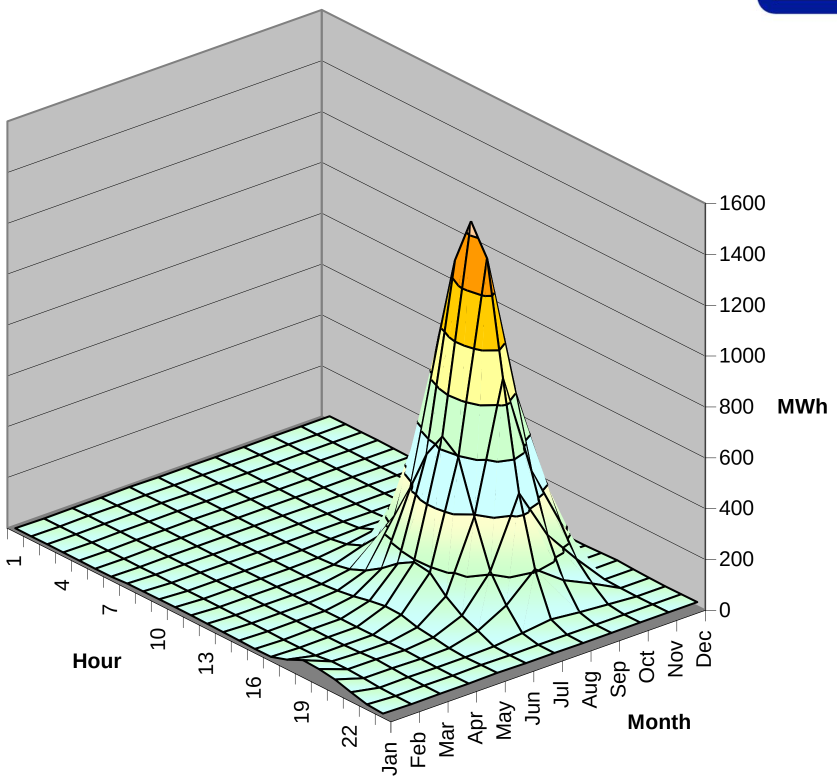


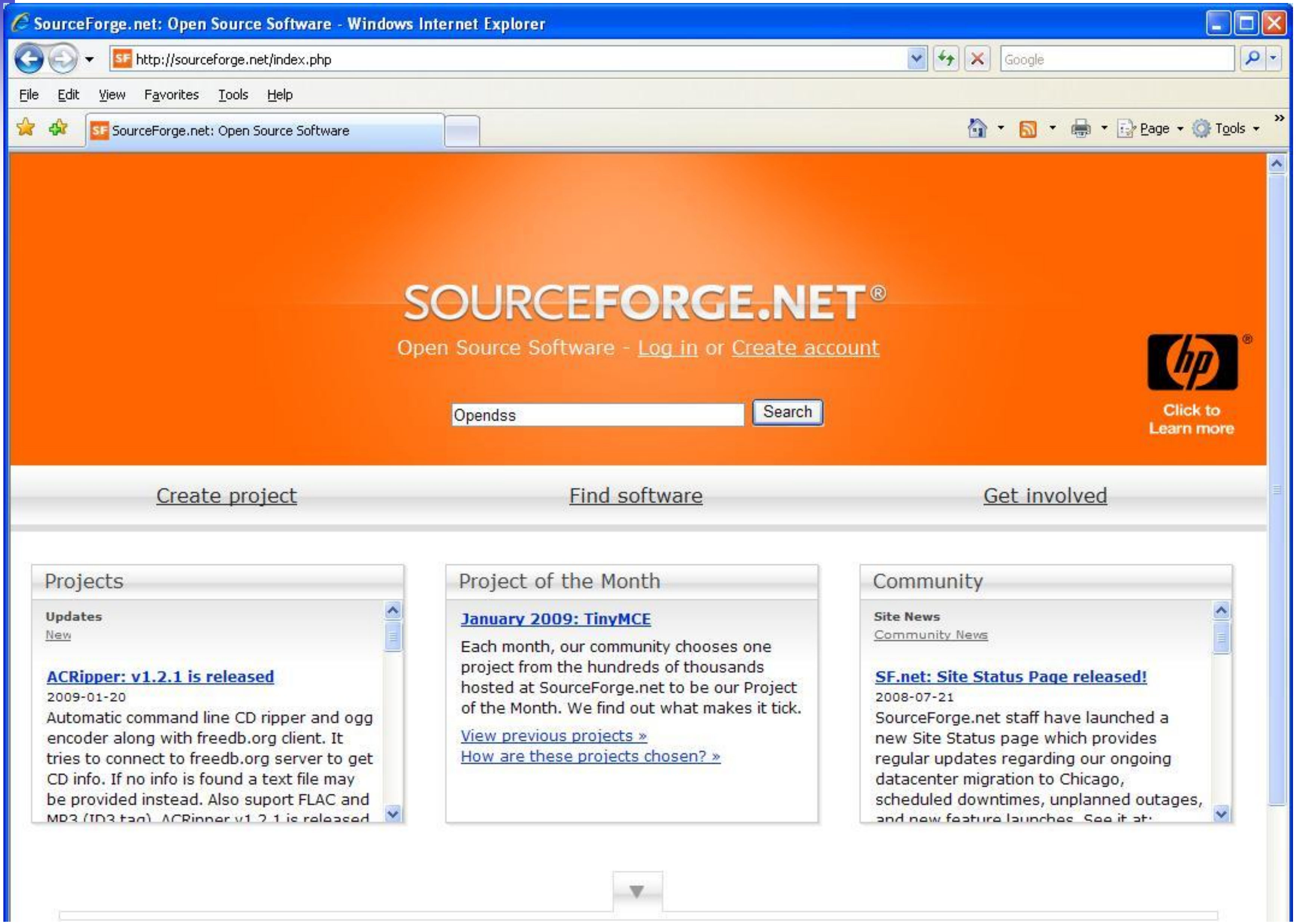
ELECTRIC POWER RESEARCH INSTITUTE

OpenDSS
Level 2 Training

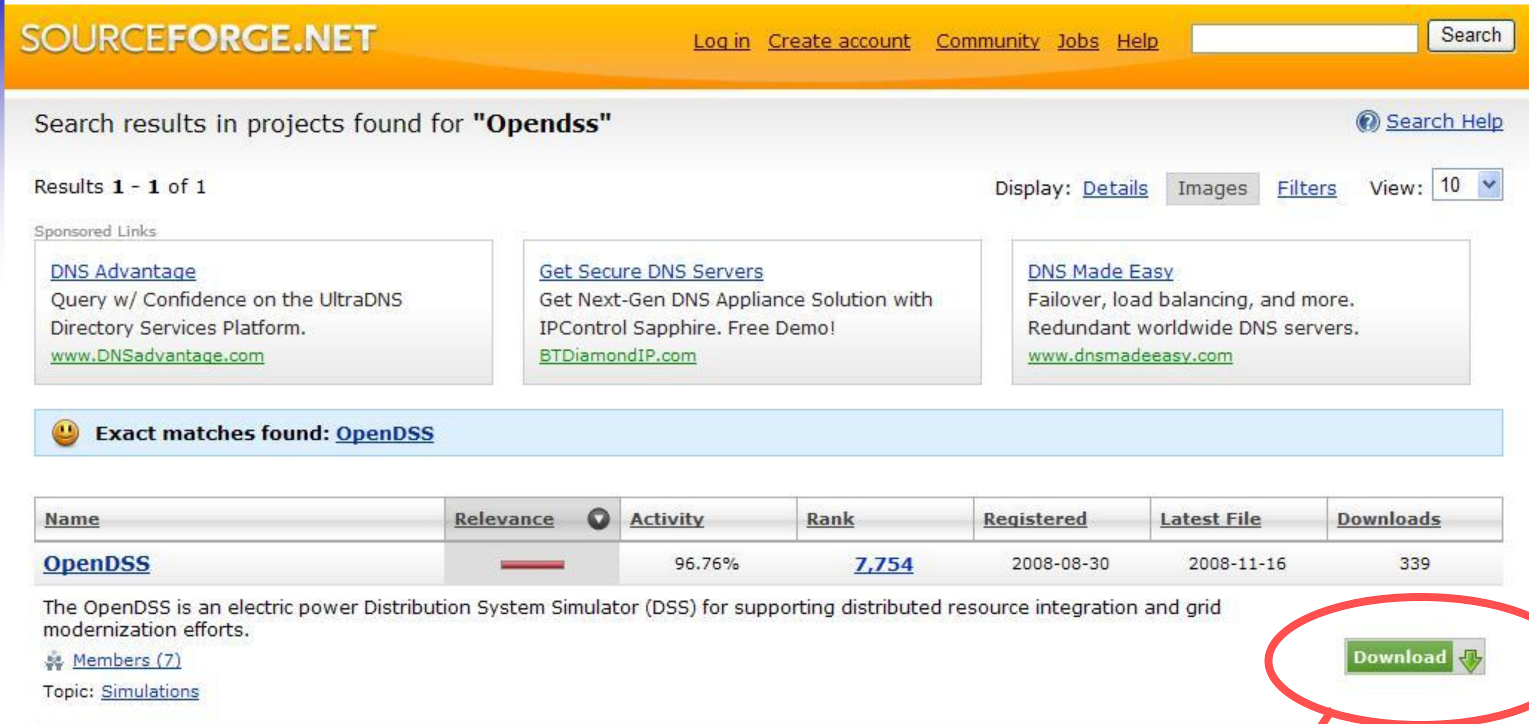
27 April 2009

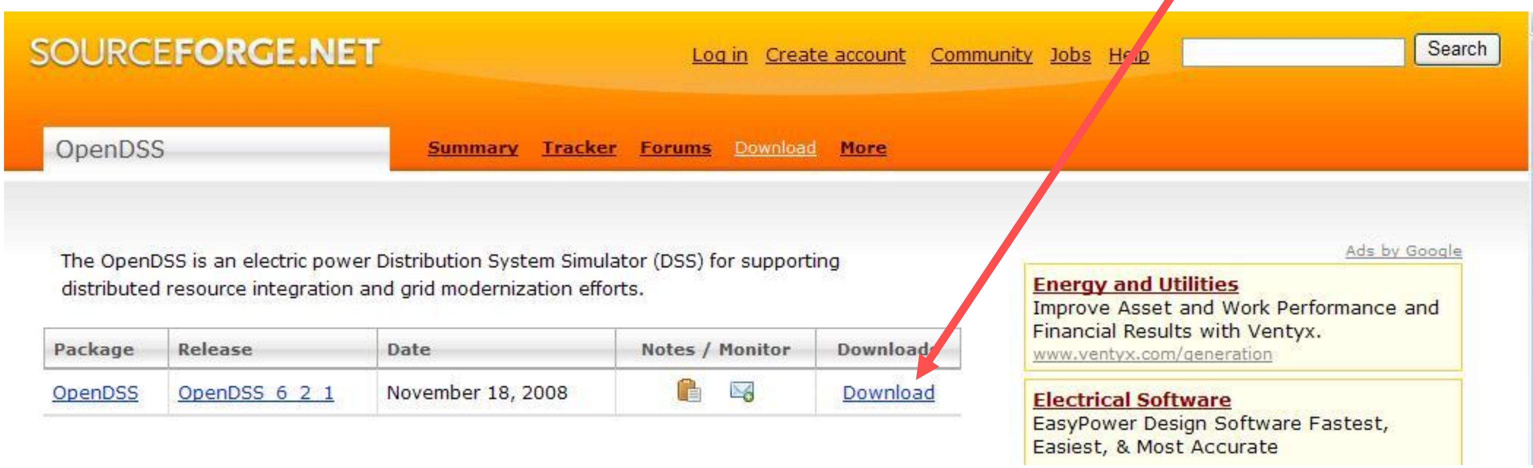
Roger Dugan
rdugan@epri.com

Getting Started:
Installation & Basic Usage



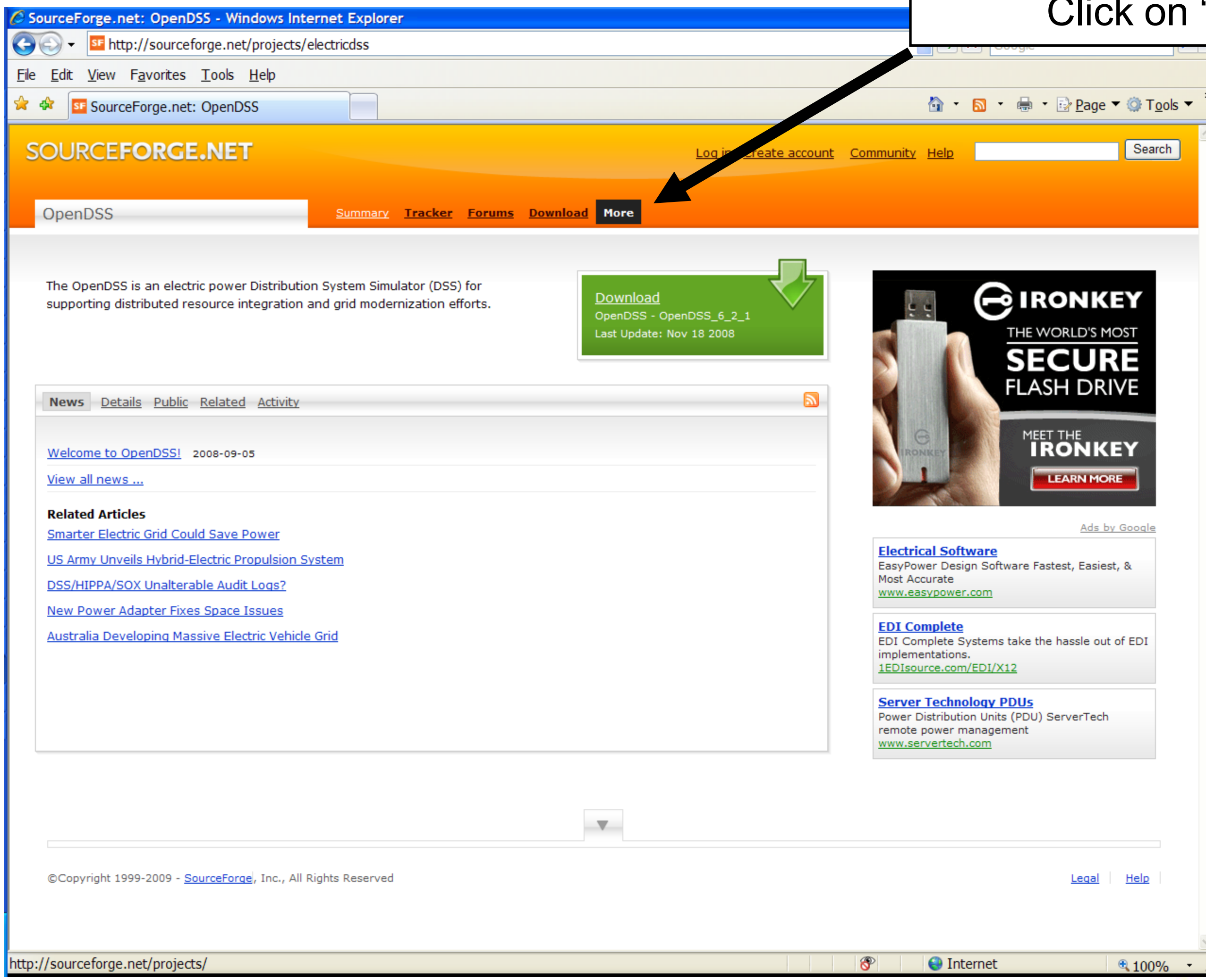
EPRI ELECTRIC POWER RESEARCH INSTITUTE


SOURCEFORGE.NET
Log in Create account Community Jobs Help
Search
Search results in projects found for "Opendss"
Search Help
Results 1 - 1 of 1
Display: Details Images Filters View: 10
Sponsored Links
DNS Advantage
Query w/ Confidence on the UltraDNS Directory Services Platform.
www.DNSadvantage.com
Get Secure DNS Servers
Get Next-Gen DNS Appliance Solution with IPControl Sapphire. Free Demo!
BTDiamondIP.com
DNS Made Easy
Failover, load balancing, and more. Redundant worldwide DNS servers.
www.dnsmadeeasy.com
Exact matches found: OpenDSS
Name Relevance Activity Rank Registered Latest File Downloads
OpenDSS 96.76% 7,754 2008-08-30 2008-11-16 339
The OpenDSS is an electric power Distribution System Simulator (DSS) for supporting distributed resource integration and grid modernization efforts.
Members (7)
Topic: Simulations
Download


SOURCEFORGE.NET
Log in Create account Community Jobs Help
Search
OpenDSS
Summary Tracker Forums Download More
The OpenDSS is an electric power Distribution System Simulator (DSS) for supporting distributed resource integration and grid modernization efforts.
Package Release Date Notes / Monitor Download
OpenDSS OpenDSS_6_2_1 November 18, 2008 Download
Ads by Google
Energy and Utilities
Improve Asset and Work Performance and Financial Results with Ventyx.
www.ventyx.com/generation
Electrical Software
EasyPower Design Software Fastest, Easiest, & Most Accurate

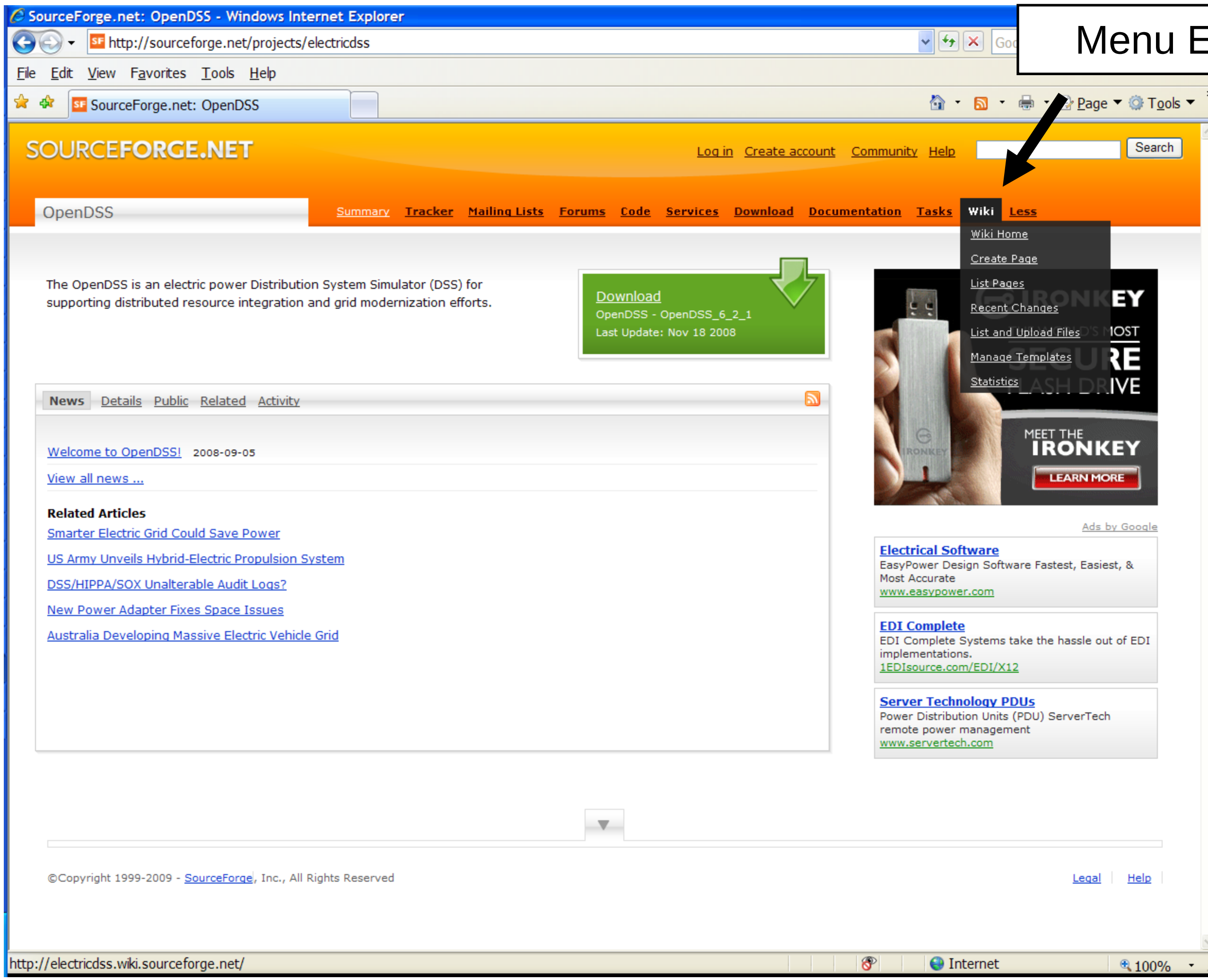
Finding the Wiki …

Click on “More”

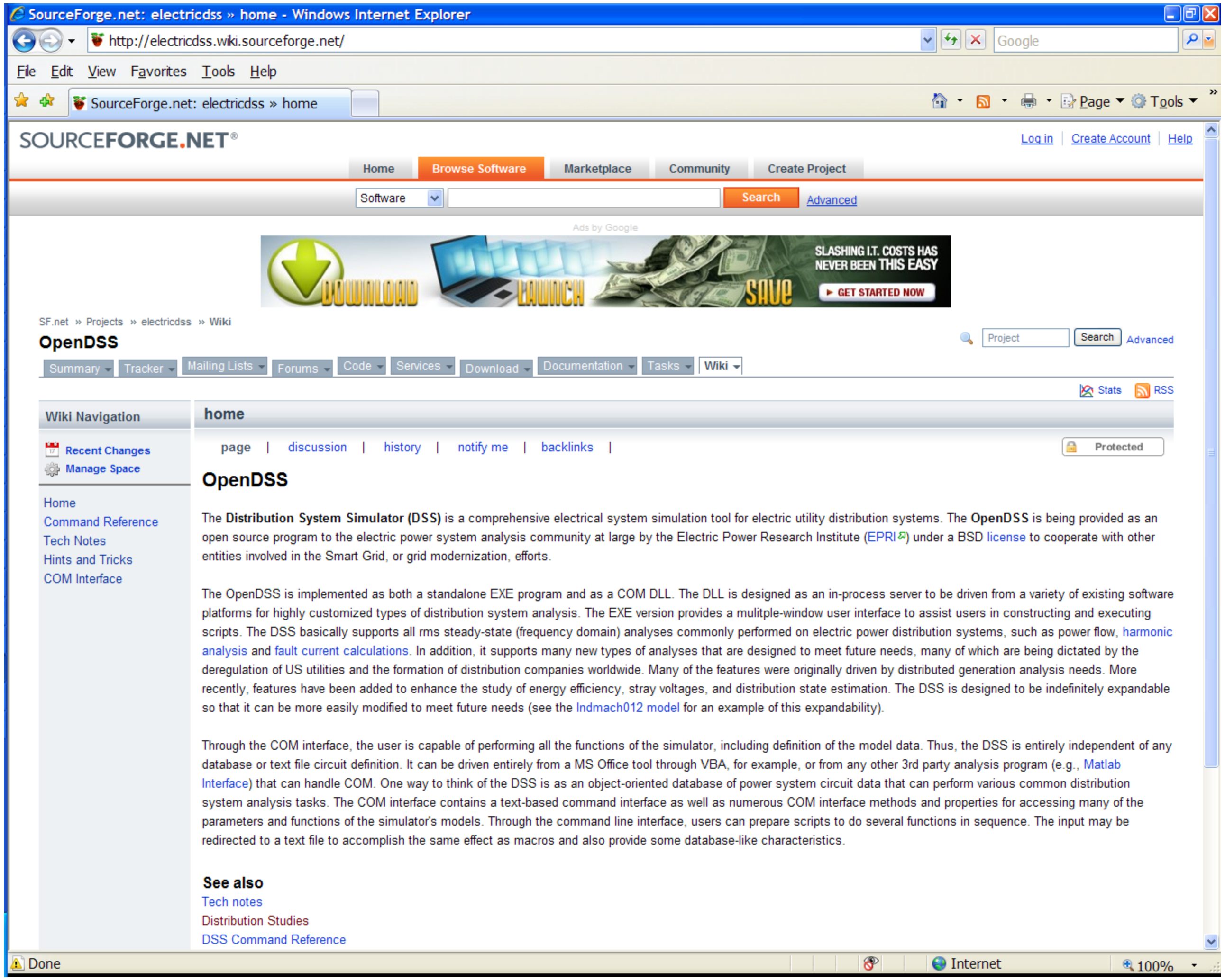


Finding the Wiki, cont'd

Menu Expands; Select Wiki



Wiki Home Page (Latest documentation)



EPRI ELECTRIC POWER RESEARCH INSTITUTE

Release Versions Vs. Source Code

- Release versions are posted irregularly
- You can keep up with the latest changes by accessing the source code and building the latest version
  - Some of the docs on the Wiki apply only to latest changes
- Compilers
  - Delphi 2007 (full IDE)
    - This is what we use for development
  - Turbo Delphi (Free)
    - https://downloads.embarcadero.com/free/turbodelphi

Accessing the SourceForge.Net Source Code Repository with TortoiseSVN

- Install a 32-bit TortoiseSVN client from tortoisesvn.net/downloads.
- Recommendation: From the TortoiseSVN General Settings dialog and click the last check box, to use "_svn" instead of ".svn" for local working directory name.

  Then, to grab the 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://electricdss.svn.sourceforge.net/svnroot/electricdss".
  - change the checkout directory if it points somewhere other than what you want.

