ELECTRIC POWER RESEARCH INSTITUTE

Results of OpenDSS CIM Interoperability Testing: Paris Interoperability Test, March 2011

2011 TECHNICAL REPORT

Results of OpenDSS CIM Interoperability Testing: Paris Interoperability Test, March 2011

EPRI Project Manager
J. Simmins

ELECTRIC POWER RESEARCH INSTITUTE

3420 Hillview Avenue
Palo Alto, CA 94304-1338
USA

PO Box 10412
Palo Alto, CA 94303-0813
USA

800.313.3774
650.855.2121

askepri@epri.com

www.epri.com

1023218
Final Report, July 2011

DISCLAIMER OF WARRANTIES AND LIMITATION OF LIABILITIES

THE FOLLOWING ORGANIZATION, UNDER CONTRACT TO EPRI, PREPARED THIS REPORT:

MelTran, Inc

NOTE

For further information about EPRI, call the EPRI Customer Assistance Center at 800.313.3774 or e-mail askepri@epri.com.

Acknowledgments

The following organization, under contract to the Electric Power Research Institute (EPRI), prepared this report:

MelTran, Inc.
90 Clairton Blvd, Suite A
Pittsburgh, PA 15236

Principal Investigator
T. McDermott

This report describes research sponsored by EPRI.

This publication is a corporate document that should be cited in the literature in the following manner:

Results of OpenDSS CIM Interoperability Testing: Paris Interoperability Test, March 2011. EPRI, Palo Alto, CA: 2011. 1023218.

Abstract

The Electric Power Research Institute (EPRI) participated in the 2011 interoperability testing of the Common Information Model (CIM) for distribution, which was conducted at the Electricité de France facilities in Clamard, France, from March 28 through April 1, 2011. The testing covered the following parts of International Electrotechnical Commission (IEC) standard 61970, Energy Management System Application Program Interface (EMS-API), and standard 61968, Application Integration at Electric Utilities, System Interfaces for Distribution Management:

- IEC 61970-301, Common Information Model (CIM) Base, a semantic model that describes the components of a power system at an electrical level and the relationships between components, originally written with transmission systems in mind
- IEC 61968-4, Interfaces for Records and Asset Management, which describes data exchange for distribution system assets and contains equipment catalog data used in load flow models
- IEC 61968-11, Common Information Model (CIM) Extensions for Distribution, which contains diagrams and documentation for the CIM for distribution
- IEC 61968-13, CIM RDF Model Exchange Format for Distribution, called the *Common Distribution Power System Model (CDPSM)*, which is an extension of the transmission-oriented CPSM (61970-452)

The EPRI researchers used OpenDSS—an open-source distribution system simulator that EPRI maintains and supports—to conduct the tests. EPRI researchers have used OpenDSS to develop IEEE test feeders that are used in the development of a DCIM for unbalanced systems. During these tests, vendors conduct model exchanges, and electric utility witnesses verify the model exchanges and sign off on test reports. The result for any test is either "passed" or "passed with errors." EPRI achieved "pass" results on the power flow tests, on the solution of an incremental power flow, and on the interoperability test with Open Grid Systems' CIMPHONY product. It achieved "pass with errors" on the basic import of instance files; Open DSS does not use or manage all the data, so there was some loss of information.

This report describes the testing and results; it also provides recommendations for further development of OpenDSS. The last section of the report presents material from IEC 61968-11.

Keywords

Common Information Model (CIM)
Interoperability tests
IEC 61968
Open DSS
IEEE 13-bus feeder
Smart Grid

Table of Contents

List of Figures

List of Tables

Section 1: Summary

This report summarizes participation in the 2011 Common Information Model (CIM) for distribution interoperability tests, using EPRI's OpenDSS software. The test was conducted at EdF in Clamard, France, from March 28 through April 1, 2011.

The distribution CIM comprises several parts that were tested:

- IEC 61970-301; the transmission-oriented CIM, consisting of a positive sequence load flow model referred to as Common Power System Model (CPSM), serving as a base for parts of the distribution CIM.
- IEC 61968-11; the full distribution CIM diagrams and documentation
- IEC 61968-13; the Common Distribution Power System Model (CDPSM), which extends CPSM to an unbalanced load flow model
- IEC 61968-4; describing distribution system assets data exchange, which contains some equipment catalog data used in load flow models

The 2011 tests in Clamard covered parts 13 and 4. Tests at other venues covered part 9 (metering), and there are several other parts of the distribution CIM that have not been fully developed or tested yet.

A CIM interoperability test covers pre-defined "profiles", which select only certain classes and attributes from the CIM, rather than the full model. A profile relies on CIM model files maintained in Enterprise Architect, particularly Unified Modeling Language (UML) diagrams, for a specific version of the CIM. Then, the profile uses a Web Ontology Language (OWL) file to select portions of the model and apply constraints. Most parties use CIMTool to work with OWL files. The test preparation consists of iterative profile development, along with development of "instance files", which are examples to be used in the test.

During the test, participating vendors either sit in the same room (as reported here), or they may conduct model exchanges remotely over a network. In both cases, electric utility witnesses must verify the model exchange and sign off on test reports, which are then published at the CIM User Group site. There were several kinds of 2011 CDPSM test:

- Basic model import; read a prepared instance file with no loss of information, but without applying the data for any purpose. Then re-export the data to CIM. This tests product compliance.

- Run load flow on the instance file, with a solution "close enough" to the base case. This tests compliance of a vendor's power flow feature, if available.
- Read one of the other vendor's re-exported CIM files, with no loss of information. This tests vendor interoperability.
- Read one of the other vendor's re-exported CIM files, and run load flow with solution "close enough" to the base case. This tests vendor interoperability of power flow.
- Read an incremental model change to one of the full instance files, such as a switching operation.

None of the vendors attempted all of these tests, because the products have specific feature sets. Also, the result for any test is either "passed" or "passed with errors", rather than simply "failed", in order to encourage vendor adoption and support of the distribution CIM. This was the second interoperability test of the CDSPM, and both tests have advanced the distribution CIM and CDPSM. Compared to the 2009 tests, in 2011 the distribution CIM and CDPSM are stable and ready for widespread adoption by vendors.

OpenDSS is an open-source distribution system simulator that EPRI maintains and supports for use in research projects. Among other features, the OpenDSS has been used to develop and promote the IEEE test feeders, which include unbalanced lines, loads, and transformers. These IEEE test feeders are used in development of the distribution CIM for unbalanced systems. Because OpenDSS can model all IEEE test feeders, a distribution CIM export command was implemented in OpenDSS. The export is implemented in the Delphi programming language for OpenDSS. An exported model resides in an XML file.

The OpenDSS importer is implemented in Java, using the open-source Jena toolkit to manage OWL and XML files. Presently, this importer runs on the command line and converts XML to DSS formats.

Key results of OpenDSS participation in these 2011 tests were:

- "Pass with errors" on the basic import of EdF and GE instance files; OpenDSS does not use or manage all of the data, so there was some loss of information.
- "Pass" the power flow tests on all 4 instance files provided by EdF. No other vendor accomplished this. (The GE instance file was not a power flow model).
- During the 2009 test, the Jena-based importer ran out of memory on very large instance files, but this did not happen during 2011. This reflects improvements in the distribution CIM and the profiles, rather than improvements in OpenDSS or its importer. It seems that Jena can be used for future versions of the OpenDSS importer.
- "Passed" the solution of an incremental power flow.

- “Passed” the interoperability test with another vendor, Open Grid Systems, with its CIMPHONY product.

Significant OpenDSS development activities associated with the tests were:

- Assisted in the modeling team’s transition from WireArrangement to WireSpacingInfo for the physical description of overhead lines. An error had been introduced at the February 2011 meeting of WG14, and this had to be corrected before the interoperability tests.
- Added cable parameter calculations for the concentric neutral and tape shield types. This involved adding new CNData and TSData object types to OpenDSS, new CNCable and TSCable attributes to the Line and LineGeometry objects, and re-factoring the line constants code. Two examples from Kersting’s textbook were implemented.
- Updated the Export CDPSM command in OpenDSS for the 2011 distribution CIM and profiles. The IEEE 13-bus instance file was then generated in two formats. One uses phase impedance matrices for the lines, and the other uses wire/cable and spacing data.
- Updated the Jena importer code, adding command-line switches for:
    - XML file character encoding
    - System nominal frequency
    - Choice of profile to import
    - Voltage multiplier (i.e., volts or kv)
    - Power multiplier (i.e., kva or MVA)
- Due to time constraints, the importer was developed only to the extent necessary for the 2011 instance files. In particular, the 13-bus IEEE instance files cannot be re-imported into OpenDSS because there are so many options for describing lines and transformers in the distribution CIM. The other participating vendors used only some of these features.

