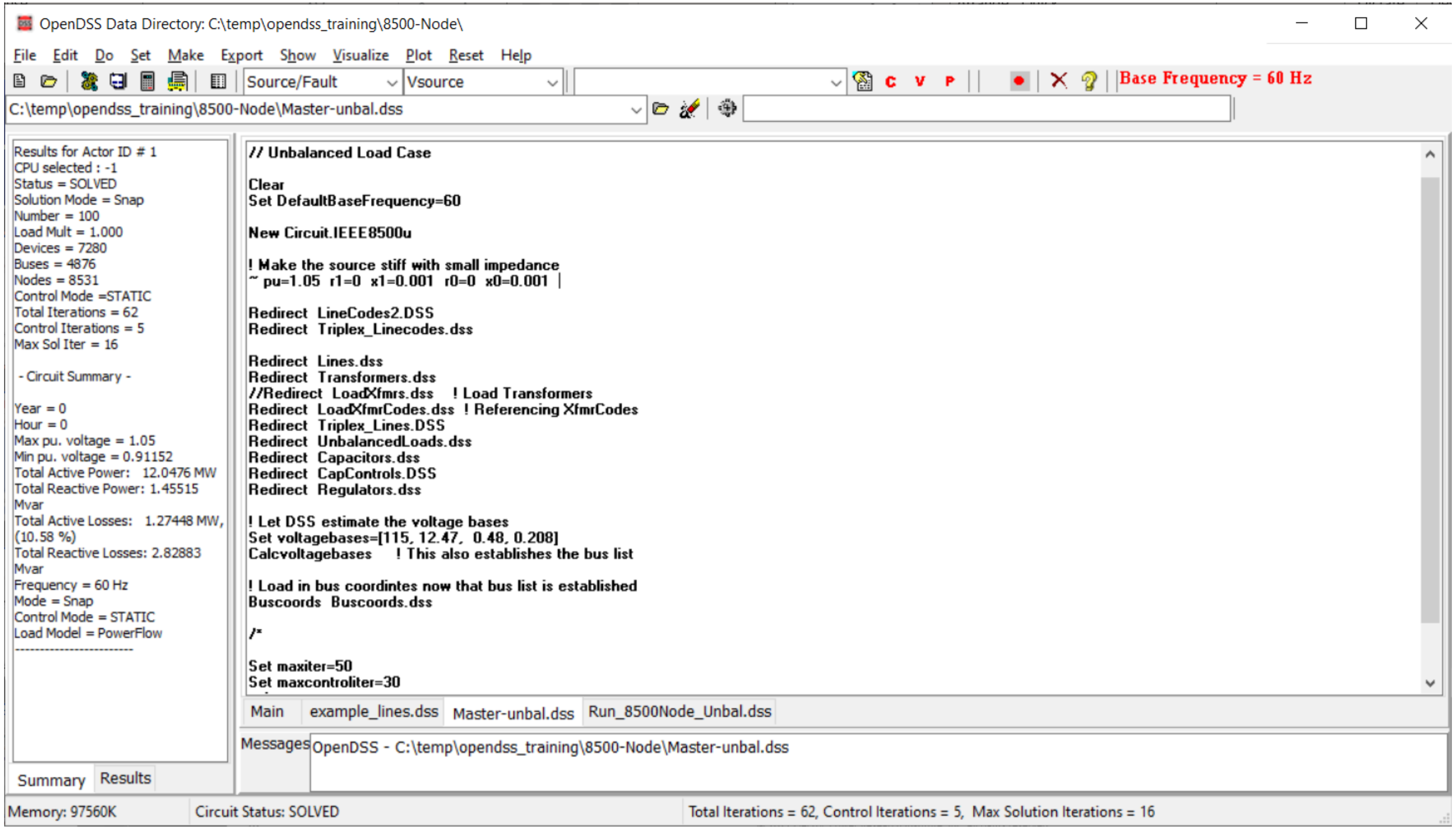
  3. The repository URL is [https://svn.code.sf.net/p/electricdss/code/trunk](https://svn.code.sf.net/p/electricdss/code/trunk)

Thereafter, to update a folder or file, right-click on the folder or file and select **SVN Update**



# Scripting Basics and Examples

# OpenDSS interface



OpenDSS Data Directory: C:\temp\opendss_training\8500-Node\

Base Frequency = 60 Hz

C:\temp\opendss_training\8500-Node\Master-unbal.dss

```
Results for Actor ID # 1
CPU selected : -1
Status = SOLVED
Solution Mode = Snap
Number = 100
Load Mult = 1.000
Devices = 7280
Buses = 4876
Nodes = 8531
Control Mode =STATIC
Total Iterations = 62
Control Iterations = 5
Max Sol Iter = 16

 - Circuit Summary -

Year = 0
Hour = 0
Max pu. voltage = 1.05
Min pu. voltage = 0.91152
Total Active Power:   12.0476 MW
Total Reactive Power: 1.45515
Mvar
Total Active Losses:   1.27448 MW,
(10.58 %)
Total Reactive Losses: 2.82883
Mvar
Frequency = 60 Hz
Mode = Snap
Control Mode = STATIC
Load Model = PowerFlow
------------------------
```

```
// Unbalanced Load Case

Clear
Set DefaultBaseFrequency=60

New Circuit.IEEE8500u

! Make the source stiff with small impedance
~ pu=1.05  r1=0  x1=0.001  r0=0  x0=0.001

Redirect  LineCodes2.DSS
Redirect  Triplex_Linecodes.dss

Redirect  Lines.dss
Redirect  Transformers.dss
//Redirect  LoadXfmrs.dss    ! Load Transformers
Redirect  LoadXfmrCodes.dss  ! Referencing XfmrCodes
Redirect  Triplex_Lines.DSS
Redirect  UnbalancedLoads.dss
Redirect  Capacitors.dss
Redirect  CapControls.DSS
Redirect  Regulators.dss

! Let DSS estimate the voltage bases
Set voltagebases=[115, 12.47,  0.48, 0.208]
Calcvoltagebases     ! This also establishes the bus list

! Load in bus coordintes now that bus list is established
Buscoords  Buscoords.dss

/*

Set maxiter=50
Set maxcontroliter=30
```

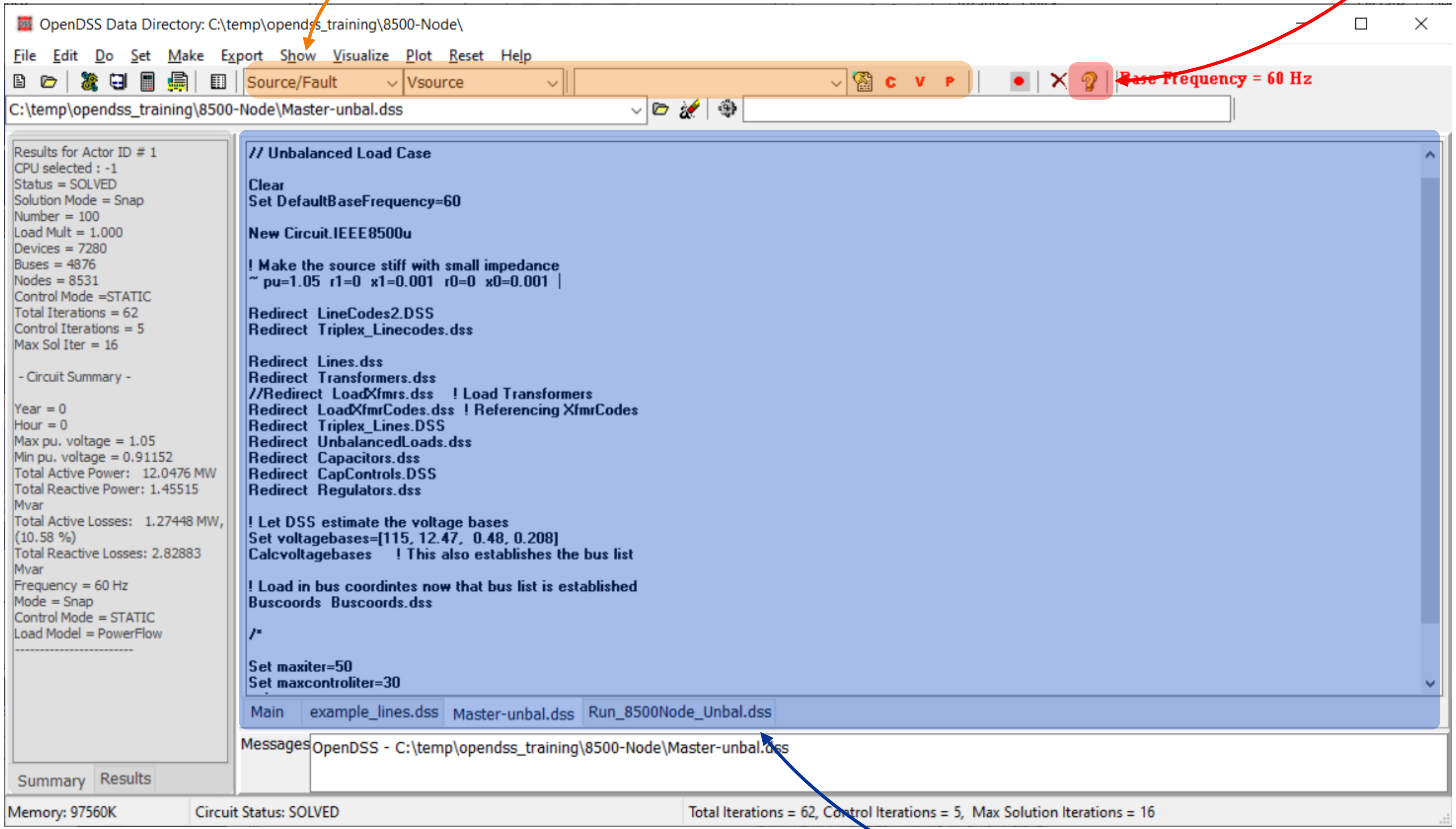
Messages: OpenDSS - C:\temp\opendss_training\8500-Node\Master-unbal.dss

Memory: 97560K | Circuit Status: SOLVED | Total Iterations = 62, Control Iterations = 5,  Max Solution Iterations = 16

# OpenDSS interface

DSS Element toolbar

Command and Properties help


OpenDSS Data Directory: C:\temp\opendss_training\8500-Node\
File Edit Do Set Make Export Show Visualize Plot Reset Help
Source/Fault
Vsource
C V P
Base Frequency = 60 Hz
C:\temp\opendss_training\8500-Node\Master-unbal.dss
Summary
Results
Main
example_lines.dss
Master-unbal.dss
Run_8500Node_Unbal.dss
Messages

```
Results for Actor ID # 1
CPU selected : -1
Status = SOLVED
Solution Mode = Snap
Number = 100
Load Mult = 1.000
Devices = 7280
Buses = 4876
Nodes = 8531
Control Mode =STATIC
Total Iterations = 62
Control Iterations = 5
Max Sol Iter = 16

 - Circuit Summary -

Year = 0
Hour = 0
Max pu. voltage = 1.05
Min pu. voltage = 0.91152
Total Active Power:   12.0476 MW
Total Reactive Power: 1.45515
Mvar
Total Active Losses:   1.27448 MW,
(10.58 %)
Total Reactive Losses: 2.82883
Mvar
Frequency = 60 Hz
Mode = Snap
Control Mode = STATIC
Load Model = PowerFlow
-----------------------
```

```
// Unbalanced Load Case

Clear
Set DefaultBaseFrequency=60

New Circuit.IEEE8500u

! Make the source stiff with small impedance
~ pu=1.05  r1=0  x1=0.001  r0=0  x0=0.001

Redirect  LineCodes2.DSS
Redirect  Triplex_Linecodes.dss

Redirect  Lines.dss
Redirect  Transformers.dss
//Redirect  LoadXfmrs.dss    ! Load Transformers
Redirect  LoadXfmrCodes.dss  ! Referencing XfmrCodes
Redirect  Triplex_Lines.DSS
Redirect  UnbalancedLoads.dss
Redirect  Capacitors.dss
Redirect  CapControls.DSS
Redirect  Regulators.dss

! Let DSS estimate the voltage bases
Set voltagebases=[115, 12.47,  0.48, 0.208]
Calcvoltagebases     ! This also establishes the bus list

! Load in bus coordintes now that bus list is established
Buscoords  Buscoords.dss

/*

Set maxiter=50
Set maxcontroliter=30
```

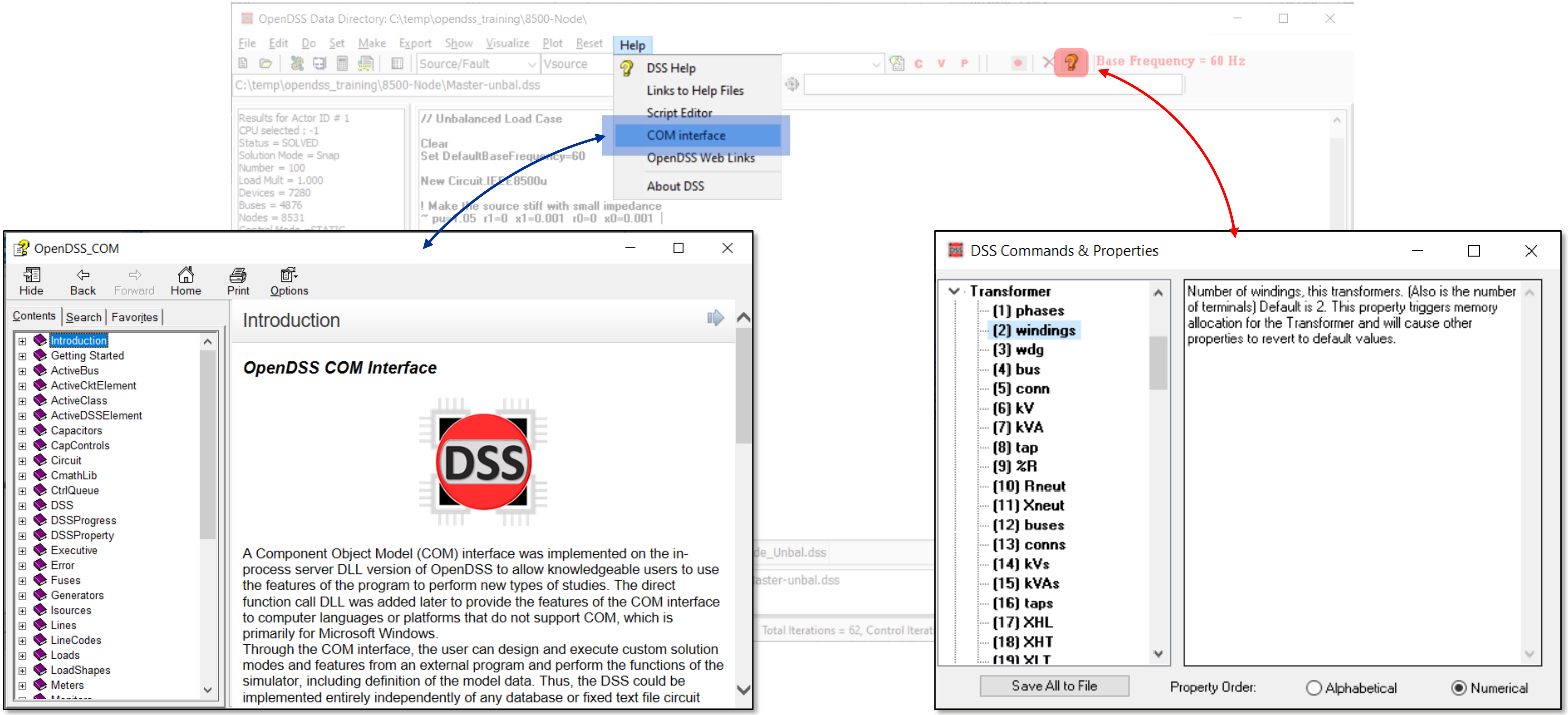
```
OpenDSS - C:\temp\opendss_training\8500-Node\Master-unbal.dss
```

```
Memory: 97560K    Circuit Status: SOLVED    Total Iterations = 62, Control Iterations = 5,  Max Solution Iterations = 16
```