ELECTRIC POWER RESEARCH INSTITUTE

Program Files

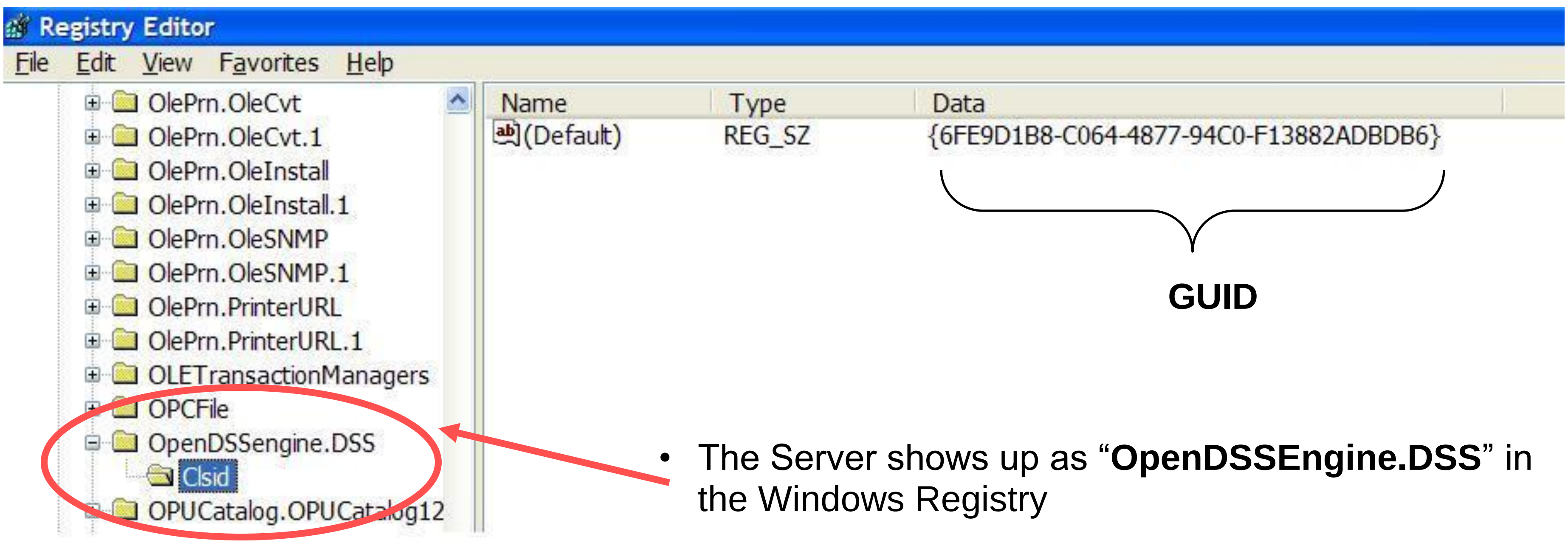
• OpenDSS.EXE	Standalone EXE
• OpenDSSEngine.DLL	In-process COM server
• KLUSolve.DLL	Sparse matrix solver
• DSSgraph.DLL	DSS graphics output

- Copy these files to the directory (folder) of your choice
  - Typically **`c:\OpenDSS`** *or* **`c:\Program Files\OpenDSS`**

- **If you intend to drive OpenDSS from another program, you will need to <u>register the COM server</u>**

Registering the COM Server

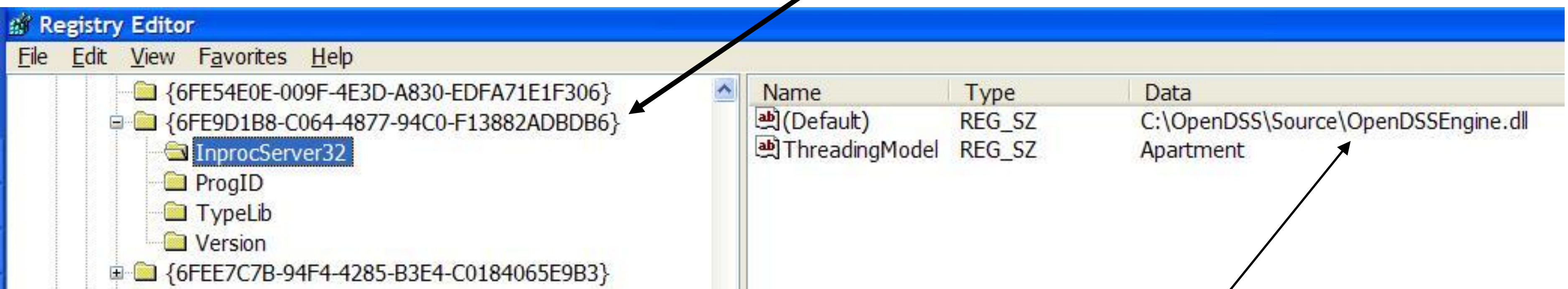
- In DOS window, change to the folder where you installed it and type:
  - **`Regsvr32 OpenDSSEngine.DLL`**



- The Server shows up as “**OpenDSSEngine.DSS**” in the Windows Registry

Registering the COM Server, cont'd

If you look up the GUID



Points to OpenDSSEngine.DLL (In-process server, Apartment Threading model)

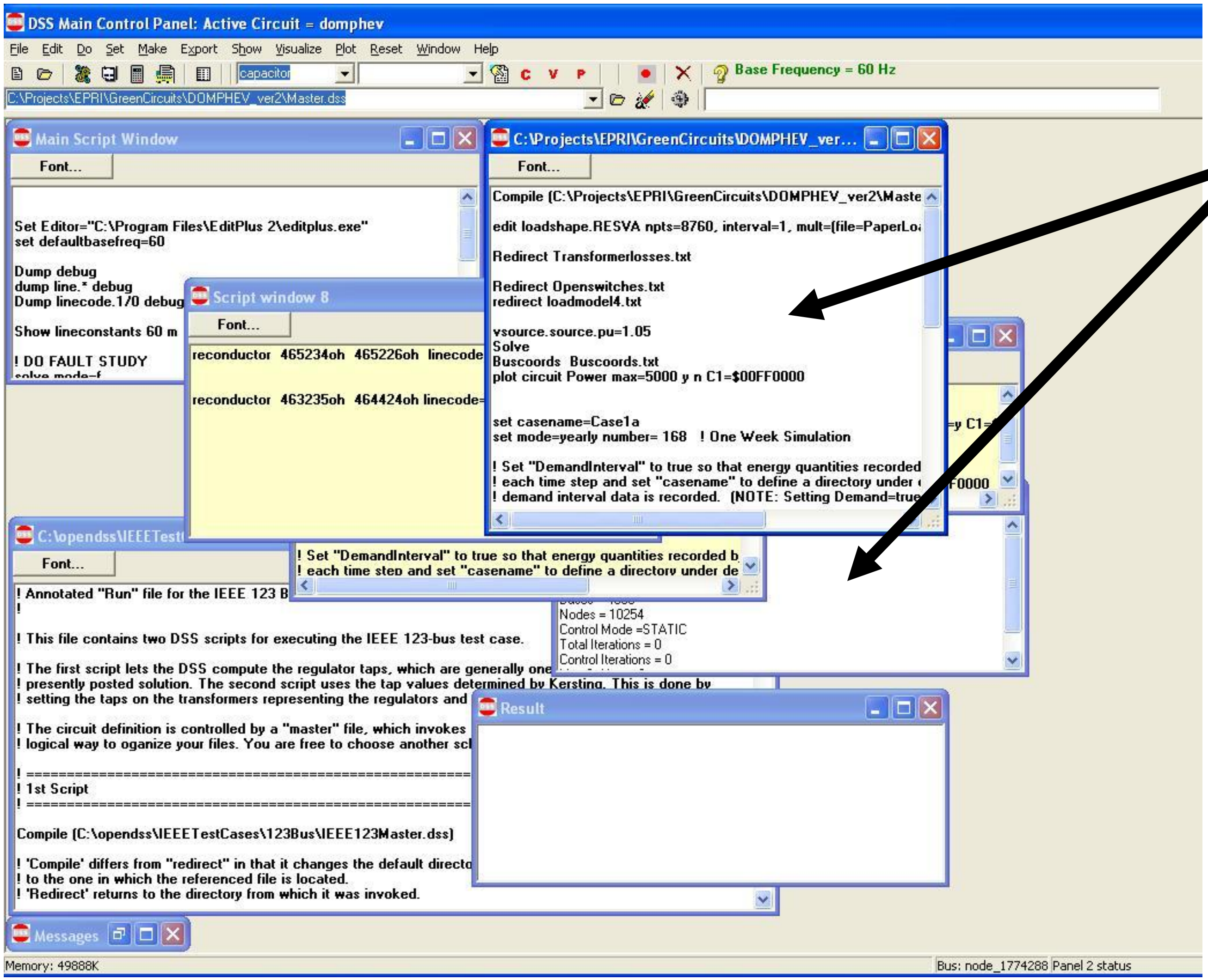
Accessing the COM Server

- In MATLAB:
  - `DSSobj = actxserver('OpenDSSEngine.DSS');`

- In VBA:
  - ```
    Public DSSobj As OpenDSSEngine.DSS
    Set DSSobj = New OpenDSSEngine.DSS
    ```

- In PYTHON:
  - `self.engine = win32com.client.Dispatch("OpenDSSEngine.DSS")`
    ```

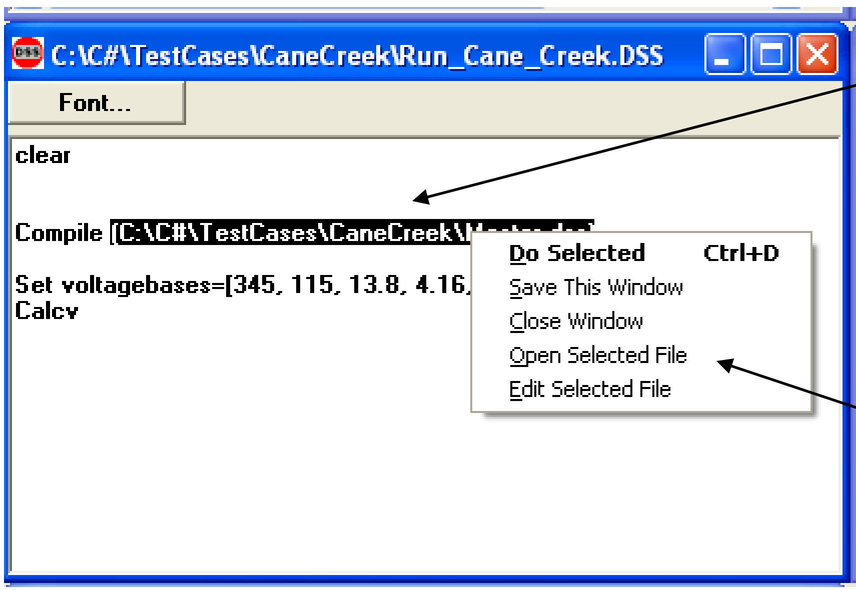
OpenDSS Standalone EXE User Interface



Multiple script windows

Any script window may be used at any time.

Executing Scripts in the Stand-alone EXE



Select all or part of a line

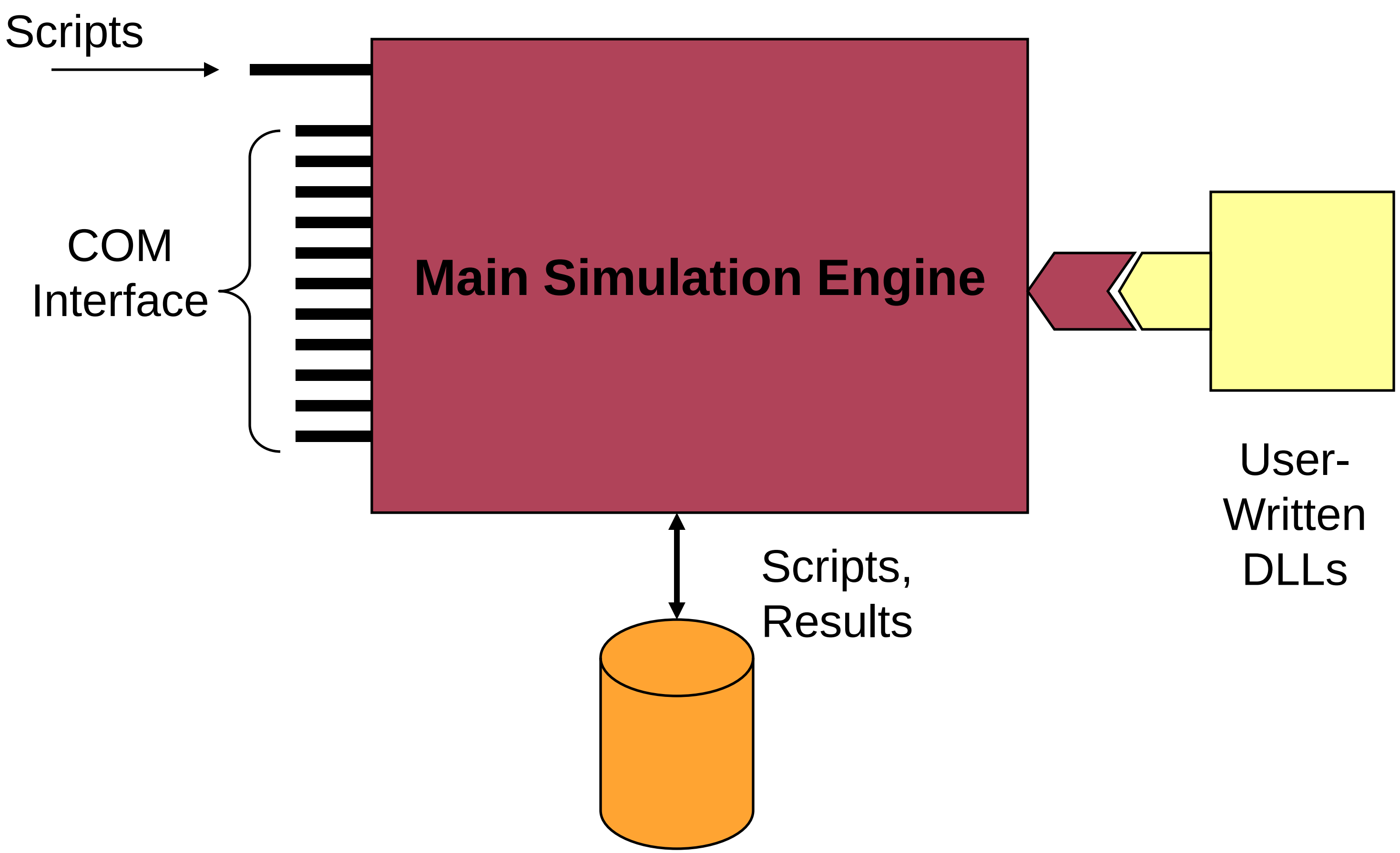
Right-Click to get this pop-up menu

DSS executes selected line or opens selected file name

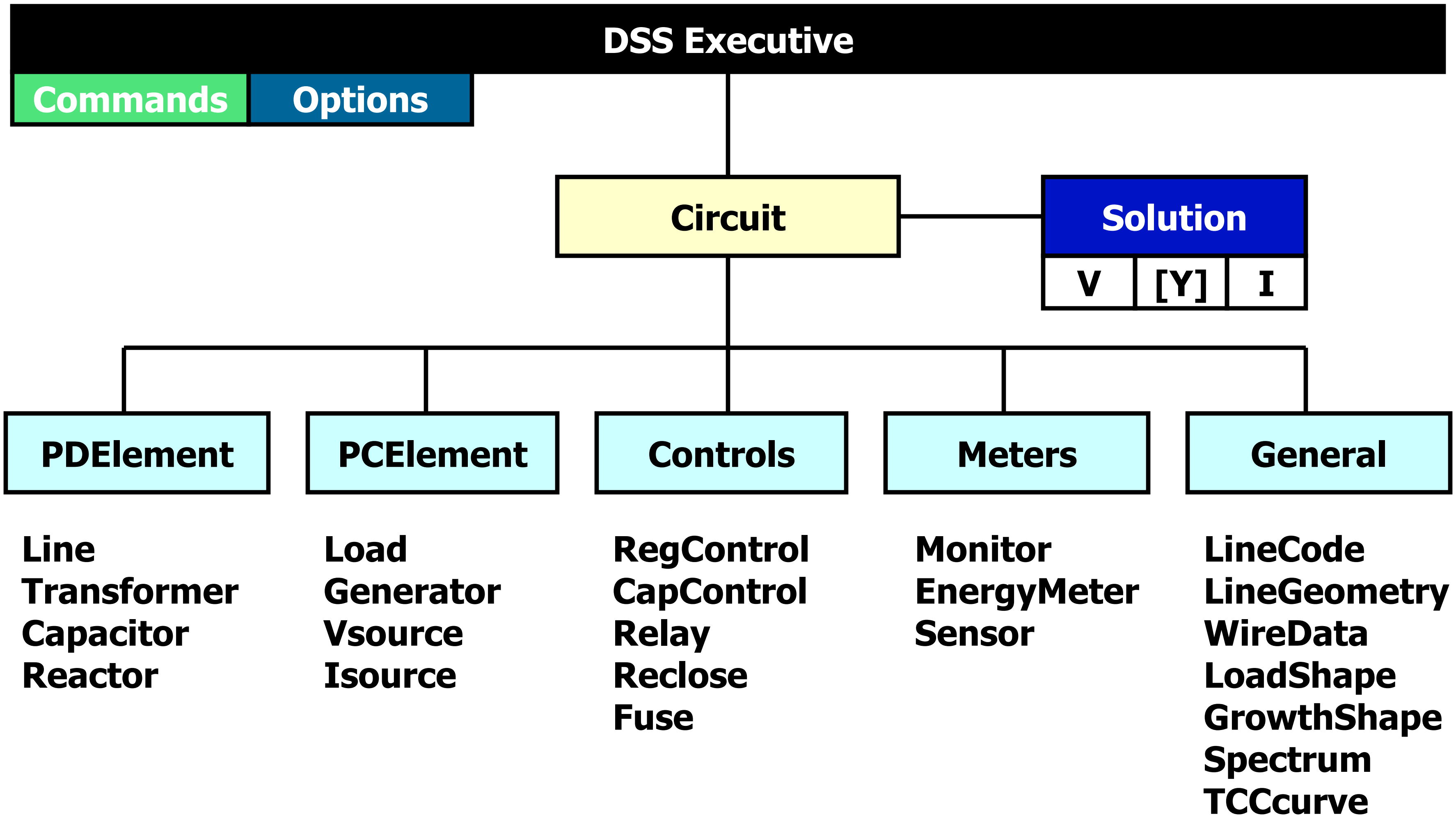
Any script window may be used at any time.

DSS Structure

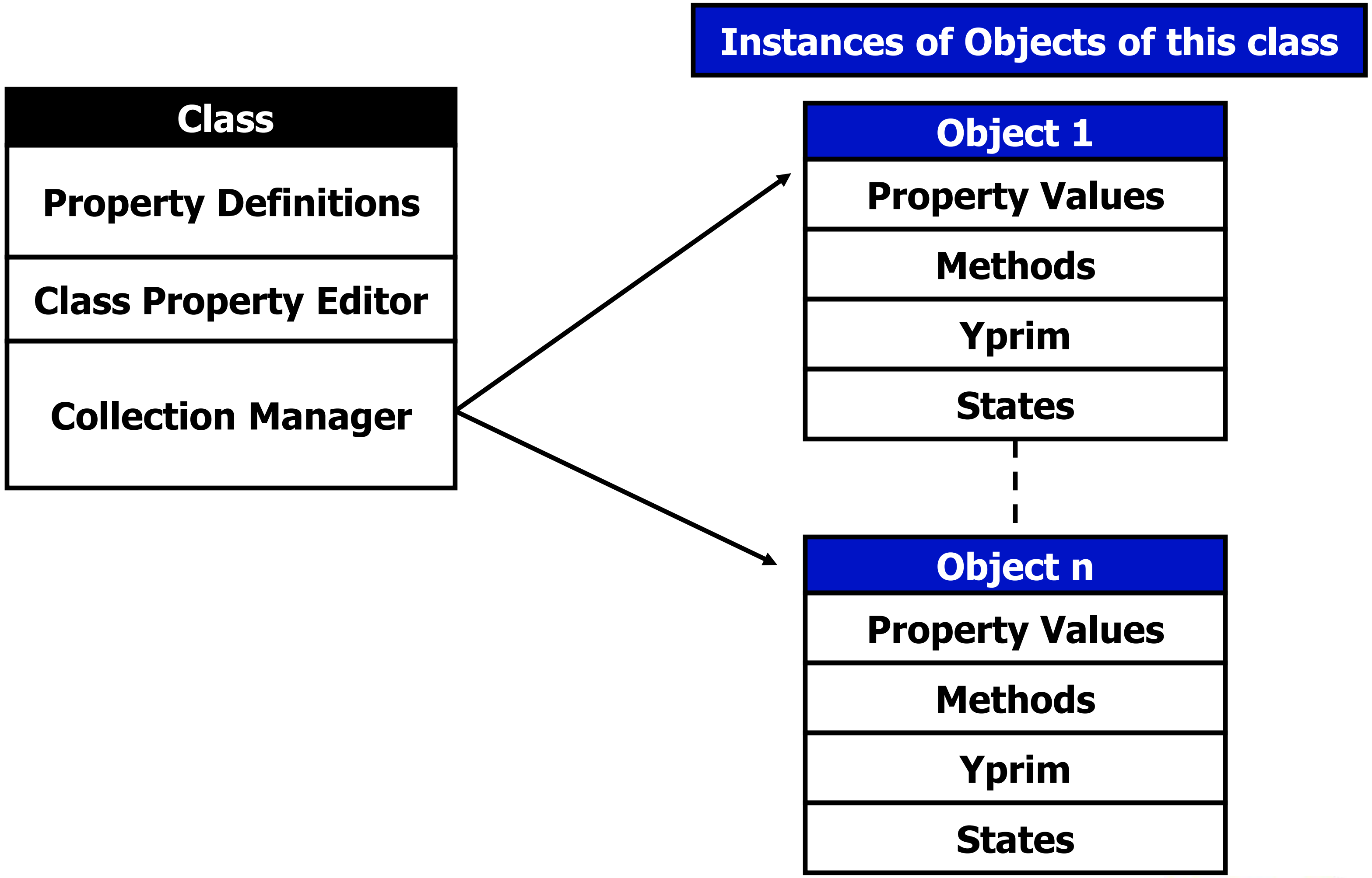
DSS Structure



DSS Object Structure



DSS Class Structure



DSS Classes (as of 2009)

- Power Delivery (PD) Elements
  - Line
  - Transformer
  - Reactor
  - Capacitor
- Power Conversion (PC) Elements
  - Load
  - Generator
  - Vsource
  - Isource
- Control Elements
  - RegControl
  - CapControl
  - Recloser
  - Relay
  - Fuse
- Metering Elements
  - Monitor
  - EnergyMeter
  - Sensor
- General
  - LineCode
  - LineGeometry
  - Loadshape
  - Growthshape
  - Wiredata
  - Spectrum
  - TCC Curves

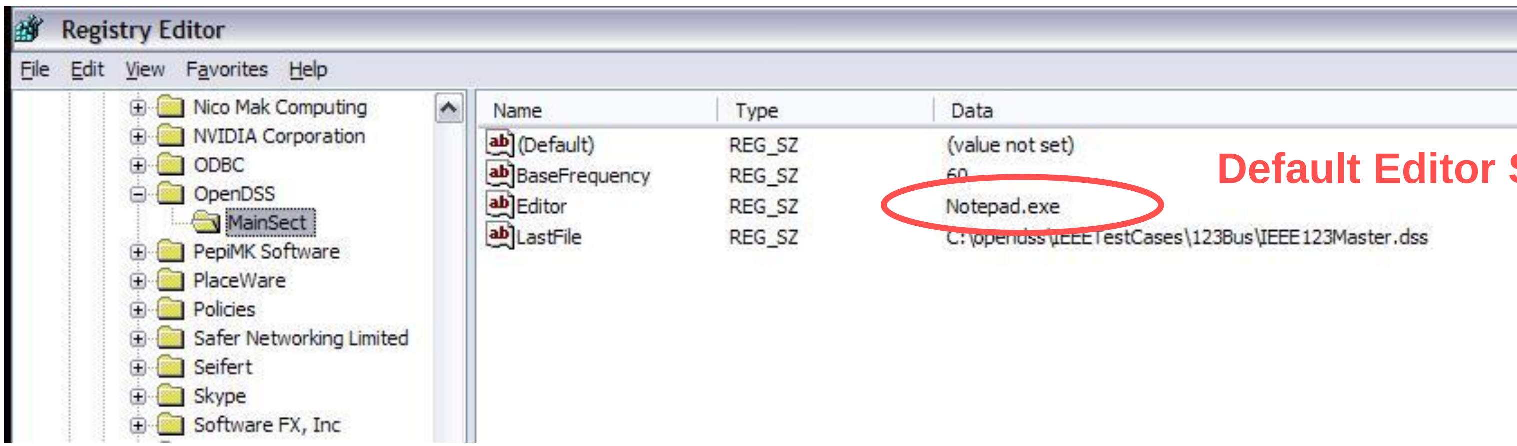
Organizing Your User Interface

Organizing Your Main Screen

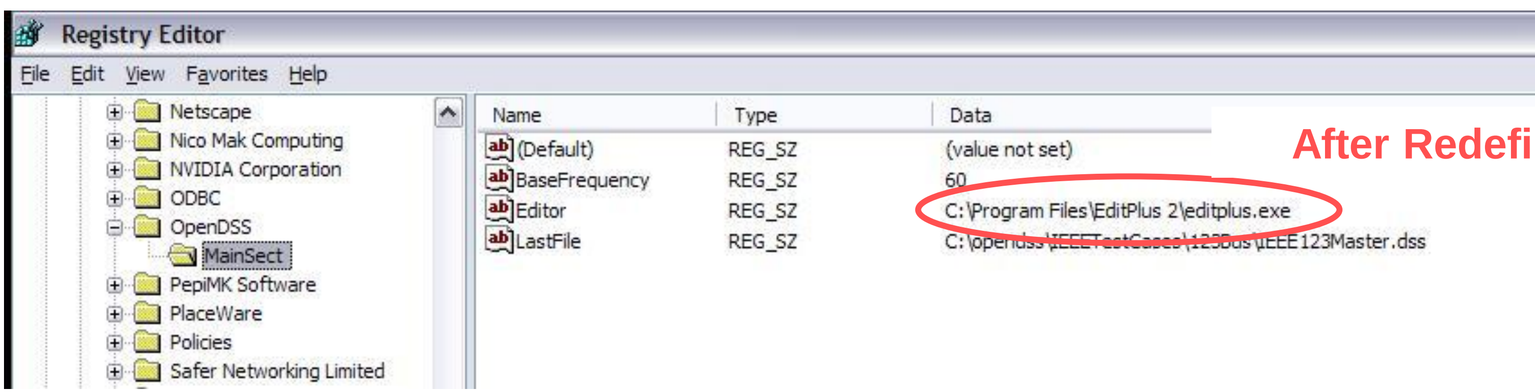
- The OpenDSS saves all windows on the main screen
- The appear where you left them when you shut down
- The next time you start up, you can resume your work
- Values are saved in a file (*OpenDSS.ini*) saved in the OpenDSS.exe folder
  - Note: You can update the program simply by copying in new exe and dll files.
  - Do not overwrite the ".ini" file if you want to preserve your workspace
    - However, if the .ini file gets corrupted, you may simply delete it.
- It is a good idea to come up with a comfortable way to organize your script windows …

OpenDSS Registry Entries

- Certain persistent values are saved to the Windows Registry upon exiting the program

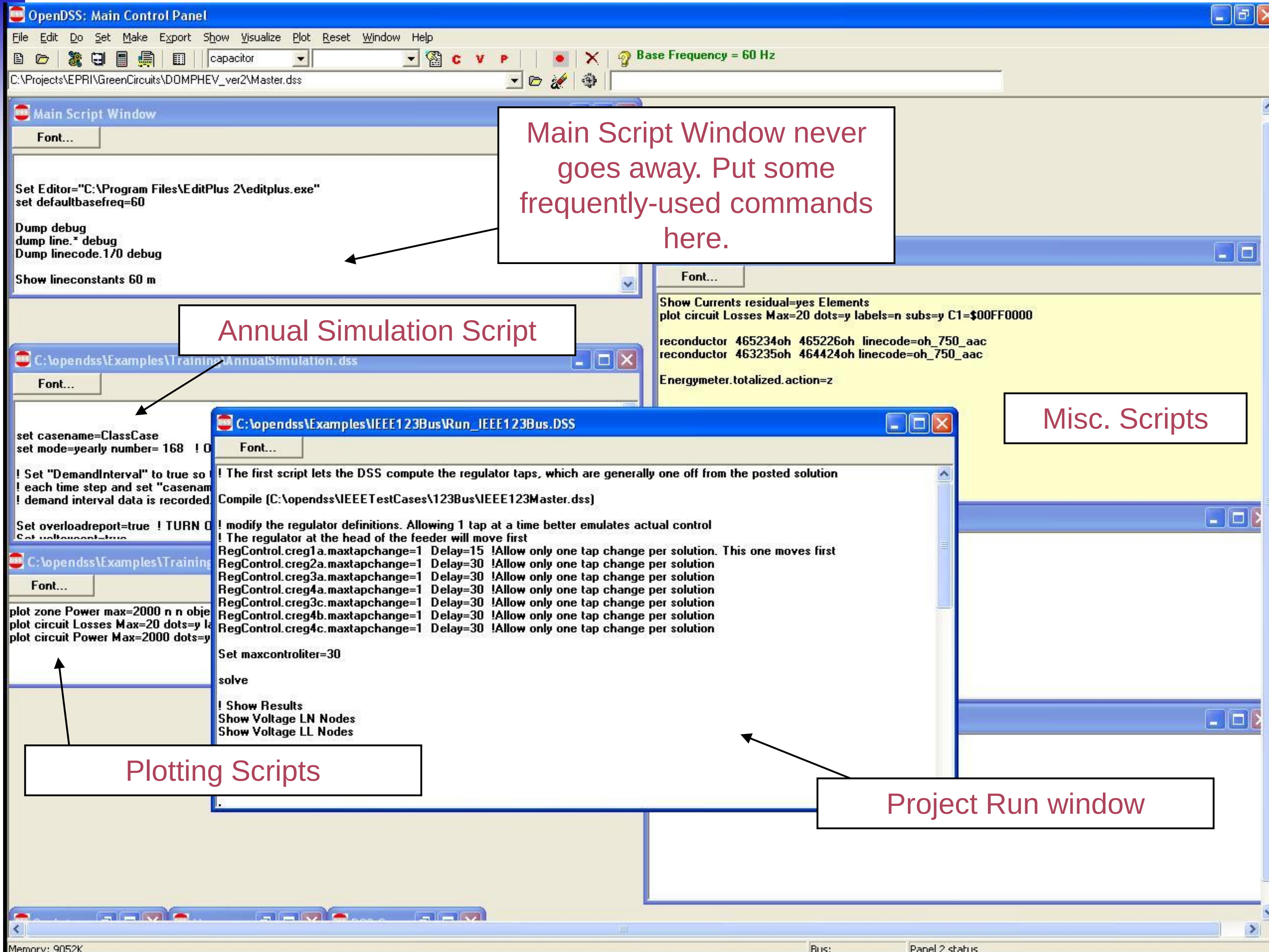


Default Editor Setting



After Redefining

EPRI ELECTRIC POWER RESEARCH INSTITUTE


OpenDSS: Main Control Panel
File Edit Do Set Make Export Show Visualize Plot Reset Window Help
capacitor
Base Frequency = 60 Hz
C:\Projects\EPRI\GreenCircuits\DOMPHEV_ver2\Master.dss
Main Script Window
Font...
Set Editor="C:\Program Files\EditPlus 2\editplus.exe"
set defaultbasefreq=60
Dump debug
dump line.* debug
Dump linecode.1/0 debug
Show lineconstants 60 m
Main Script Window never goes away. Put some frequently-used commands here.
Annual Simulation Script
C:\opendss\Examples\Training\AnnualSimulation.dss
set casename=ClassCase
set mode=yearly number= 168
! Set "DemandInterval" to true so
! each time step and set "casenam
! demand interval data is recorded
Set overloadreport=true ! TURN O
C:\opendss\Examples\Training
plot zone Power max=2000 n n obje
plot circuit Losses Max=20 dots=y la
plot circuit Power Max=2000 dots=y
Plotting Scripts
Show Currents residual=yes Elements
plot circuit Losses Max=20 dots=y labels=n subs=y C1=$00FF0000
reconductor 465234oh 465226oh linecode=oh_750_aac
reconductor 463235oh 464424oh linecode=oh_750_aac
Energymeter.totalized.action=z
Misc. Scripts
C:\opendss\Examples\IEEE123Bus\Run_IEEE123Bus.DSS
! The first script lets the DSS compute the regulator taps, which are generally one off from the posted solution
Compile (C:\opendss\IEEETestCases\123Bus\IEEE123Master.dss)
! modify the regulator definitions. Allowing 1 tap at a time better emulates actual control
! The regulator at the head of the feeder will move first
RegControl.creg1a.maxtapchange=1 Delay=15 !Allow only one tap change per solution. This one moves first
RegControl.creg2a.maxtapchange=1 Delay=30 !Allow only one tap change per solution
RegControl.creg3a.maxtapchange=1 Delay=30 !Allow only one tap change per solution
RegControl.creg4a.maxtapchange=1 Delay=30 !Allow only one tap change per solution
RegControl.creg3c.maxtapchange=1 Delay=30 !Allow only one tap change per solution
RegControl.creg4b.maxtapchange=1 Delay=30 !Allow only one tap change per solution
RegControl.creg4c.maxtapchange=1 Delay=30 !Allow only one tap change per solution
Set maxcontroliter=30
solve
! Show Results
Show Voltage LN Nodes
Show Voltage LL Nodes
Project Run window
Memory: 9052K
Bus:
Panel 2 status

Organizing Run Scripts

C:\opendss\Examples\IEEE123Bus\Run_IEEE123Bus.DSS

Font...