Work products and deliverables included:

- Identified CIM and Profile issues as described in Section 4 of this report. This includes a revised version of the combined profile, in OpenDSS.owl, which properly supports the IEEE 13-bus feeder.
- Contributed IEC 61968-11 material in Section 5 of this report
- OpenDSS source code checked in to SourceForge, supporting the cable parameter calculations and the new exporter version.
- Java source code and instance files checked in to SourceForge, supporting the new importer version.

The following recommendations are made for EPRI:

- Continue active participation in the IEC model and profile development processes. Without this, the model can degenerate because very few people involved in the process use unbalanced load flow models. During 2010, both the line and transformer models had some errors introduced, and EPRI could avoid this with continuous participation by its experts.

- In OpenDSS, implement the import of tank-based transformer models. This is the main feature needed to support two-way exchange of the IEEE 13-bus instance files.
- In OpenDSS, implement the export and import features for separate, granular profiles. The 2011 test allowed use of a single combined profile, but future tests are planned to require use of separate profiles.
- In OpenDSS, add support for incremental model exchanges. This is important because the XML file is very large for a complete feeder model in the distribution CIM.
- In OpenDSS, add support for the state variable profile, which allows the exchange of power flow solution results with other applications, like GIS.
- In OpenDSS, add a transmission-style power flow solution report, summarizing generators, loads, and shunts. This will streamline the process of solution validation during future interoperability tests.

Section 2: IEEE 13-Bus Instance Files

During this test period, the IEEE 13-bus test feeder was exported from OpenDSS under the 2011 distribution CIM model, in two different forms. One version uses phase impedance matrices to describe the lines and cables, while the other version uses physical wire, spacing and cable data.

This task was accomplished by modifying the Delphi source code of OpenDSS. During this process, some gaps in the 2011 model and profiles were identified, as described in Section 4 of this report. Overall, the 2011 model is much more mature and ready for widespread use than the 2009 model. Because these instance files were generated from the OpenDSS executable, after reading the IEEE test case, it is now possible to export any OpenDSS model of any size in the latest distribution CIM.

Compared to the 2009 interoperability tests, there were major changes in the distribution CIM line and transformer models, with smaller changes in the capacitor and load models. Most of these changes were made to combine the WG13 (transmission) and WG14 (distribution) models, and the end result is that these unbalanced line and transformer models are more stable and mature because both working groups appear to agree on them. However, the unbalanced line model changes were not settled until the week before the 2011 test. In addition, cable model support was added in 2011.

Information that applies to both versions of the IEEE 13-bus feeder in the distribution CIM:

- Units of Length or PerLength: meters
- Units of Voltage: kV
- Units of Current: Amperes
- Units of Power: kW, kVAR, kVA
- System frequency: 60 Hz
- Soil resistivity: 100 ohm-meter

Further information may be found at http://ewh.ieee.org/soc/pes/dsacom/testfeeders/index.html

Wires Package Example: OpenDSS_IEEE13_MTX

This instance file demonstrates:

- line segment parameters using PerLengthPhaseImpedance
- transformer parameters using TransformerMeshImpedance and TransformerCoreAdmittance

Table 2-1 shows the instance counts using CIMSpy version 2.2. The newer classes are highlighted in **red**; CIMSpy uses an older CIM schema and is able to identify these new classes, but cannot interpret them.

Table 2-1
Instance Counts for the 13-Bus Feeder Using Wires Package

Summary		
CIM Version:		http://iec.ch/TC57/2010/CIM-schema-cim15#
Class	**# of Objects**	**Description**
ACLineSegment	11	A wire or combination of wires, with consistent electrical characteristics, building a single electrical system, used to carry alternating current between points in the power system.
ACLineSegmentPhase	8	Class ACLineSegmentPhase is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
Asset	3	
BaseVoltage	3	Collection of BaseVoltages which is used to verify that the BusbarSection.BaseVoltage and other voltage attributes in the CIM are given a value existing in the collection.
ConnectivityNode	16	Connectivity nodes are points where terminals of conducting equipment are connected together with zero impedance.
CoordinateSystem	1	Class CoordinateSystem is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
EnergyConsumer	9	Generic user of energy - a point of consumption on the power system model
EnergyConsumerPhase	19	Class EnergyConsumerPhase is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
EnergySource	1	A generic equivalent for an energy supplier on a transmission or distribution voltage level.

Table 2-1 (continued)
Instance Counts for the 13-Bus Feeder Using Wires Package

Summary		
CIM Version:		http://iec.ch/TC57/2010/CIM-schema-cim15#
Class	**# of Objects**	**Description**
GeographicalRegion	1	A geographical region of a power system network model.
IEC61970CIMVersion	1	This is the IEC 61970 CIM version number assigned to this UML model file.
Line	1	A component part of a system extending between adjacent substations or from a substation to an adjacent interconnection point.
LoadBreakSwitch	1	A mechanical switching device capable of making, carrying, and breaking currents under normal operating conditions.
LoadResponseCharacteristic	7	Models the characteristic response of the load demand due to to changes in system conditions such as voltage and frequency. This is not related to demand response.
Location	30	
PerLengthPhaseImpedance	7	Class PerLengthPhaseImpedance is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
PhaseImpedanceData	26	Class PhaseImpedanceData is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
PositionPoint	46	Class PositionPoint is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
PowerTransformer	3	An electrical device consisting of two or more coupled windings, with or without a magnetic core, for introducing mutual coupling between electric circuits. Transformers can be used to control voltage and phase shift (active power flow).
RatioTapChanger	3	Class RatioTapChanger is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#

Table 2-1 (continued)
Instance Counts for the 13-Bus Feeder Using Wires Package

Summary		
ShuntCompensator	2	A shunt capacitor or reactor or switchable bank of shunt capacitors or reactors. A section of a shunt compensator is an individual capacitor or reactor. Negative values for mVArPerSection and nominalMVAr indicate that the compensator is a reactor.
ShuntCompensatorPhase	4	Class ShuntCompensatorPhase is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
SubGeographicalRegion	1	A subset of a geographical region of a power system network model.
SvTapStep	3	State variable for transformer tap step. Normally a profile specifies only one of the attributes "position"or "tapRatio".
TapChangerControl	3	Class TapChangerControl is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
TapChangerInfo	3	Class TapChangerInfo is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
Terminal	46	An electrical connection point to a piece of conducting equipment. Terminals are connected at physical connection points called "connectivity nodes".
TransformerCoreAdmittance	5	Class TransformerCoreAdmittance is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
TransformerMeshImpedance	5	Class TransformerMeshImpedance is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
TransformerTank	5	Class TransformerTank is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
TransformerTankEnd	10	Class TransformerTankEnd is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#