Solution / Results summaries

DSS script editor / navigator

# OpenDSS interface



```
Results for Actor ID # 1
CPU selected : -1
Status = SOLVED
Solution Mode = Snap
Number = 100
Load Mult = 1.000
Devices = 7280
Buses = 4876
Nodes = 8531
Control Mode =STATIC
```

```
// Unbalanced Load Case

Clear
Set DefaultBaseFrequency=60

New Circuit.IEEE8500u

! Make the source stiff with small impedance
~ pu=1.05  r1=0  x1=0.001  r0=0  x0=0.001
```

## Introduction

### *OpenDSS COM Interface*

A Component Object Model (COM) interface was implemented on the in-process server DLL version of OpenDSS to allow knowledgeable users to use the features of the program to perform new types of studies. The direct function call DLL was added later to provide the features of the COM interface to computer languages or platforms that do not support COM, which is primarily for Microsoft Windows.
Through the COM interface, the user can design and execute custom solution modes and features from an external program and perform the functions of the simulator, including definition of the model data. Thus, the DSS could be implemented entirely independently of any database or fixed text file circuit

Number of windings, this transformers. (Also is the number of terminals) Default is 2. This property triggers memory allocation for the Transformer and will cause other properties to revert to default values.

Property Order: ○ Alphabetical ● Numerical

# Summary of Commands and Scripting Characters

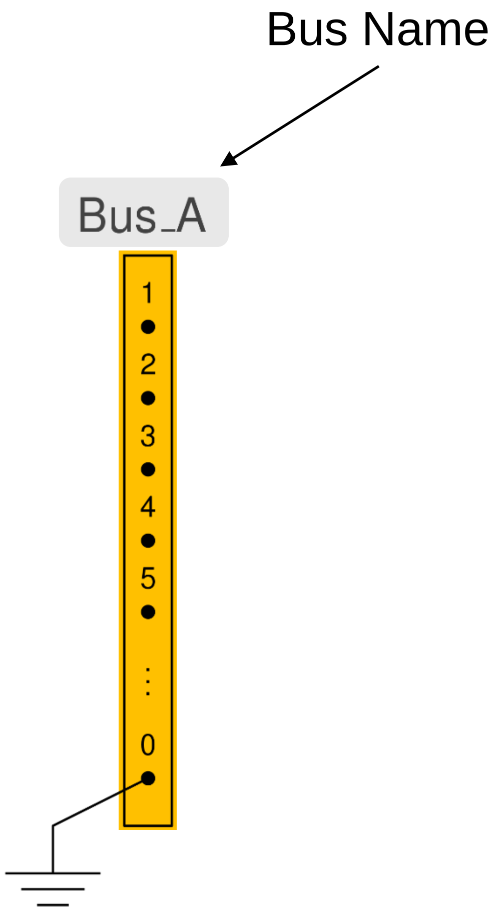
## Scripting

| Character | Description |
|---:|---|
| ~ | Line continuation |
| [] () {} “” ‘’ | Arrays, Matrices, Strings, Math Expressions |
| \| | Row delimiter in matrices |
| , space tab | Delimiter command parameters and array/matrix entries |
| . | Delimiter class-object, bus-node |
| = | Delimiter parameter tag and parameter value |
| ! // | In line comments |
| /*…*/ | Block comments |

## Commands (common)

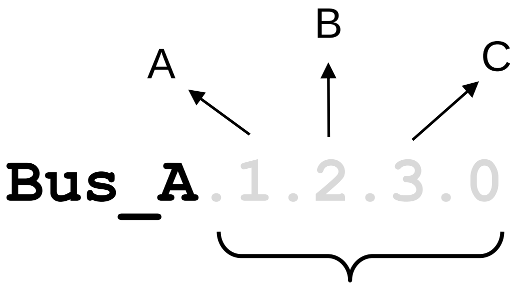
| Command | Description |
|---:|---|
| New | Create new DSS elements |
| Edit | Edit existing DSS elements |
| Set | Define solution modes and options |
| Solve | Perform current solution mode |
| Show | Write selected results to text files |
| Export | Export solution variables in CSV or XML format |
| Plot | Plots results with built-in plotting features |

# OpenDSS Buses and Nodes



Buses are a collection of 1 or more nodes

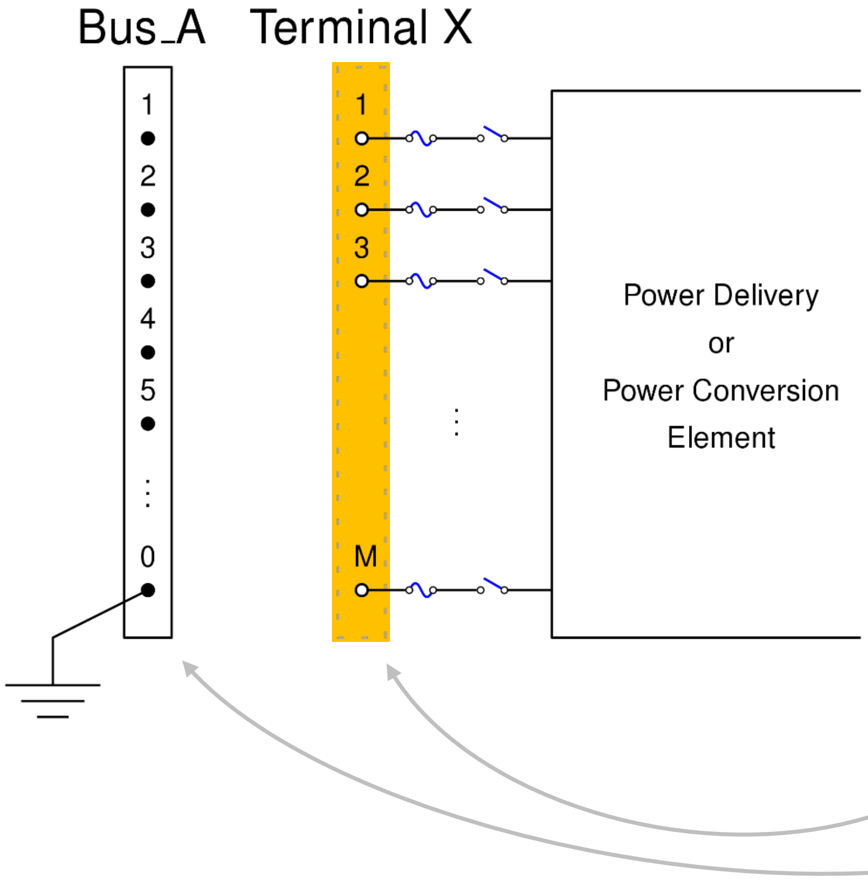
- A bus can host several nodes
- Node 0 of the bus is always grounded (0 V)
- Conventionally nodes 1 2 and 3 are associated to phases A B and C (but it is not mandatory)



Assumed by default for 3-ph elements if not specified

# OpenDSS Terminals

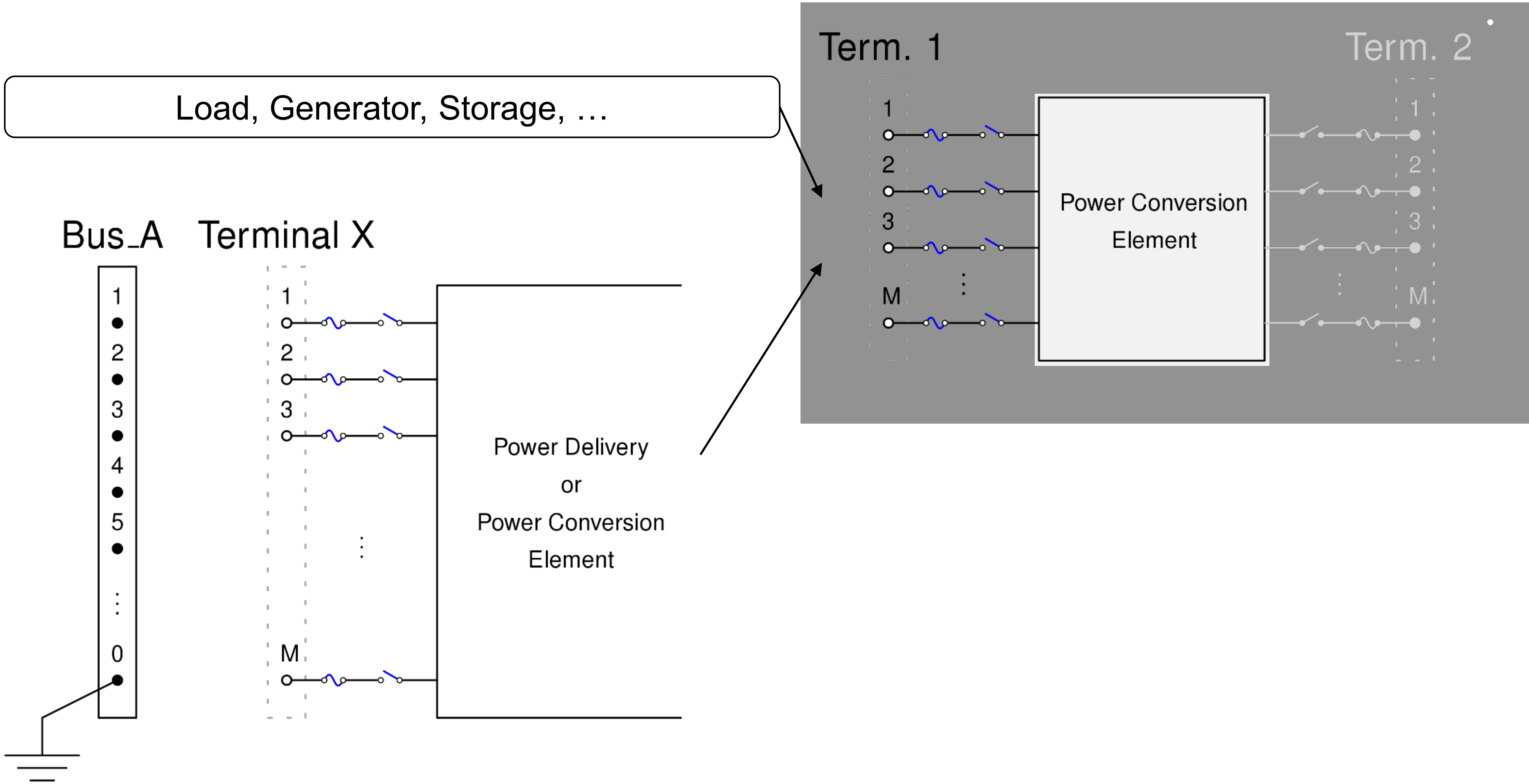
A Terminal can be associated to <u>only one</u> Bus.
A Bus can be associated to multiple terminals



- Circuit Elements can have one or more terminals
- A Terminal is a collection of *M* conductors
- Circuit Elements have fuses in series with switches on each conductor of each of its terminals

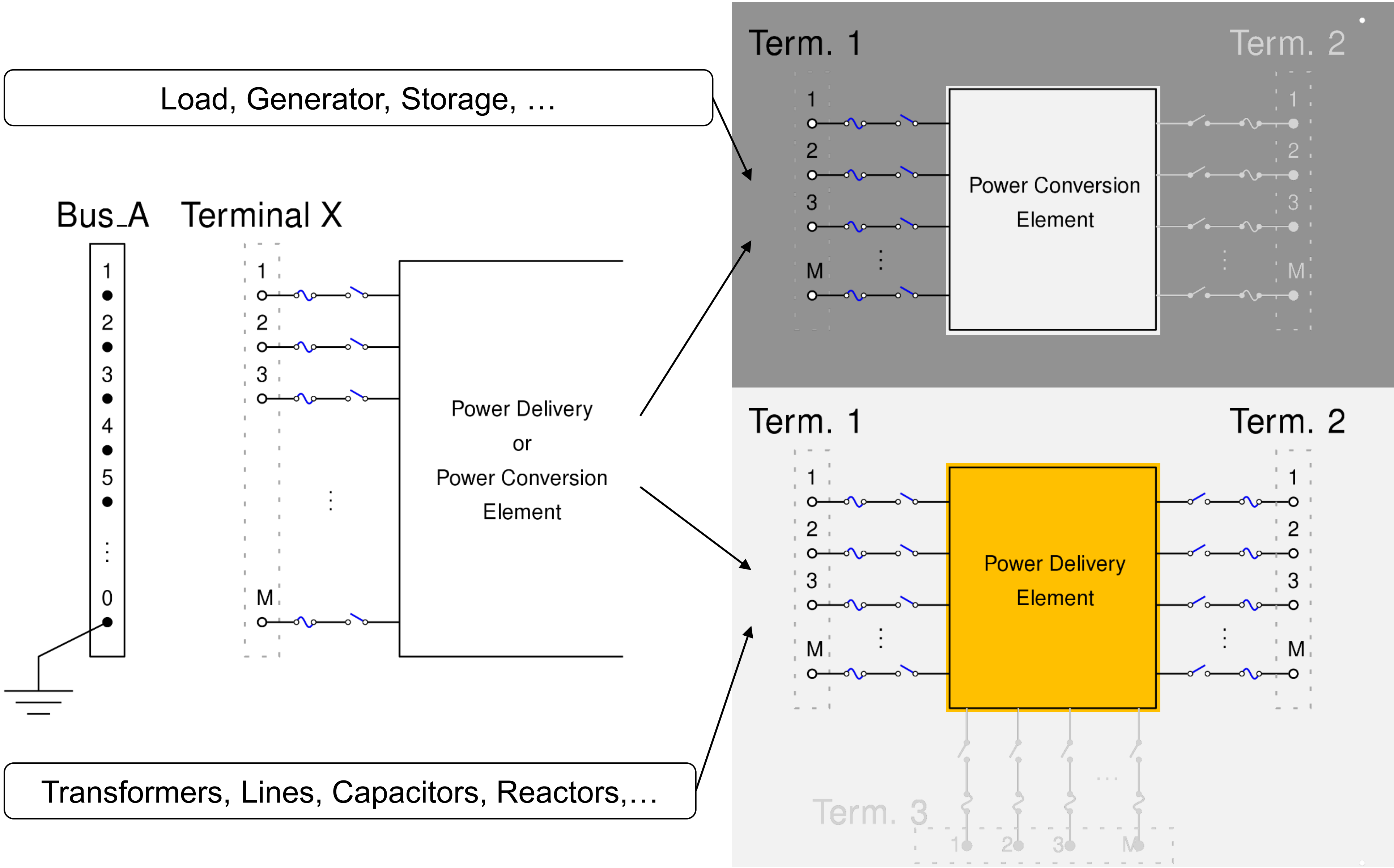
```
Bus1=Bus_A.1.2.3.0
```

# Power Delivery (PDE) & Power Conversion (PCE) Elements



- Typically, one terminal but not always
- Connected in *shunt* with PDEs
- Function of V
- May be nonlinear