```
! The first script lets the DSS compute the regulator taps, which are generally one off from the posted solution

Compile (C:\opendss\IEEETestCases\123Bus\IEEE123Master.dss)

! modify the regulator definitions. Allowing 1 tap at a time better emulates actual control
! The regulator at the head of the feeder will move first
RegControl.creg1a.maxtapchange=1  Delay=15  !Allow only one tap change per solution. This
RegControl.creg2a.maxtapchange=1  Delay=30  !Allow only one tap change per solution
RegControl.creg3a.maxtapchange=1  Delay=30  !Allow only one tap change per solution
RegControl.creg4a.maxtapchange=1  Delay=30  !Allow only one tap change per solution
RegControl.creg3c.maxtapchange=1  Delay=30  !Allow only one tap change per solution
RegControl.creg4b.maxtapchange=1  Delay=30  !Allow only one tap change per solution
RegControl.creg4c.maxtapchange=1  Delay=30  !Allow only one tap change per solution

Set maxcontroliter=30

solve

! Show Results
Show Voltage LN Nodes
Show Voltage LL Nodes

Show Currents Elements
Show Powers kva Elements
Show taps   ! shows regulator taps
```

Compiles the Circuit Description

Override Some Property Settings and/or Define Some Additional Circuit Element

Change an option

Solve Snapshot Power Flow

Selected Results Display

EPRI ELECTRIC POWER RESEARCH INSTITUTE

Organizing Master File

So Compile Doesn't Fail

```
Clear

New Circuit.ExampleCircuit BasekV=138 pu=1.05 MVASC3 = 2000  MVASC1=2000

! Master file examples

! Library files
Redirect LineCode.dss
Redirect LoadShape.dss
Redirect GrowthShape.dss
Redirect TCC_Curve.dss
Redirect Spectrum.dss

! Circuit element descriptions are in a subdirectory "Feeders"
Redirect Feeders\Transformers.dss
Redirect Feeders\Branches.dss
Redirect Feeders\Loads.dss
Redirect Feeders\Capacitors.dss

Set Voltagebases=(69, 12.1, 4.16, 0.48)    ! define legal voltage bases
calcv        ! Abbrev for CalcVoltageBases

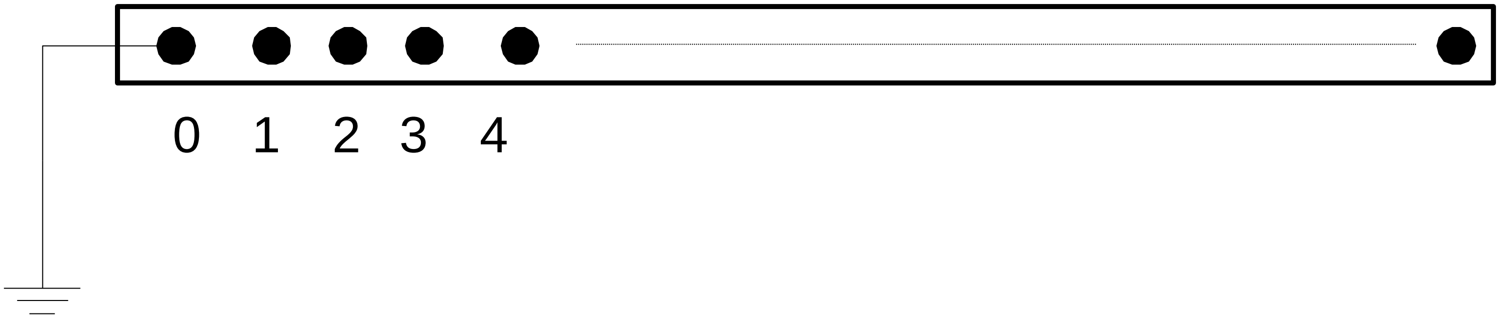
! Buses exit now so define coordinates
Buscoords buscoords.txt     ! Load bus x,y coordinates

! Define energy meters after voltage bases so they will know voltage bases
Redirect EnergyMeter.dss