Power Flow Solution in OpenDSS

```
Max pu. voltage = 1.0684
Min pu. voltage = 0.9878
Total Active Power:   3.58158 MW
Total Reactive Power: 1.72064 Mvar
Total Active Losses:   0.10986 MW, (3.067 %)
Total Reactive Losses: 0.314585 Mvar

CONTROLLED TRANSFORMER TAP SETTINGS

Name            Tap      Min       Max     Step  Position

reg1         1.06250  0.90000  1.10000  0.00625     10
reg2         1.05000  0.90000  1.10000  0.00625     8
reg3         1.06875  0.90000  1.10000  0.00625     11

NODE-GROUND VOLTAGES BY BUS & NODE

Bus       Node    V (kV)    Angle      p.u.   Base kV

sourcebus  1     66.387 /_   30.0   0.99987   115.000
   -       2     66.388 /_  -90.0   0.99989   115.000
   -       3     66.385 /_  150.0   0.99985   115.000
650 .....  1     2.4013 /_    0.0   0.99981     4.160
   -       2     2.4015 /_ -120.0   0.99987     4.160
   -       3     2.4014 /_  120.0   0.99983     4.160
rg60 ....  1     2.5511 /_    0.0    1.0622     4.160
   -       2     2.5213 /_ -120.0    1.0498     4.160
   -       3     2.5662 /_  120.0    1.0684     4.160
633 .....  1     2.4557 /_   -2.8    1.0225     4.160
   -       2     2.4553 /_ -121.3    1.0223     4.160
   -       3     2.4587 /_  117.9    1.0237     4.160
634 .....  1    0.27674 /_   -3.5   0.99859     0.480
   -       2    0.27813 /_ -121.8    1.0036     0.480
   -       3    0.27852 /_  117.4     1.005     0.480
671 .....  1     2.3947 /_   -5.7   0.99703     4.160
   -       2     2.4522 /_ -121.7     1.021     4.160
   -       3      2.382 /_  116.2   0.99176     4.160
645 .....  2     2.4381 /_ -121.5    1.0151     4.160
   -       3     2.4599 /_  117.9    1.0242     4.160
646 .....  2      2.434 /_ -121.6    1.0134     4.160
   -       3      2.455 /_  117.9    1.0221     4.160
692 .....  3      2.382 /_  116.2   0.99176     4.160
   -       1     2.3947 /_   -5.7   0.99703     4.160
   -       2     2.4522 /_ -121.7     1.021     4.160
675 .....  1     2.3793 /_   -5.9   0.99066     4.160
   -       2     2.4576 /_ -121.8    1.0232     4.160
   -       3     2.3776 /_  116.2   0.98992     4.160
611 .....  3     2.3725 /_  116.0    0.9878     4.160
652 .....  1     2.3764 /_   -5.6   0.98945     4.160
670 .....  1     2.4407 /_   -3.7    1.0162     4.160
   -       2     2.4553 /_ -121.4    1.0223     4.160
   -       3     2.4343 /_  117.2    1.0135     4.160
632 .....  1     2.4629 /_   -2.8    1.0255     4.160
   -       2     2.4604 /_ -121.3    1.0244     4.160
   -       3     2.4646 /_  117.9    1.0262     4.160
680 .....  1     2.3947 /_   -5.7   0.99703     4.160
   -       2     2.4522 /_ -121.7     1.021     4.160
   -       3      2.382 /_  116.2   0.99176     4.160
684 .....  1     2.3899 /_   -5.7   0.99507     4.160
   -       3     2.3772 /_  116.1   0.98978     4.160
```

```
CIRCUIT ELEMENT CURRENTS

(Currents into element from indicated bus)

Power Delivery Elements

  Bus     Phase    Magnitude, A     Angle

ELEMENT = "Vsource.source"
sourcebus     1           18.027 /_  179.5
sourcebus     2           19.782 /_   71.5
sourcebus     3           22.286 /_  -58.1
------------
sourcebus     0           18.027 /_   -0.5
sourcebus     0           19.782 /_ -108.5
sourcebus     0           22.286 /_  121.9

ELEMENT = "Transformer.sub"
sourcebus     1           18.027 /_   -0.5
sourcebus     2           19.782 /_ -108.5
sourcebus     3           22.286 /_  121.9
sourcebus     0                0 /_    0.0
------------
650           1           592.54 /_  151.3
650           2           440.98 /_   38.9
650           3           623.29 /_  -86.0
650           0           145.99 /_   24.5

ELEMENT = "Transformer.reg1"
650           1           592.54 /_  -28.7
650           0           592.54 /_  151.3
------------
rg60          1           557.69 /_  151.3
rg60          0           557.69 /_  -28.7

ELEMENT = "Transformer.reg2"
650           2           440.98 /_ -141.1
650           0           440.98 /_   38.9
------------
rg60          2           419.98 /_   38.9
rg60          0           419.98 /_ -141.1

ELEMENT = "Transformer.reg3"
650           3           623.29 /_   94.0
650           0           623.29 /_  -86.0
------------
rg60          3           583.19 /_  -86.0
rg60          0           583.19 /_   94.0

ELEMENT = "Transformer.xfm1"
633           1           80.957 /_  -38.0
633           2            62.23 /_ -158.7
633           3           62.146 /_   80.5
633           0           19.546 /_  145.1
------------
634           1           701.62 /_  142.0
634           2           539.33 /_   21.3
634           3           538.59 /_  -99.5
634           0            169.4 /_  -34.9

ELEMENT = "Capacitor.cap1"
675           1           82.494 /_   84.1
675           2           85.206 /_  -31.8
675           3           82.432 /_ -153.8
------------
675           0           82.494 /_  -95.9
675           0           85.206 /_  148.2
675           0           82.432 /_   26.2
```

```
ELEMENT = "Capacitor.cap2"
611           3           41.189 /_ -154.0
------------
611           0           41.189 /_   26.0

ELEMENT = "Line.650632"
rg60          1           557.69 /_  -28.7
rg60          2           419.98 /_ -141.1
rg60          3           583.19 /_   94.0
------------
632           1           557.69 /_  151.3
632           2           419.98 /_   38.9
632           3           583.19 /_  -86.0

ELEMENT = "Line.632670"
632           1           477.98 /_  -27.1
632           2           217.57 /_ -135.4
632           3           472.73 /_  100.4
------------
670           1           477.98 /_  152.9
670           2           217.57 /_   44.6
670           3           472.73 /_  -79.6

ELEMENT = "Line.670671"
670           1           469.96 /_  -27.0
670           2           187.94 /_ -132.8
670           3           418.82 /_  102.1
------------
671           1           469.96 /_  153.0
671           2           187.94 /_   47.2
671           3           418.82 /_  -77.9

ELEMENT = "Line.671680"
671           1       0.00057873 /_   85.1
671           2        0.0005895 /_  -32.2
671           3       0.00058629 /_ -154.0
------------
680           1      3.7499E-012 /_  -76.0
680           2      1.2862E-012 /_   45.0
680           3       1.819E-012 /_  -90.0

ELEMENT = "Line.632633"
632           1           80.957 /_  -38.0
632           2            62.23 /_ -158.7
632           3           62.145 /_   80.5
------------
633           1           80.957 /_  142.0
633           2            62.23 /_   21.3
633           3           62.146 /_  -99.5

ELEMENT = "Line.632645"
632           3           65.041 /_   58.5
632           2            144.2 /_ -142.3
------------
645           3           65.041 /_ -121.5
645           2            144.2 /_   37.7

ELEMENT = "Line.645646"
645           3           65.041 /_   58.5
645           2           65.041 /_ -121.5
------------
646           3           65.041 /_ -121.5
646           2           65.041 /_   58.5

ELEMENT = "Line.692675"
692           1           203.86 /_   -5.2
692           2           66.793 /_  -56.3
692           3           122.17 /_  113.1
------------
675           1           203.86 /_  174.8
```

```
675            2           66.792 /_  123.7
675            3           122.17 /_  -66.9

ELEMENT = "Line.671684"
671            1           63.622 /_  -39.5
671            3           71.271 /_  122.3
------------
684            1           63.622 /_  140.5
684            3           71.271 /_  -57.7

ELEMENT = "Line.684611"
684            3           71.271 /_  122.3
------------
611            3           71.271 /_  -57.7

ELEMENT = "Line.684652"
684            1           63.622 /_  -39.5
------------
652            1           63.622 /_  140.5

ELEMENT = "Line.671692"
671            1           227.53 /_  -18.3
671            2           66.793 /_  -56.3
671            3           176.33 /_  110.2
------------
692            1           227.53 /_  161.7
692            2           66.793 /_  123.7
692            3           176.33 /_  -69.8

= = = = = = = = = = = = = = = = = = =  = = = = = = = = = = = =  = =

Power Conversion Elements

  Bus     Phase    Magnitude, A     Angle

ELEMENT = "Load.671"
671            1           184.08 /_  -33.4
671            2           184.15 /_ -153.4
671            3           184.07 /_   86.5

ELEMENT = "Load.634a"
634            1           701.62 /_  -38.0
634            0           701.62 /_  142.0

ELEMENT = "Load.634b"
634            2           539.32 /_ -158.7
634            0           539.32 /_   21.3

ELEMENT = "Load.634c"
634            3           538.57 /_   80.5
634            0           538.57 /_  -99.5

ELEMENT = "Load.645"
645            2           86.546 /_ -157.8
645            0           86.546 /_   22.2

ELEMENT = "Load.646"
646            2            65.04 /_ -121.5
646            3            65.04 /_   58.5

ELEMENT = "Load.692"
692            3           54.658 /_  103.7
692            1           54.658 /_  -76.3

ELEMENT = "Load.675a"
675            1           218.92 /_  -27.3
675            0           218.92 /_  152.7

ELEMENT = "Load.675b"
675            2           36.901 /_ -163.3
```

```
675           0           36.901 /_   16.7

ELEMENT = "Load.675c"
675           3            151.1 /_   80.0
675           0            151.1 /_ -100.0

ELEMENT = "Load.611"
611           3           78.285 /_   90.8
611           0           78.285 /_  -89.2

ELEMENT = "Load.652"
652           1           63.622 /_  -39.5
652           0           63.622 /_  140.5

ELEMENT = "Load.670a"
670           1           8.0809 /_  -34.2
670           0           8.0809 /_  145.8

ELEMENT = "Load.670b"
670           2           31.018 /_ -151.3
670           0           31.018 /_   28.7

ELEMENT = "Load.670c"
670           3           55.593 /_   87.1
670           0           55.593 /_  -92.9
```

Assets Package Example: OpenDSS_IEEE13_ASSETS

This instance file demonstrates:

- line segment parameters using WireSpacingInfo, WireInfo, TapeShieldCableInfo, and ConcentricNeutralCableInfo
- transformer parameters using TransformerTankInfo, TransformerEndInfo, ShortCircuitTest, and NoLoadTest

Table 2-2 shows the instance counts using CIMSpy version 2.2. The newer classes are highlighted in **red**; CIMSpy uses an older CIM schema and is able to identify these new classes, but cannot interpret them.