# Power Delivery (PDE) & Power Conversion (PCE) Elements



Load, Generator, Storage, …

Transformers, Lines, Capacitors, Reactors,…

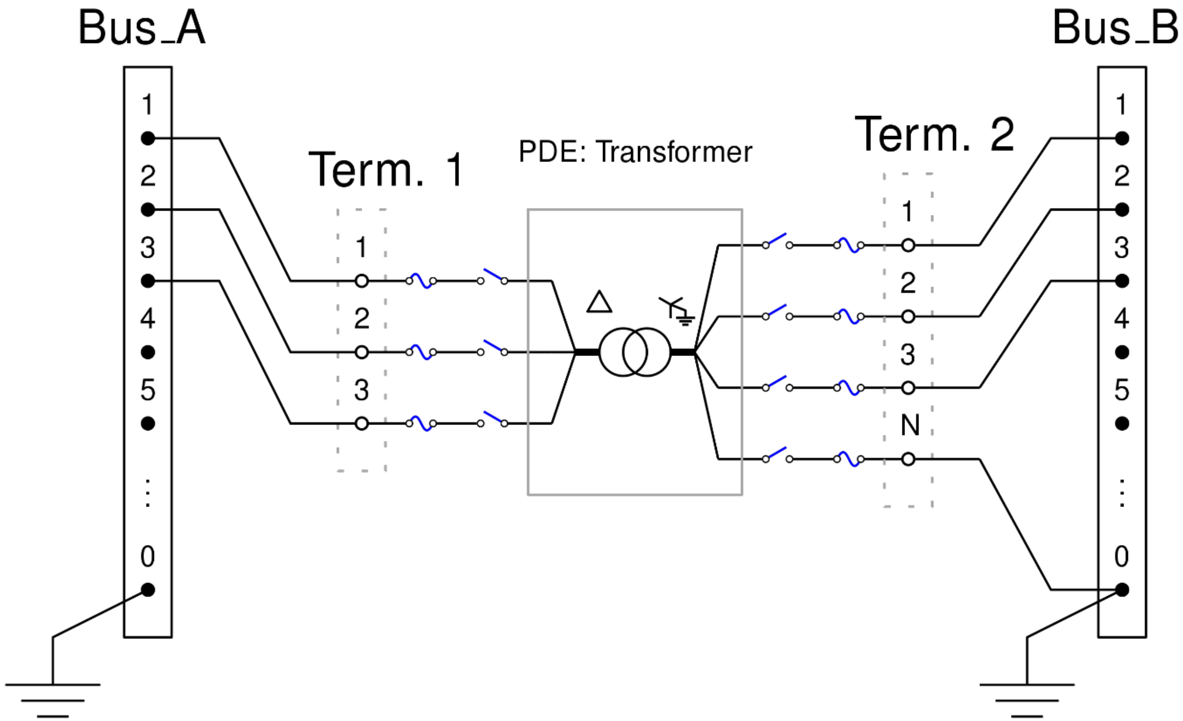
- Typically, one terminal but not always
- Connected in *shunt* with PDEs
- Function of V
- May be nonlinear

- Typically, two or more terminals
- Described by a *Yprim* matrix

# Example: Solidly Grounded Transformer

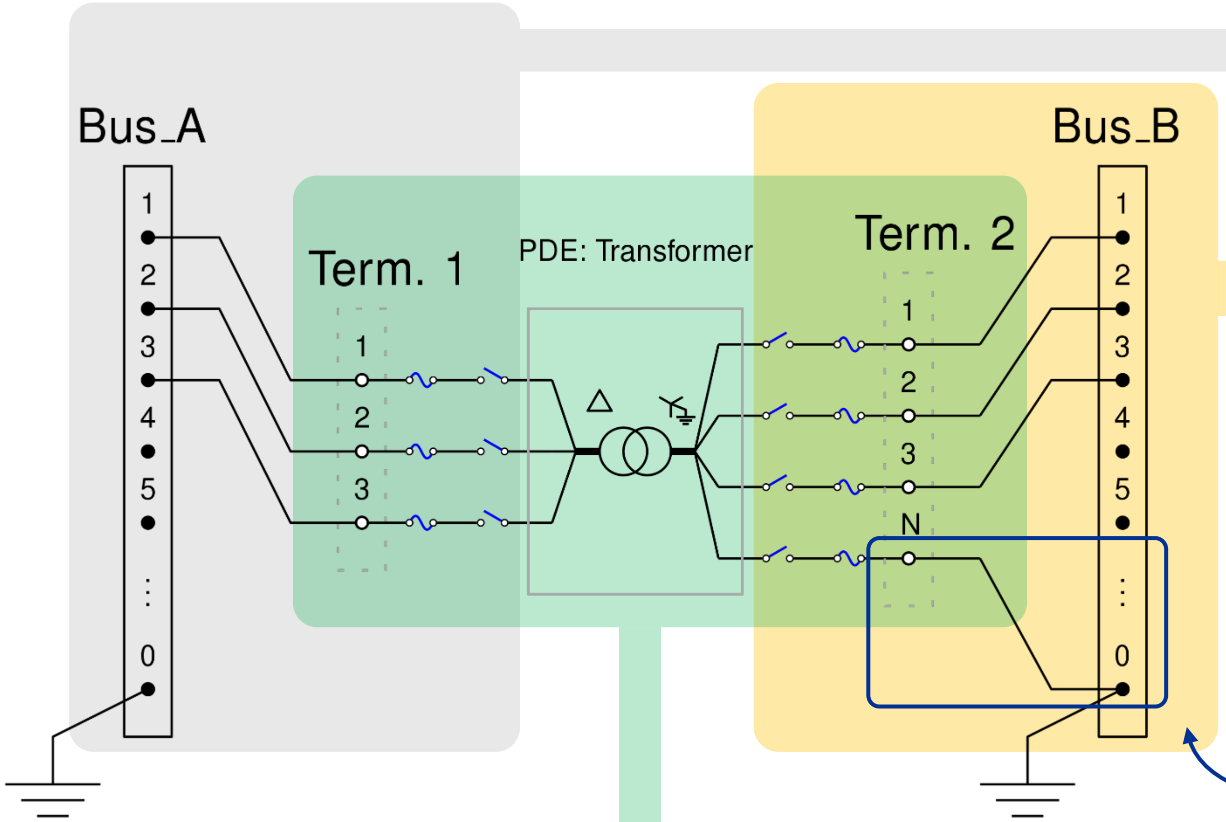
Part of the OpenDSS script that defines this transformer

```
/*
My first DSS transformer
*/

! This is a Dy xfmr
New transformer.xfmr_name
~ phases=3 windings=2
~ Bus1=Bus_A.1.2.3
~ Bus2=Bus_B.1.2.3.0
~ conns=[delta, wye]
...
```



# Example: Solidly Grounded Transformer

Circuit Element instantiation



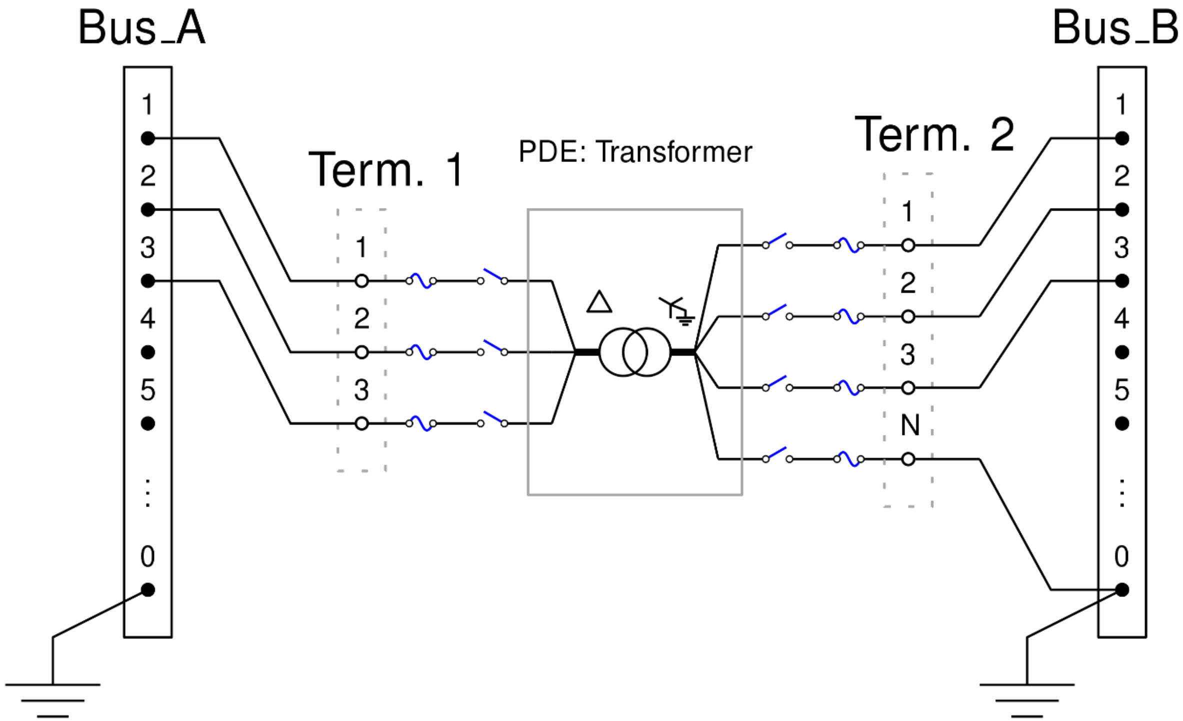
Terminal to bus (conductor to node) assignment

```
/*
My first DSS transformer
*/

! This is a Dy xfmr
New transformer.xfmr_name
~ phases=3 windings=2
~ Bus1=Bus_A.1.2.3
~ Bus2=Bus_B.1.2.3.0
~ conns=[delta, wye]
...
```

# Example: Solidly Grounded Transformer

Use /*…*/, // or ! for line or block comments

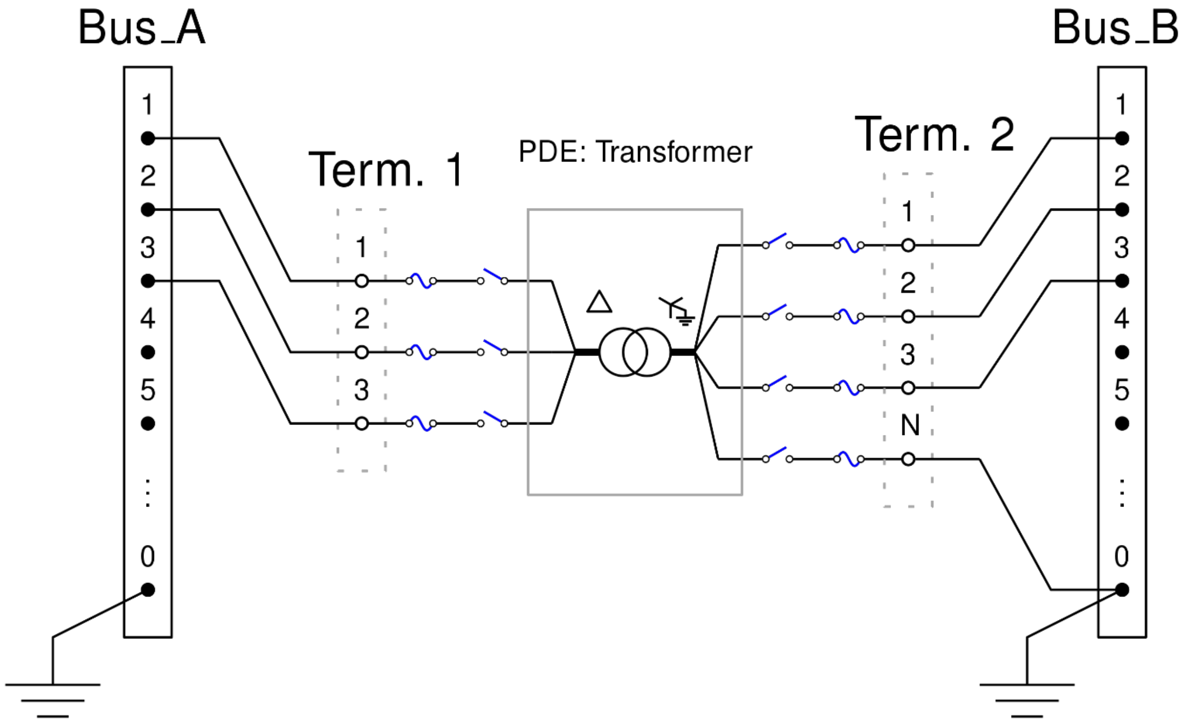
```
/*
My first DSS transformer
*/

! This is a Dy xfmr
New transformer.xfmr_name
~ phases=3 windings=2
~ Bus1=Bus_A.1.2.3
~ Bus2=Bus_B.1.2.3.0
~ conns=[delta, wye]
...
```