! Don't do Solve here ... better to do it in Run File
```

Circuit Modeling Basics

DSS Bus Model



Referring to Buses and Nodes

Bus1=*BusName.1.2.3.0*

(This is the default for a 3-phase circuit element)

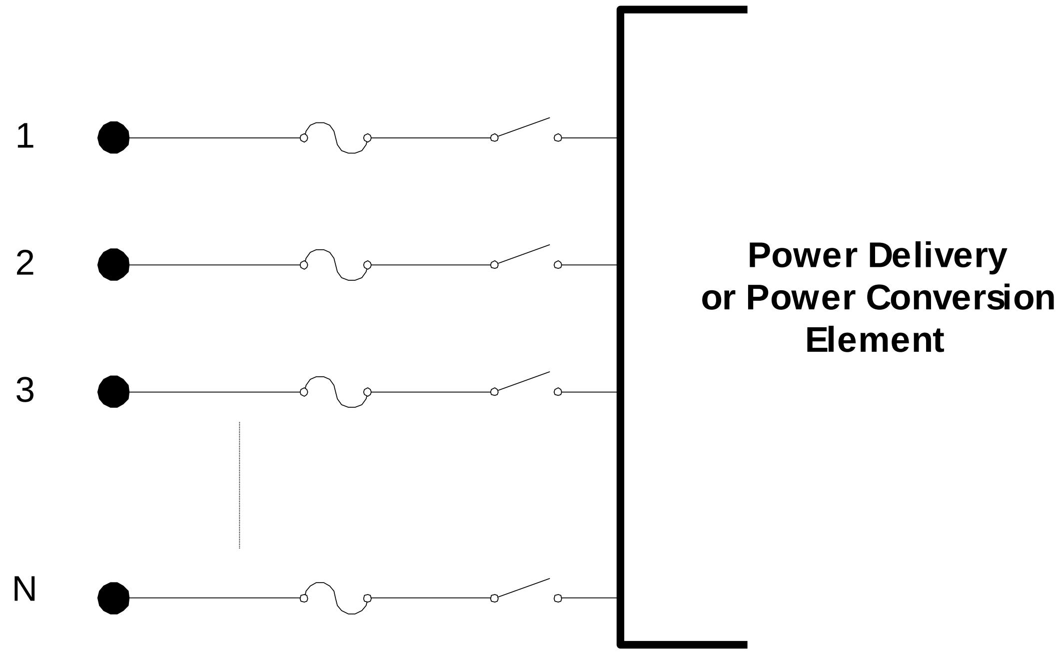
Shorthand notation for taking the default

Bus1=*BusName*

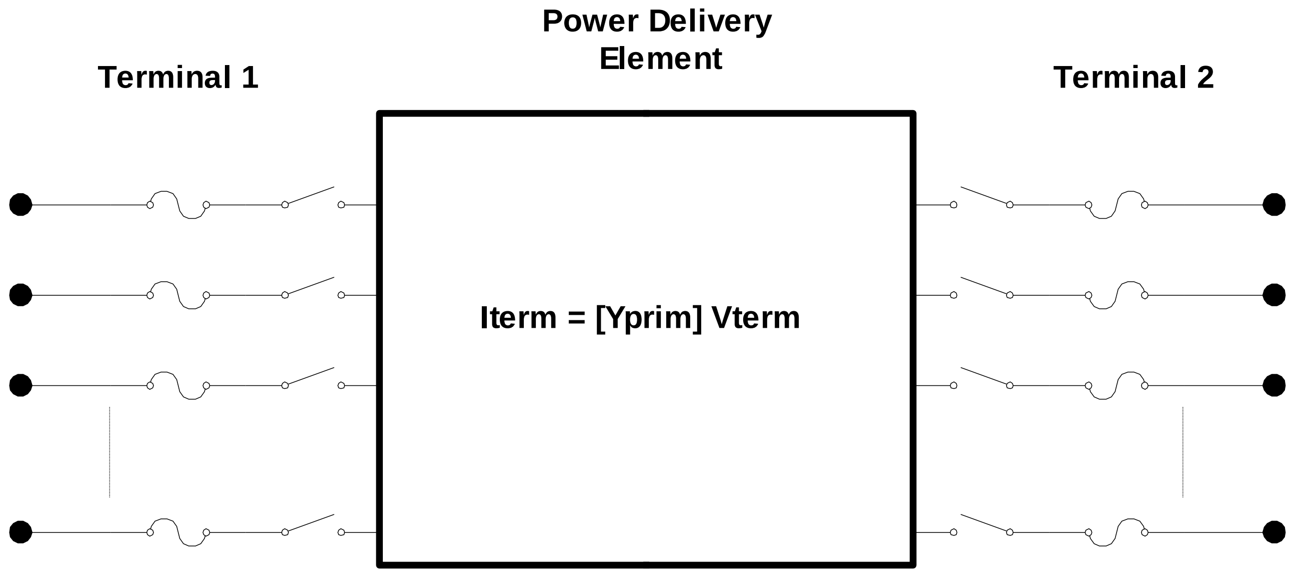
Note: Sometimes this can bite you (e.g. – Transformers, or capacitors with ungrounded neutrals)

ELECTRIC POWER RESEARCH INSTITUTE

DSS Terminal Definition



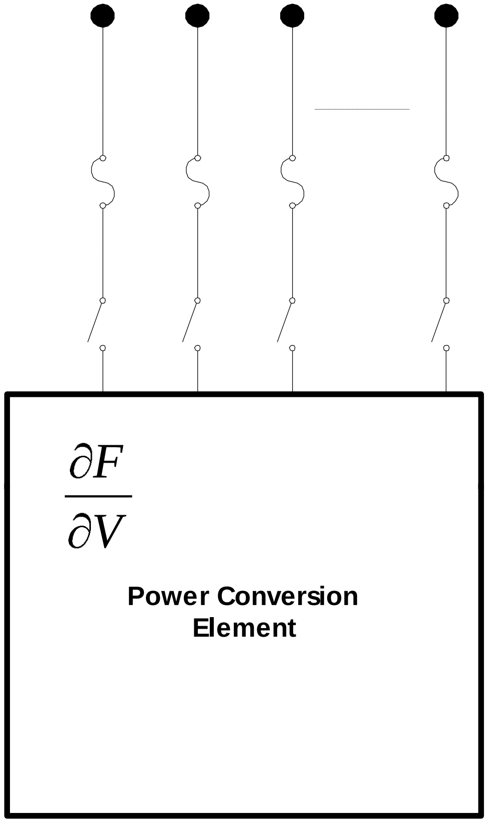
Power Delivery Elements



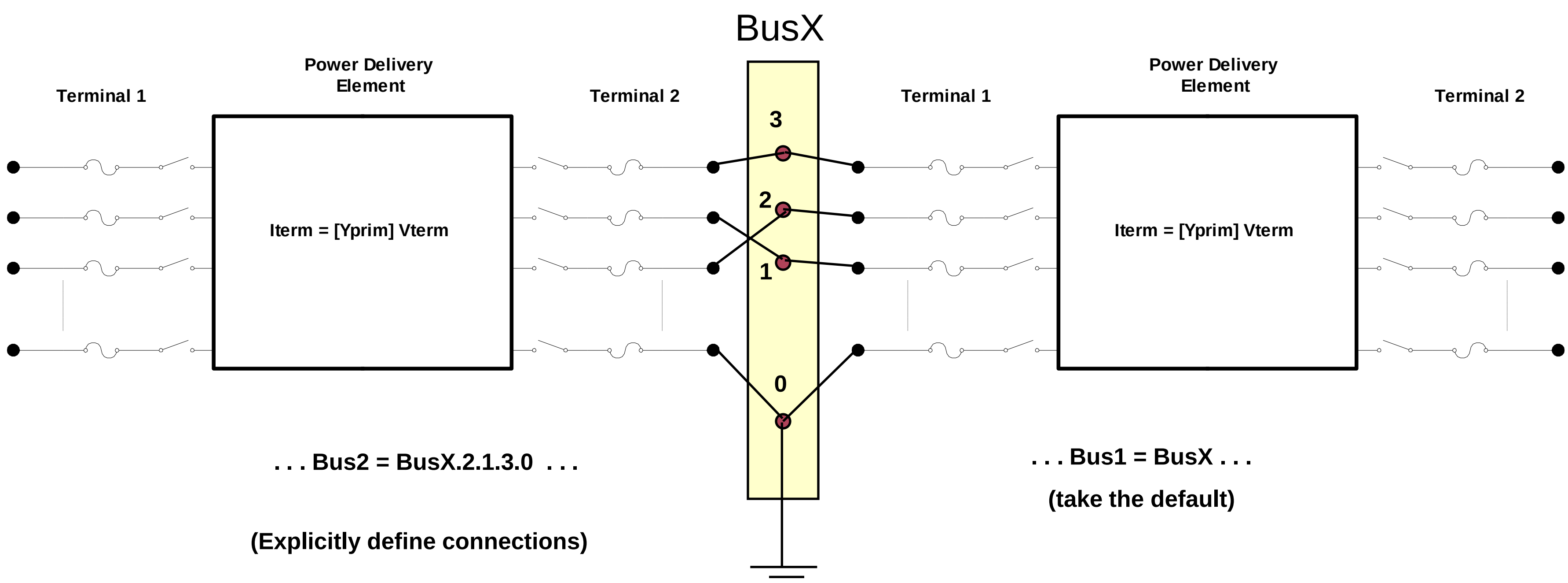
ELECTRIC POWER RESEARCH INSTITUTE

Power Conversion Elements

$I_{Term}(t) = \mathbf{F(V_{Term}}, [State], t)$



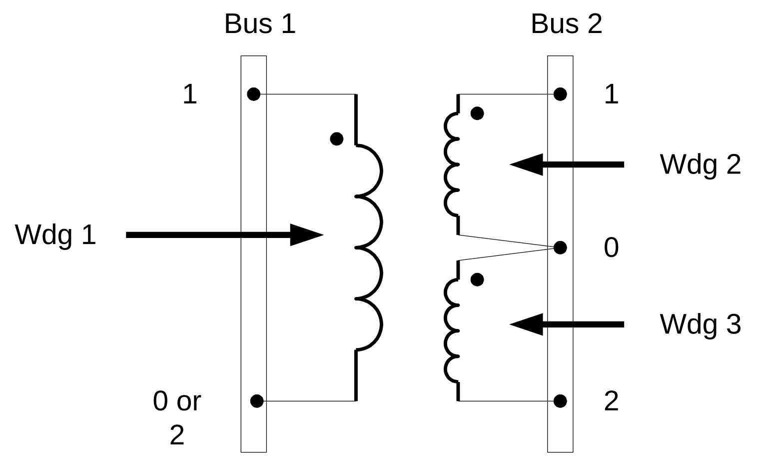
Circuit Elements are Connected together at the Nodes of Buses



DSS Convention: A *Terminal* can be connected to only one *Bus*. You can have any number of *Nodes* at a bus.

Connections for 1-Phase Residential Transformer

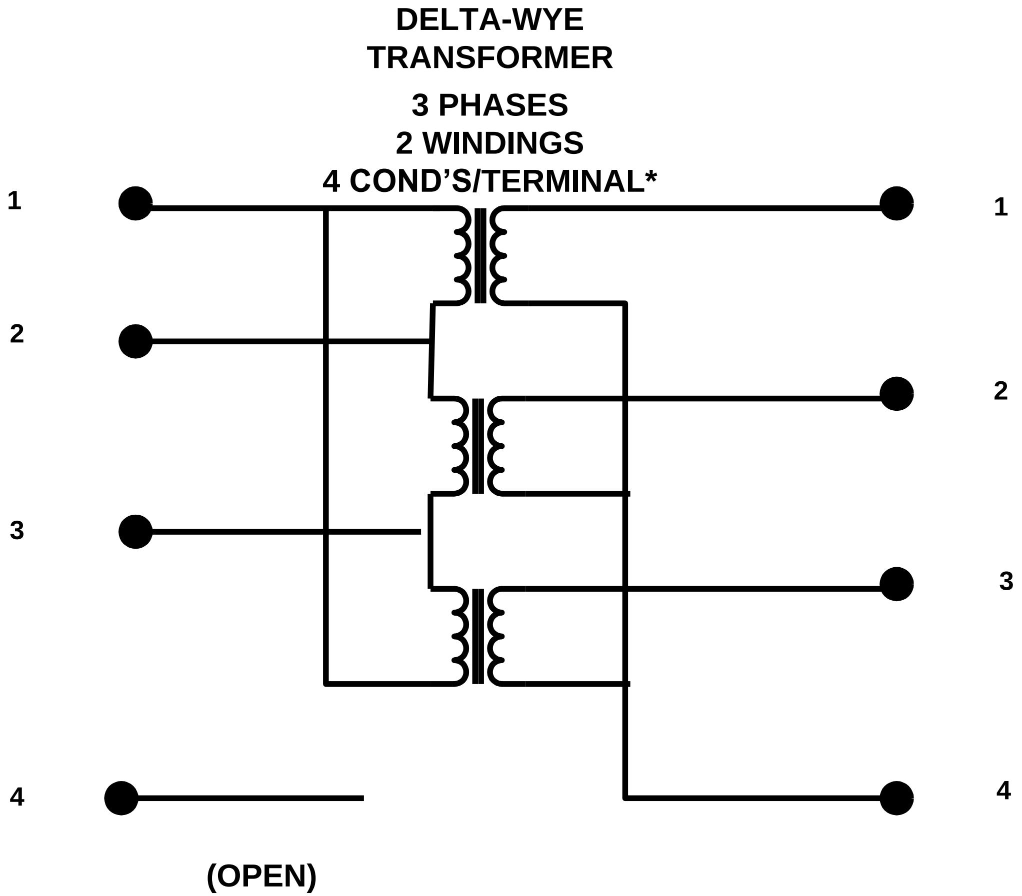
```
! Line-to-Neutral Connected 1-phase Center-tapped transformer
New  Transformer.Example1-ph  phases=1  Windings=3
~ Xhl=2.04  Xht=2.04  Xlt=1.36  %noloadloss=.2
~ Buses=[bus1.1  bus2.1.0   bus2.0.2]    !!! Note polarity
~ kVs=[7.2 .12  .12]     ! ratings of windings
~ kVAs=[25 25 25]
~ %Rs = [0.6  1.2  1.2]
~ conns=[wye wye wye]    ! default
```



Center-Tapped 1-Phase Transformer Model

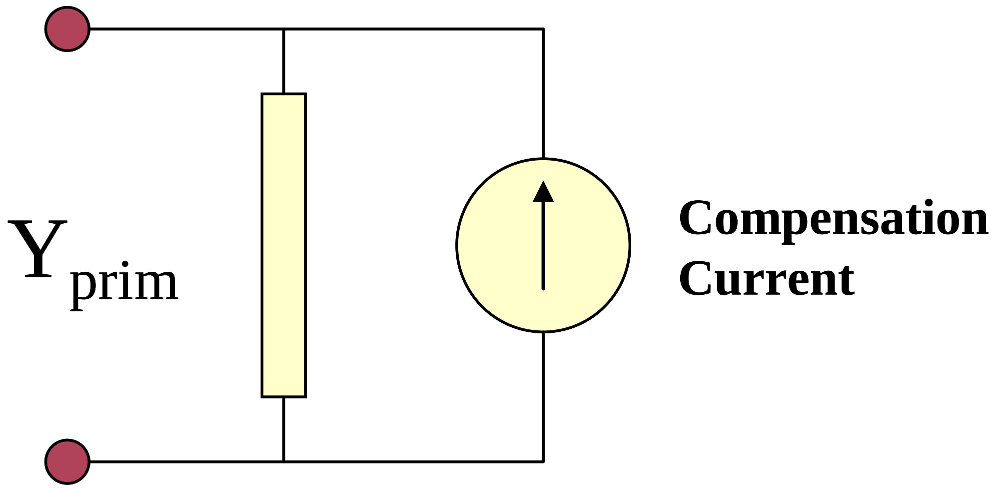
All Terminals of a Circuit Element Have Same Number of Conductors

3-Phase Transformer



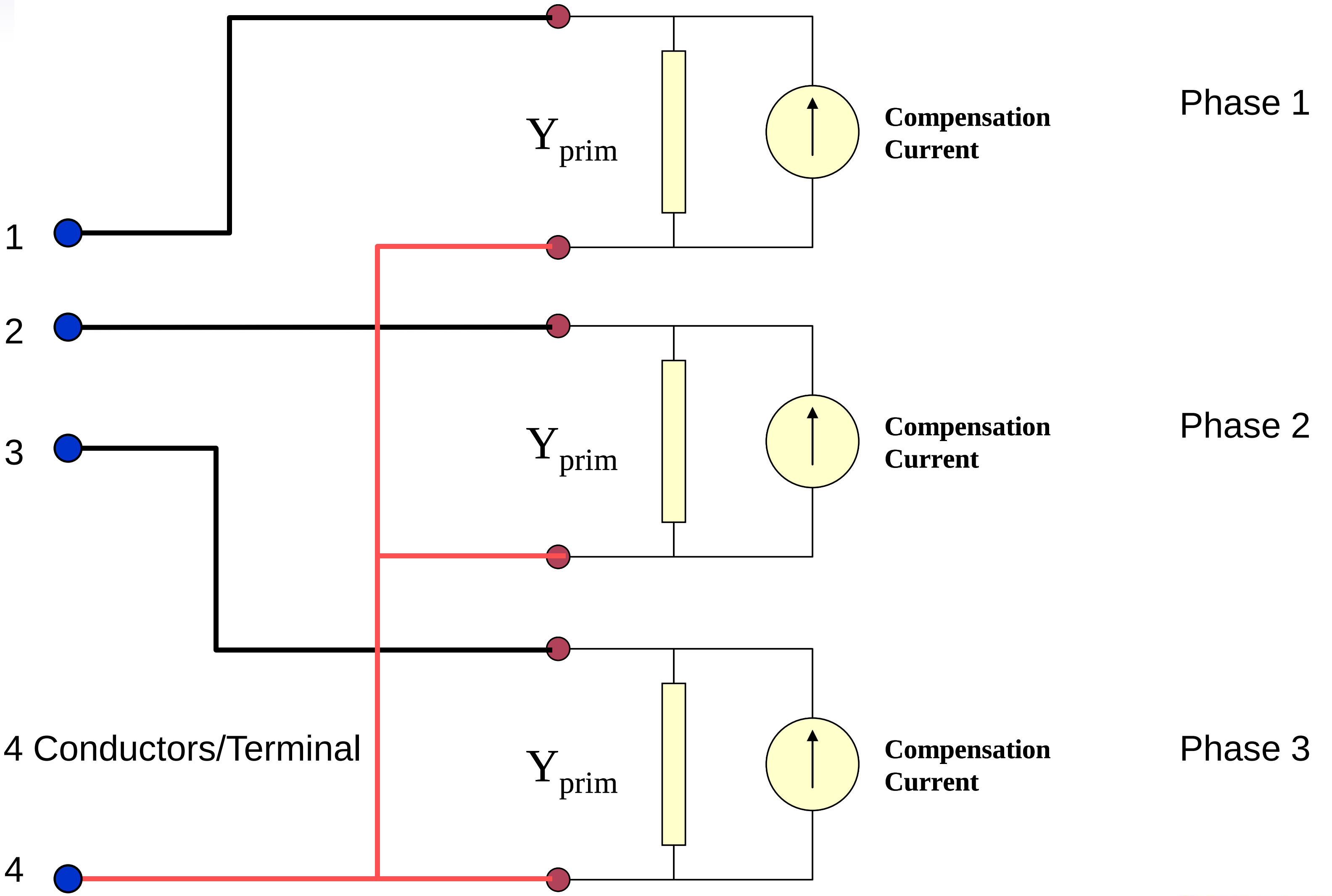
* MUST HAVE THE SAME NUMBER OF CONDUCTORS FOR EACH TERMINAL

Load (a PC Element)

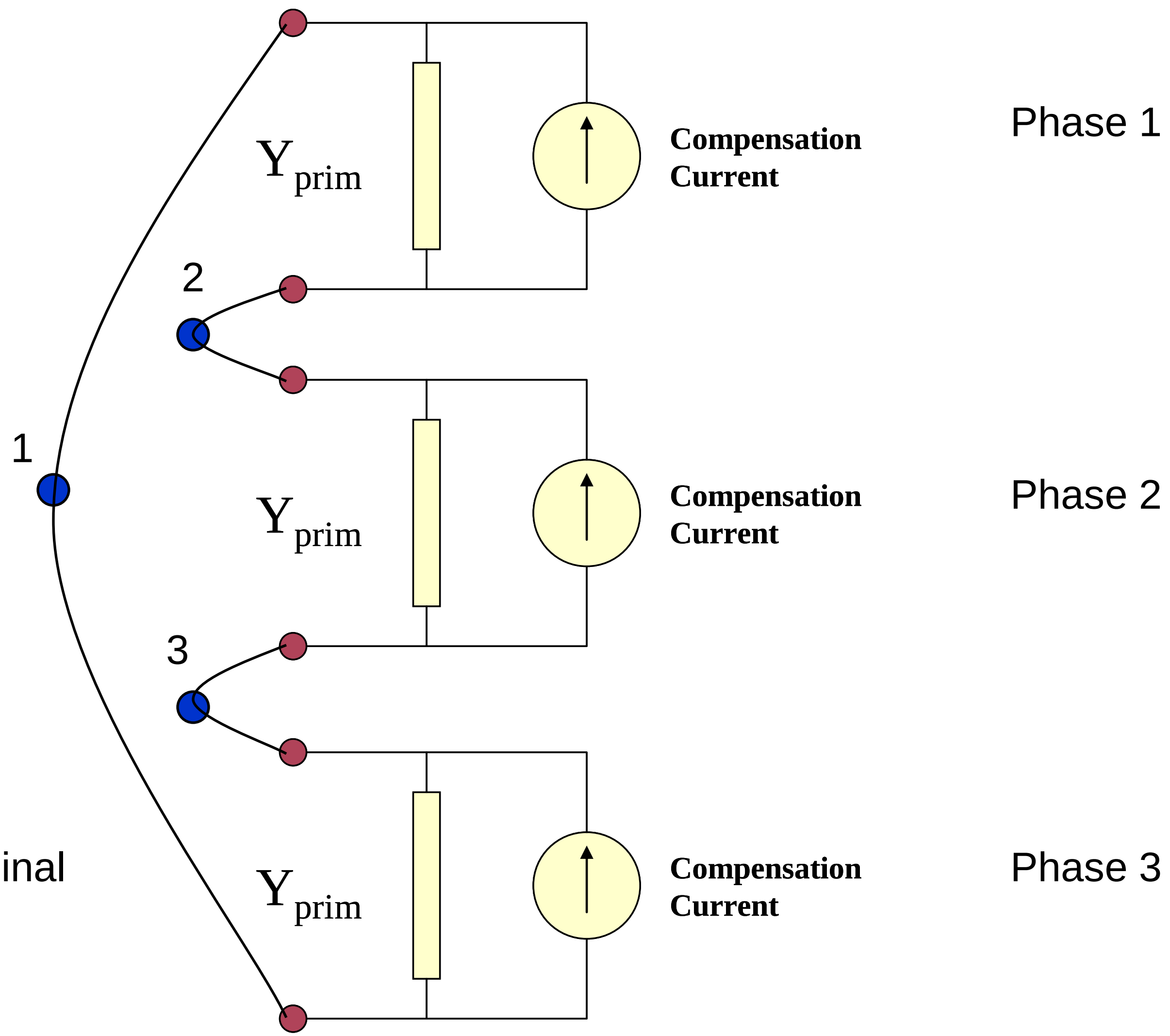


(One-Line Diagram)

Load - 3-phase Y connected

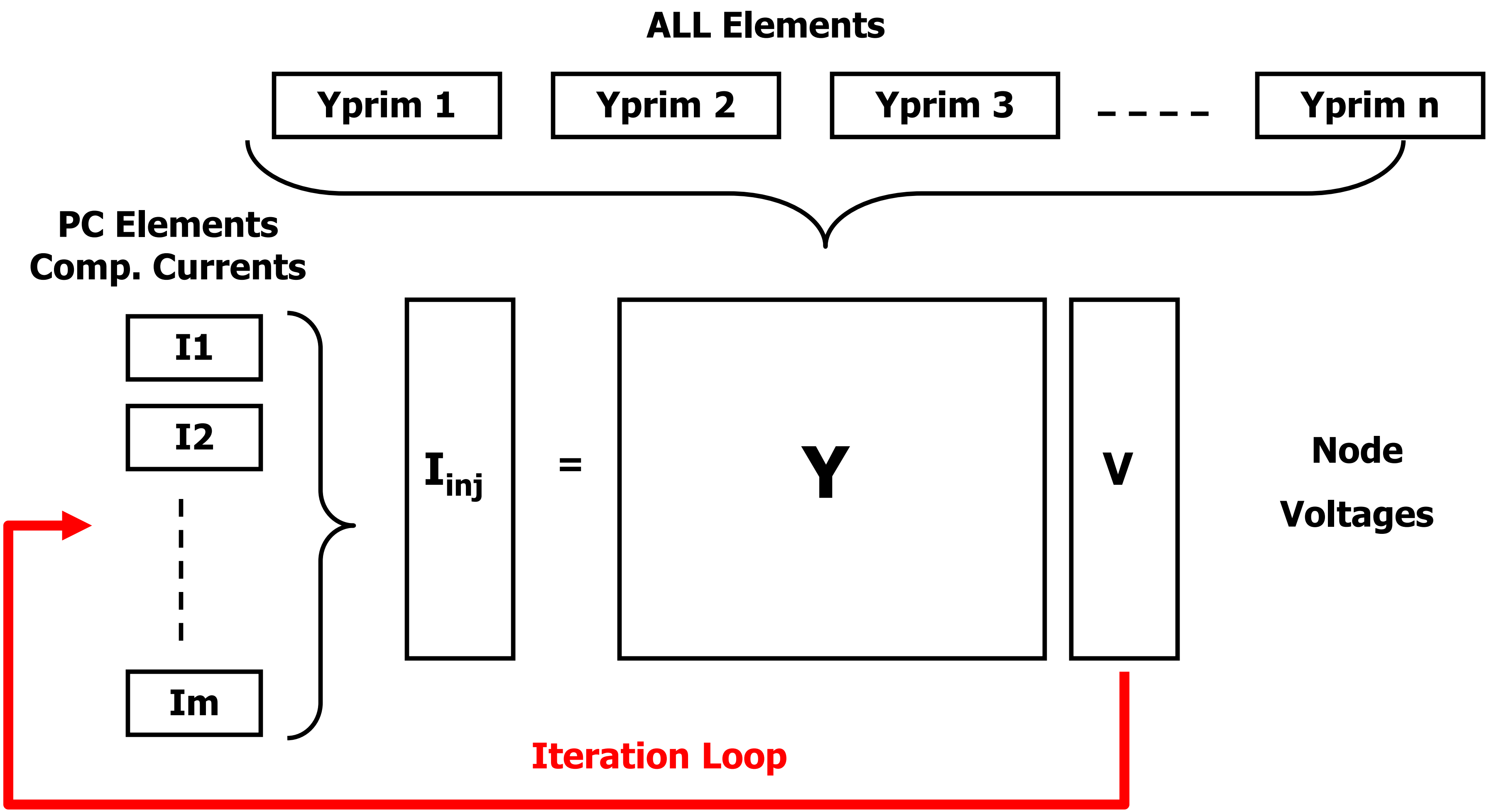


Load - 3-phase Delta connected



3 Conductors/Terminal

Putting it All Together



Solution Speed

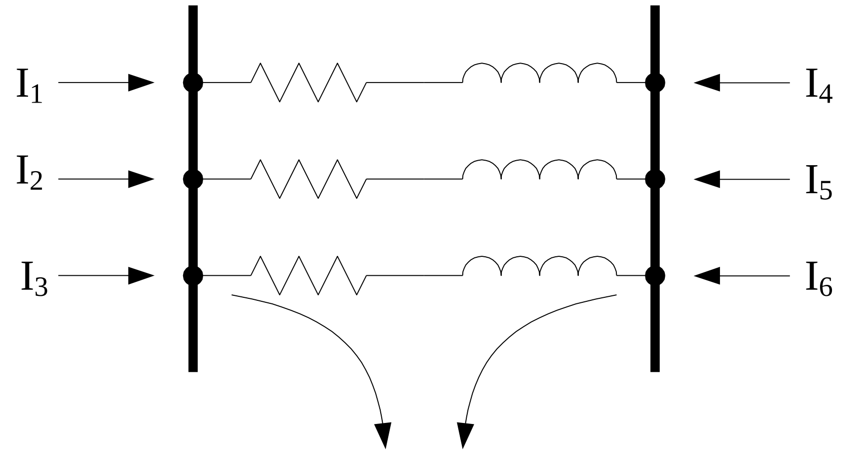
- Distribution systems generally converge quite rapidly with this method.
- The OpenDSS program seems to be on par with the faster commercial programs – or faster
- It is set up to run annual simulations easily
  - Our recommendation:
    - *Err on the side of running more power flow simulations*
    - That is, don't worry about the solution time until it proves to be a problem
    - That reveals more information about the problem.

How Do You Get Currents and Power If You Only Solve for Node Voltages?

- One thing that troubles some users who are accustomed to other ways of solving power flows is how the branch currents (and powers) are determined when only the Node voltages and Compensation currents are known.
- If the Y matrix is properly formed, and convergence is achieved, the currents will be correct (obey Kirchoff's law at nodes)
- Currents and powers are determined by post processing
- Power criteria are matched by converging with the specified Load criteria
  - i.e., compensation currents

ELECTRIC POWER RESEARCH INSTITUTE

Computing Currents in a Branch



$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \end{bmatrix} = \underset{(6 \times 6)}{\left[Y_{prim} \right]} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{bmatrix}$$

Yprim

- You can obtain the Primitive Y matrix for each element a number of ways (after a Solve)
- Dump command
  - **Dump class.name debug**
    - Or, Dump Class.* debug
- Script
  - **Show Yprim** ! Of active element
  - **Export Yprims** ! All Yprims
- COM Interface
  - **V = DSSCircuit.ActiveElement.NumPhases**

Possible Source of Error!

- If the branch is extremely short (impedance is very low), currents may be incorrectly computed
  - Convergence tolerance is generally 0.0001 pu
  - Voltage solution will be correct enough
- 64-bit math is used throughout
  - You have a fair amount of leeway
  - **However, if voltages at both ends of branch are nearly the same, you will be taking the difference between two nearly equal numbers and the multiplying it by a large number (very high conductance)**
    - **This will magnify any error**
- Do not use impractically short branches

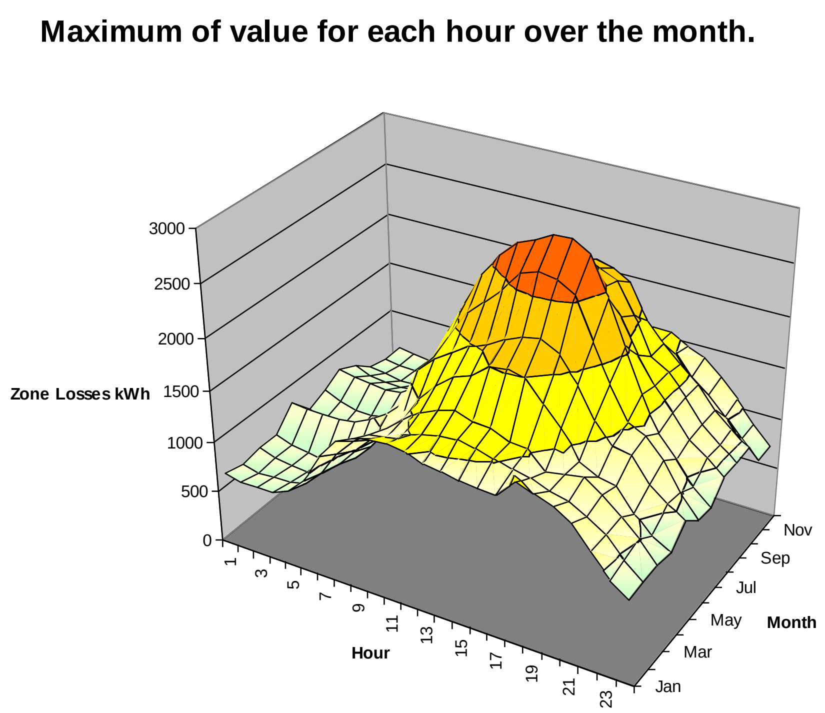
Advanced Topics

Plotting

Ways to Plot

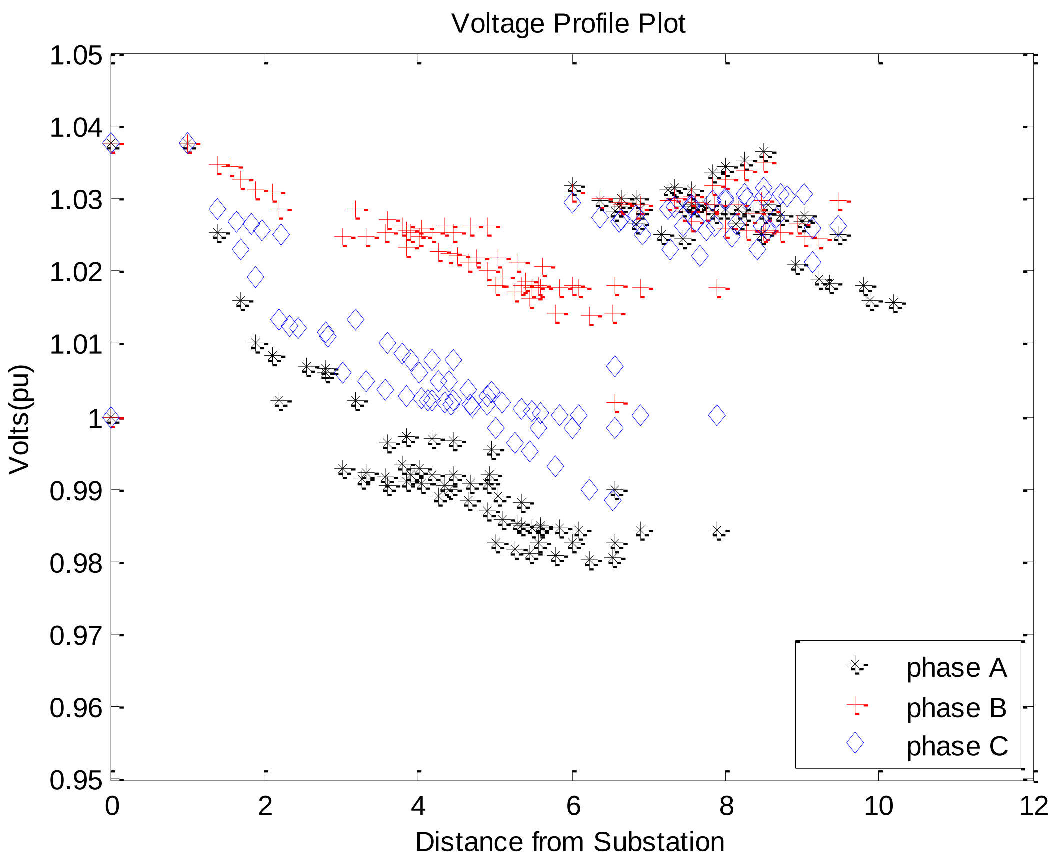
- Use the built-in plotting capabilities

- Plot in an external program, such as Excel or MATLAB



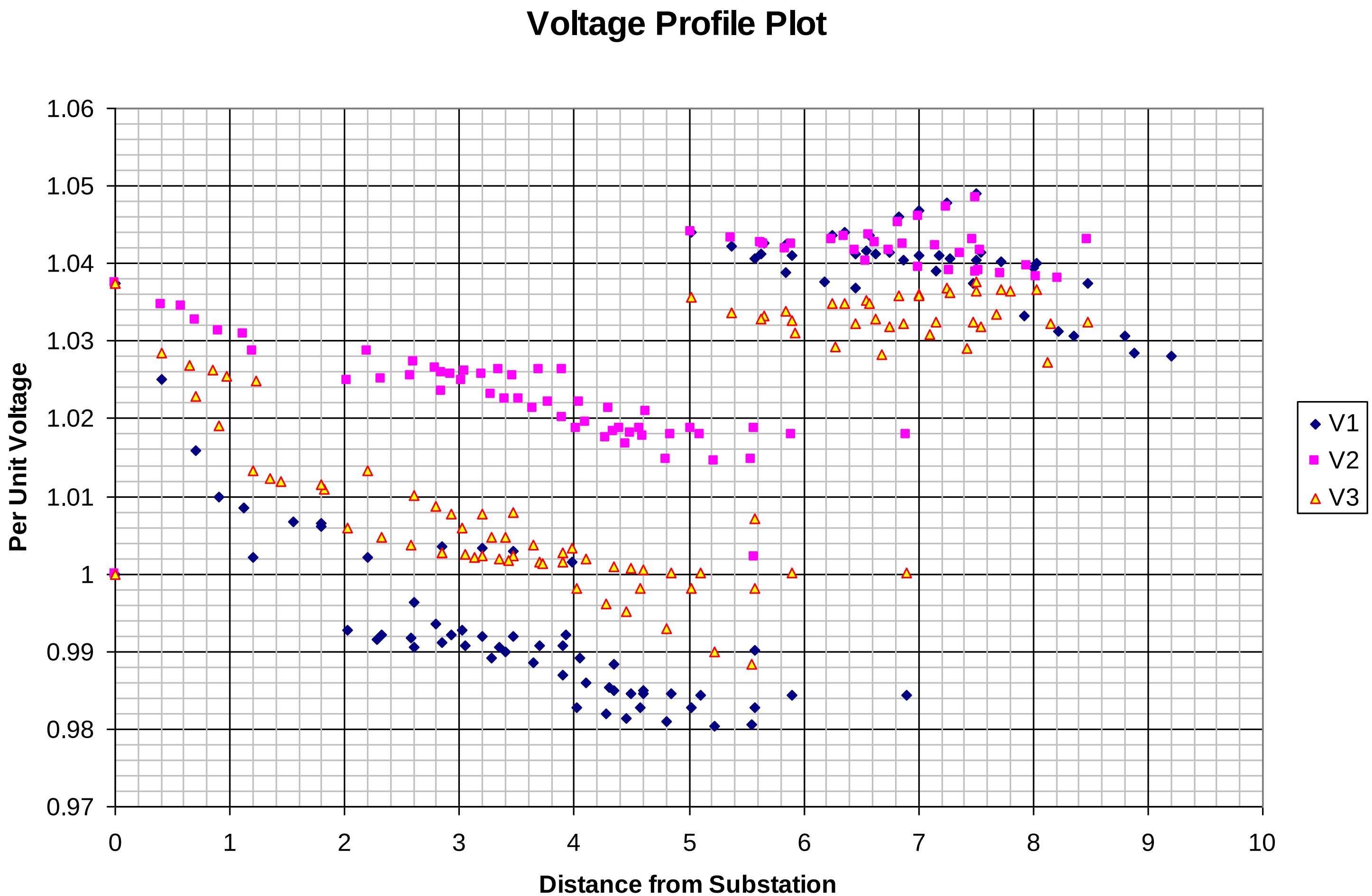
From Excel
(See Example)

From Matlab …



Voltage Profile Plot
1.05
1.04
1.03
1.02
1.01
1
0.99
0.98
0.97
0.96
0.95
Volts(pu)
0
2
4
6
8
10
12
Distance from Substation
phase A
phase B
phase C

From Excel …



Voltage Profile Plot
Per Unit Voltage
1.06
1.05
1.04
1.03
1.02
1.01
1
0.99
0.98
0.97
0
1
2
3
4
5
6
7
8
9
10
Distance from Substation
V1
V2
V3

The Plot Command

- **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)

The Plot command, cont'd

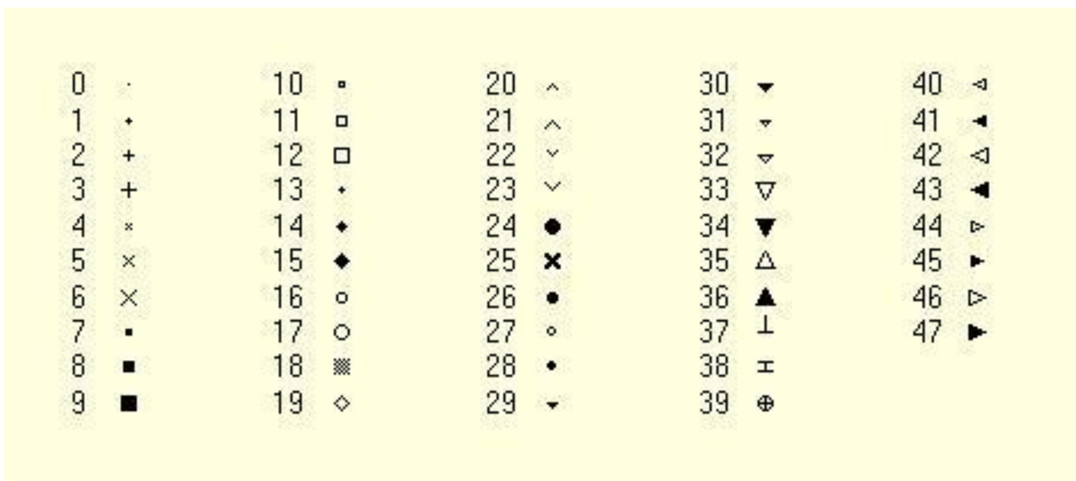
- Power and Losses in kW.
- C1 used for default color (RGB).
  - Hex Format: $00FF00000
- C2, C3 used for gradients, tri-color plots.
- Scale determined automatically if Max = 0 or not specified.

- Examples:
- Plot type=daisy quantity=power max=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

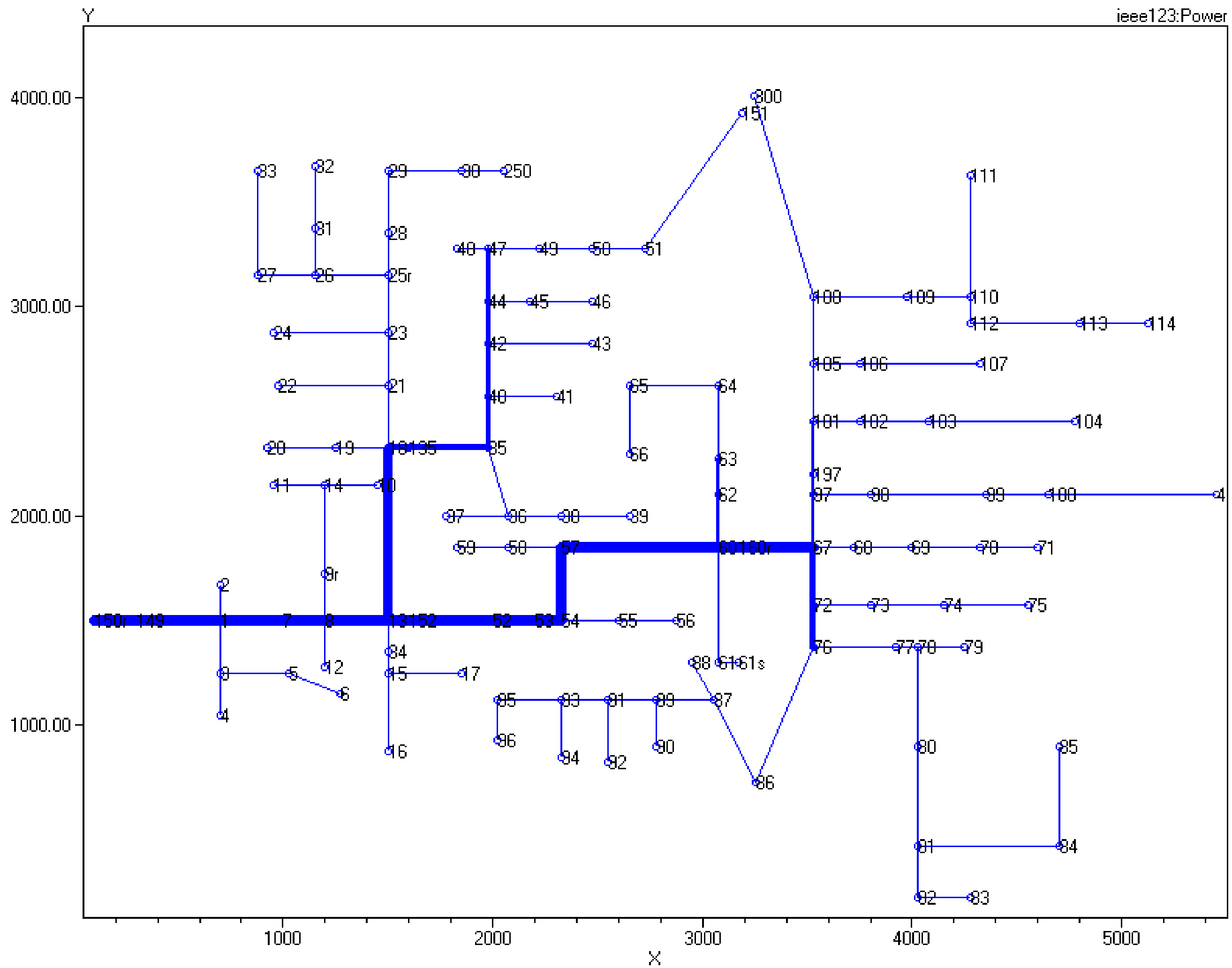
Commands/Options Associated with Plot

- AddMarker Bus=busname code=nn color=$00FF0000 size=3
- Set Nodewidth = nn
- Set MarkerCode = nn

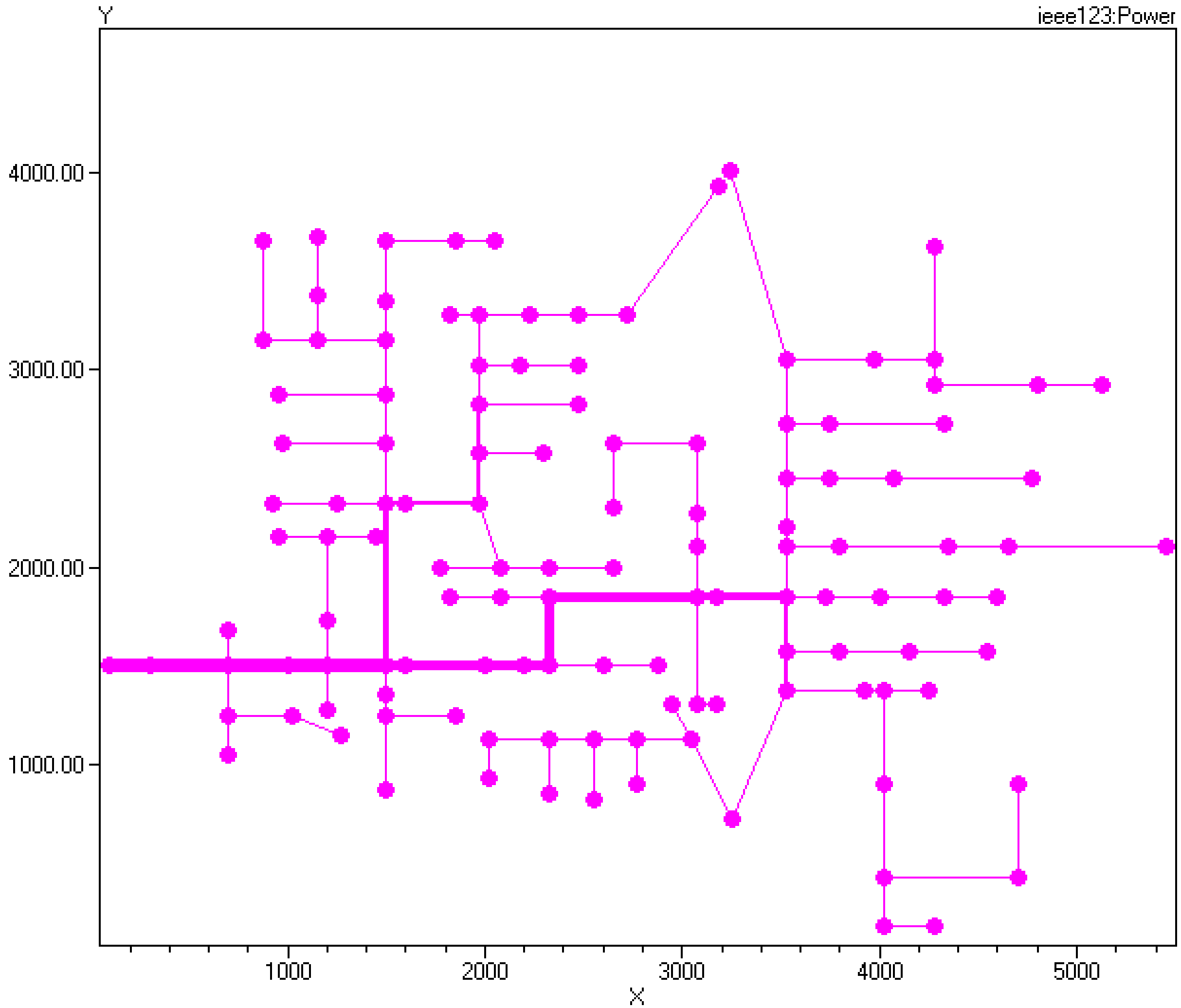
Marker Codes



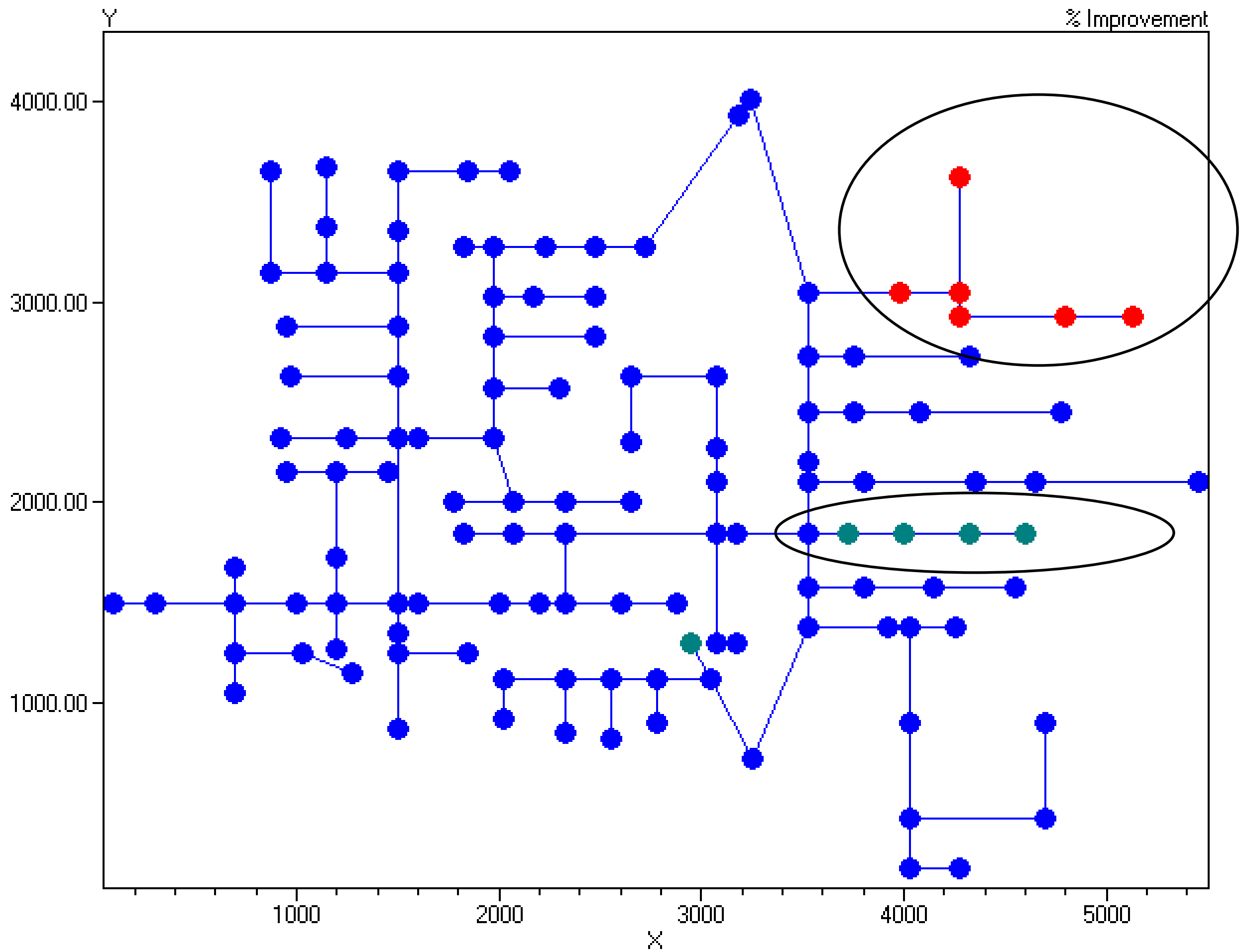
plot circuit Power max=1000 dots=y labels=y C1=$00FF0000



set nodewidth=3 markercode=24
plot circuit Power Max=2000 dots=y labels=n subs=n C1=$00FF00FF

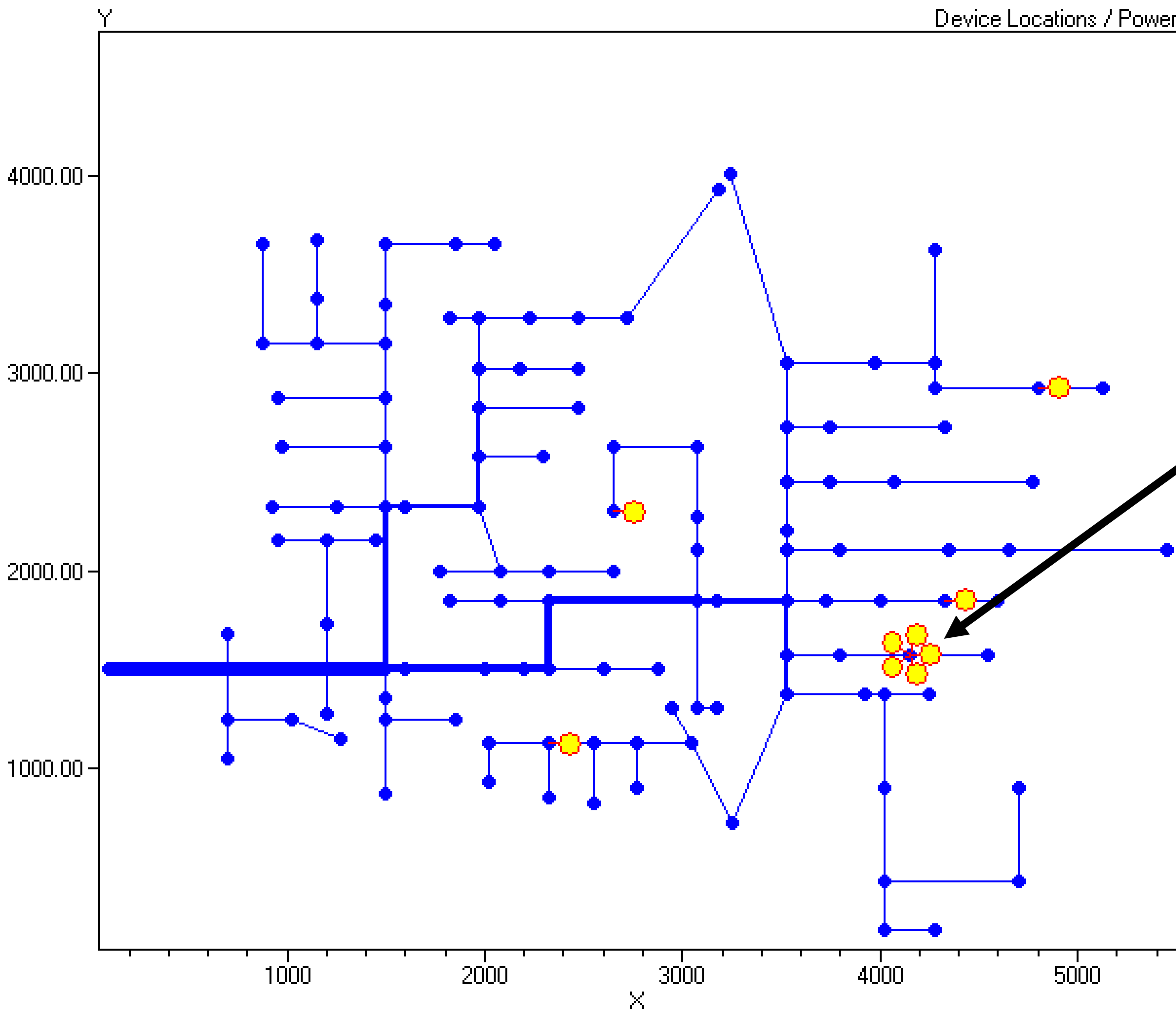


Set Genkw=100
set mode=autoadd
solve
Set nodewidth=7
plot Auto 3 dots=y labels=n subs=n C1=16711680 C2=8421376 C3=255 R3=0.95 R2=0.9



Possibly best areas for adding DG

Set nodewidth=1 daisysize=2
plot daisy Power max=2000 y n C1=$00FF0000



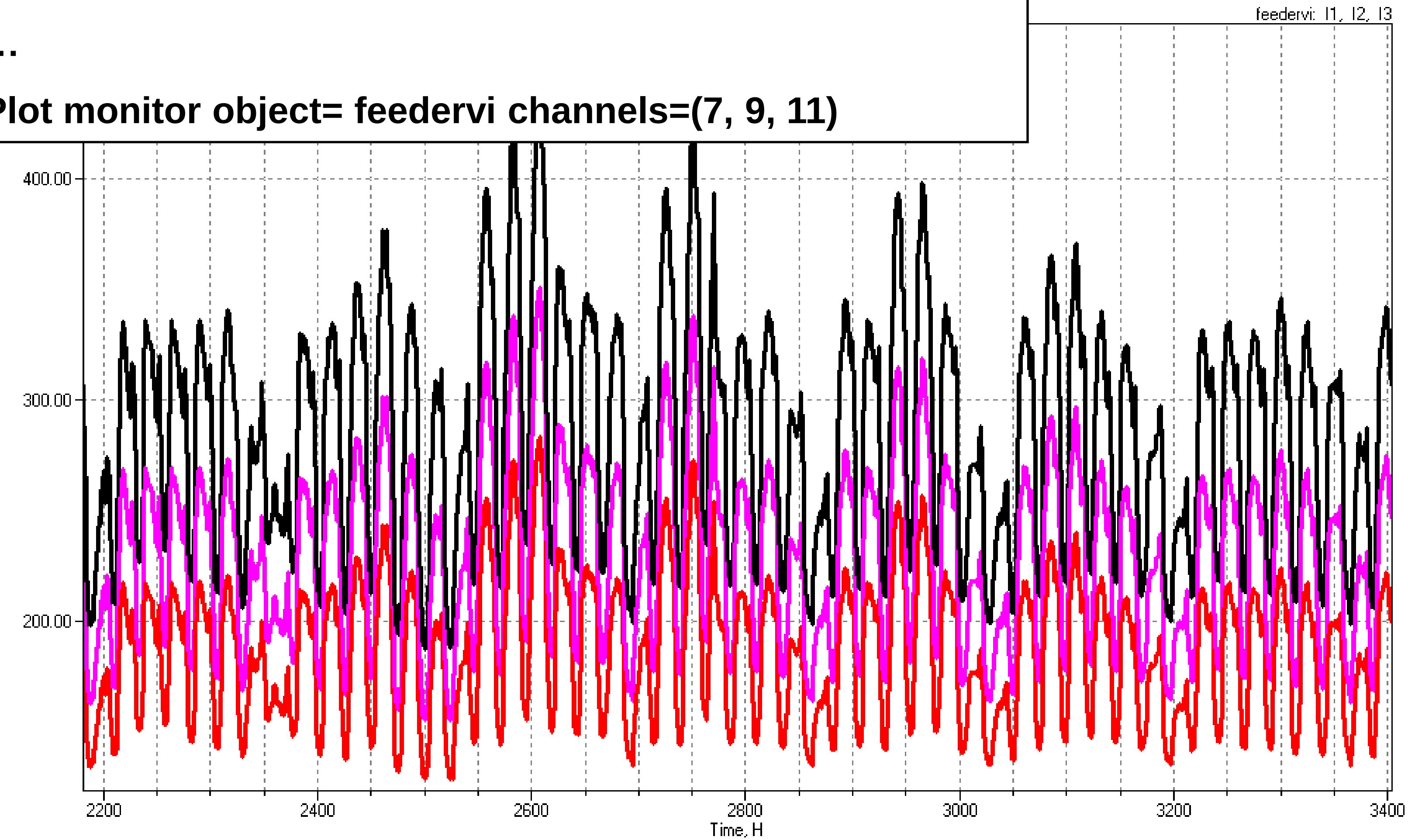
Need 500 kW here

Monitor Plot Of Feeder Currents

New Monitor.FeederVI Line.l115 1 Mode=0

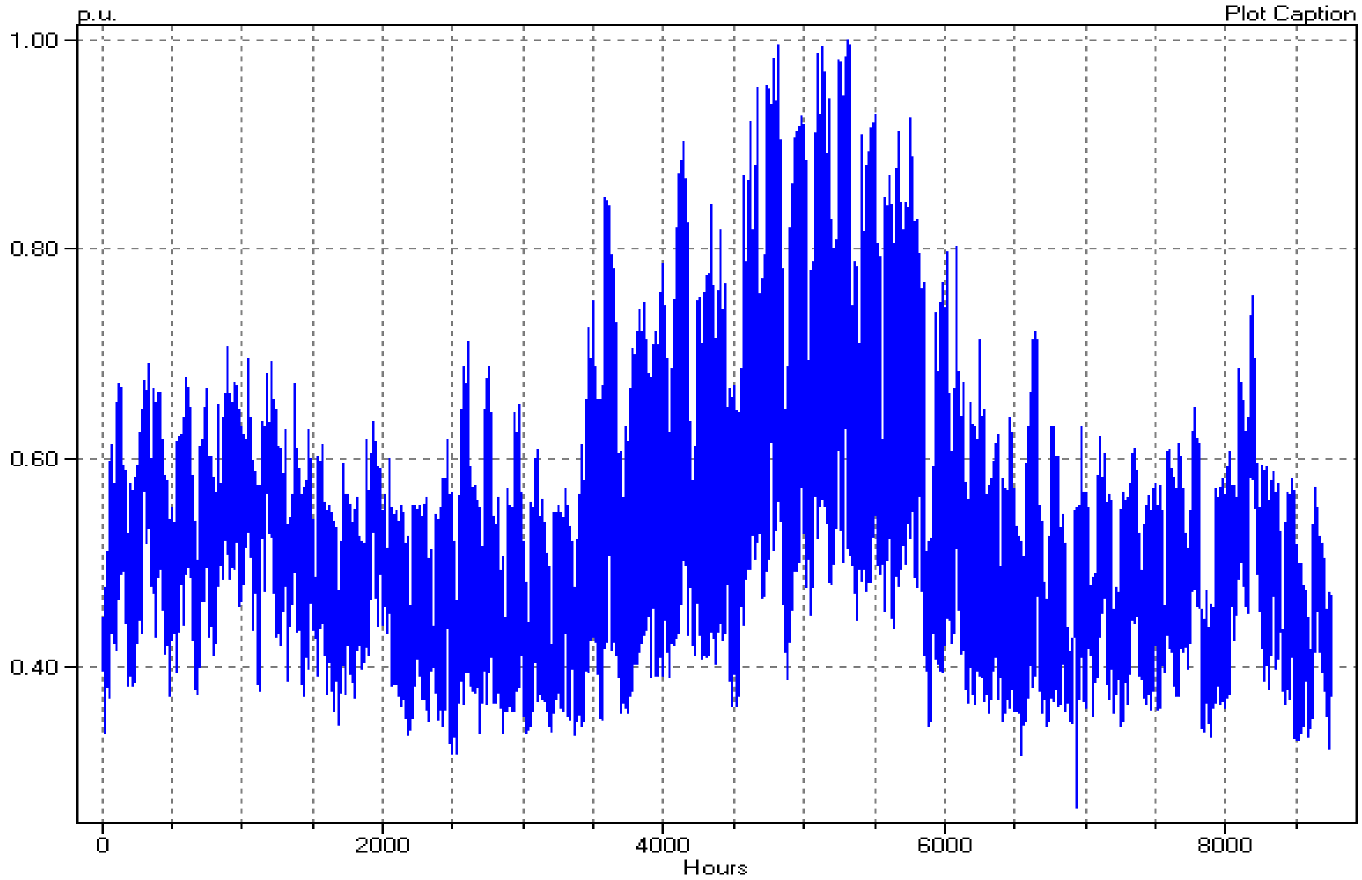
…

Plot monitor object= feedervi channels=(7, 9, 11)



LoadShape Plot

(Special plot in EXE version only)



EnergyMeter Object

EnergyMeter

- Perhaps the most complex object presently in the DSS
- Emulates an actual energy meter
  - Except it can measure things elsewhere in the meter *zone*.
- Has multiple registers
  - Registers cleared on
    - reset meters (or reset)
    - set mode = ….
    - Set year=
  - Two types: accumulators and "drag hand"

EnergyMeter Registers (Jan 2009)

1. KWh at the meter location.
2. Kvarh at the meter location.
3. Maximum kW at the meter location.
4. Maximum kVA at the meter location.
5. KWh in the meter zone.
6. Kvarh in the meter zone.
7. Maximum kW in the meter zone.
8. Maximum kVA in the meter zone.
9. Overload kWh in the meter zone, normal ratings.
10. Overload kWh in the meter zone, emergency ratings.
11. Energy Exceeding Normal (EEN) in the loads in the meter zone.
12. Unserved Energy (UE) in the loads in the meter zone.
13. Losses (kWh) in power delivery elements in the meter zone.
14. Reactive losses (kvarh) in power delivery elements in the meter zone.
15. Maximum losses (kW) in power delivery elements in the meter zone.
16. Maximum reactive losses (kvar) in power delivery elements in the meter zone.
17. Load Losses kWh. I2R Losses in power delivery elements
18. Load Losses kvarh. I2X Losses in power delivery elements
19. No Load Losses kWh in shunt elements, principally transformers.
20. No Load Losses kvarh in shunt elements.
21. Max kW Load Losses during the simulation
22. Max kW No Load Losses during the simulation
23. Line Losses: Losses in LINE elements.
24. Transformer Losses: Losses in TRANSFORMER elements.
25. Line Mode Line Losses (3X Pos and neg seq losses)
26. Zero Mode Line Losses (3X zero sequence losses)
27. 3-phase Line Losses
28. 1- and 2-phase Line Losses
29. Gen kWh
30. Gen kvarh
31. Gen Max kW
32. Gen Max kVA
33. Aux1 (used for segregating losses by voltage level)
34. Aux2
35. Aux3
36. Aux4
37. Aux5
38. Aux6
39. Aux7

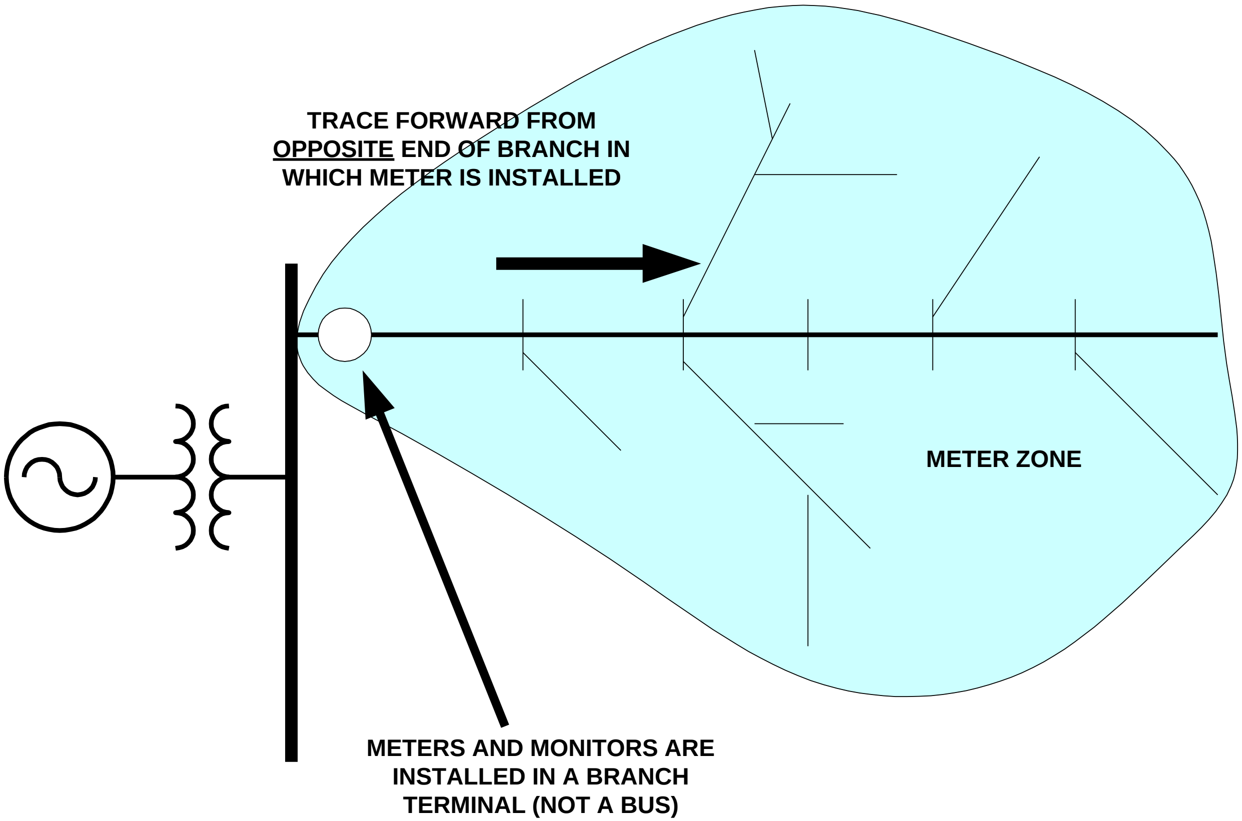
ELECTRIC POWER RESEARCH INSTITUTE

Meter Zone

- Collection of circuit elements “downline” from meter.
- Only element in DSS that knows about radial circuits
- Zone is established first time solution is executed
  - May be more time-consuming than actual solving for very large circuits.
  - Rebuilt whenever bus list is rebuilt
- EnergyMeter and Monitor objects are installed in a branch terminal
  - `New Energymeter.example Element=Line.Line1 Terminal=1`

Meter Zone, cont'd

- Zone is traced from the opposite end of the branch



Meter Zone, cont'd

- Plotting Meter Zone
  - `plot zone Power max=2000 n n object=(metername) C1=$00FF0000`
- Showing Meter Zone
  - `Show zone metername`
- Zone dump
  - `energymeter.metername.action=zonedump`
  - `Or`
  - `Edit energymeter.metername   action=zonedump`

Some Things That Require a Meter Zone

- Loss Analysis
- Excess load analysis
- Plotting zones if different colors
- Distance from substation (distance from meter)
- Reconductor Command (needs to trace back)

Monitor or Meter?

- **Monitor** measures quantities only where it is located
  - Takes a sample of quantity
  - Voltage and current (several options)
  - Powers
  - Transformer taps
  - State vars

- **EnergyMeter** measures power and integrates some, samples others
  - Samples quantities throughout its zone

Introduction to Driving the COM Server from another Application

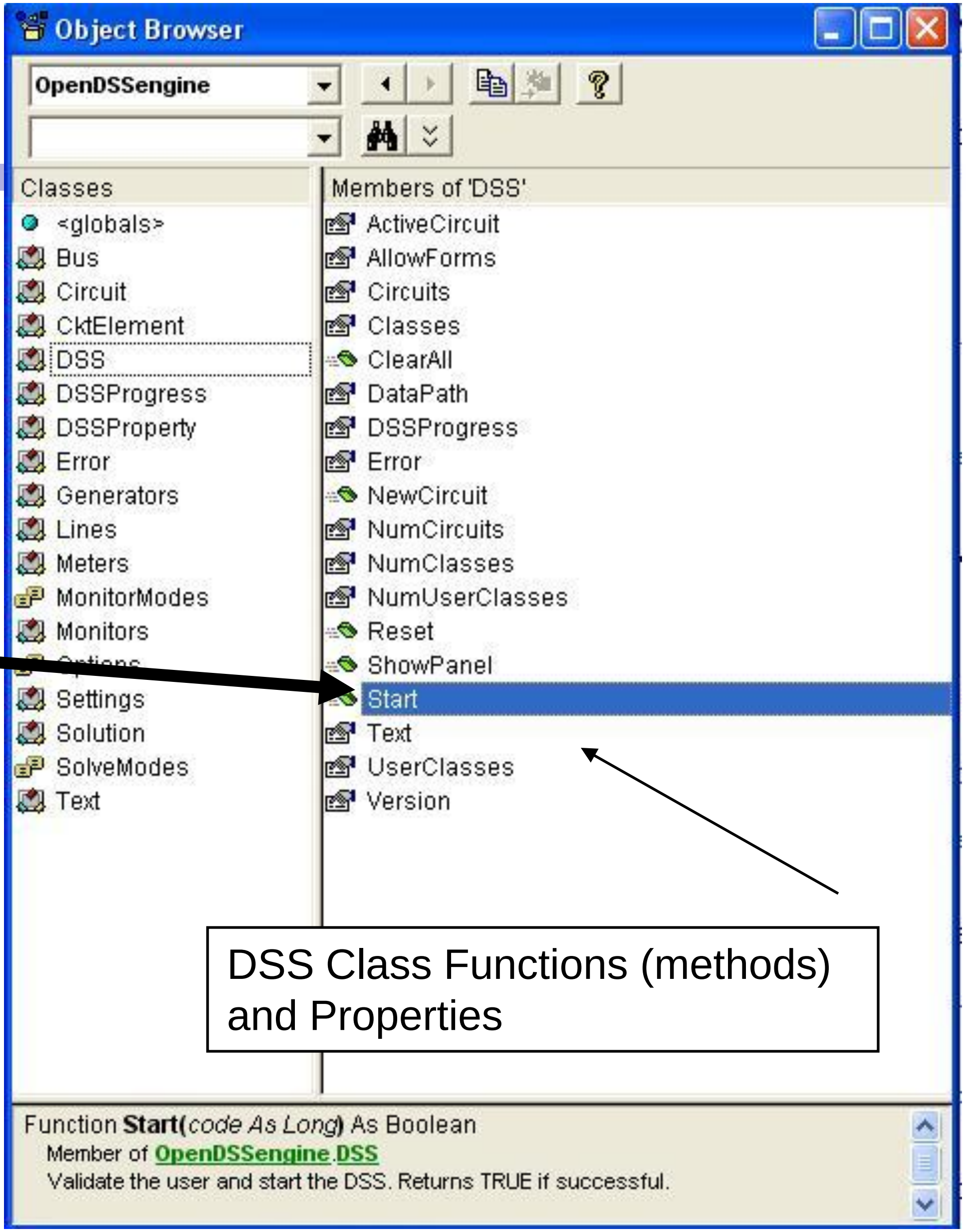
Active objects concept

- There is one registered In-Process COM interface:
  - *OpenDSSEngine.DSS*
    - That is, the DSS interface is the one you instantiate
    - The DSS interface creates all the others.
- The interfaces generally employ the idea of an **ACTIVE object**
  - Active circuit,
  - Active circuit element,
  - Active bus, etc.
  - The interfaces generally point to the active object
    - To work with another object, change the active object.

DSS Interface

This interface is instantiated upon loading OpenDSSEngine.DSS and then instantiates all other interfaces

Call the Start(0) method to initialize the DSS



Instantiate the DSS Interface and Attempt Start