Table 2-2
Instance Counts for the 13-Bus Feeder Using Assets Package

Summary		
CIM Version:		http://iec.ch/TC57/2010/CIM-schema-cim15#
Class	**# of Objects**	**Description**
ACLineSegment	11	A wire or combination of wires, with consistent electrical characteristics, building a single electrical system, used to carry alternating current between points in the power system.
ACLineSegmentPhase	26	Class ACLineSegmentPhase is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
Asset	22	
BaseVoltage	3	Collection of BaseVoltages which is used to verify that the BusbarSection.BaseVoltage and other voltage attributes in the CIM are given a value existing in the collection.
ConcentricNeutralCableInfo	1	Class ConcentricNeutralCableInfo is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
ConnectivityNode	16	Connectivity nodes are points where terminals of conducting equipment are connected together with zero impedance.
CoordinateSystem	1	Class CoordinateSystem is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
EnergyConsumer	9	Generic user of energy - a point of consumption on the power system model
EnergyConsumerPhase	19	Class EnergyConsumerPhase is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
EnergySource	1	A generic equivalent for an energy supplier on a transmission or distribution voltage level.
GeographicalRegion	1	A geographical region of a power system network model.
IEC61970CIMVersion	1	This is the IEC 61970 CIM version number assigned to this UML model file.

Table 2-2 (continued)
Instance Counts for the 13-Bus Feeder Using Assets Package

Summary		
CIM Version:		http://iec.ch/TC57/2010/CIM-schema-cim15#
Class	**# of Objects**	**Description**
Line	1	A component part of a system extending between adjacent substations or from a substation to an adjacent interconnection point.
LoadBreakSwitch	1	A mechanical switching device capable of making, carrying, and breaking currents under normal operating conditions.
LoadResponseCharacteristic	7	Models the characteristic response of the load demand due to to changes in system conditions such as voltage and frequency. This is not related to demand response.
Location	30	
NoLoadTest	3	Class NoLoadTest is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
OverheadWireInfo	4	Class OverheadWireInfo is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
PerLengthPhaseImpedance	1	Class PerLengthPhaseImpedance is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
PhaseImpedanceData	3	Class PhaseImpedanceData is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
PositionPoint	46	Class PositionPoint is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
PowerTransformer	3	An electrical device consisting of two or more coupled windings, with or without a magnetic core, for introducing mutual coupling between electric circuits. Transformers can be used to control voltage and phase shift (active power flow).

Table 2-2 (continued)
Instance Counts for the 13-Bus Feeder Using Assets Package

Summary		
CIM Version:		http://iec.ch/TC57/2010/CIM-schema-cim15#
Class	**# of Objects**	**Description**
PowerTransformerInfo	3	Class PowerTransformerInfo is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
RatioTapChanger	3	Class RatioTapChanger is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
ShortCircuitTest	3	Class ShortCircuitTest is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
ShuntCompensator	2	A shunt capacitor or reactor or switchable bank of shunt capacitors or reactors. A section of a shunt compensator is an individual capacitor or reactor. Negative values for mVArPerSection and nominalMVAr indicate that the compensator is a reactor.
ShuntCompensatorPhase	4	Class ShuntCompensatorPhase is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
SubGeographicalRegion	1	A subset of a geographical region of a power system network model.
SvTapStep	3	State variable for transformer tap step. Normally a profile specifies only one of the attributes "position"or "tapRatio".
TapChangerControl	3	Class TapChangerControl is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
TapChangerInfo	3	Class TapChangerInfo is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
TapeShieldCableInfo	1	Class TapeShieldCableInfo is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#

Table 2-2 (continued)
Instance Counts for the 13-Bus Feeder Using Assets Package

Summary		
CIM Version:		http://iec.ch/TC57/2010/CIM-schema-cim15#
Class	**# of Objects**	**Description**
Terminal	46	An electrical connection point to a piece of conducting equipment. Terminals are connected at physical connection points called "connectivity nodes".
TransformerEndInfo	6	Class TransformerEndInfo is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
TransformerTank	5	Class TransformerTank is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
TransformerTankEnd	10	Class TransformerTankEnd is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
TransformerTankInfo	3	Class TransformerTankInfo is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
WirePosition	30	Class WirePosition is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#
WireSpacingInfo	10	Class WireSpacingInfo is undeclared in http://iec.ch/TC57/2008/CIM-schema-cim13#