# Example: Solidly Grounded Transformer

Command syntax: `command param_1, param_2 param_3`

*Comma, space or tab separated*



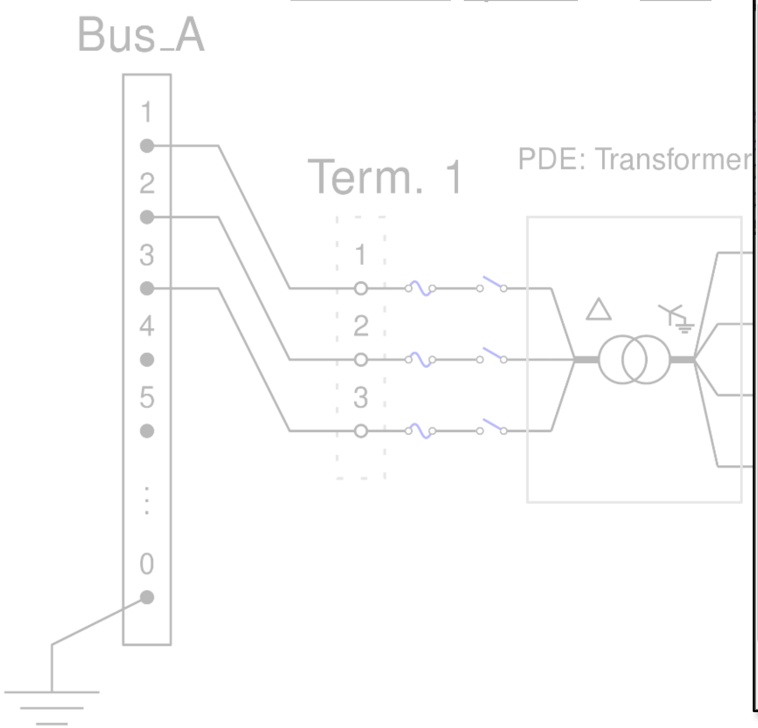
Use = to separate keywords from values

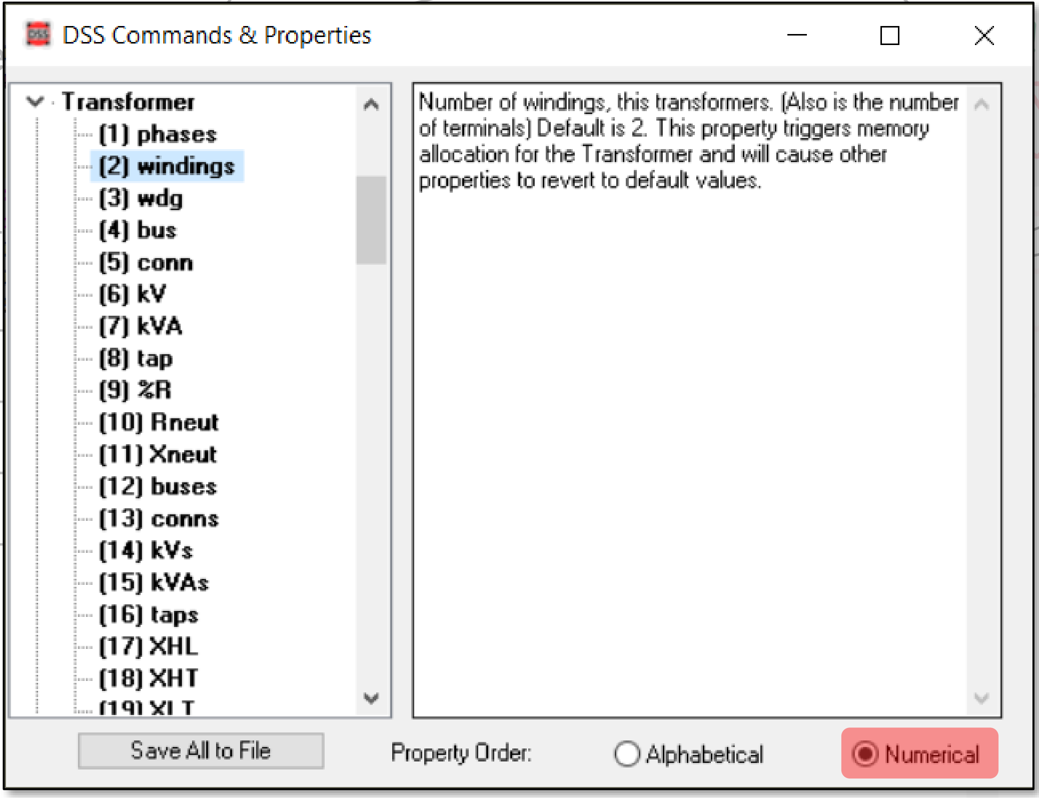
```
/*
My first DSS transformer
*/

! This is a Dy xfmr
New transformer.xfmr_name
~ phases=3 windings=2
~ Bus1=Bus_A.1.2.3
~ Bus2=Bus_B.1.2.3.0
~ conns=[delta, wye]
...
```

# Example: Solidly Grounded Transformer

Command syntax: `command param_1, param_2 param_3`

Comma, space or tab se





- Transformer
  - (1) phases
  - (2) windings
  - (3) wdg
  - (4) bus
  - (5) conn
  - (6) kV
  - (7) kVA
  - (8) tap
  - (9) %R
  - (10) Rneut
  - (11) Xneut
  - (12) buses
  - (13) conns
  - (14) kVs
  - (15) kVAs
  - (16) taps
  - (17) XHL
  - (18) XHT
  - (19) XLT

Number of windings, this transformers. (Also is the number of terminals) Default is 2. This property triggers memory allocation for the Transformer and will cause other properties to revert to default values.

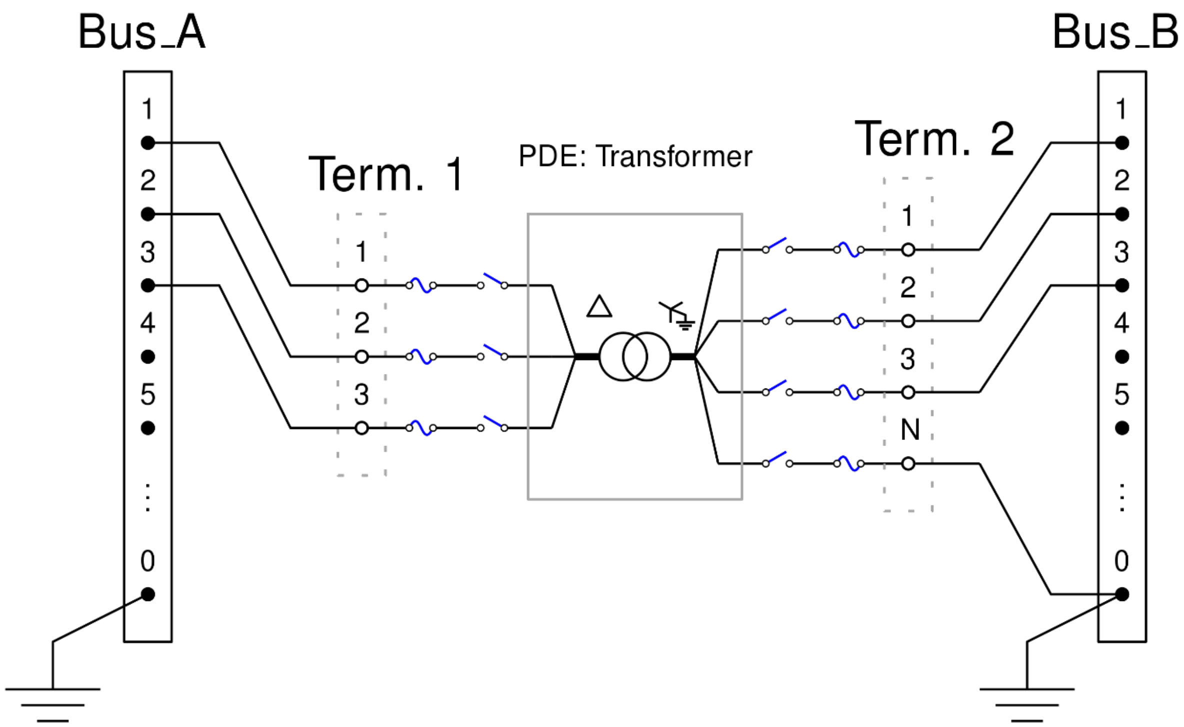
```
/*
My first DSS transformer
*/

  This is a Dy xfmr
New transformer.xfmr_name
~ phases=3 windings=2
~ Bus1=Bus_A.1.2.3
~ Bus2=Bus_B.1.2.3.0
~ conns=[delta, wye]
...
```

The parameter tag and equals sign in `keyword=value` are not needed if values are passed in numerical order. (See numerical order in OpenDSS command help)

# Example: Solidly Grounded Transformer

Command syntax: `command param_1, param_2 param_3`

*Comma, space or tab separated*



Use ~ for line continuation

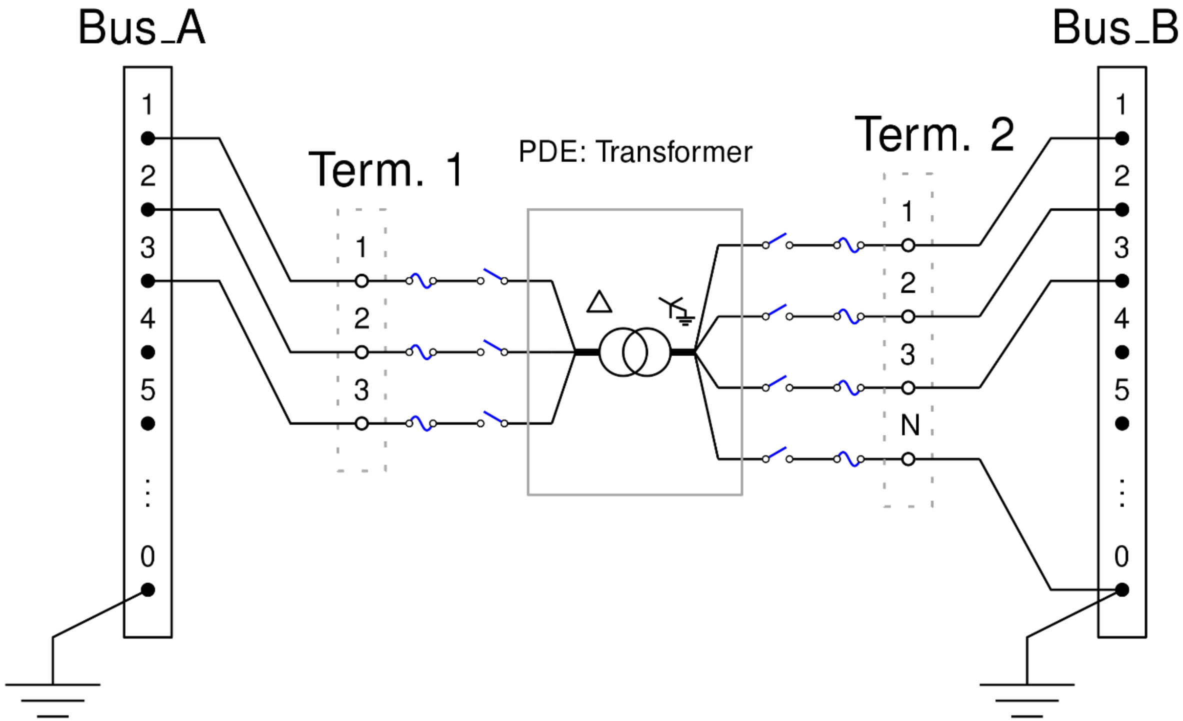
Use . for delimiting class, object, bus, node

```
/*
My first DSS transformer
*/

! This is a Dy xfmr
New transformer.xfmr_name
~ phases=3 windings=2
~ Bus1=Bus_A.1.2.3
~ Bus2=Bus_B.1.2.3.0
~ conns=[delta, wye]
...
```

# Example: Solidly Grounded Transformer

Command syntax: `command param_1, param_2 param_3`

*Comma, space or tab separated*

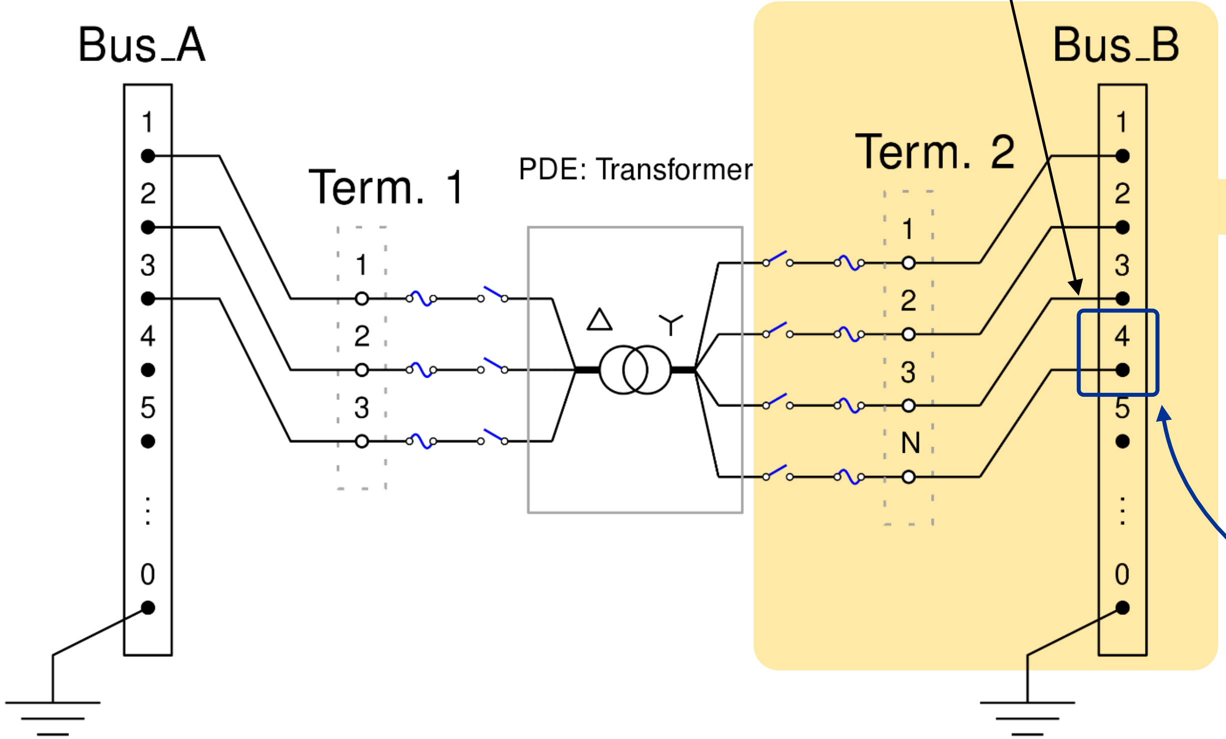


Use [] to create arrays. Comma or space to separate elements

```
/*
My first DSS transformer
*/

! This is a Dy xfmr
New transformer.xfmr_name
~ phases=3 windings=2
~ Bus1=Bus_A.1.2.3
~ Bus2=Bus_B.1.2.3.0
~ conns=[delta, wye]
. . .
```