```
Public Sub StartDSS()

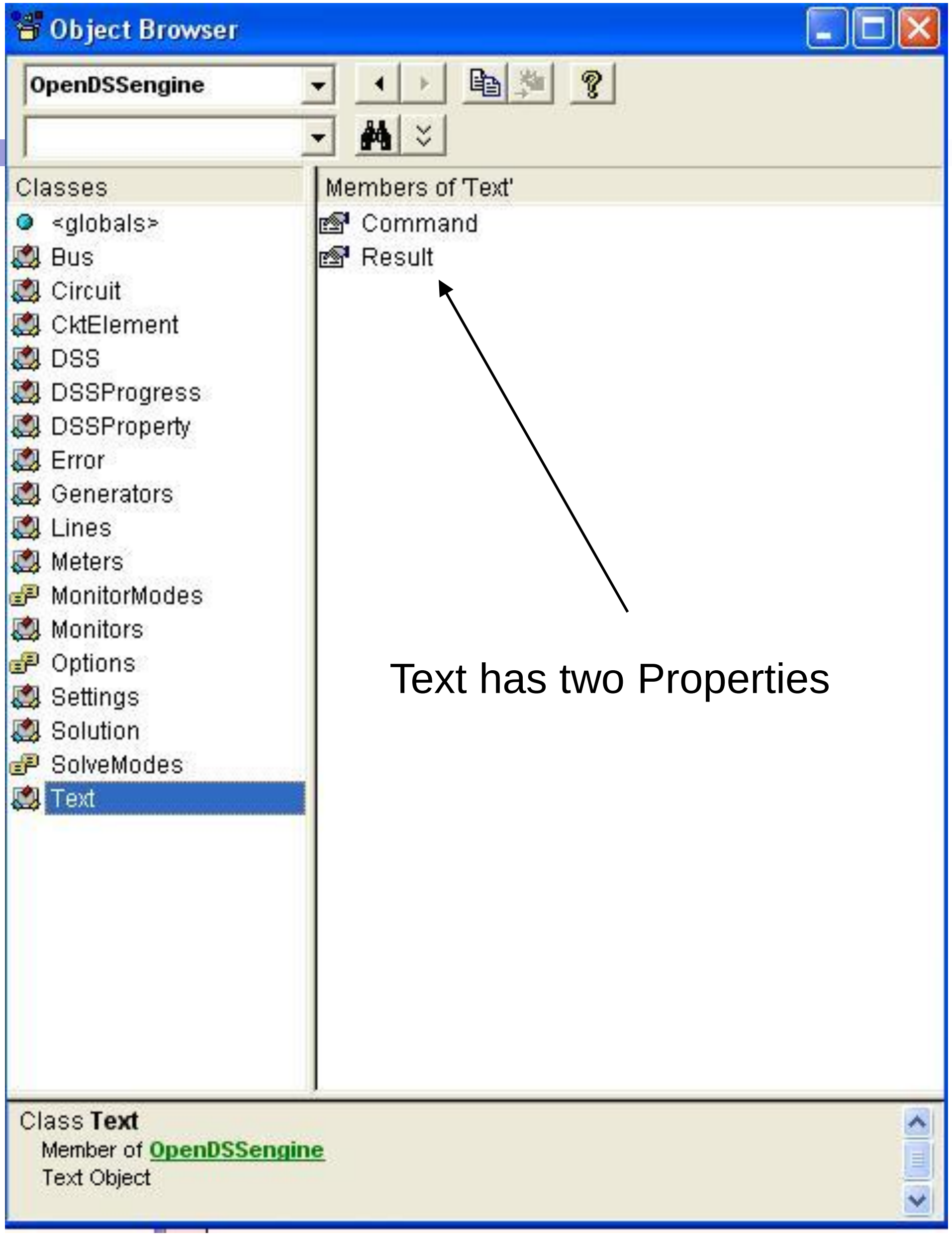
' Create a new instance of the DSS
    Set DSSobj = New OpenDSSengine.DSS
' Start the DSS
    If Not DSSobj.Start(0) Then
        MsgBox "DSS Failed to Start"
    Else
        MsgBox "DSS Started successfully"
      ' Assign a variable to the Text interface for easier access
        Set DSSText = DSSobj.Text
    End If


End Sub
```

COM Interface

Interfaces as Exposed by VBA Object Browser in MS Excel

Text interface is simplest



Text has two Properties

Assign a Variable to the Text Interface

```
Public Sub StartDSS()

' Create a new instance of the DSS
    Set DSSobj = New OpenDSSengine.DSS
' Start the DSS
    If Not DSSobj.Start(0) Then
        MsgBox "DSS Failed to Start"
    Else
        MsgBox "DSS Started successfully"
      ' Assign a variable to the Text interface for easier access
        Set DSSText = DSSobj.Text
    End If


End Sub
```


EPRI ELECTRIC POWER RESEARCH INSTITUTE

Now Use the Text Interface …

- You can issue any of the DSS script commands from the Text interface

```
` Always a good idea to clear the DSS when loading a new circuit
    DSSText.Command = "clear"
' Compile the script in the file listed under "fname" cell on the main form
    DSSText.Command = "compile " + fname
` Set regulator tap change limits for IEEE 123 bus test case
With DSSText
     .Command = "RegControl.creg1a.maxtapchange=1  Delay=15  !Allow only one tap change per solution.
   This one moves first"
     .Command = "RegControl.creg2a.maxtapchange=1  Delay=30  !Allow only one tap change per solution"
     .Command = "RegControl.creg3a.maxtapchange=1  Delay=30  !Allow only one tap change per solution"
     .Command = "RegControl.creg4a.maxtapchange=1  Delay=30  !Allow only one tap change per solution"
     .Command = "RegControl.creg3c.maxtapchange=1  Delay=30  !Allow only one tap change per solution"
     .Command = "RegControl.creg4b.maxtapchange=1  Delay=30  !Allow only one tap change per solution"
     .Command = "RegControl.creg4c.maxtapchange=1  Delay=30  !Allow only one tap change per solution"
     .Command = "Set MaxControlIter=30"
  End With
```

Result Property

- The Result property is a Read Only property that contains any result messages the most recent command may have issued.
  - Error messages
  - Requested values

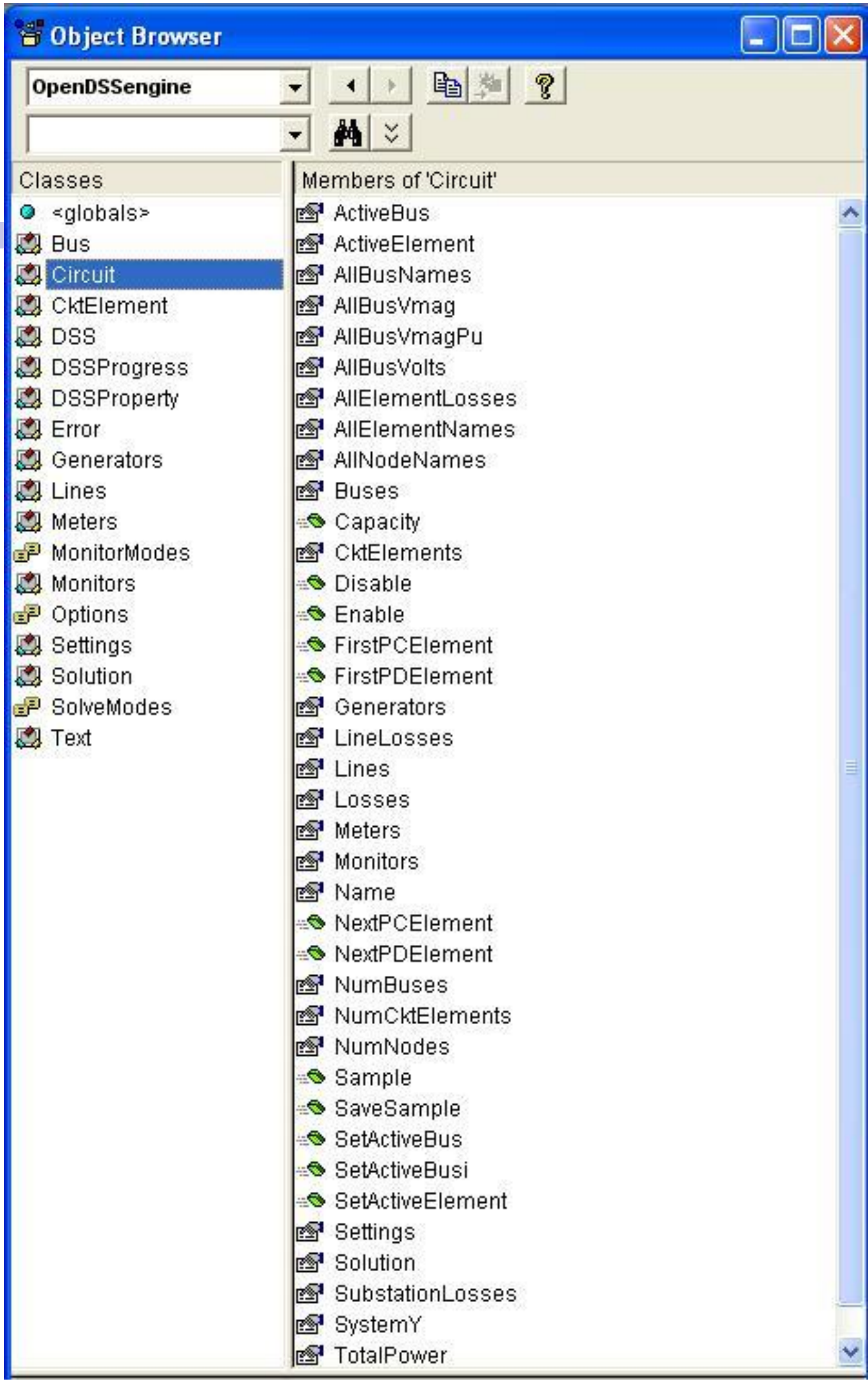
```
‘ Example: Query line length
DSSText.Command = “? Line.L1.Length”
S = DSSText.Result       ‘ Get the answer
MsgBox S           ‘ Display the answer
```


EPRI | ELECTRIC POWER RESEARCH INSTITUTE

Circuit Interface

This interface is used to

1) Get many of the results for the most recent solution of the circuit
2) Select individual circuit elements in a variety of ways
3) Select the active bus
4) Enable/Disable circuit elements



Circuit Interface

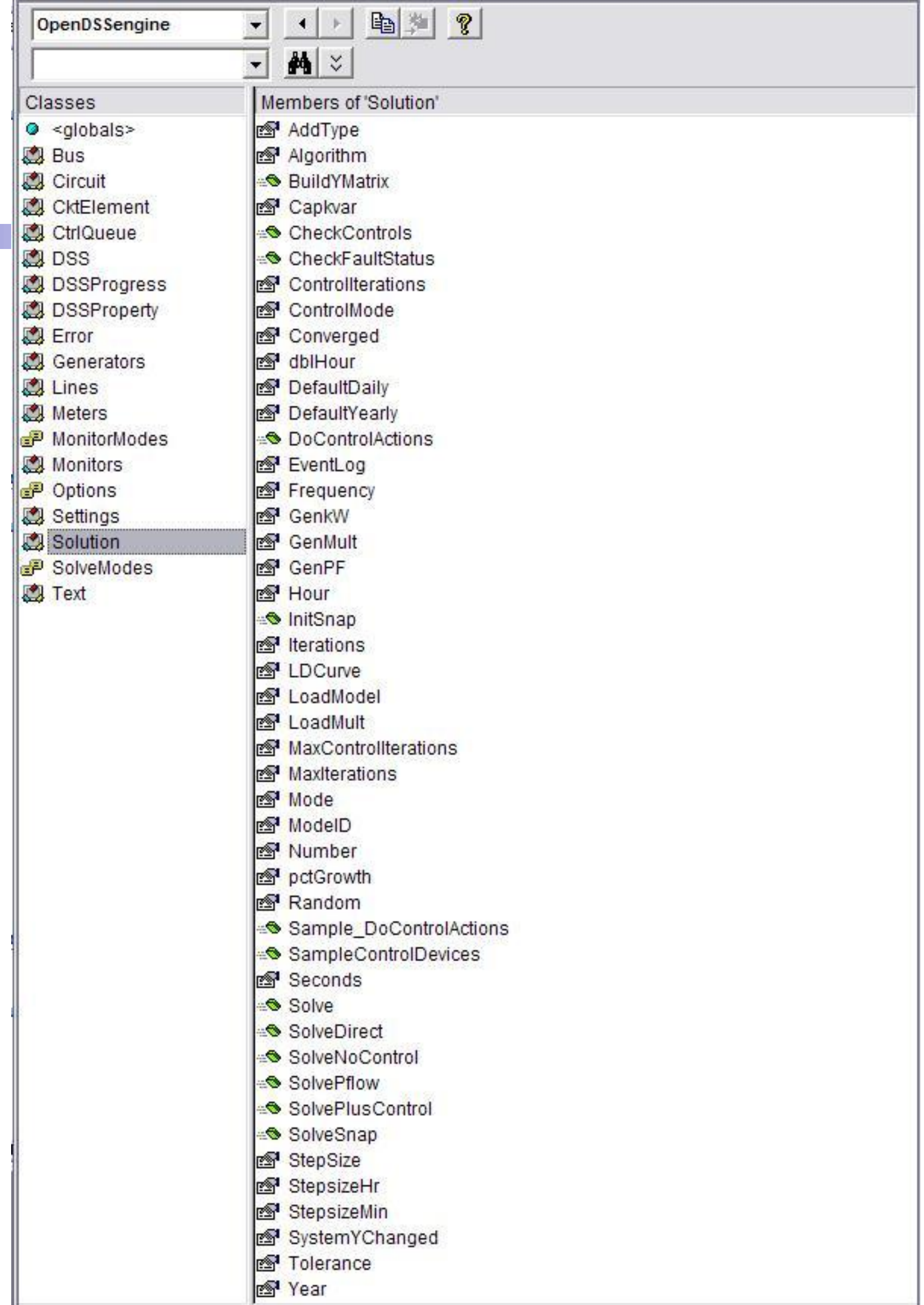
Since the Circuit interface is used often, it is recommended that a special variable be assigned to it:

```
Public DSSCircuit As OpenDSSengine.Circuit
…
DSSText.Command = "Compile xxxx.dss"
Set DSSCircuit = DSSobj.ActiveCircuit
DSSCircuit.Solution.Solve
…  ' Retrieving array quantities into variants
V = DSSCircuit.AllBusVmagPu
VL =DSSCircuit.AllElementLosses
```

Solution Interface

The Solution Interface is used to

1) Execute a solution
2) Set the solution mode
3) Set solution parameters (iterations, control iterations, etc.)
4) Set the time and time step size



Solution Interface

Assuming the existence of a DSSCircuit variable referencing the Circuit interface

```
Set DSSSolution = DSSCircuit.Solution
With DSSSolution
…
    .LoadModel=dssAdmittance
    .dblHour = 750.75
    .solve

End With
```

Use the With statement in VBA to simplify coding

CktElement Interface

This interface provides specific values of the Active Circuit Element

Some values are returned as variant arrays

```
V = DSSCircuit.ActiveElement.Powers

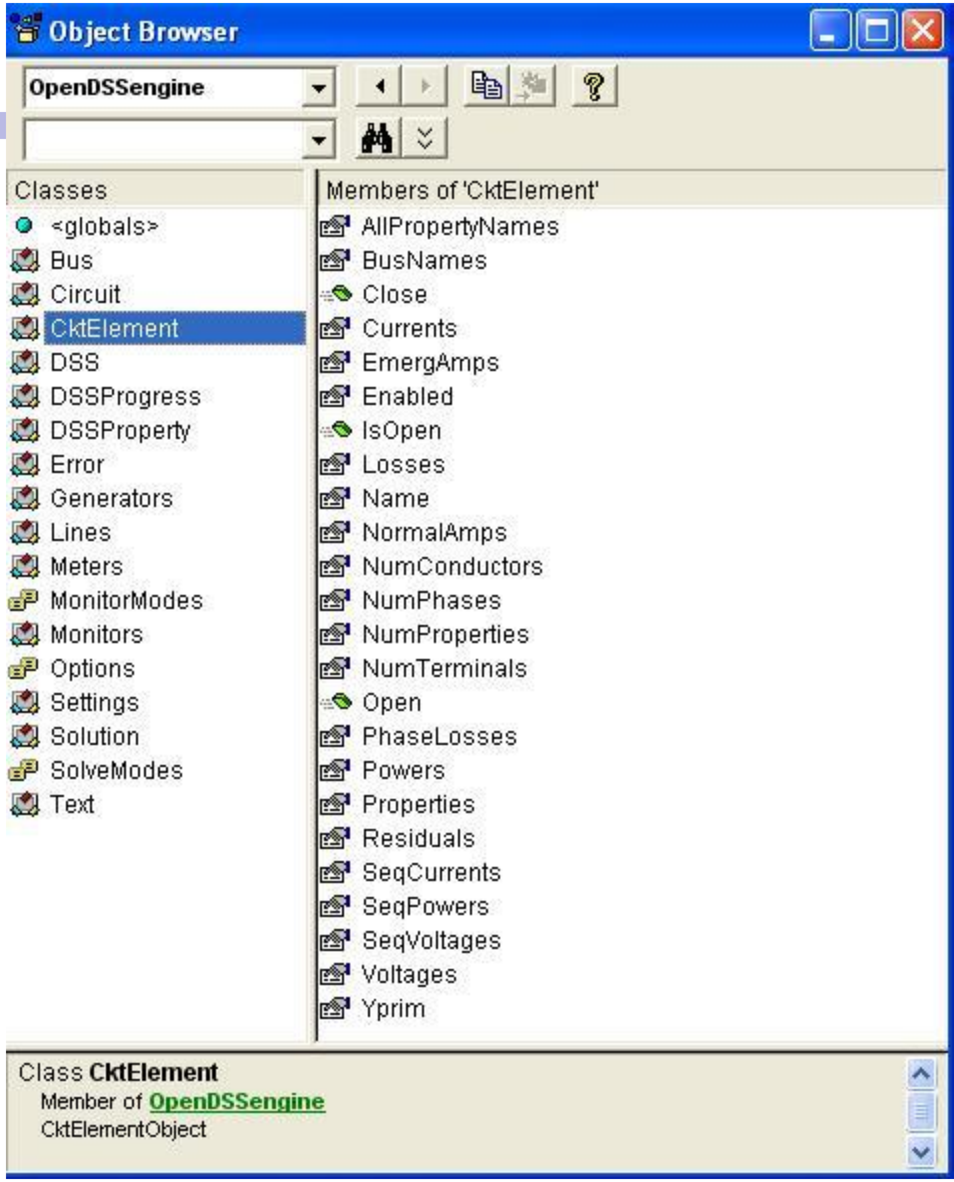
V = DSSCircuit.ActiveElement.seqCurrents

V = DSSCircuit.ActiveElement.Yprim
```

Other values are scalars

```
Name = DSSCircuit.ActiveElement.Name

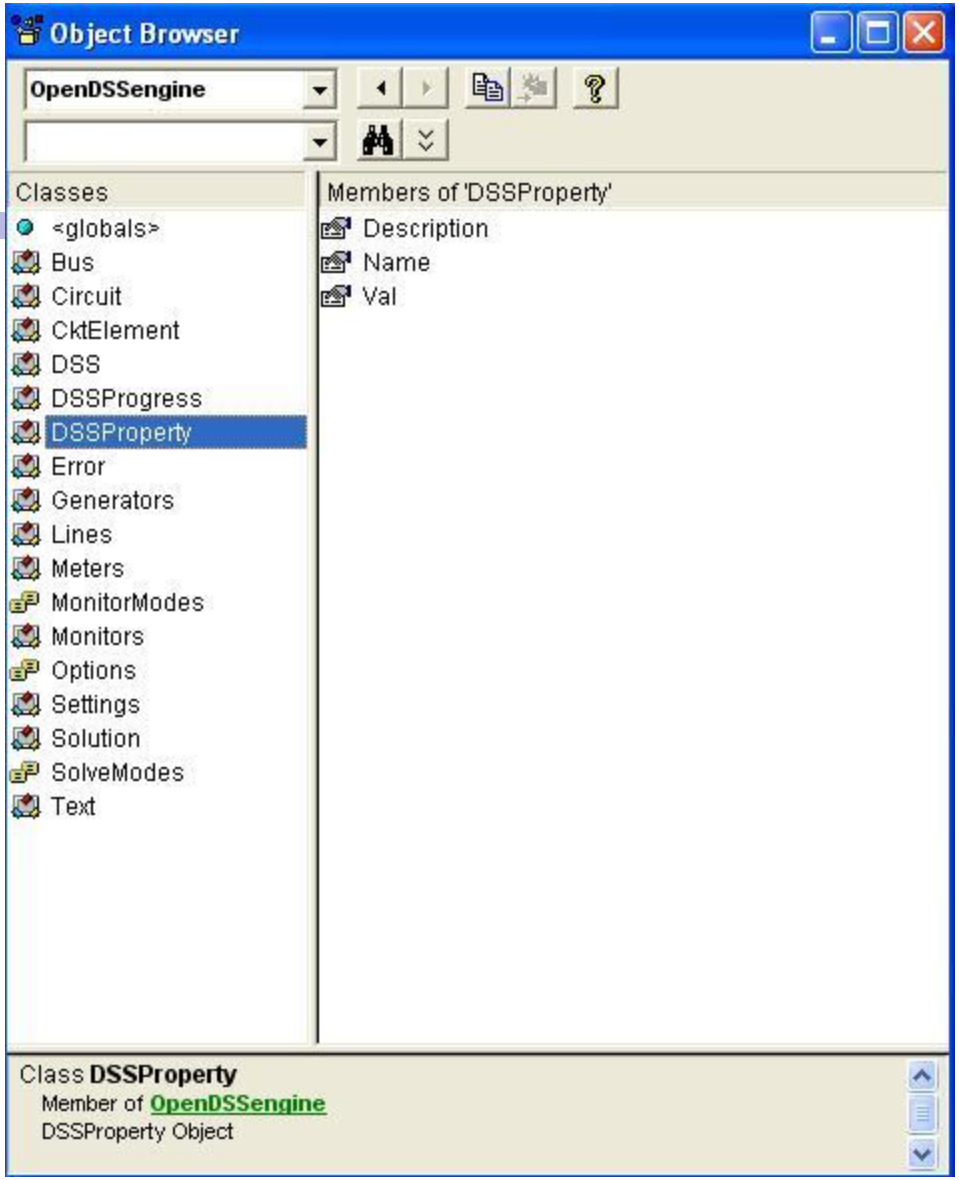
Nph = DSSCircuit.ActiveElement.NumPhases
```



Properties Interface

This interface gives access to a String value of each public property of the active element

“Val” is a read/write property



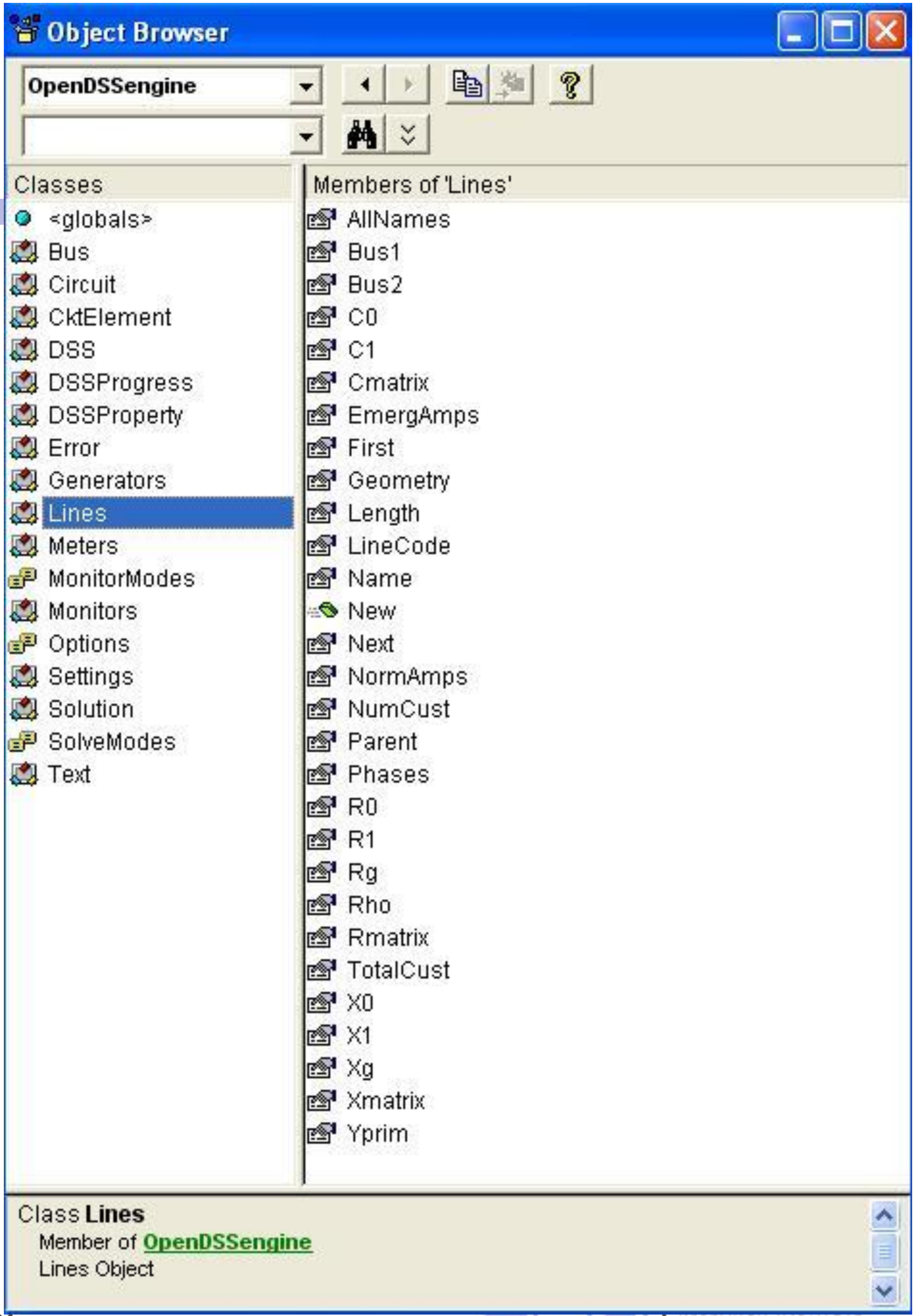
Properties Interface

```
With DSSCircuit.ActiveElement
    ` Get all the property names
    VS = .AllPropertyNames
    ` Get a property value by numeric index
    V = .Properties(2).Val
    ` Get same property value by name (VS is 0 based)
    V = .Properties(VS(1)).Val
    ` Set Property Value by Name
    DSSCircuit.SetActiveElement("Line.L1")
    .Properties('R1').Val = ".068"
End With
```

The last two statements are equivalent to:

DSSText.Command = "Line.L1.R1=.068"

Lines Interface



This interface is provided to iterate through all the lines in the circuit and change the most common properties of Lines.

Example: Setting all LineCodes to a Value

```
Set DSSLines = DSSCircuit.Lines
.  .  .
iL = DSSLines.First  'sets active
Do While iL>0
 DSSLines.LineCode = MyNewLineCode
 iL = DSSLines.Next  ' get next line
Loop
```


EPRI ELECTRIC POWER RESEARCH INSTITUTE

VBA Example

```
Option Explicit

Public DSSobj As OpenDSSengine.DSS
Public DSSText As OpenDSSengine.Text
Public DSSCircuit As OpenDSSengine.Circuit

Public Sub StartDSS()

' Create a new instance of the DSS
   Set DSSobj = New OpenDSSengine.DSS

' Assign a variable to the Text interface for easier
  access
   Set DSSText = DSSobj.Text

' Start the DSS
   If Not DSSobj.Start(0) Then MsgBox "DSS
  Failed to Start"

End Sub
```

Define some public variables that are used throughout the project

This routine instantiates the DSS and starts it. It is also a good idea at this time to assign the text interface variable.

VBA Example

```
Public Sub LoadCircuit(fname As String)

' Always a good idea to clear the DSS when loading a new
  circuit
   DSSText.Command = "clear"

' Compile the script in the file listed under "fname" cell on the
  main form
   DSSText.Command = "compile " + fname

' The Compile command sets the current directory the that of
  the file
' Thats where all the result files will end up.