Power Flow Solution

```
Max pu. voltage = 1.0684
Min pu. voltage = 0.97508
Total Active Power:   3.57846 MW
Total Reactive Power: 1.73172 Mvar
Total Active Losses:   0.110585 MW, (3.09 %)
Total Reactive Losses: 0.330217 Mvar

CONTROLLED TRANSFORMER TAP SETTINGS

Name            Tap       Min       Max      Step   Position

reg1          1.06250  0.90000  1.10000  0.00625     10
reg2          1.05000  0.90000  1.10000  0.00625     8
reg3          1.06875  0.90000  1.10000  0.00625     11


NODE-GROUND VOLTAGES BY BUS & NODE

Bus       Node    V (kV)    Angle      p.u.    Base kV

sourcebus  1     66.387 /_   30.0   0.99987   115.000
   -       2     66.388 /_  -90.0   0.99989   115.000
   -       3     66.385 /_  150.0   0.99985   115.000
650 .....  1     2.4013 /_    0.0   0.99981     4.160
   -       2     2.4015 /_ -120.0   0.99987     4.160
   -       3     2.4014 /_  120.0   0.99983     4.160
633 .....  1     2.4338 /_   -2.8    1.0133     4.160
   -       2     2.4874 /_ -121.5    1.0356     4.160
   -       3       2.45 /_  117.7    1.0201     4.160
```

```
634 .....  1    0.27414 /_   -3.5   0.98922     0.480
   -       2    0.28189 /_ -121.9    1.0172     0.480
   -       3     0.2775 /_  117.2    1.0014     0.480
rg60 ....  1     2.5511 /_    0.0    1.0622     4.160
   -       2     2.5213 /_ -120.0    1.0498     4.160
   -       3     2.5662 /_  120.0    1.0684     4.160
671 .....  1     2.3599 /_   -5.7   0.98258     4.160
   -       2     2.5094 /_ -121.9    1.0448     4.160
   -       3     2.3703 /_  115.8   0.98691     4.160
645 .....  2     2.4694 /_ -121.6    1.0281     4.160
   -       3     2.4509 /_  117.7    1.0204     4.160
646 .....  2     2.4652 /_ -121.7    1.0264     4.160
   -       3     2.4459 /_  117.8    1.0184     4.160
692 .....  3     2.3703 /_  115.8   0.98691     4.160
   -       1     2.3599 /_   -5.7   0.98258     4.160
   -       2     2.5094 /_ -121.9    1.0448     4.160
675 .....  1     2.3443 /_   -6.0   0.97608     4.160
   -       2      2.515 /_ -122.0    1.0471     4.160
   -       3      2.366 /_  115.9   0.98512     4.160
611 .....  3     2.3626 /_  115.6    0.9837     4.160
652 .....  1     2.3419 /_   -5.7   0.97508     4.160
670 .....  1     2.4149 /_   -3.7    1.0054     4.160
   -       2     2.4954 /_ -121.6     1.039     4.160
   -       3     2.4241 /_  117.0    1.0093     4.160
632 .....  1      2.442 /_   -2.7    1.0168     4.160
   -       2     2.4915 /_ -121.4    1.0374     4.160
   -       3     2.4556 /_  117.7    1.0224     4.160
680 .....  1     2.3599 /_   -5.7   0.98258     4.160
   -       2     2.5094 /_ -121.9    1.0448     4.160
   -       3     2.3703 /_  115.8   0.98691     4.160
684 .....  1     2.3558 /_   -5.7   0.98084     4.160
   -       3     2.3674 /_  115.8   0.98569     4.160


CIRCUIT ELEMENT CURRENTS

(Currents into element from indicated bus)

Power Delivery Elements

  Bus     Phase    Magnitude, A     Angle

ELEMENT = "Vsource.source"
sourcebus     1          17.982 /_  179.2
sourcebus     2            19.8 /_   71.6
sourcebus     3          22.356 /_  -58.4
------------
sourcebus     0          17.982 /_   -0.8
sourcebus     0            19.8 /_ -108.4
sourcebus     0          22.356 /_  121.6

ELEMENT = "Transformer.sub"
sourcebus     1          17.982 /_   -0.8
sourcebus     2            19.8 /_ -108.4
sourcebus     3          22.356 /_  121.6
sourcebus     0               0 /_    0.0
------------
650           1          595.79 /_  151.1
650           2          437.24 /_   39.1
650           3          624.89 /_  -86.4
650           0           154.8 /_   22.8

ELEMENT = "Transformer.xfm1"
633           1          81.725 /_  -38.0
633           2          61.399 /_ -158.8
633           3          62.374 /_   80.3
633           0           20.83 /_  148.1
------------
634           1          708.28 /_  142.0
634           2          532.12 /_   21.2
```

```
634           3           540.57 /_  -99.7
634           0           180.53 /_  -31.9

ELEMENT = "Transformer.reg1"
650           1           595.79 /_  -28.9
650           0           595.79 /_  151.1
------------
rg60          1           560.75 /_  151.1
rg60          0           560.75 /_  -28.9

ELEMENT = "Transformer.reg2"
650           2           437.24 /_ -140.9
650           0           437.24 /_   39.1
------------
rg60          2           416.42 /_   39.1
rg60          0           416.42 /_ -140.9

ELEMENT = "Transformer.reg3"
650           3           624.89 /_   93.6
650           0           624.89 /_  -86.4
------------
rg60          3           584.69 /_  -86.4
rg60          0           584.69 /_   93.6

ELEMENT = "Capacitor.cap1"
675           1            81.28 /_   84.0
675           2           87.196 /_  -32.0
675           3           82.033 /_ -154.1
------------
675           0            81.28 /_  -96.0
675           0           87.196 /_  148.0
675           0           82.033 /_   25.9

ELEMENT = "Capacitor.cap2"
611           3           41.018 /_ -154.4
------------
611           0           41.018 /_   25.6

ELEMENT = "Line.650632"
rg60          1           560.75 /_  -28.9
rg60          2           416.42 /_ -140.9
rg60          3           584.69 /_   93.6
------------
632           1           560.75 /_  151.1
632           2           416.43 /_   39.1
632           3           584.69 /_  -86.4

ELEMENT = "Line.632670"
632           1           480.22 /_  -27.4
632           2           215.67 /_ -134.9
632           3           473.74 /_  100.0
------------
670           1           480.22 /_  152.6
670           2           215.68 /_   45.1
670           3           473.74 /_  -80.0

ELEMENT = "Line.670671"
670           1           472.11 /_  -27.2
670           2           186.64 /_ -132.2
670           3           419.57 /_  101.7
------------
671           1           472.11 /_  152.8
671           2           186.64 /_   47.8
671           3           419.57 /_  -78.3

ELEMENT = "Line.671680"
671           1        0.0032035 /_   92.9
671           2         0.003634 /_  -34.5
671           3        0.0031063 /_ -159.7
------------
```

```
680          1      2.6516E-012 /_ -149.0
680          2      6.8212E-012 /_  180.0
680          3       1.819E-012 /_  180.0

ELEMENT = "Line.632633"
632          1           81.724 /_  -38.0
632          2           61.398 /_ -158.8
632          3           62.373 /_   80.3
------------
633          1           81.725 /_  142.0
633          2           61.399 /_   21.2
633          3           62.374 /_  -99.7

ELEMENT = "Line.632645"
632          3           65.359 /_   58.1
632          2           143.57 /_ -142.4
------------
645          3            65.36 /_ -121.9
645          2           143.57 /_   37.6

ELEMENT = "Line.645646"
645          3            65.36 /_   58.1
645          2            65.36 /_ -121.9
------------
646          3            65.36 /_ -121.9
646          2            65.36 /_   58.1

ELEMENT = "Line.692675"
692          1           206.89 /_   -5.9
692          2           68.889 /_  -55.1
692          3           122.81 /_  112.3
------------
675          1           206.89 /_  174.1
675          2           68.868 /_  124.8
675          3           122.81 /_  -67.7

ELEMENT = "Line.671684"
671          1            62.68 /_  -39.6
671          3           35.445 /_  120.9
671          3           35.816 /_  122.7
------------
684          1            62.68 /_  140.4
684          3           35.445 /_  -59.1
684          3           35.816 /_  -57.3

ELEMENT = "Line.684611"
684          3           71.253 /_  121.8
------------
611          3           71.252 /_  -58.2

ELEMENT = "Line.684652"
684          1            62.68 /_  -39.6
------------
652          1           62.698 /_  140.4

ELEMENT = "Line.671692"
671          1           230.78 /_  -18.8
671          2           68.889 /_  -55.1
671          3           177.01 /_  109.6
------------
692          1           230.78 /_  161.2
692          2           68.889 /_  124.9
692          3           177.01 /_  -70.4

= = = = = = = = = = = = = = = = = = =  = = = = = = = = = = = =  = =

Power Conversion Elements

  Bus     Phase    Magnitude, A     Angle
```

```
ELEMENT = "Load.671"
671           1           183.72 /_  -33.6
671           2           183.87 /_ -153.6
671           3           183.88 /_   86.4

ELEMENT = "Load.634a"
634           1           708.28 /_  -38.0
634           0           708.28 /_  142.0

ELEMENT = "Load.634b"
634           2           532.12 /_ -158.8
634           0           532.12 /_   21.2

ELEMENT = "Load.634c"
634           3           540.55 /_   80.3
634           0           540.55 /_  -99.7

ELEMENT = "Load.645"
645           2           85.451 /_ -157.9
645           0           85.451 /_   22.1

ELEMENT = "Load.646"
646           2            65.36 /_ -121.9
646           3            65.36 /_   58.1

ELEMENT = "Load.692"
692           3           54.658 /_  103.4
692           1           54.658 /_  -76.6

ELEMENT = "Load.675a"
675           1           222.19 /_  -27.4
675           0           222.19 /_  152.6

ELEMENT = "Load.675b"
675           2           36.059 /_ -163.5
675           0           36.059 /_   16.5

ELEMENT = "Load.675c"
675           3           151.83 /_   79.7
675           0           151.83 /_ -100.3

ELEMENT = "Load.611"
611           3           78.285 /_   90.4
611           0           78.285 /_  -89.6

ELEMENT = "Load.652"
652           1           62.697 /_  -39.6
652           0           62.697 /_  140.4

ELEMENT = "Load.670a"
670           1           8.1675 /_  -34.2
670           0           8.1675 /_  145.8

ELEMENT = "Load.670b"
670           2            30.52 /_ -151.5
670           0            30.52 /_   28.5

ELEMENT = "Load.670c"
670           3           55.827 /_   86.8
670           0           55.827 /_  -93.2
```

Section 3: Interop Test Documentation

The interoperability tests involve data exchange using defined "profiles", which are selections of distribution CIM classes and attributes to suit a particular purpose. The test must be witnessed and verified by a third party (electric utility) witness in order to pass. For the 2011 part 4 (asset) and part 13 (power flow model) tests, several profiles were specified:

- Functional – the base connectivity and identification
- Electrical Properties – values for power flow models
- Asset – includes "data sheet" values for equipment libraries used in power flow
- Geographic – object locations
- Topology – collections of terminals into nodes for topology processing; this is not necessary for a power flow model, but can be important in state estimation
- State Variable – exchange switch states, tap settings, and solved voltage/current values

A power flow model relies on the first two or three of these profiles. For 2011 only, all six were combined into a single "combined" profile, and OpenDSS was tested only against the combined profile. The 2009 test only used combined profiles. In the future, OpenDSS will have to support the more granular profiles listed above, which means that model import could occur in several steps.

The 2011 tests used five instance files. Four were provided by EdF, and supported power flow solutions. A fifth file was provided by GE, with only functional, geographic, and partial asset information.

During the test, two instance files of the IEEE 13-bus feeder were provided by exporting from OpenDSS. However, these were not part of the formal test protocol and only one participated attempted to import them.

Partial Import Tests

It is not possible for OpenDSS to pass the CIM basic import test, because the OpenDSS importer ignores any data that isn't relevant to a power flow solution. OpenDSS does not purport to be a full DMS. In 2011, these tests were "passed with errors" because a partial match of instance counts can be obtained, and also, OpenDSS properly imports the data relevant to power flow.

OpenDSS collects ACLineSegments, LoadBreakSwitches, Disconnectors, Fuses, and Breakers into "Lines" for the power flow solution. In OpenDSS, all power delivery components have a built-in switch at each terminal. A Line, however, can be given special status as a switch for display purposes, and that is how CIM switches are represented in OpenDSS.