# Example: Ungrounded Transformer

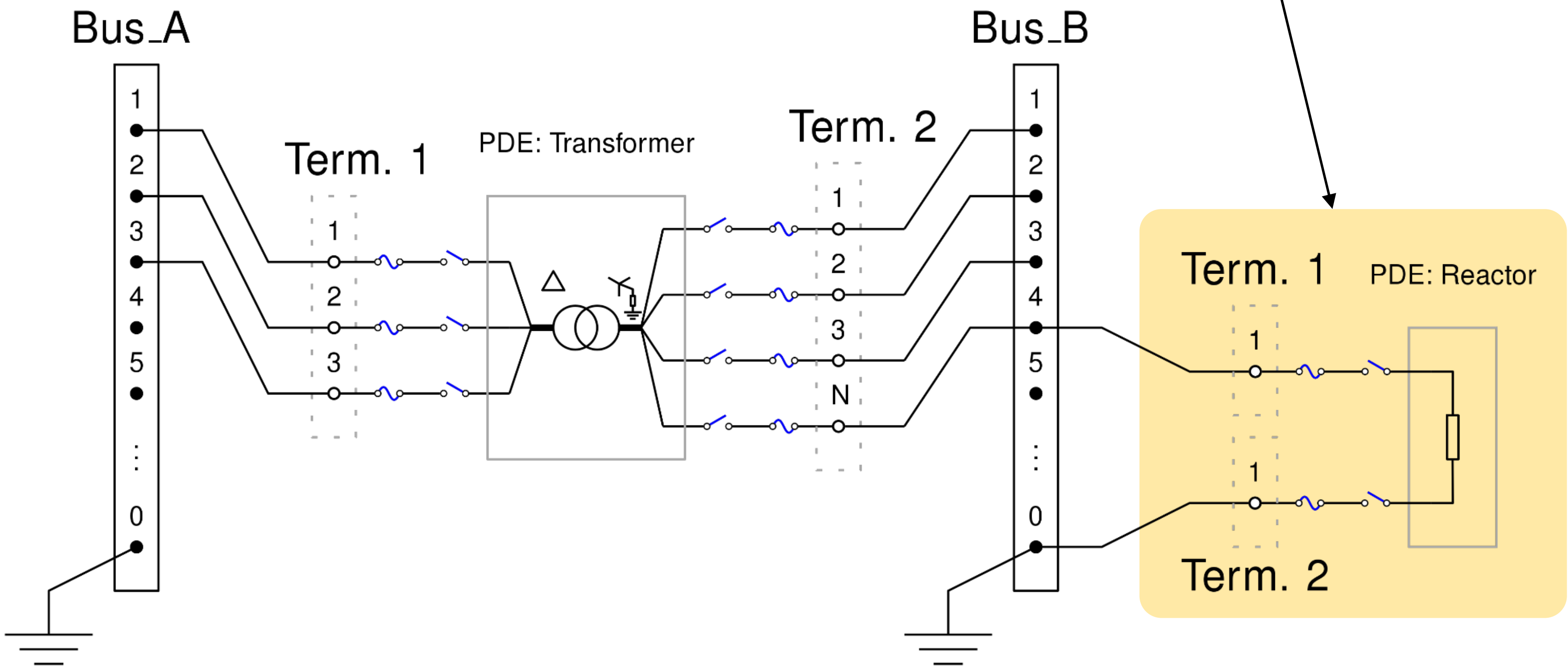
Connect neutral conductor to node 4 of the bus.
Neutral is floating now



```
/*
My first DSS transformer
*/

! This is a Dy xfmr
New transformer.xfmr_name
~ phases=3 windings=2
~ Bus1=Bus_A.1.2.3
~ Bus2=Bus_B.1.2.3.4
~ conns=[delta, wye]
...
```

# Example: Grounding Impedance

Connect <u>both terminals of a reactor</u> (another PDE) to *Bus_B None 4* and *Node 0*. The transformer is grounded again.

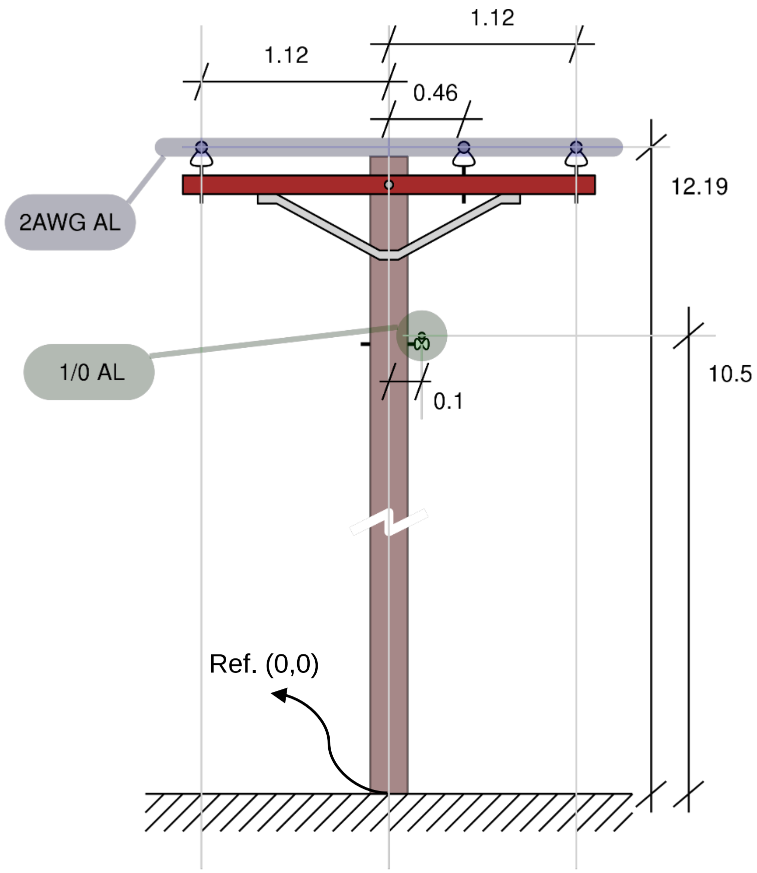


```
/*
My first DSS transformer
*/

! This is a Dy xfmr
New transformer.xfmr_name
~ phases=3 windings=2
~ Bus1=Bus_A.1.2.3
~ Bus2=Bus_B.1.2.3.4
~ conns=[delta, wye]
...
```

```
New reactor.rg_name
~ phases=1
~ Bus1=Bus_B.4
~ Bus2=Bus_B.0
~ R=0.1 X=700.0
...
```

# Example: Distribution Line Segment

If you know the per-length impedance matrices..



Use [ ] to create matrices.

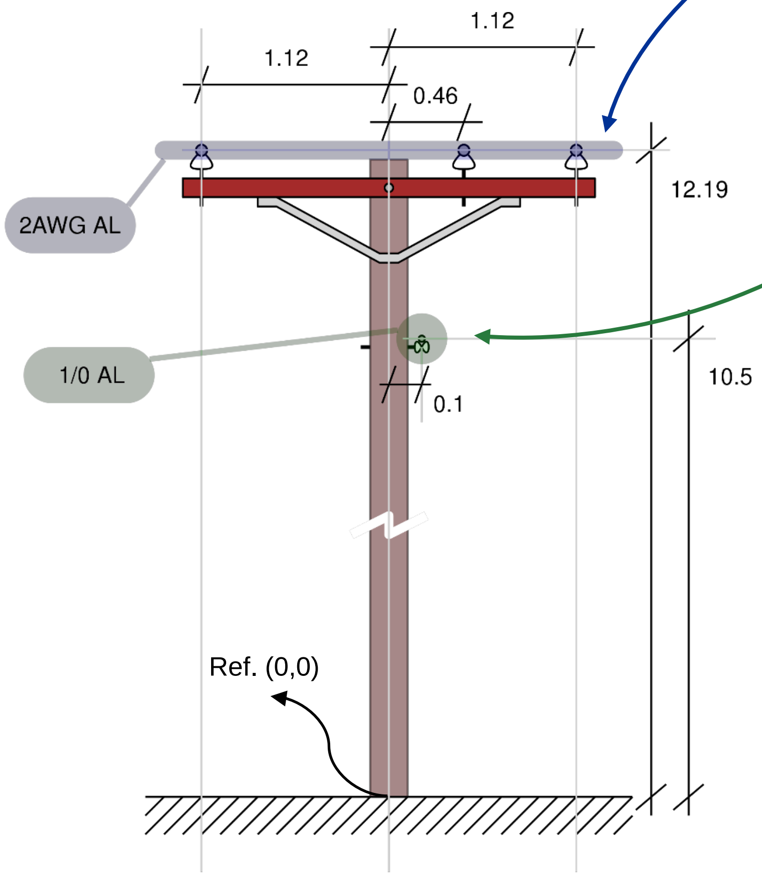
Use | to separate rows.

```
! My first DSS line
New line.L1 bus1=A bus2=B phases=3
~ rmatrix=[0.9699 | 0.1200 0.9753 | 0.1181 0.1208 0.9714]
~ xmatrix=[0.7766 | 0.2905 0.7650 | 0.2682 0.3543 0.7733]
~ cmatrix=[7.4518 | -1.493 8.2716 | -1.029 -2.602 8.0307]
~ length=1 units=km
```

You can define all the matrix entries or just those of the lower triangle

# Example: Distribution Line Segment

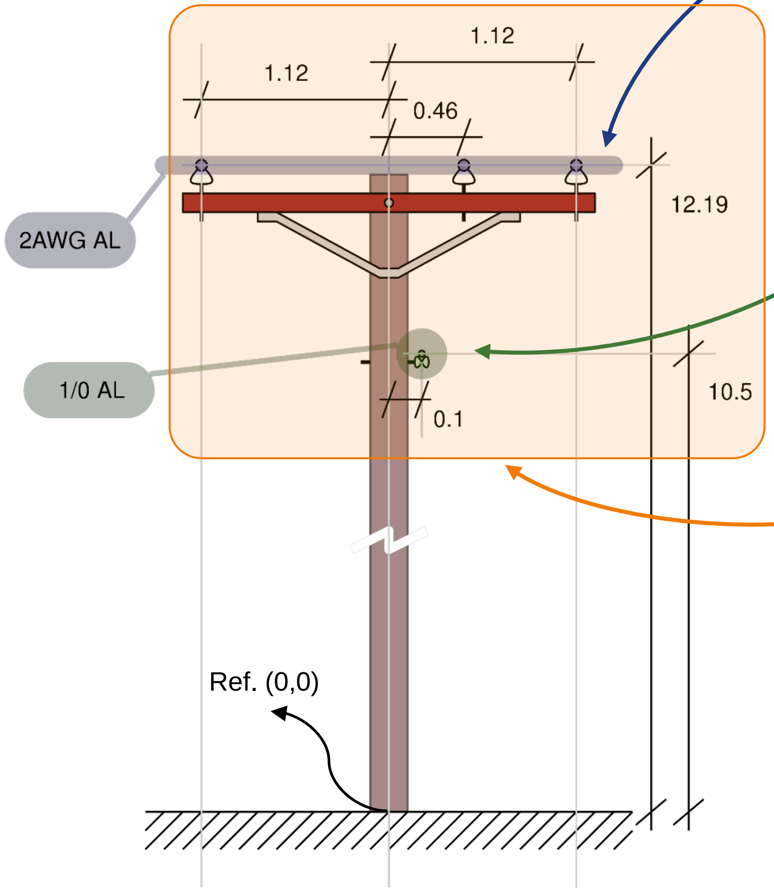
… or if you know the conductor and geometry details...



```
! Phase Conductor
New "wiredata.2AWG AL" Rac=0.86942 Runits=km GMRac=0.26924
~ GMRunits=cm diam=0.74168 radunits=cm

! Neutral Conductor
New wiredata.1/0AL Rac=0.54463 Runits=km GMRac=0.33833
~ GMRunits=cm diam=0.93472 radunits=cm

! Cross arm pole geometry details
New linespacing.4KV_3PH_3CH nphases=3 nconds=4
~ x=[-1.1176 0.4572 1.1176 0.10160]
~ h=[12.1920 12.192 12.192 10.5664] units=m

! My second DSS line
New line.L2 bus1=A bus2=B phases=3 spacing=4KV_3PH_3CH
~ wires=["2AWG AL" "2AWG AL" "2AWG AL" 1/0AL] length=1
~ units=km
```

# Example: Distribution Line Segment

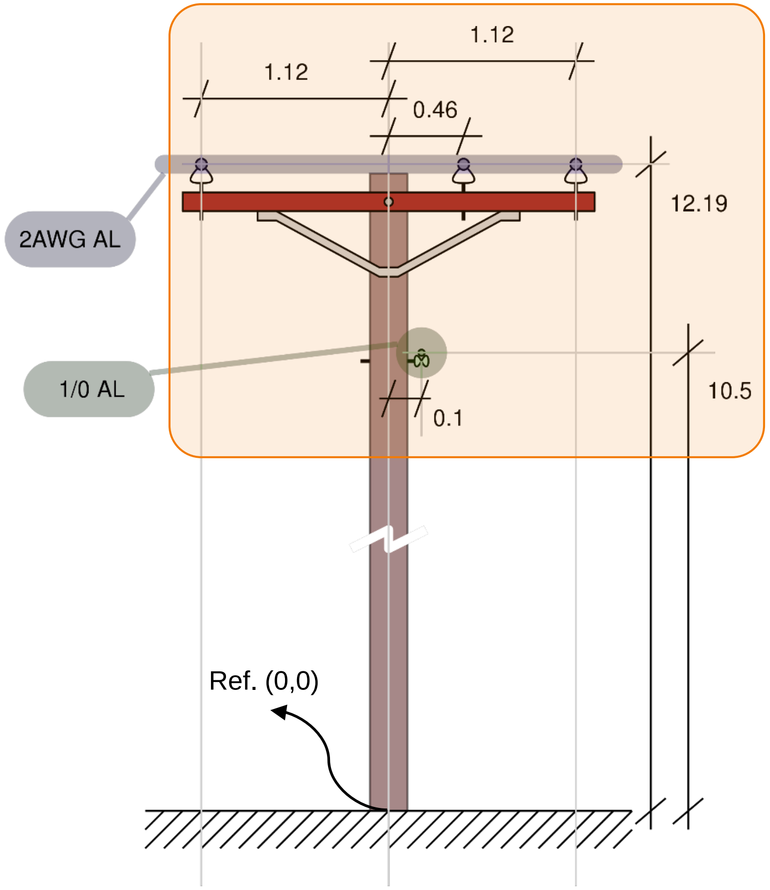
… or if you know the conductor and geometry details...



```
! Phase Conductor
New "wiredata.2AWG AL" Rac=0.86942 Runits=km GMRac=0.26924
~ GMRunits=cm diam=0.74168 radunits=cm

! Neutral Conductor
New wiredata.1/0AL Rac=0.54463 Runits=km GMRac=0.33833
~ GMRunits=cm diam=0.93472 radunits=cm

! Cross arm pole geometry details
New linespacing.4KV_3PH_3CH nphases=3 nconds=4
~ x=[-1.1176 0.4572 1.1176 0.10160]
~ h=[12.1920 12.192 12.192 10.5664] units=m

! My second DSS line
New line.L2 bus1=A bus2=B phases=3 spacing=4KV_3PH_3CH
~ wires=["2AWG AL" "2AWG AL" "2AWG AL" 1/0AL] length=1
~ units=km
```

# Example: Distribution Line Segment

… or if you know the conductor and geometry details...