' Assign a variable to the Circuit interface for easier access
   Set DSSCircuit = DSSobj.ActiveCircuit

End Sub
```

This subroutine loads the circuit from the base script files using the Compile command through the Text interface. “fname” is a string contains the name of the master file.

There is an active circuit now, so assign the DSSCircuit variable.

EPRI | ELECTRIC POWER RESEARCH INSTITUTE

VBA Example

```
Public Sub LoadSeqVoltages()

' This Sub loads the sequence voltages onto Sheet1 starting in Row 2

    Dim DSSBus As OpenDSSengine.Bus
    Dim iRow As Long, iCol As Long, i As Long, j As Long
    Dim V As Variant
    Dim WorkingSheet As Worksheet

    Set WorkingSheet = Sheet1   'set to Sheet1 (target sheet)

    iRow = 2
    For i = 1 To DSSCircuit.NumBuses  ' Cycle through all buses

        Set DSSBus = DSSCircuit.Buses(i)  ' Set ith bus active

    ' Bus name goes into Column 1
        WorkingSheet.Cells(iRow, 1).Value = DSSCircuit.ActiveBus.Name

    ' Load sequence voltage magnitudes of active bus into variant array
        V = DSSBus.SeqVoltages

    ' Put the variant array values into Cells
    ' Use Lbound and UBound because you don't know the actual range
        iCol = 2
        For j = LBound(V) To UBound(V)
            WorkingSheet.Cells(iRow, iCol).Value = V(j)
            iCol = iCol + 1
        Next j
       iRow = iRow + 1
    Next i

End Sub
```

This Sub puts the sequence voltage onto a spreadsheet

Define a variable for the Bus interface

Define a variant to pick up the arrays

Cycle through all the buses

Get the bus name

Get the voltages into the variant array

Put them on the spreadsheet

Running OpenDSS From Matlab

- Starting the DSS

```
 %Start up the DSS
[DSSStartOK, DSSObj, DSSText] = DSSStartup;
```

```
    function [Start,Obj,Text] = DSSStartup
    % Function for starting up the DSS
%
    %instantiate the DSS Object
    Obj = actxserver('OpenDSSEngine.DSS');
    %
    %Start the DSS.   Only needs to be executed the first time w/in a
    %Matlab session
    Start = Obj.Start(0);

    % Define the text interface to return
    Text = Obj.Text;
```

Using the DSS through the DSSText Interface from Matlab (harmonics example)

```
%Compile the DSS circuit script
DSSText.Command = 'compile master.dss';

% get an interface to the active circuit called "DSSCircuit"
DSSCircuit = DSSObj.ActiveCircuit;

%Determine which connection type for the source and call
%appropriate DSS file
switch XFMRType
case 1
 DSSText.Command = 'redirect directconnectsource.DSS';
case 2
 DSSText.Command = 'redirect deltadelta.DSS';
case 3
 DSSText.Command = 'redirect deltawye.DSS';
otherwise
 disp('Unknown source Connection Type')
end

%Set the system frequency and vsource frequency for harmonic requested
DSSText.Command = ['set frequency=(' num2str(Freq) ' 60 *)'];
DSSText.Command = ['vsource.source.frequency=(' num2str(Freq) ' 60 *)'];
```

Using the DSS through the DSSText Interface from Matlab (harmonics example) (cont'd)

```
% Vary the parameters according to a random distribution
% If more parameters need to be varied, just add them to the below
% list.  Set ParamNum to total number of parameters varied
 ParamNum = 6;   %ParamNum used for sorting/plotting
 for Case_Count = 1:Max_Cases
%Create index in the OutputData matrix to keep the cases in order
 OutputData(Case_Count,1) = Case_Count;
 % Generate random new coordinates for each conductor
 [x1 y1 x2 y2 x3 y3 geomean] = RandomGeometry(8,0.75,30);
            (... etc. etc. )
%define a new line geometry with random spacing
DSSText.Command = ['New LineGeometry.OHMOD nconds=3 nphases=3  cond=1  wire=acsr336    x=' num2str(x1) '   ' num2str(y1) '   units=ft  cond=2  wire=acsr336    x=' num2str(x2) '   ' num2str(y2) '   units=ft cond=3  wire=acsr336    x=' num2str(x3) '   ' num2str(y3) ' units=ft'];
%Solve the circuit
 DSSText.Command = 'solve';
               (etc. etc.)
```

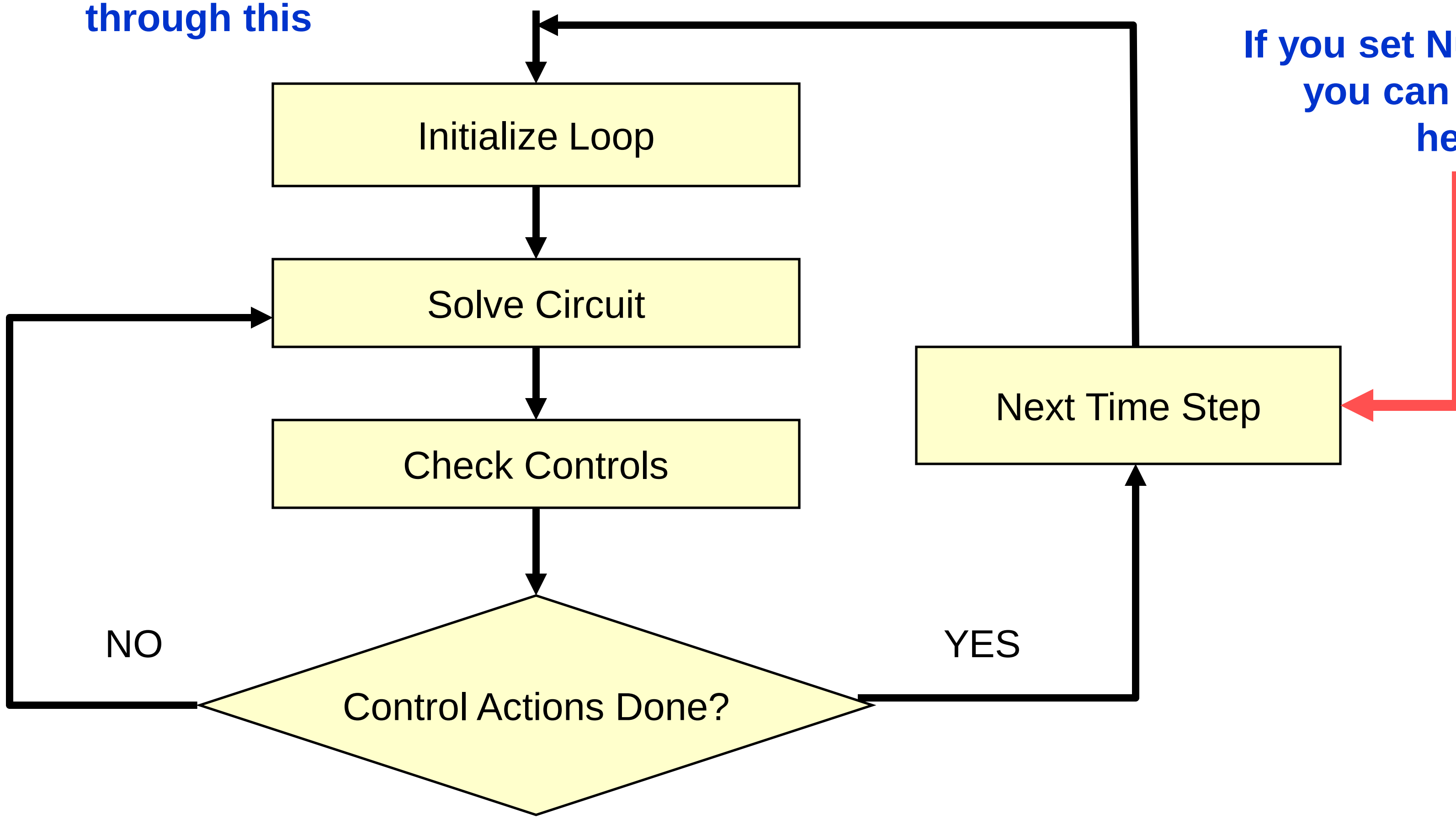
User-Written Controls

From the COM Interface

Basic Control Loop Flow Chart

You can single-step through this

If you set Number=1, you can break in here



Control Loop (Actual Pascal Code)

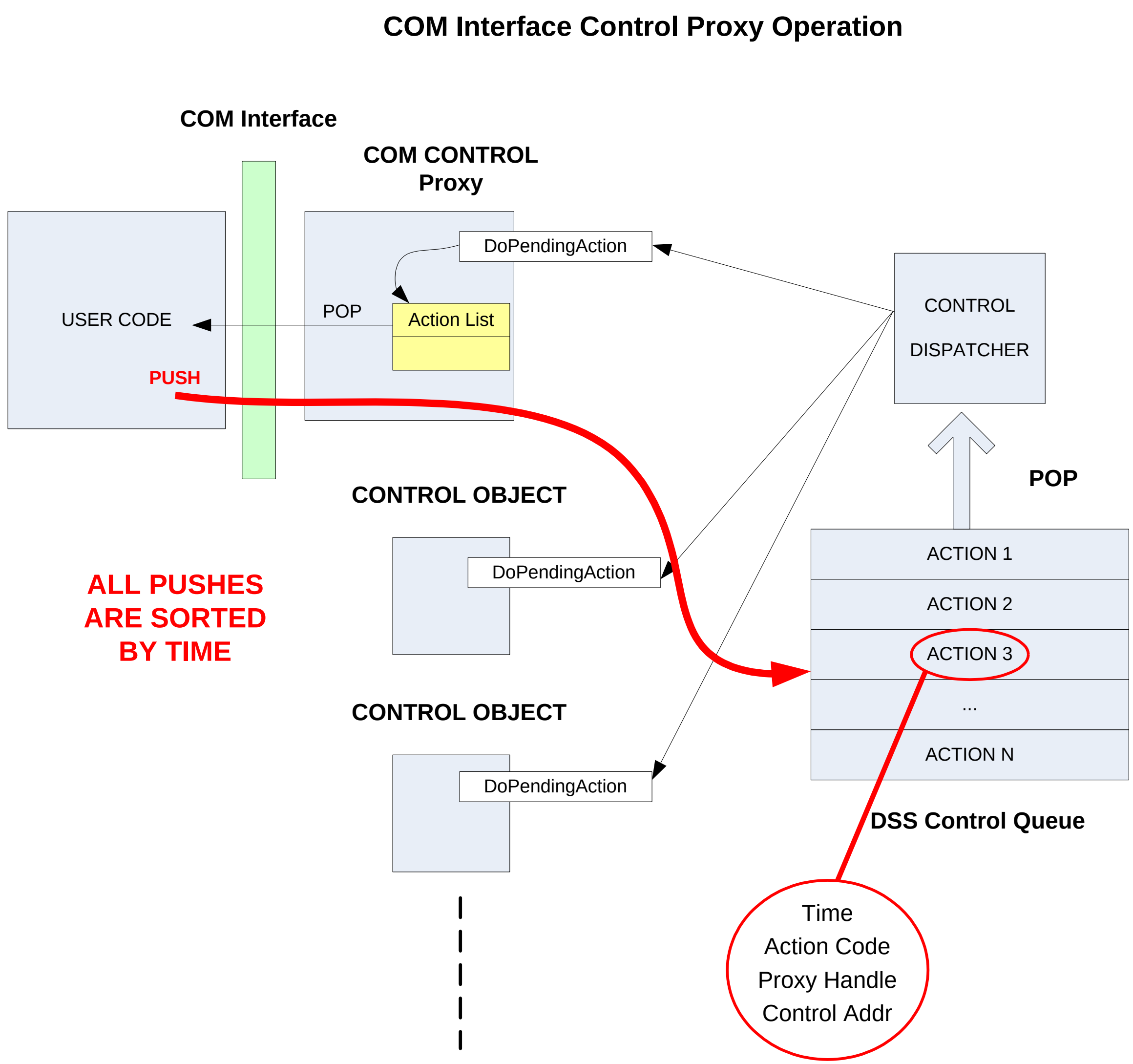
```
FUNCTION TSolutionObj.SolveSnap:Integer;  // solve for now once
VAR
   TotalIterations   :Integer;
Begin
   SnapShotInit;
   TotalIterations     := 0;
   REPEAT
       Inc(ControlIteration);
       Result := SolveCircuit;  // Do circuit solution w/o checking controls
       {Now Check controls}
       CheckControls;
       {For reporting max iterations per control iteration}
       If Iteration > MostIterationsDone  THEN MostIterationsDone := Iteration;
       TotalIterations := TotalIterations + Iteration;
   UNTIL ControlActionsDone or (ControlIteration >= MaxControlIterations);
   If Not ControlActionsDone and (ControlIteration >= MaxControlIterations) then  Begin
       DoSimpleMsg('Warning Max Control Iterations Exceeded. ' + CRLF + 'Tip: Show
  Eventlog to debug control settings.', 485);
       SolutionAbort := TRUE;   // this will stop this message in dynamic power flow modes
   End;
   If ActiveCircuit.LogEvents Then LogThisEvent('Solution Done');
   Iteration := TotalIterations;  { so that it reports a more interesting number }
End;
```


EPRI ELECTRIC POWER RESEARCH INSTITUTE

External Script and COM Interface Options

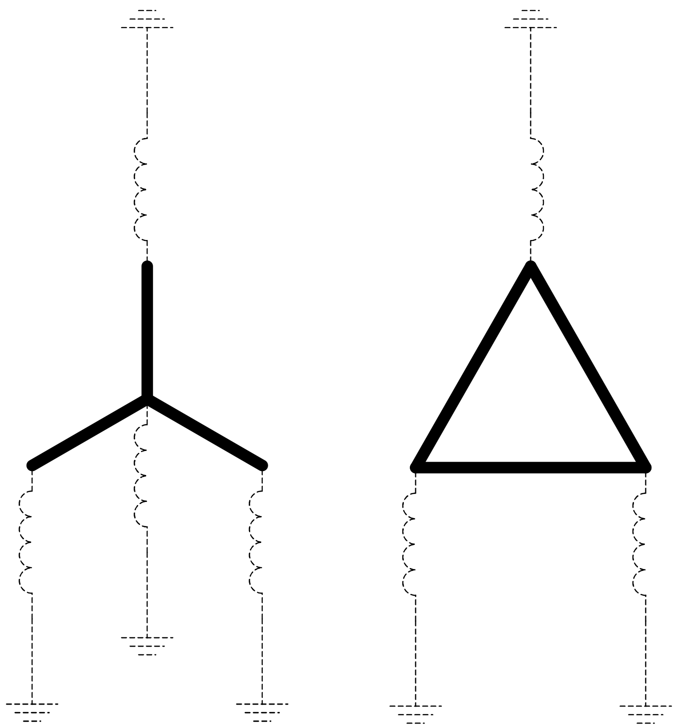
- Take Immediate action or keep track of time yourself
  - Set Number=1
  - Sample after solution step
  - Execute command to change element state
- Use the DSS Control Queue through COM Proxy
  - Set Number=1
  - Step through solution
  - Push control commands onto DSS control queue
    - (Allows DSS to keep track of when control actions happen)
  - Write routines to handle pending actions

Control Proxy in COM Interface



Misc. Hints and Tips, Known Issues, etc.

Transformer PPM_antifloat Property



Admittance matrix formulations can suffer from Y matrix singularities if part of the circuit is isolated from ground voltage reference, such as for a floating delta winding. The DSS by default increases the diagonal elements of the Yprim matrix by 1 part in 1 million (1 ppm) which is equivalent to attaching a small reactance to ground at each terminal as shown.

The DSS uses 64-bit arithmetic throughout, so this is usually not a problem with precision.

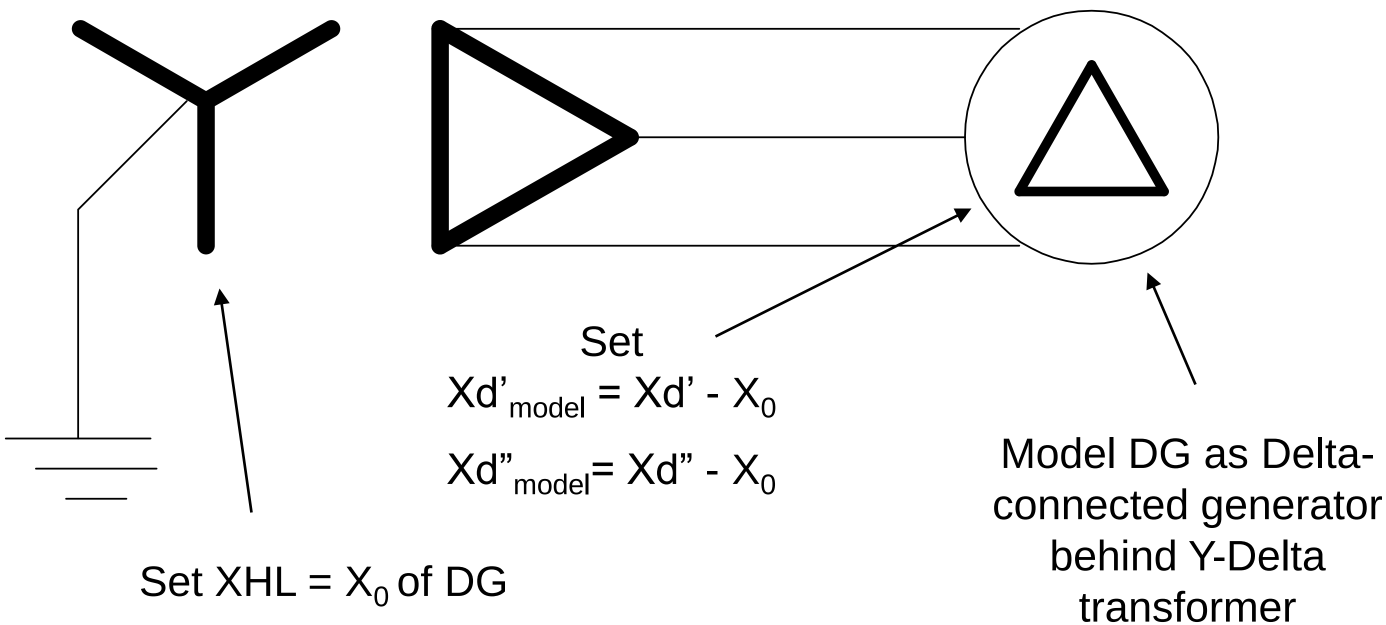
If you don't want this, you can set PPM_Antifloat = 0.

Transformer PPM_antifloat Property, cont'd

- A common error is to specify a very large transformer (e.g., 1000000 kVA or higher) to represent a very low impedance transformer
- This works for the leakage impedance
- However, the "small" anti-float reactances are no longer small !!!
  - This will sometimes draw more current than the load!
- Better approach
  - Use a reasonable kVA value and set XHL= small value
  - Set PPM_Antifloat = 0 or a small number
    - Make sure you can do this without getting floating networks!

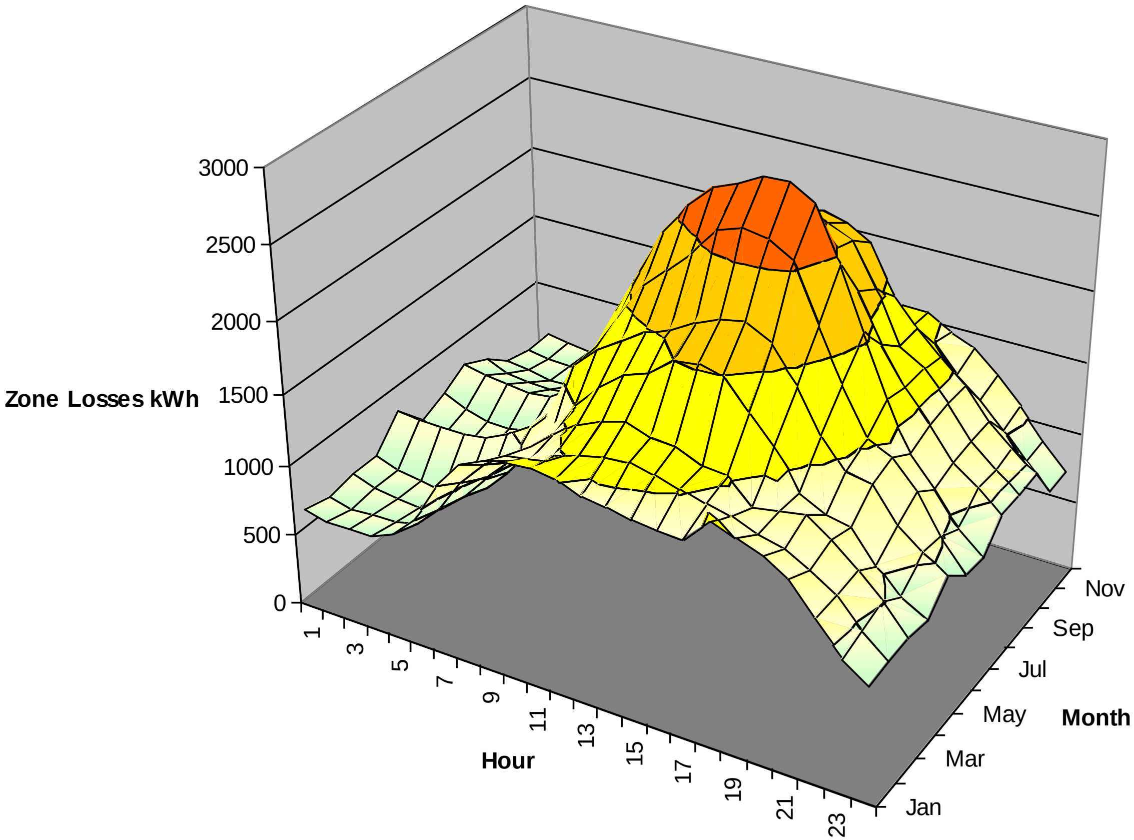
Wye-connected Generator

- Sometimes the normal power flow will not converge well for direct-connect Y-connected DG.
- Workaround:



How Do I Make One of Those 3-D Plots?

Maximum of value for each hour over the month.



We call this an “E-3” plot after the San Francisco economics firm who taught us how to do it (see http://ethree.com)

ELECTRIC POWER RESEARCH INSTITUTE

Procedure for 3D Annual Plot

- Run an annual simulation (Set Mode=Yearly)
- Turn on the Demand Interval feature of the EnergyMeter object
  - Or, simply use a Monitor object
- Import the resulting CSV file into MS Excel where a 3-D plot has been defined
- (See SampleDSSDriver.XLS for an example macro for importing a field from a CSV file.)

Annual Simulation Script

```
set casename=MyCaseName   ! This will be the folder name + year number

set mode=yearly    ! Sets Number=8760, 1 hr time step, resets meters

! Set "DemandInterval" to true so that energy quantities recorded by energy
  meters are recorded for
! each time step and set "casename" to define a directory under default
  directory in which all of the
! demand interval data is recorded.  (NOTE: Setting DemandInterval=true
  resets all energymeters.)

Set overloadreport=true  ! TURN OVERLOAD REPORT ON  (optional)
Set voltexcept=true      ! TURN VOLT Exception REPORT ON
set DemandInterval=true  ! Capture demand interval data
set DIVerbose=true

Set Year=1
solve
Set Year=2
solve

Closedi  ! Do this after final year
```

Executing Part of Annual Simulation

```
set casename=MyCaseName   ! This will be the folder name + year number

set mode=yearly    ! Sets Number=8760, 1 hr time step, resets meters

! Set "DemandInterval" to true so that energy quantities recorded by energy
  meters are recorded for
! each time step and set "casename" to define a directory under default
  directory in which all of the
! demand interval data is recorded.  (NOTE: Setting DemandInterval=true
  resets all energymeters.)