OpenDSS instance counts for the EdF Aigue system are:

```
Format: DSS Class Name = Instance Count
Solution = 1
LineCode = 3
LoadShape = 1
TShape = 0
PriceShape = 0
XYcurve = 0
GrowthShape = 1
TCC_Curve = 4
Spectrum = 7
WireData = 0
CNData = 0
TSData = 0
LineGeometry = 0
LineSpacing = 0
XfmrCode = 0
Line = 395
Vsource = 1
Isource = 0
Load = 101
Transformer = 1
RegControl = 0
Capacitor = 1
Reactor = 0
CapControl = 0
Fault = 0
Generator = 4
GenDispatcher = 0
Storage = 0
StorageController = 0
Relay = 0
Recloser = 0
Fuse = 0
SwtControl = 0
PVSystem = 0
Monitor = 0
EnergyMeter = 0
Sensor = 0
```

OpenDSS instance counts for the EdF C10 system are:

```
Format: DSS Class Name = Instance Count
Solution = 1
LineCode = 3
```

```
LoadShape = 1
TShape = 0
PriceShape = 0
XYcurve = 0
GrowthShape = 1
TCC_Curve = 4
Spectrum = 7
WireData = 0
CNData = 0
TSData = 0
LineGeometry = 0
LineSpacing = 0
XfmrCode = 0
Line = 169
Vsource = 2
Isource = 0
Load = 52
Transformer = 4
RegControl = 0
Capacitor = 6
Reactor = 0
CapControl = 0
Fault = 0
Generator = 2
GenDispatcher = 0
Storage = 0
StorageController = 0
Relay = 0
Recloser = 0
Fuse = 0
SwtControl = 0
PVSystem = 0
Monitor = 0
EnergyMeter = 0
Sensor = 0
```

OpenDSS instance counts for the EdF Rural system are:

```
Format: DSS Class Name = Instance Count
Solution = 1
LineCode = 3
LoadShape = 1
TShape = 0
PriceShape = 0
XYcurve = 0
GrowthShape = 1
TCC_Curve = 4
Spectrum = 7
WireData = 0
CNData = 0
TSData = 0
LineGeometry = 0
LineSpacing = 0
XfmrCode = 0
Line = 4938
Vsource = 4
```

```
Isource = 0
Load = 1093
Transformer = 8
RegControl = 0
Capacitor = 4
Reactor = 0
CapControl = 0
Fault = 0
Generator = 2
GenDispatcher = 0
Storage = 0
StorageController = 0
Relay = 0
Recloser = 0
Fuse = 0
SwtControl = 0
PVSystem = 0
Monitor = 0
EnergyMeter = 0
Sensor = 0
```

OpenDSS instance counts for the EdF Urban system are:

```
Format: DSS Class Name = Instance Count
Solution = 1
LineCode = 3
LoadShape = 1
TShape = 0
PriceShape = 0
XYcurve = 0
GrowthShape = 1
TCC_Curve = 4
Spectrum = 7
WireData = 0
CNData = 0
TSData = 0
LineGeometry = 0
LineSpacing = 0
XfmrCode = 0
Line = 8061
Vsource = 5
Isource = 0
Load = 2056
Transformer = 13
RegControl = 0
Capacitor = 5
Reactor = 0
CapControl = 0
Fault = 0
Generator = 0
GenDispatcher = 0
Storage = 0
StorageController = 0
Relay = 0
Recloser = 0
Fuse = 0
```

```
SwtControl = 0
PVSystem = 0
Monitor = 0
EnergyMeter = 0
Sensor = 0
```

Power Flow Tests

Figures 3-1 through 3-10 document power flow solutions on the four instance files provided by EdF; these were all "passed" during the test. Some of these EdF systems include local generation. Figures 3-7 and 3-8 illustrate the OpenDSS component plots used to verify generator output against the EdF base case solution. The validation process can be streamlined in future tests by adding transmission-style load flow outputs (e.g. generation, shunt, and load reports) to OpenDSS.