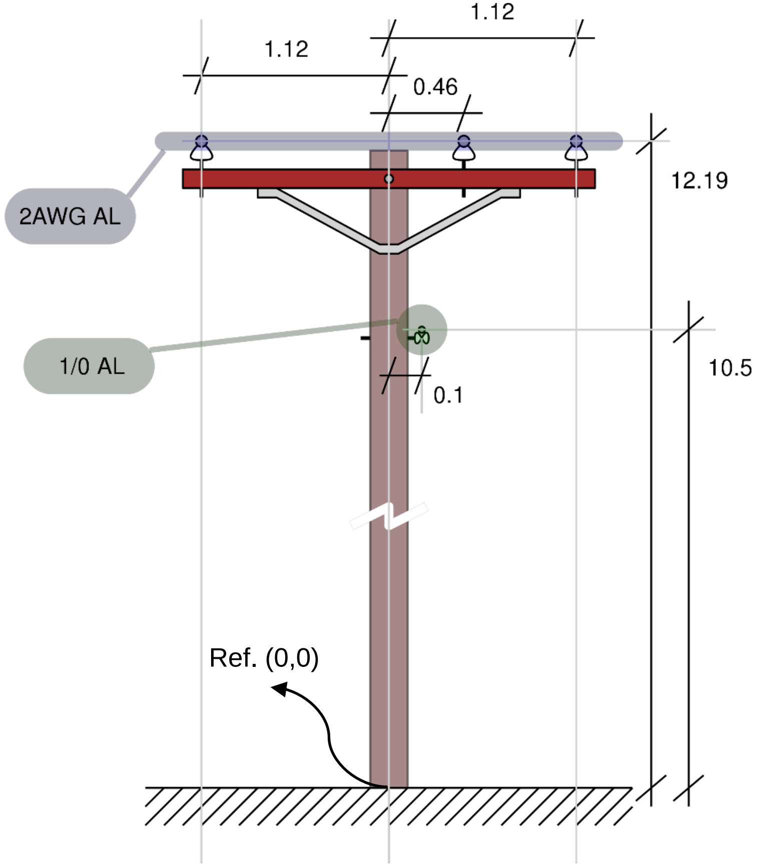


```
! Phase Conductor
New "wiredata.2AWG AL" Rac=0.86942 Runits=km GMRac=0.26924
~ GMRunits=cm diam=0.74168 radunits=cm

! Neutral Conductor
New wiredata.1/0AL Rac=0.54463 Runits=km GMRac=0.33833
~ GMRunits=cm diam=0.93472 radunits=cm

! Cross arm pole geometry details
New linespacing.4KV_3PH_3CH nphases=3 nconds=4
~ x=[-1.1176 0.4572 1.1176 0.10160]
~ h=[12.1920 12.192 12.192 10.5664] units=m

! My second DSS line
New line.L2 bus1=A bus2=B phases=3 spacing=4KV_3PH_3CH
~ wires=["2AWG AL" "2AWG AL" "2AWG AL" 1/0AL] length=1
~ units=km
```

# Example: Distribution Line Segment

Use "" if you want to have space characters in strings or names of DSS objects

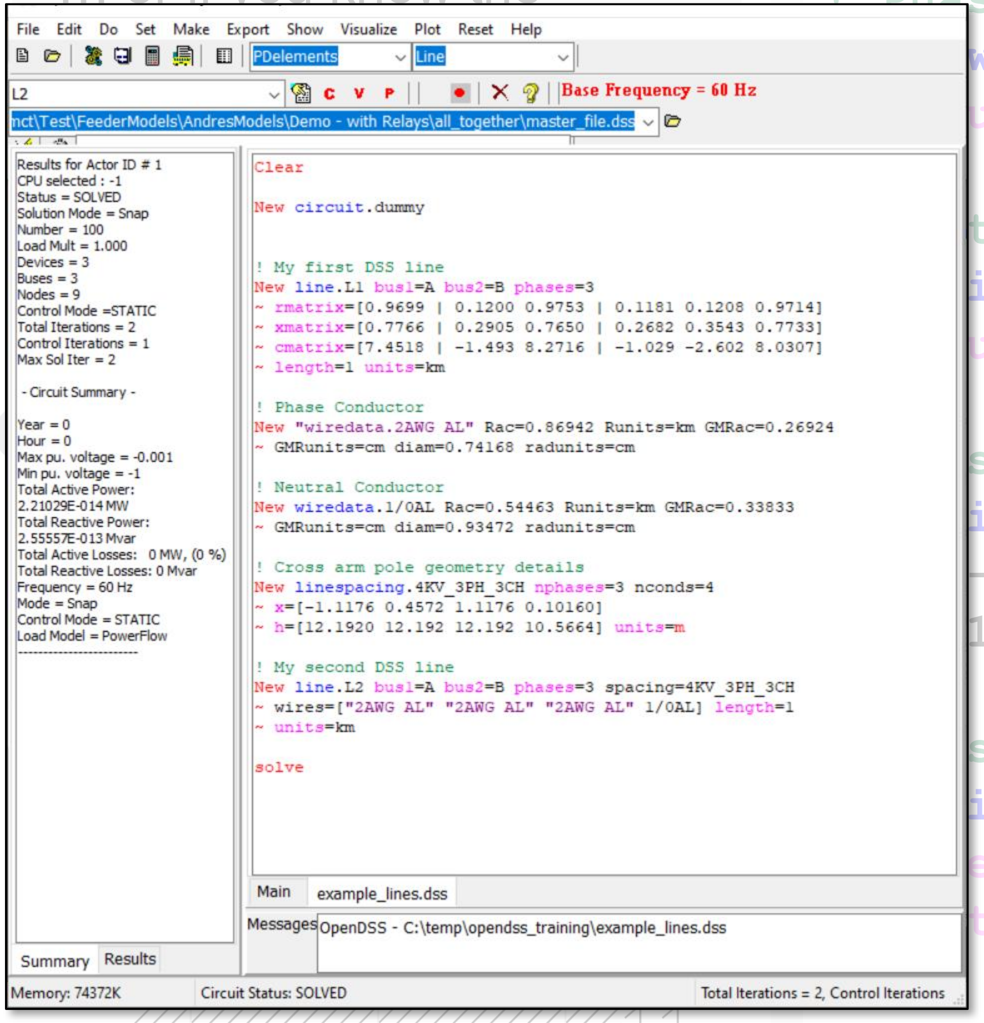
… or if you know the conductor and geometry details...



```
! Phase Conductor
New "wiredata.2AWG AL" Rac=0.86942 Runits=km GMRac=0.26924
~ GMRunits=cm diam=0.74168 radunits=cm

! Neutral Conductor
New wiredata.1/0AL Rac=0.54463 Runits=km GMRac=0.33833
~ GMRunits=cm diam=0.93472 radunits=cm

! Cross arm pole geometry details
New linespacing.4KV_3PH_3CH nphases=3 nconds=4
~ x=[-1.1176 0.4572 1.1176 0.10160]
~ h=[12.1920 12.192 12.192 10.5664] units=m

! My second DSS line
New line.L2 bus1=A bus2=B phases=3 spacing=4KV_3PH_3CH
~ wires=["2AWG AL" "2AWG AL" "2AWG AL" 1/0AL] length=1
~ units=km
```

# Example: Distribution Line Segment

... or if you know the

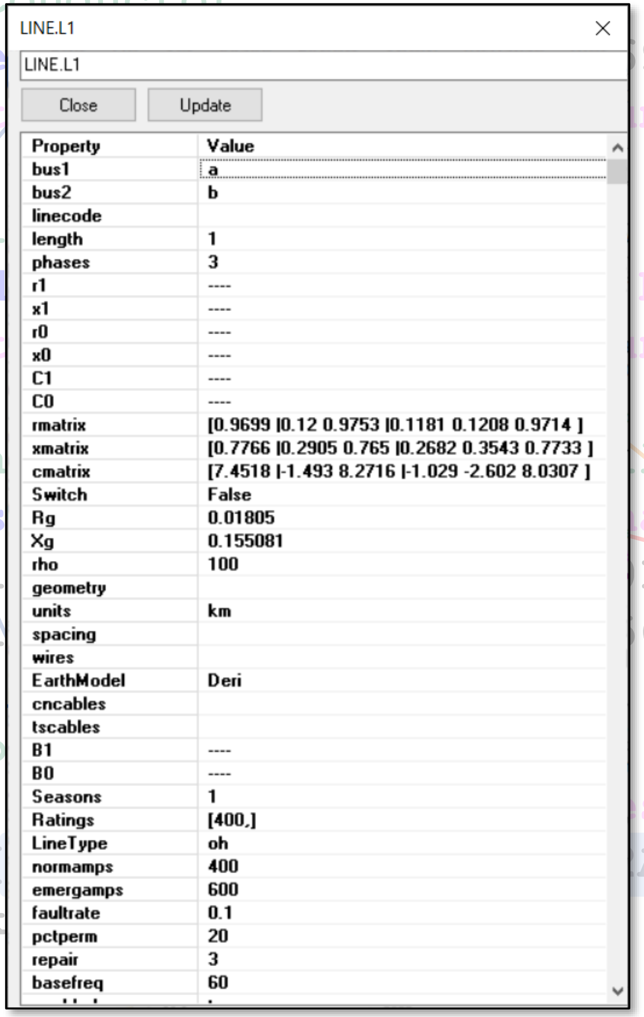


```
Results for Actor ID # 1
CPU selected : -1
Status = SOLVED
Solution Mode = Snap
Number = 100
Load Mult = 1.000
Devices = 3
Buses = 3
Nodes = 9
Control Mode =STATIC
Total Iterations = 2
Control Iterations = 1
Max Sol Iter = 2

 - Circuit Summary -

Year = 0
Hour = 0
Max pu. voltage = -0.001
Min pu. voltage = -1
Total Active Power:
2.21029E-014 MW
Total Reactive Power:
2.55557E-013 Mvar
Total Active Losses:   0 MW, (0 %)
Total Reactive Losses: 0 Mvar
Frequency = 60 Hz
Mode = Snap
Control Mode = STATIC
Load Model = PowerFlow
------------------------
```

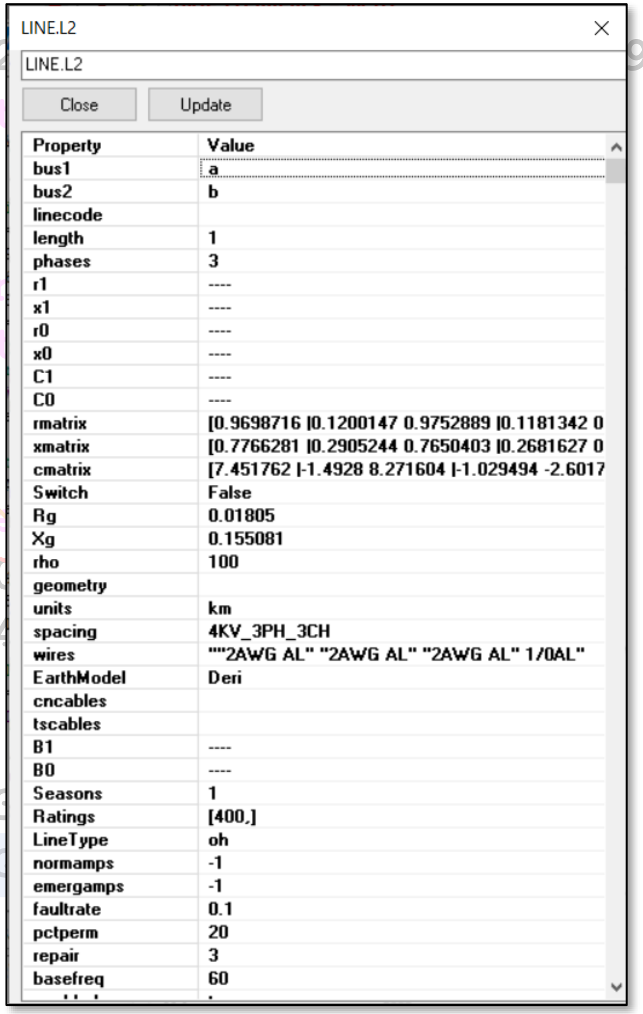
```
Clear

New circuit.dummy


! My first DSS line
New line.L1 bus1=A bus2=B phases=3
~ rmatrix=[0.9699 | 0.1200 0.9753 | 0.1181 0.1208 0.9714]
~ xmatrix=[0.7766 | 0.2905 0.7650 | 0.2682 0.3543 0.7733]
~ cmatrix=[7.4518 | -1.493 8.2716 | -1.029 -2.602 8.0307]
~ length=1 units=km

! Phase Conductor
New "wiredata.2AWG AL" Rac=0.86942 Runits=km GMRac=0.26924
~ GMRunits=cm diam=0.74168 radunits=cm

! Neutral Conductor
New wiredata.1/0AL Rac=0.54463 Runits=km GMRac=0.33833
~ GMRunits=cm diam=0.93472 radunits=cm

! Cross arm pole geometry details
New linespacing.4KV_3PH_3CH nphases=3 nconds=4
~ x=[-1.1176 0.4572 1.1176 0.10160]
~ h=[12.1920 12.192 12.192 10.5664] units=m

! My second DSS line
New line.L2 bus1=A bus2=B phases=3 spacing=4KV_3PH_3CH
~ wires=["2AWG AL" "2AWG AL" "2AWG AL" 1/0AL] length=1
~ units=km

solve
```

```
OpenDSS - C:\temp\opendss_training\example_lines.dss
```

```
Memory: 74372K     Circuit Status: SOLVED     Total Iterations = 2, Control Iterations
```



| Property | Value |
|---|---|
| bus1 | a |
| bus2 | b |
| linecode | |
| length | 1 |
| phases | 3 |
| r1 | ---- |
| x1 | ---- |
| r0 | ---- |
| x0 | ---- |
| C1 | ---- |
| C0 | ---- |
| rmatrix | [0.9699 \|0.12 0.9753 \|0.1181 0.1208 0.9714 ] |
| xmatrix | [0.7766 \|0.2905 0.765 \|0.2682 0.3543 0.7733 ] |
| cmatrix | [7.4518 \|-1.493 8.2716 \|-1.029 -2.602 8.0307 ] |
| Switch | False |
| Rg | 0.01805 |
| Xg | 0.155081 |
| rho | 100 |
| geometry | |
| units | km |
| spacing | |
| wires | |
| EarthModel | Deri |
| cncables | |
| tscables | |
| B1 | ---- |
| B0 | ---- |
| Seasons | 1 |
| Ratings | [400,] |
| LineType | oh |
| normamps | 400 |
| emergamps | 600 |
| faultrate | 0.1 |
| pctperm | 20 |
| repair | 3 |
| basefreq | 60 |



| Property | Value |
|---|---|
| bus1 | a |
| bus2 | b |
| linecode | |
| length | 1 |
| phases | 3 |
| r1 | ---- |
| x1 | ---- |
| r0 | ---- |
| x0 | ---- |
| C1 | ---- |
| C0 | ---- |
| rmatrix | [0.9698716 \|0.1200147 0.9752889 \|0.1181342 0 |
| xmatrix | [0.7766281 \|0.2905244 0.7650403 \|0.2681627 0 |
| cmatrix | [7.451762 \|-1.4928 8.271604 \|-1.029494 -2.6017 |
| Switch | False |
| Rg | 0.01805 |
| Xg | 0.155081 |
| rho | 100 |
| geometry | |
| units | km |
| spacing | 4KV_3PH_3CH |
| wires | ""2AWG AL" "2AWG AL" "2AWG AL" 1/0AL" |
| EarthModel | Deri |
| cncables | |
| tscables | |
| B1 | ---- |
| B0 | ---- |
| Seasons | 1 |
| Ratings | [400,] |
| LineType | oh |
| normamps | -1 |
| emergamps | -1 |
| faultrate | 0.1 |
| pctperm | 20 |
| repair | 3 |
| basefreq | 60 |