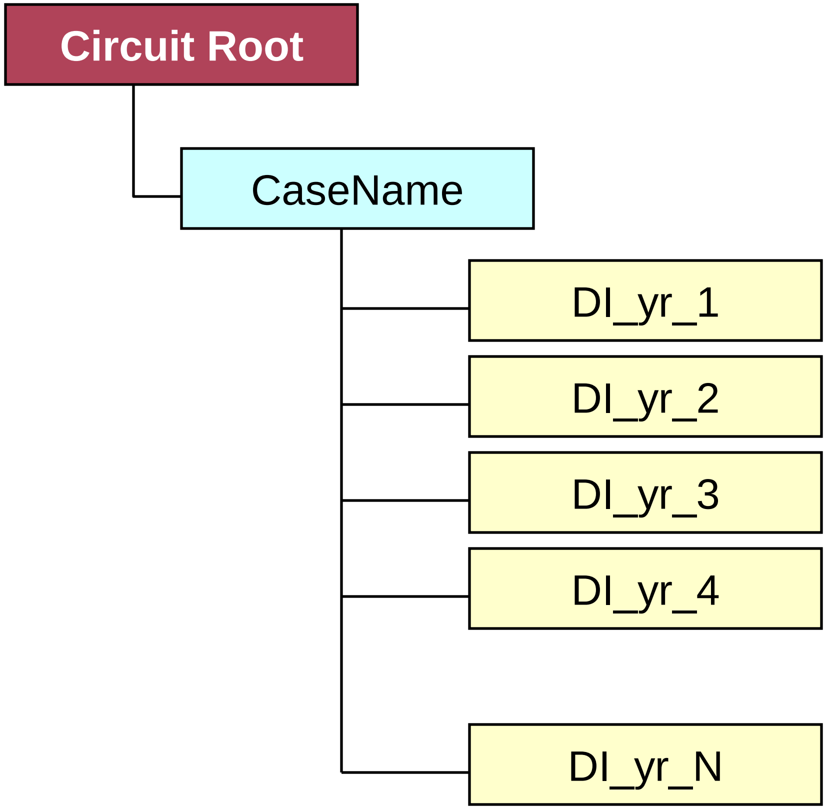
Set overloadreport=true  ! TURN OVERLOAD REPORT ON  (optional)
Set voltexcept=true      ! TURN VOLT Exception REPORT ON
set DemandInterval=true  ! Capture demand interval data
set DIVerbose=true

Set Year=1
Set hour=5000        ! Start at hour 5000 into loadshapes
solve Number=168     ! Solve one week; redefine Number

Closedi   ! Do this after final year
```

Demand Interval Files

- After an annual simulation, the results are saved in the directory structure as a collection of CSV files



Inside a DI_Yr_nn Directory

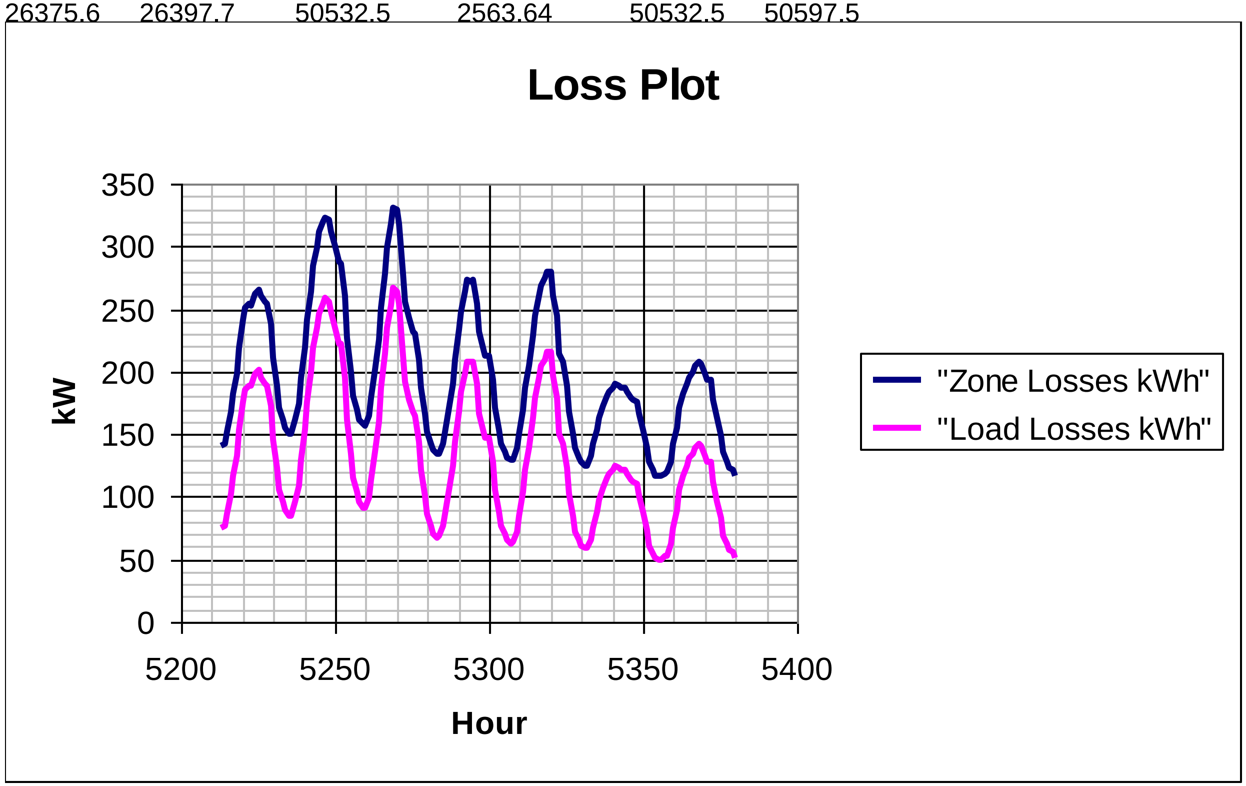
Name	Size	Type	Date Modified
DI_Overloads.CSV	79 KB	Microsoft Office Exc...	1/8/2009 8:34 PM
DI_SystemMeter.CSV	22 KB	Microsoft Office Exc...	1/8/2009 8:34 PM
DI_Totals.CSV	52 KB	Microsoft Office Exc...	1/8/2009 8:34 PM
DI_VoltExceptions.CSV	6 KB	Microsoft Office Exc...	1/8/2009 8:34 PM
EnergyMeterTotals.CSV	2 KB	Microsoft Office Exc...	1/8/2009 8:34 PM
SystemMeter.CSV	1 KB	Microsoft Office Exc...	1/8/2009 8:34 PM
totalized.CSV	52 KB	Microsoft Office Exc...	1/8/2009 8:34 PM
Totals.CSV	2 KB	Microsoft Office Exc...	1/8/2009 8:34 PM

The results are saved as a series of CSV files.

(one week simulation)

Snip from CSV file loaded into Excel

Hour	"kWh"	"kvarh"	"Max kW"	"Max kVA'	"Zone kWh"	"Zone kvarh'	"Zone Max kV	"Zone Max "
5213	20430.7	775.005	20430.7	20445.4	39147.9	1908.31	39147.9	39194.4
5214	21438	744.841	21438	21450.9	41086.8	1875.58	41086.8	41129.5
5215	22450.5	810.443	22450.5	22465.2	43024.8	2013.08	43024.8	43071.9
5216	23559.8	928.416	23559.8	23578.1	45143	2243.88	45143	45198.7
5217	24688.2	1012.56	24688.2	24708.9	47300.2	2413.44	47300.2	47361.7
5218	26375.6	1079.88	26375.6	26397.7	50532.5	2563.64	50532.5	50597.5
5219	28516.4	1149.75						
5220	30231	1242.84						
5221	31459.5	1263.09						
5222	32448.5	1232.52						
5223	33111.5	1189.47						
5224	33959.2	1222.36						
5225	34387.3	1225.1						
5226	34678.5	1175.29						



Enable/Disable

- Once a circuit element is defined in a script, it can be
  - Enabled (default)
  - Disabled
  - Redefined (Edit)
- Use Enable/Disable to temporarily add or delete elements from the circuit
- Enable/Disable Commands
  - Disable Line.Line1
  - Line.Line1.Enabled = No
  - Edit Line.Line1 …. Enabled=No

Open/Close

- You can open any terminal of any device that is active in the circuit:
  - Open Line.Line1 2   (opens Terminal 2)
  - Open Line.Line1 2 3 (opens phase 3 of terminal 2)
- Caveat: Voltage at open terminal is inaccessible (does not exist)
- Alternative:
  - Line.Line1.Bus2=Term2_Open
    - Redefine Bus2 to another bus with nothing else
    - "Term2_open" voltage is accessible

Load Models (Present version)

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

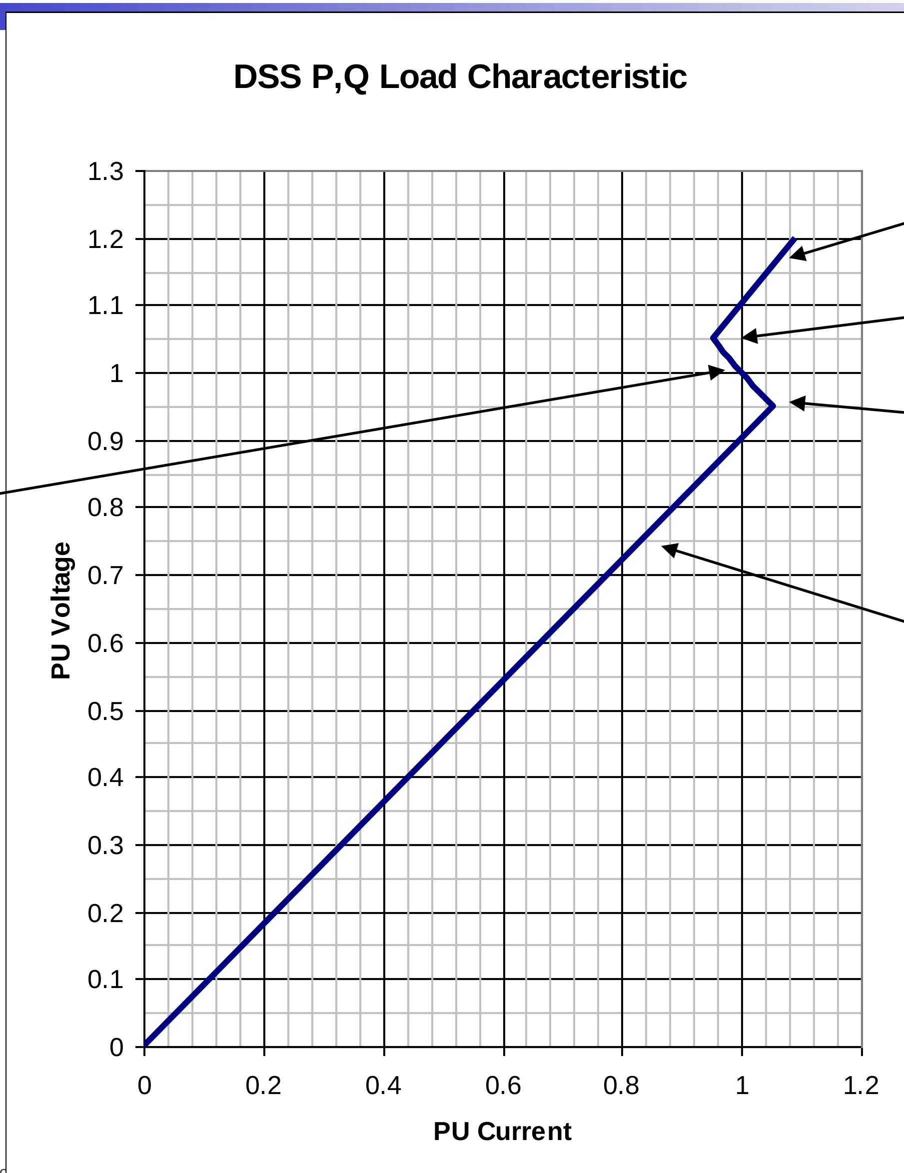
6:Const P, Fixed Q

7:Const P, Fixed Impedance Q

Standard P + jQ Load Model

- When the voltage goes out of the normal range for a load the model reverts to a linear load model
  - This generally guarantees convergence
    - Even when a fault is applied
  - To change break points to +/- 10%:
    - Load.Load1.Vmaxpu=1.10
    - Load.Load1.Vminpu=0.90
  - Note: to solve some of the IEEE Radial Test feeders and match the published results, you have to set Vminpu to less than the lowest voltage published

Standard P + jQ Load Model (Model=1)



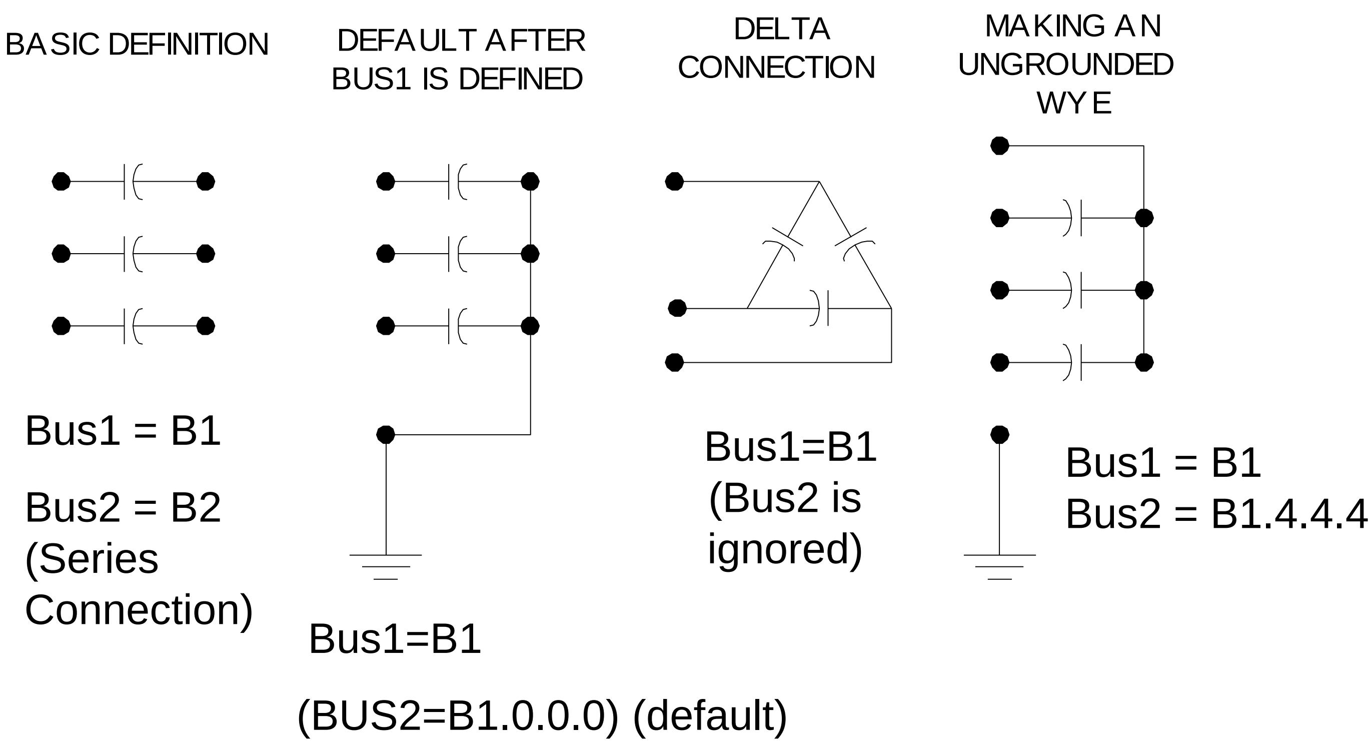
* Change by setting *Vminpu* and *Vmaxpu* Properties

Other Load Models

- 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

Two Terminal Caps, Reactors, Faults

- Capacitors, Reactors, and Faults are special 2-terminal PDElements with special defaults for bus connections



Common Errors

A Bus's Life

- In contrast to other power system analysis programs, Bus objects do not exist until required
- These commands will force building of the bus list
  - Solve
  - CalcVoltageBases
  - MakeBusList

- The Bus list is built from the currently enabled devices
- Editing circuit element properties often requires rebuilding the bus list
  - If no changes, the bus list is not re-built

Specifying Transformer Neutral Reactor

What's Wrong With This?

New Transformer.T1 phases=3 Wind=2

~ Buses=[DeltaBus WyeBus]

~ Conns=[Delta Wye]

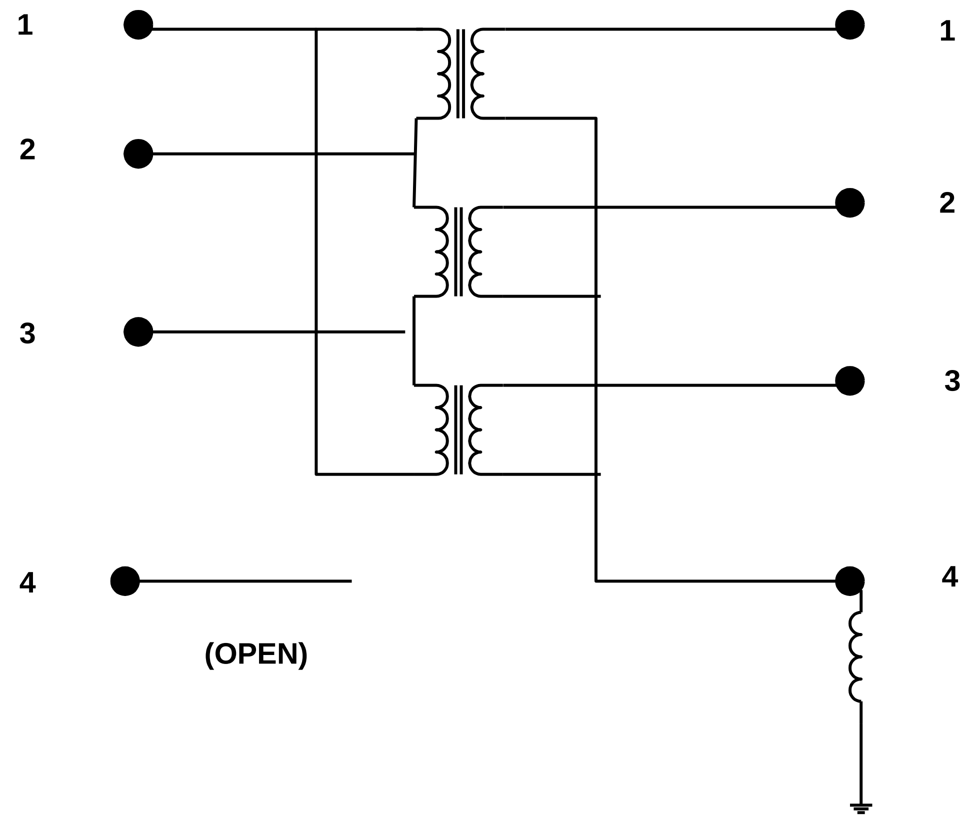
~ kVAs=[10000 10000]

~ kVs=[115 12.47]

~ Wdg=2 Rneut=0 Xneut=4

DELTA-WYE TRANSFORMER

With Neutral Grounding Reactor



EPRI | ELECTRIC POWER RESEARCH INSTITUTE

Specifying Transformer Neutral Reactor

What's Wrong With This?

New Transformer.T1 phases=3 Wind=2

~ Buses=[DeltaBus WyeBus]

~ Conns=[Delta Wye]

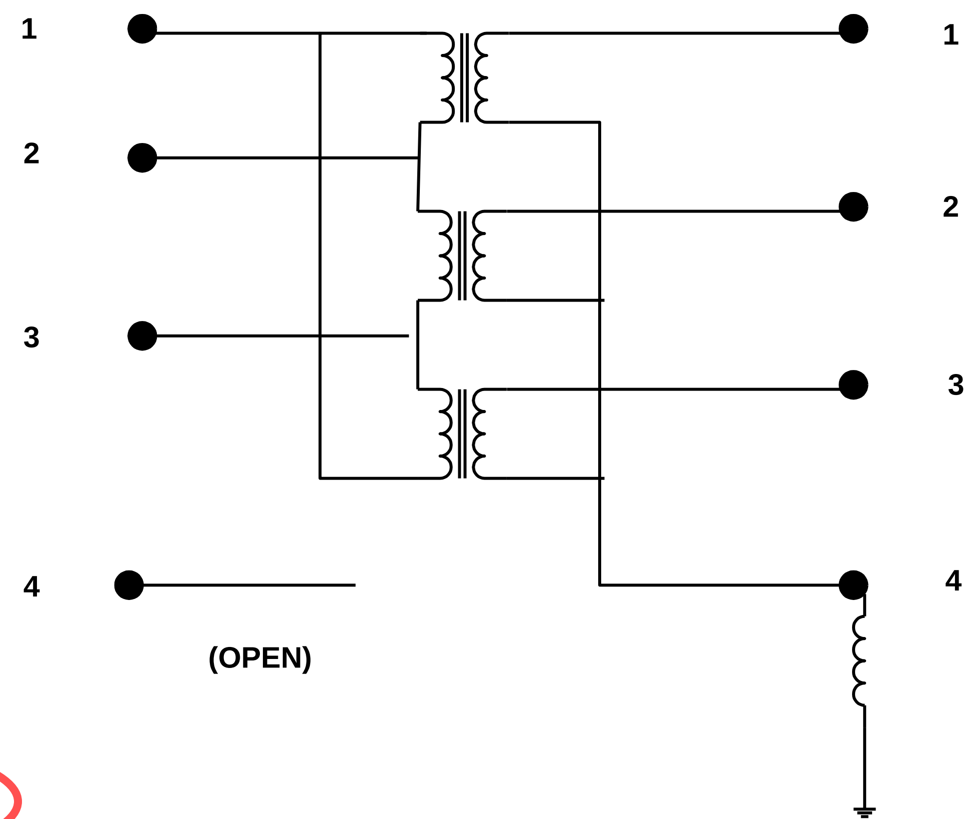
~ kVAs=[10000 10000]

~ kVs=[115 12.47]

~ Wdg=2 Rneut=0 Xneut=4

Expands to

WyeBus.1.2.3.0

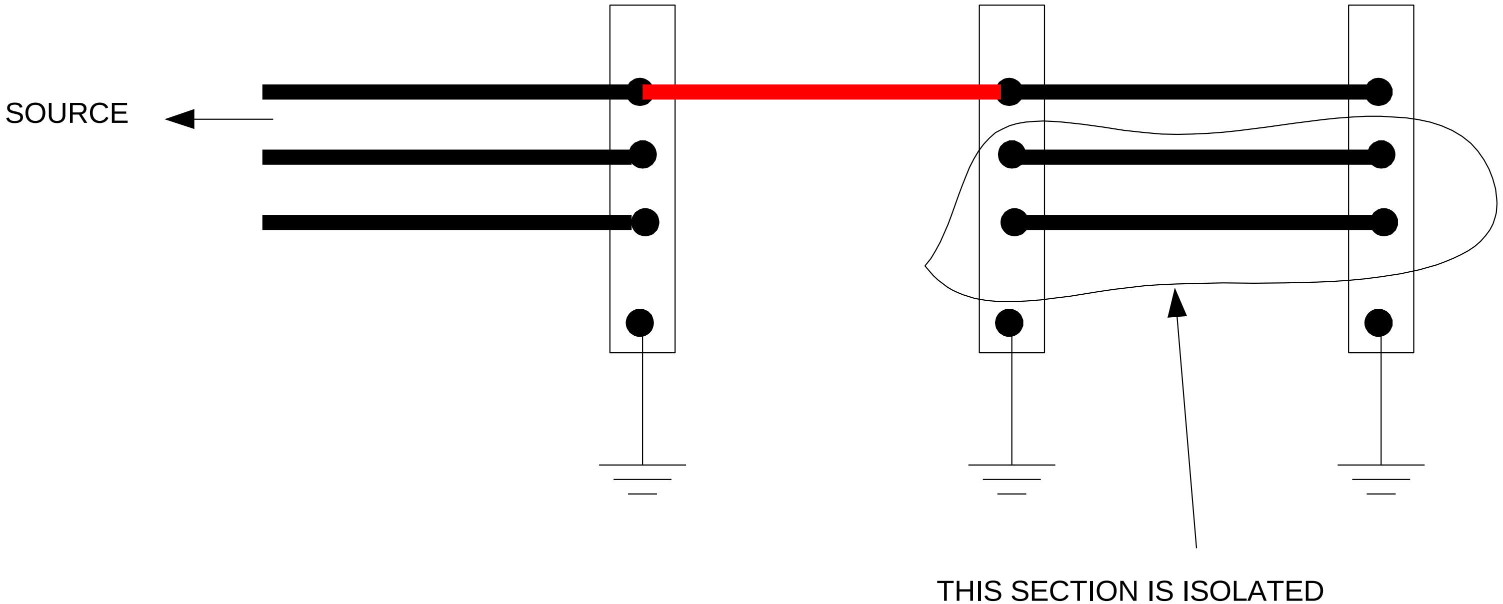


Shorts The Reactor !!!

Line Length and Distance from Meter

- Distance from the upstream meter is accumulated for each branch and bus as the meter zone is traced
- The **Length=** and **Units=** properties of the Line definition are used to convert the length to km
- Problem: **Unit-less lines**
  - Line impedance may be computed by entering the impedance value in actual ohms and setting the length=1.
  - However, this will be interpreted as a <u>1 km line</u>!!
    - Usage of the distance value will be suspect
- Distance is available from the COM Bus interface

Single-Phase to 3-phase Line



Diagnosing: This may cause a floating point error in normal snapshot power flow.
1) Look at voltage after CalcVoltageBases
2) Do Solve Mode=Direct; then look at voltages (Show Voltage LN Nodes)

Wrong Voltage Base on Load or Generator

- Don't forget to specify "kv=…" property!

- 3-phase Load
  - `New Load.Load1 Phases=3 Bus1=B1 kV=12.47 kW=100 PF=.95`

- 3 Single-phase Loads
  - `New Load.Load1a Phases=1 Bus1=B1.1 kV=7.2 kW=33.33 PF=.95`
  - `New Load.Load1b Phases=1 Bus1=B1.2 kV=7.2 kW=33.33 PF=.95`
  - `New Load.Load1c Phases=1 Bus1=B1.3 kV=7.2 kW=33.33 PF=.95`

Make sure these have different names

ELECTRIC POWER RESEARCH INSTITUTE

A Tricky One – Sequence dependencies

```
Compile xxxx
Set VoltageBases=[12.47  .48  .208]
CalcVoltageBases
! Now define energymeter to get voltage base register
New EnergyMeter.Main Line.Line1 1
Reconductor Line50  Line60   LineCode=795ACSR
```

Causes an error that says Line50 and Line60 not in a meter zone or not in same meter zone

Why?

Nothing has happened since the definition of the EnergyMeter object to force it to compute its zone.

Resolution:

Issue **Solve** or **MakeBusList** to force meterzone. Then issue Reconductor

ELECTRIC POWER RESEARCH INSTITUTE

Questions?