# Recap: Commands and Scripting Characters

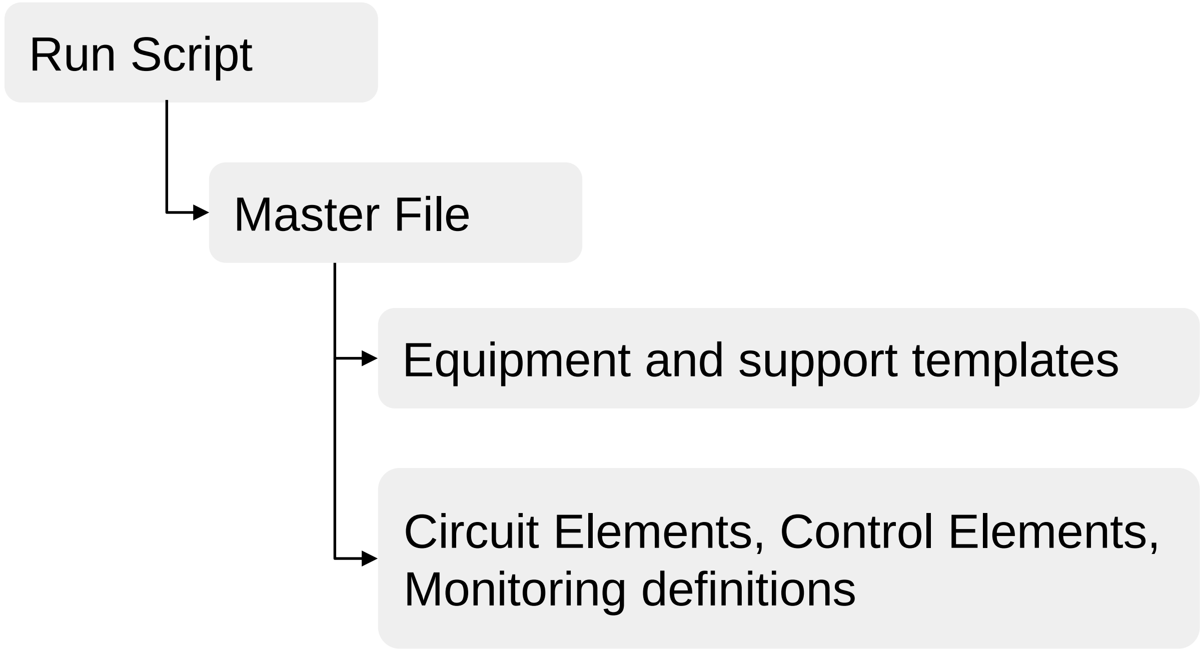
Scripting

| Character | Description |
|---|---|
| ~ | Line continuation |
| [] () {} “” ‘’ | Arrays, Matrices, Strings, Math Expressions |
| \| | Row delimiter in matrices |
| , space tab | Delimiter command parameters and array/matrix entries |
| . | Delimiter class-object, bus-node |
| = | Delimiter parameter tag and parameter value |
| ! // | In line comments |
| /*…*/ | Block comments |

Commands (common)

| Command | Description |
|---|---|
| New | Create new DSS elements |
| Edit | Edit existing DSS elements |
| Set | Define solution modes and options |
| Solve | Perform current solution mode |
| Show | Write selected results to text files |
| Export | Export solution variables in CSV or XML format |
| Plot | Plots results with built-in plotting features |

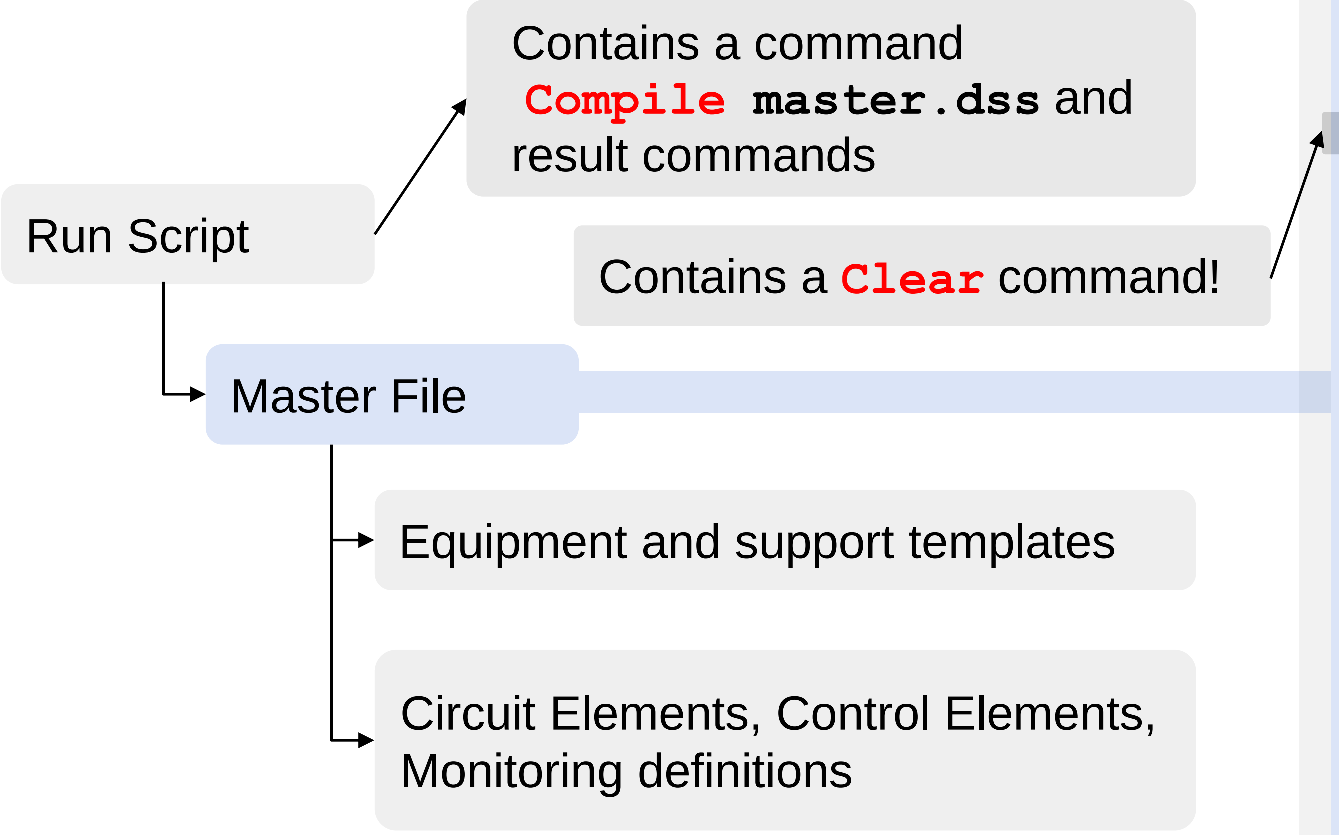
# Large Circuits – Script structure


Run Script
Master File
Equipment and support templates
Circuit Elements, Control Elements, Monitoring definitions

```
// Master file for 8500-Node IEEE Test Feeder Case
// Unbalanced Load Case

Clear
Set DefaultBaseFrequency=60

! Definition of main source (mandatory!)
New Circuit.IEEE8500
~ pu=1.05  r1=0  x1=0.001  r0=0  x0=0.001

! Templates
Redirect  LineCodes2.dss
Redirect  Triplex_Linecodes.dss
Redirect  LoadXfmrCodes.dss

! Circuit Elements, Controls, Monitors
Redirect  Lines.dss
Redirect  Transformers.dss
Redirect  Triplex_Lines.dss
Redirect  UnbalancedLoads.dss
Redirect  Capacitors.dss
Redirect  CapControls.dss
Redirect  Regulators.dss

! Let DSS estimate the voltage bases
Set voltagebases=[115, 12.47,  0.48, 0.208]
Calcvoltagebases     ! This also establishes the bus list

! Load in bus coordinates now that bus list is established
Buscoords  Buscoords.dss
```

This structure is not required but it helps to explore and organize your model. A single script can host an entire model.

# Large Circuits – Script structure



Run Script

Contains a command `Compile master.dss` and result commands

Contains a `Clear` command!

Master File

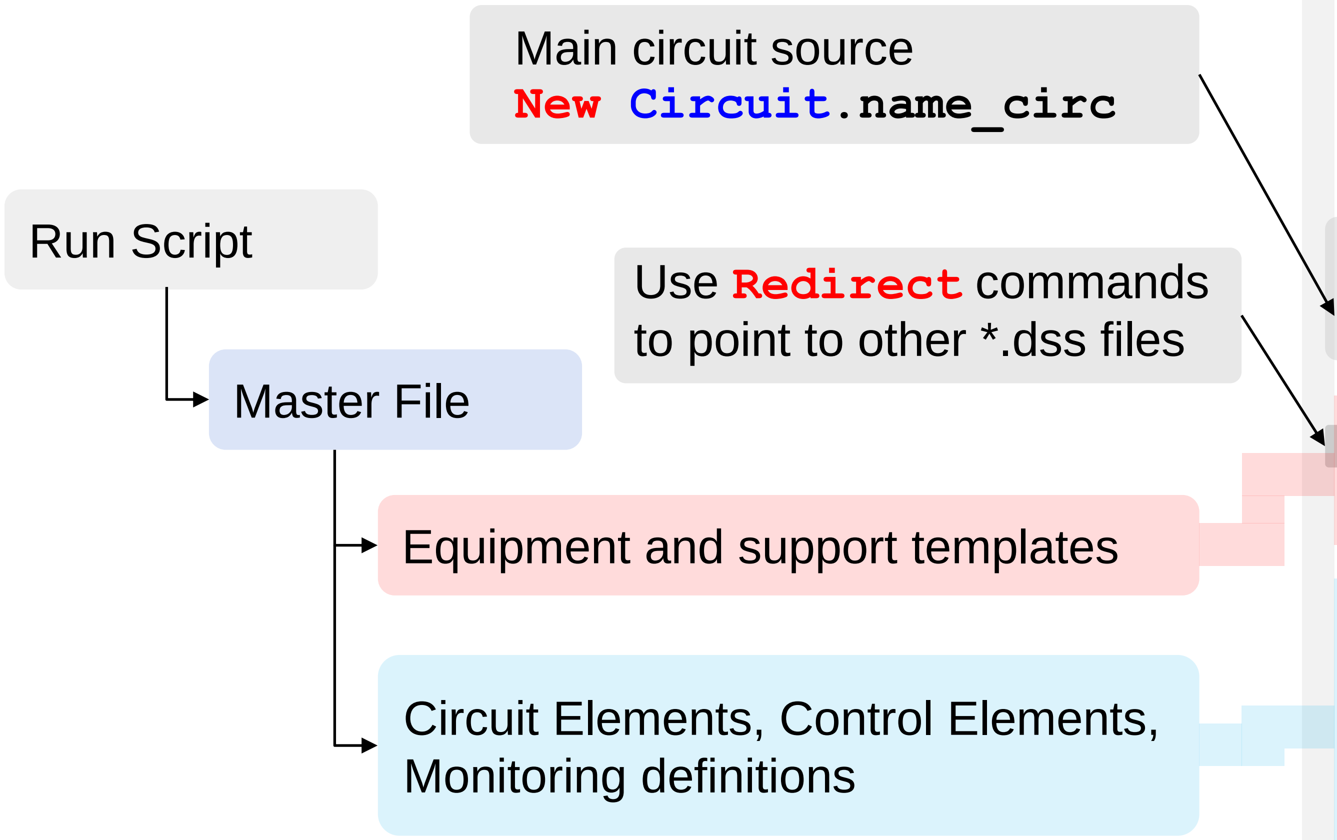
Equipment and support templates

Circuit Elements, Control Elements, Monitoring definitions

```
// Master file for 8500-Node IEEE Test Feeder Case
// Unbalanced Load Case

Clear
Set DefaultBaseFrequency=60

! Definition of main source (mandatory!)
New Circuit.IEEE8500
~ pu=1.05  r1=0  x1=0.001  r0=0  x0=0.001

! Templates
Redirect  LineCodes2.dss
Redirect  Triplex_Linecodes.dss
Redirect  LoadXfmrCodes.dss

! Circuit Elements, Controls, Monitors
Redirect  Lines.dss
Redirect  Transformers.dss
Redirect  Triplex_Lines.dss
Redirect  UnbalancedLoads.dss
Redirect  Capacitors.dss
Redirect  CapControls.dss
Redirect  Regulators.dss

! Let DSS estimate the voltage bases
Set voltagebases=[115, 12.47,  0.48, 0.208]
Calcvoltagebases     ! This also establishes the bus list

! Load in bus coordinates now that bus list is established
Buscoords  Buscoords.dss
```

This structure is not required but it helps to explore and organize your model. A single script can host an entire model.

# Large Circuits – Script structure



Main circuit source

New Circuit.name_circ

Run Script

Master File

Use Redirect commands to point to other *.dss files

Equipment and support templates

Circuit Elements, Control Elements, Monitoring definitions

```
// Master file for 8500-Node IEEE Test Feeder Case
// Unbalanced Load Case

Clear
Set DefaultBaseFrequency=60

! Definition of main source (mandatory!)
New Circuit.IEEE8500
~ pu=1.05  r1=0  x1=0.001  r0=0  x0=0.001

! Templates
Redirect  LineCodes2.dss
Redirect  Triplex_Linecodes.dss
Redirect  LoadXfmrCodes.dss

! Circuit Elements, Controls, Monitors
Redirect  Lines.dss
Redirect  Transformers.dss
Redirect  Triplex_Lines.dss
Redirect  UnbalancedLoads.dss
Redirect  Capacitors.dss
Redirect  CapControls.dss
Redirect  Regulators.dss

! Let DSS estimate the voltage bases
Set voltagebases=[115, 12.47,  0.48, 0.208]
Calcvoltagebases     ! This also establishes the bus list

! Load in bus coordinates now that bus list is established
Buscoords  Buscoords.dss
```

This structure is not required but it helps to explore and organize your model. A single script can host an entire model.

# Large Circuits – Script structure

Main circuit source

```
New Linecode.1ph-x4_acsrx4_acsr ...
New Linecode.1ph-xx4_acsr4_acsr ...
New Linecode.1ph-x2_acsrx2_acsr ...
New Linecode.1ph-x4_acsrx4_wpal ...
New Linecode.3ph_h-4_acsr4_acsr4_acsr4_acsr ...
...
```

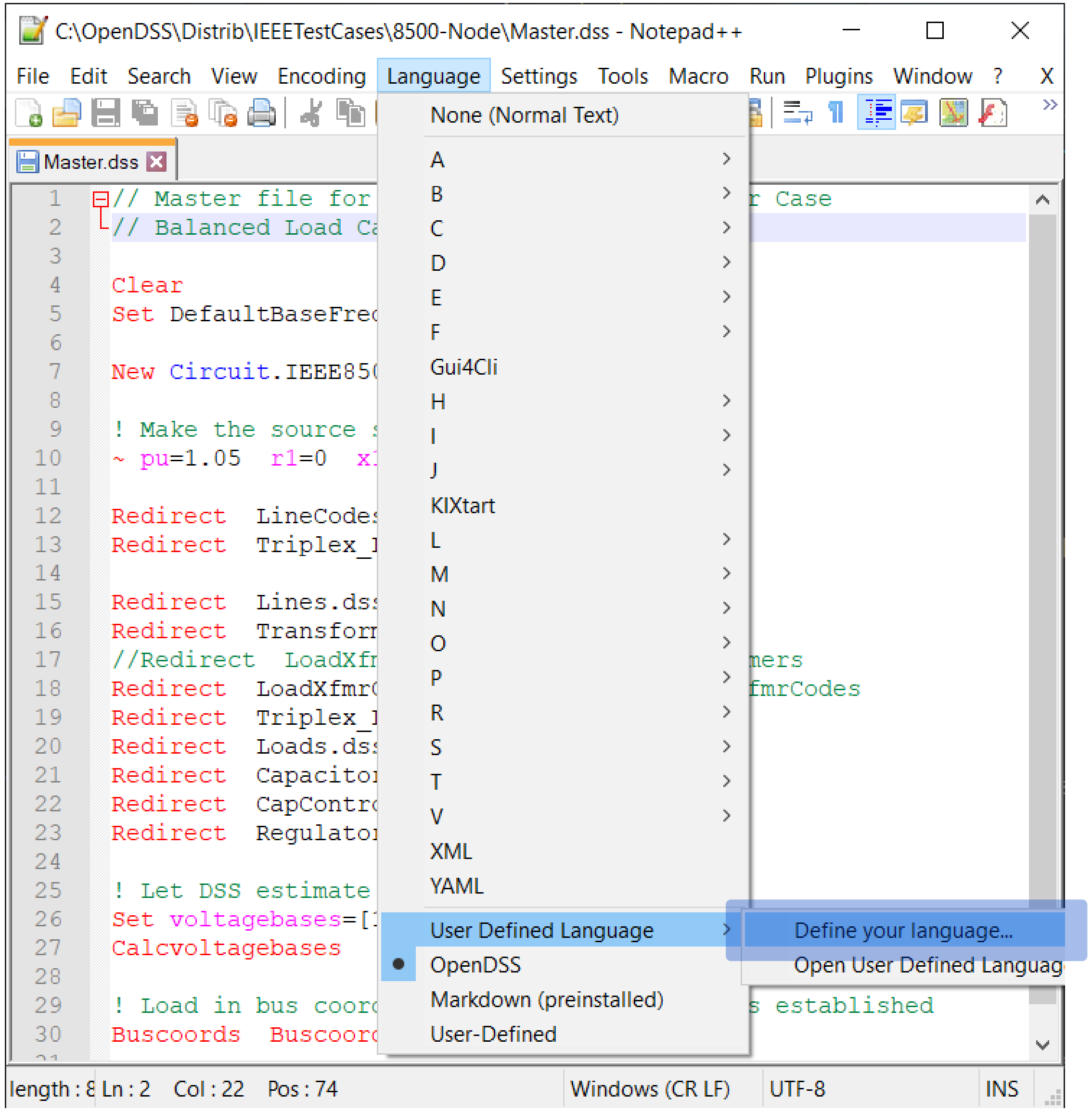
LineCodes2.dss

```
New Line.HVMV_Sub_connector ...
New Line.LN5502549-1 ...
New Line.LN6259988-1 ...
New Line.LN6077796-1 ...
New Line.LN5835135-2 ...
New Line.LN5896826-1 ...
New Line.LN5714038-1 ...
...
```

Lines.dss

```
// Master file for 8500-Node IEEE Test Feeder Case
// Unbalanced Load Case

Clear
Set DefaultBaseFrequency=60

! Definition of main source (mandatory!)
New Circuit.IEEE8500
~ pu=1.05  r1=0  x1=0.001  r0=0  x0=0.001

! Templates
Redirect  LineCodes2.dss
Redirect  Triplex_Linecodes.dss
Redirect  LoadXfmrCodes.dss

! Circuit Elements, Controls, Monitors
Redirect  Lines.dss
Redirect  Transformers.dss
Redirect  Triplex_Lines.dss
Redirect  UnbalancedLoads.dss
Redirect  Capacitors.dss
Redirect  CapControls.dss
Redirect  Regulators.dss

! Let DSS estimate the voltage bases
Set voltagebases=[115, 12.47,  0.48, 0.208]
Calcvoltagebases     ! This also establishes the bus list

! Load in bus coordinates now that bus list is established
Buscoords  Buscoords.dss
```

# DSS Scripts in Text editor



```
// Master file for                                r Case
// Balanced Load Ca

Clear
Set DefaultBaseFre

New Circuit.IEEE85

! Make the source s
~ pu=1.05  r1=0  x1

Redirect  LineCode
Redirect  Triplex_

Redirect  Lines.ds
Redirect  Transfor
//Redirect  LoadXfr                               mers
Redirect  LoadXfmr                                fmrCodes
Redirect  Triplex_
Redirect  Loads.ds
Redirect  Capacito
Redirect  CapContr
Redirect  Regulato

! Let DSS estimate
Set voltagebases=[
Calcvoltagebases

! Load in bus coor                                s established
Buscoords  Buscoor
```

Use a text editor like **Notepad++** or **EditPlus** to better visualize your OpenDSS scripts

Import the syntax file from this folder OpenDSS/Examples/SyntaxFiles

A restart of the text editor might be required

# Demo

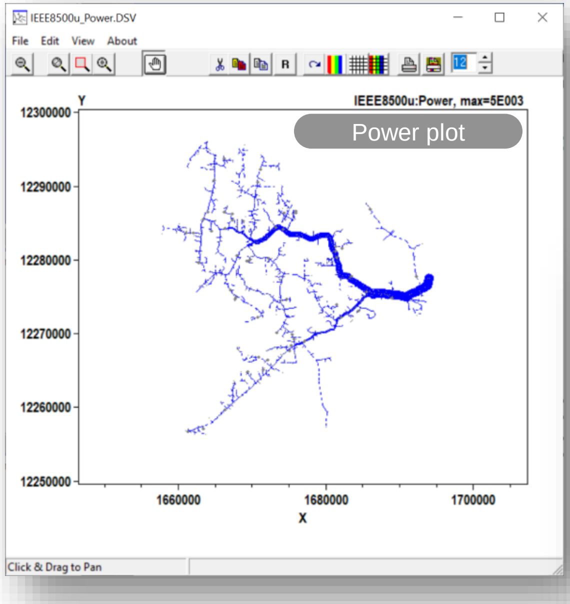
# 8500-node case – Some useful plotting commands

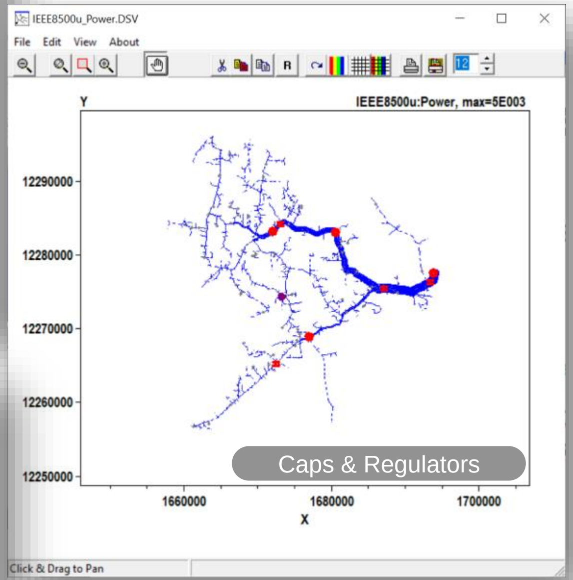
```
! Highlight a bus in the circuit plot

AddBusMarker Bus=12973156 code=15 color=Purple
~ size=4
```

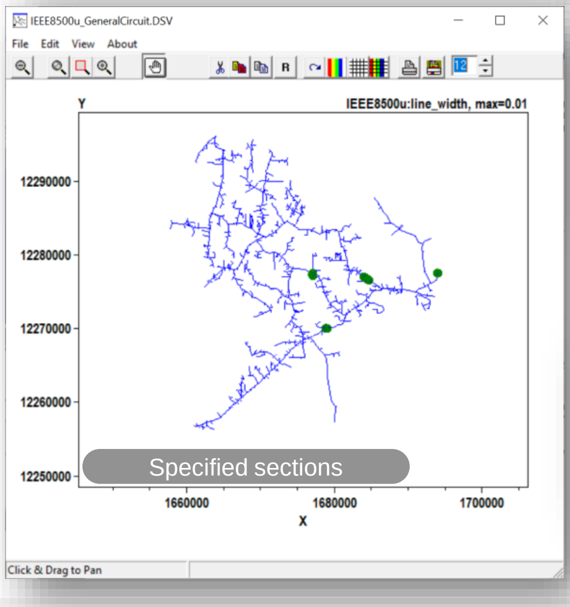
```
! Highlight circuit elements

Set markCapacitors=yes  CapMarkersize=3
Set markRegulators=yes  RegMarkersize=5

Interpolate
```

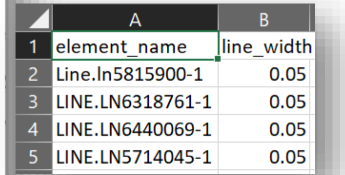
```
! Plot circuit. Line width based on power

Plot Circuit Power Max=5000 dots=n labels=n
~ C1=Blue  1ph=3 ! $00FF0000
```

```
/* Plot circuit. Highlight circuit sections of
interest (quantity refers to the column (0 is
column 1) of the CSV defining the line width) */

Plot Circuit quantity=1 max=0.01
~ object=lines_to_highlight.csv
```





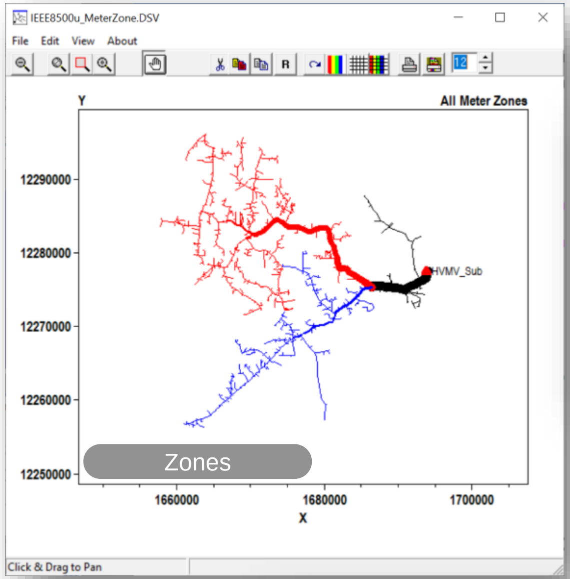


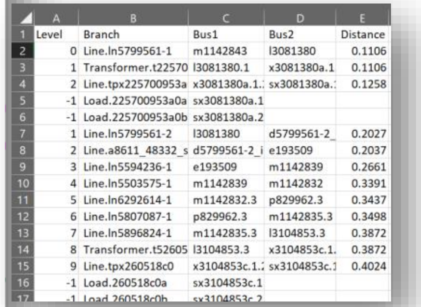


|   | A | B |
|---|---|---|
| 1 | element_name | line_width |
| 2 | Line.ln5815900-1 | 0.05 |
| 3 | LINE.LN6318761-1 | 0.05 |
| 4 | LINE.LN6440069-1 | 0.05 |
| 5 | LINE.LN5714045-1 | 0.05 |

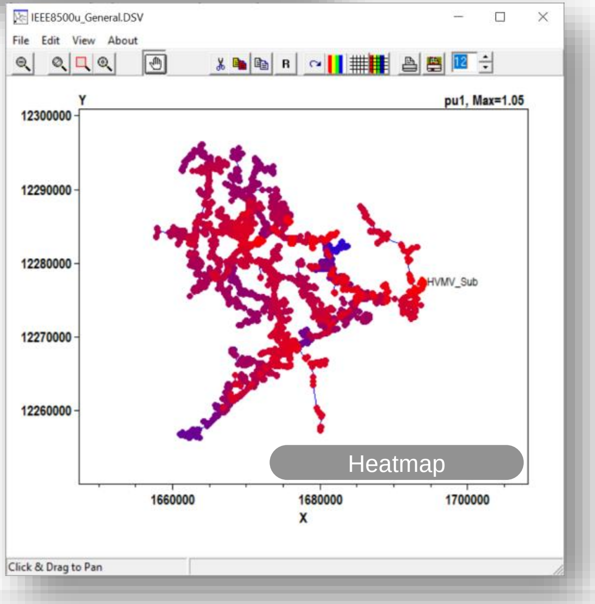
# 8500-node case – Some useful plotting commands

```
! Additional energy meters to split the feeders

New Energymeter.m2 element=LINE.LN6381853-1 terminal=1
New Energymeter.m3 element=LINE.LN5799561-1 terminal=1


/* Command to export the PD elements in the zone of
energy meter m3 to a CSV file */
Edit energymeter.m3 action=zonedump
solve

! Plot energy meter zones
plot zones Power max=5000 labels=n subs=y C1=$00FF0000


/* Plot heatmap of voltage p.u. quantities (quantity
refers to the column (0 is column 1) of the CSV
defining the marker size) */

Set markercode=24  nodewidth=0.005

Export voltages

plot General quantity=5  Max=0 dots=y labels=n subs=y
~ object=IEEE8500u_EXP_VOLTAGES.CSV C1=blue
~ C2=$000000FF ! C1=$0080FFFF C2=$000000FF
```





| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Level | Branch | Bus1 | Bus2 | Distance |
| 2 | 0 | Line.ln5799561-1 | m1142843 | l3081380 | 0.1106 |
| 3 | 1 | Transformer.t22570 | l3081380.1 | x3081380a.1 | 0.1106 |
| 4 | 2 | Line.tpx225700953a | x3081380a.1.: | sx3081380a.: | 0.1258 |
| 5 | -1 | Load.225700953a0a | sx3081380a.1 | | |
| 6 | -1 | Load.225700953a0b | sx3081380a.2 | | |
| 7 | 1 | Line.ln5799561-2 | l3081380 | d5799561-2_ | 0.2027 |
| 8 | 2 | Line.a8611_48332_s | d5799561-2_i | e193509 | 0.2037 |
| 9 | 3 | Line.ln5594236-1 | e193509 | m1142839 | 0.2661 |
| 10 | 4 | Line.ln5503575-1 | m1142839 | m1142832 | 0.3391 |
| 11 | 5 | Line.ln6292614-1 | m1142832.3 | p829962.3 | 0.3437 |
| 12 | 6 | Line.ln5807087-1 | p829962.3 | m1142835.3 | 0.3498 |
| 13 | 7 | Line.ln5896824-1 | m1142835.3 | l3104853.3 | 0.3872 |
| 14 | 8 | Transformer.t52605 | l3104853.3 | x3104853c.1. | 0.3872 |
| 15 | 9 | Line.tpx260518c0 | x3104853c.1.: | sx3104853c.1 | 0.4024 |
| 16 | -1 | Load.260518c0a | sx3104853c.1 | | |
| 17 | -1 | Load.260518c0b | sx3104853c.2 | | |



# Questions ??

Together…Shaping the Future of Energy®