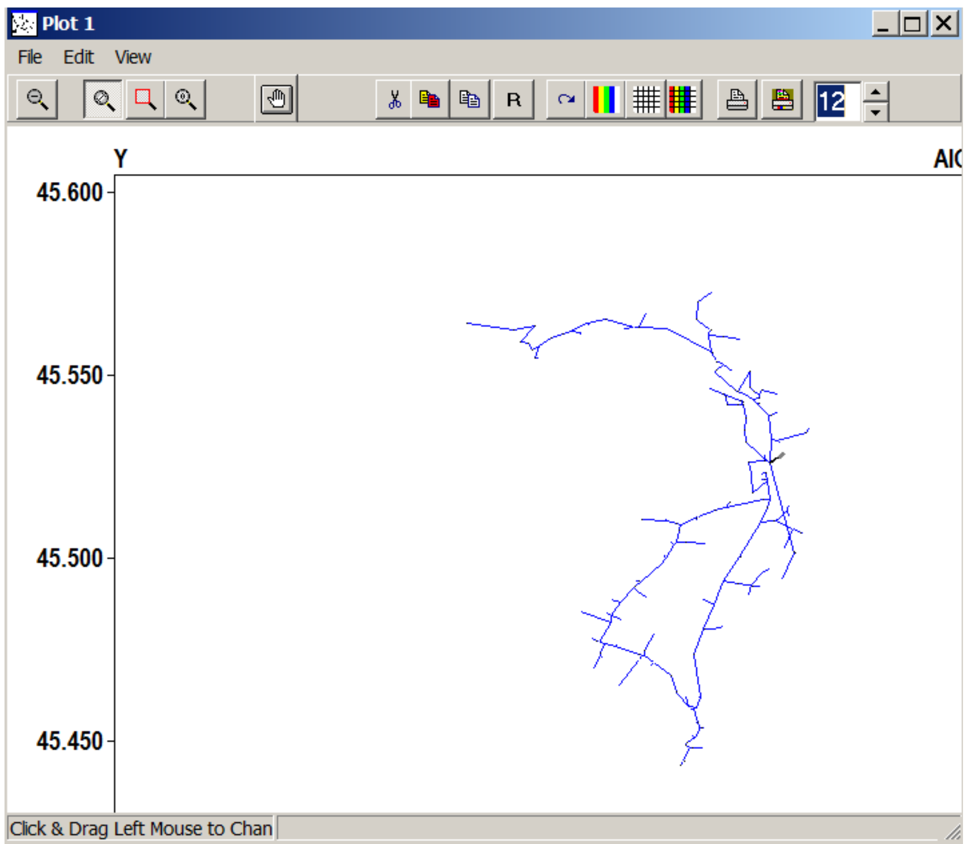


Figure 3-1
Circuit Plot of EdF Aigue System

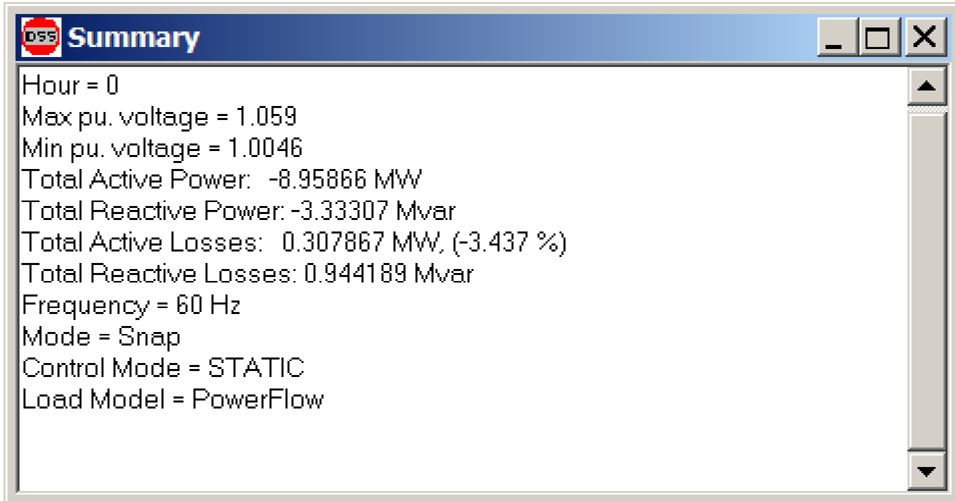


Figure 3-2
OpenDSS Solution Summary for Aigue

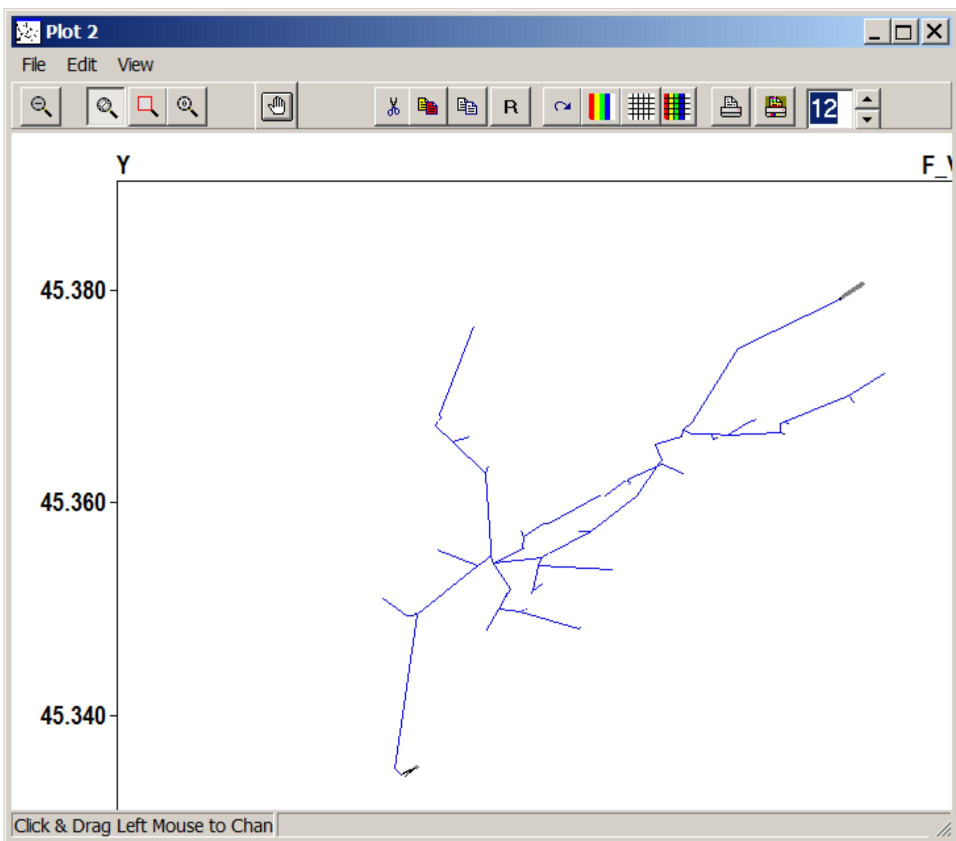


Figure 3-3
Circuit Plot of EdF C10 System

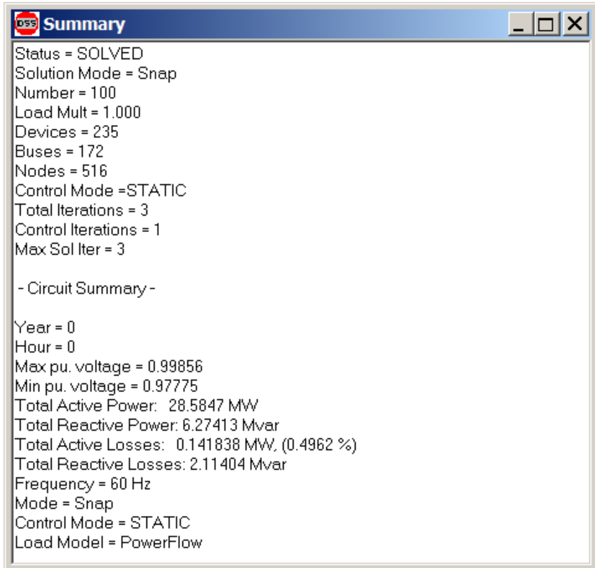


Figure 3-4
OpenDSS Solution Summary for C10

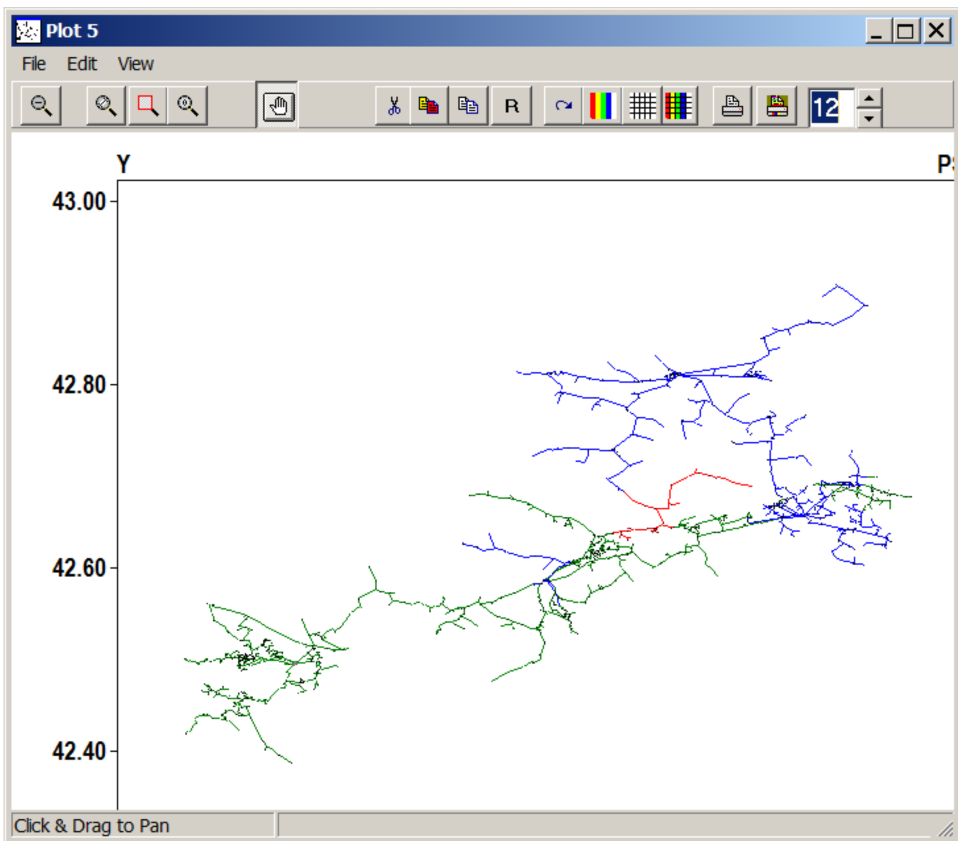


Figure 3-5
Circuit Plot of EdF Rural System

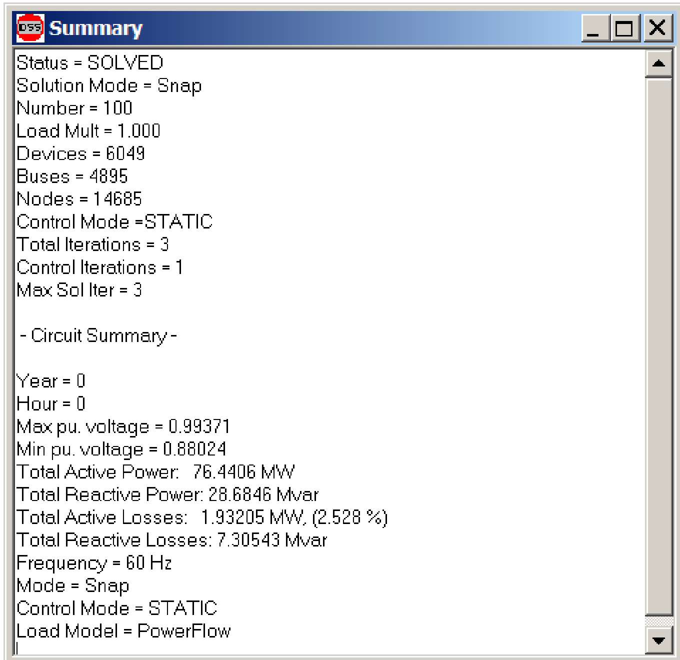


Figure 3-6
OpenDSS Solution Summary for EdF Rural System

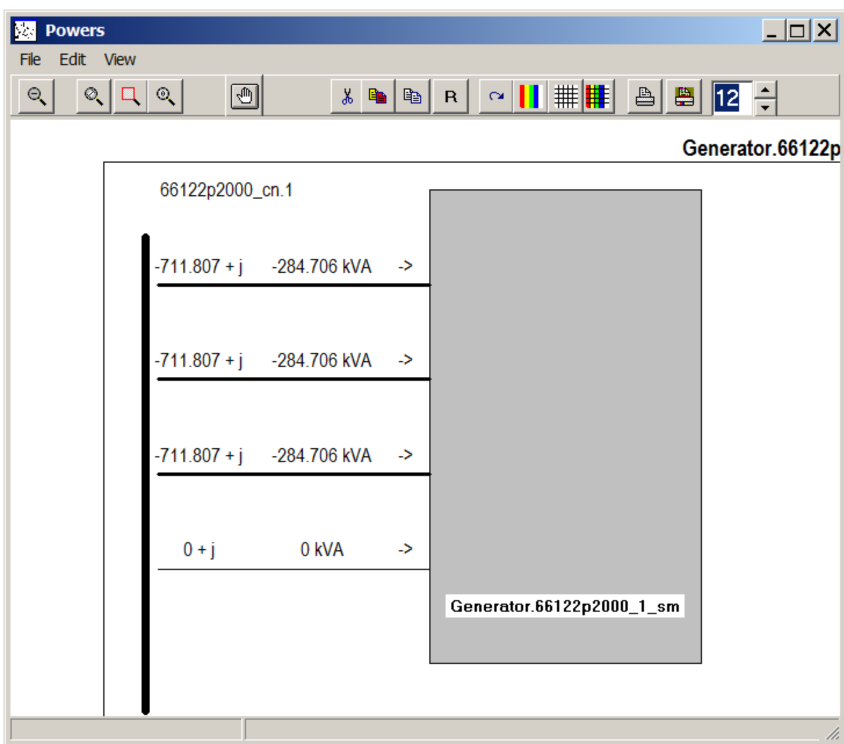


Figure 3-7
First DG Solution for EdF Rural System

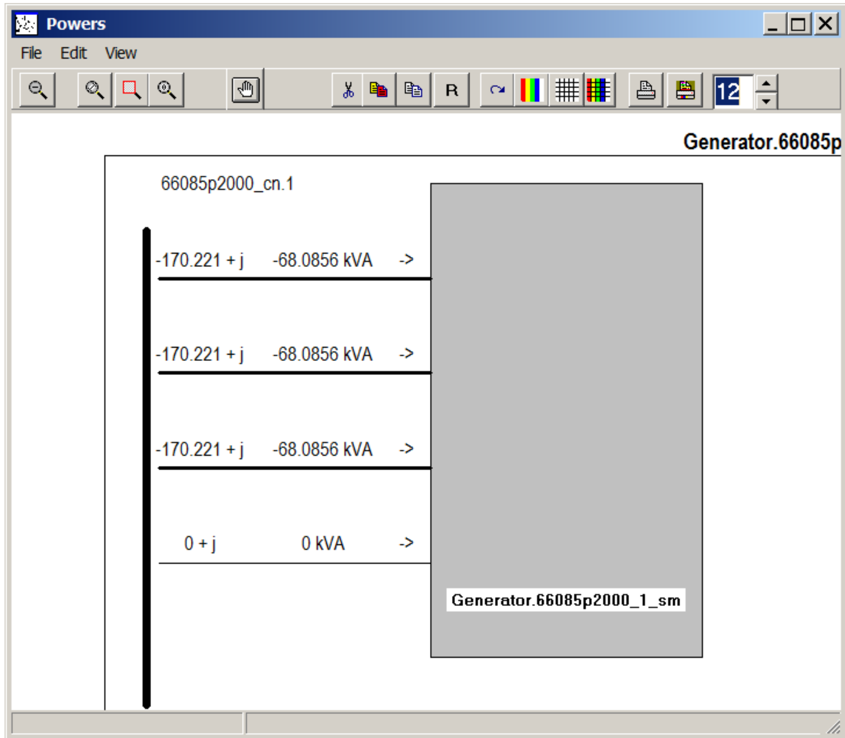


Figure 3-8
Second DG Solution for EdF Rural System

EdF Urban Power Flow

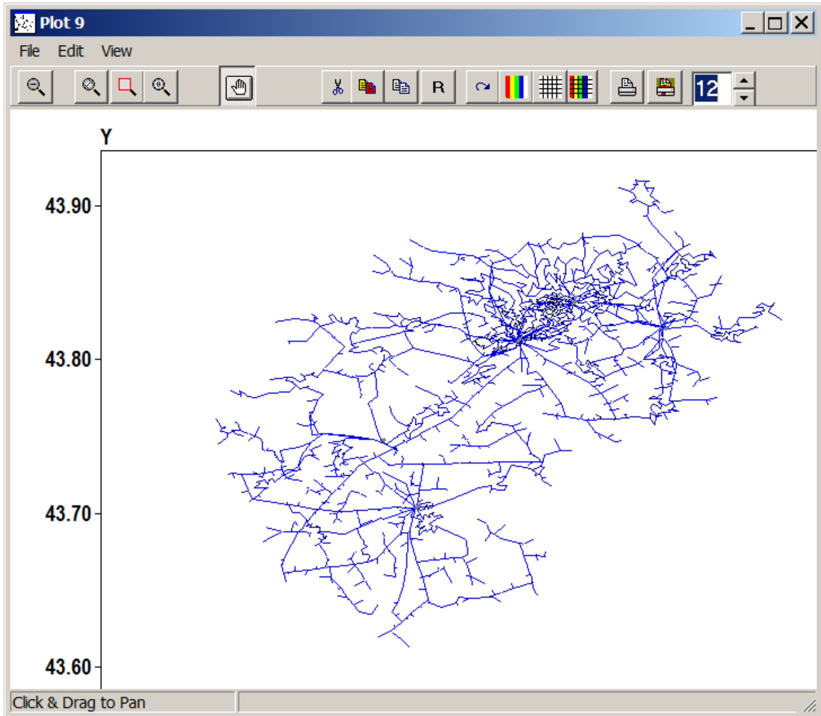


Figure 3-9
Circuit Plot for EdF Urban System

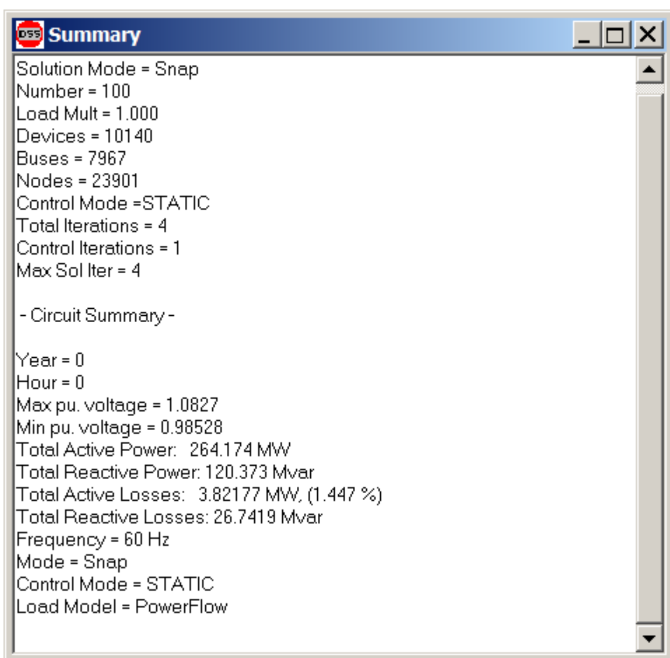


Figure 3-10
OpenDSS Solution Summary for EdF Urban System

Partial GIS Import Test

Instance counts for the GE CIRC system are:

```
Format: DSS Class Name = Instance Count
Solution = 1
LineCode = 3
LoadShape = 1
TShape = 0
PriceShape = 0
XYcurve = 0
GrowthShape = 1
TCC_Curve = 4
Spectrum = 7
WireData = 0
CNData = 0
TSData = 0
LineGeometry = 0
LineSpacing = 0
XfmrCode = 0
Line = 1711
Vsource = 1
Isource = 0
Load = 593
Transformer = 136
RegControl = 0
Capacitor = 0
Reactor = 0
CapControl = 0
Fault = 0
Generator = 0
GenDispatcher = 0
Storage = 0
StorageController = 0
Relay = 0
Recloser = 0
Fuse = 0
SwtControl = 0
PVSystem = 0
Monitor = 0
EnergyMeter = 0
Sensor = 0
```

These matched the input to the extent possible, so the test was "passed with errors". As before, OpenDSS collects ACLineSegments, LoadBreakSwitches, Disconnectors, Fuses, and Breakers into "Lines" for the power flow solution. In OpenDSS, all power delivery components have a built-in switch at each terminal. A Line, however, can be given special status as a switch for display purposes, and that is how CIM switches are represented in OpenDSS.

In the 2009 test, an OpenDSS power flow solution produced a visual no-load circuit plot on GE's instance file, because the 2009 system included geographic locations. The geographic locations were not provided in GE's 2011 example, so the OpenDSS power flow executed, but could not produce a circuit plot.

Figure 3-11 illustrates partial data extracted for a single-phase transformer tank, particularly the voltage and kVA ratings. The impedance values did not come from the CIM model; they are defaults.

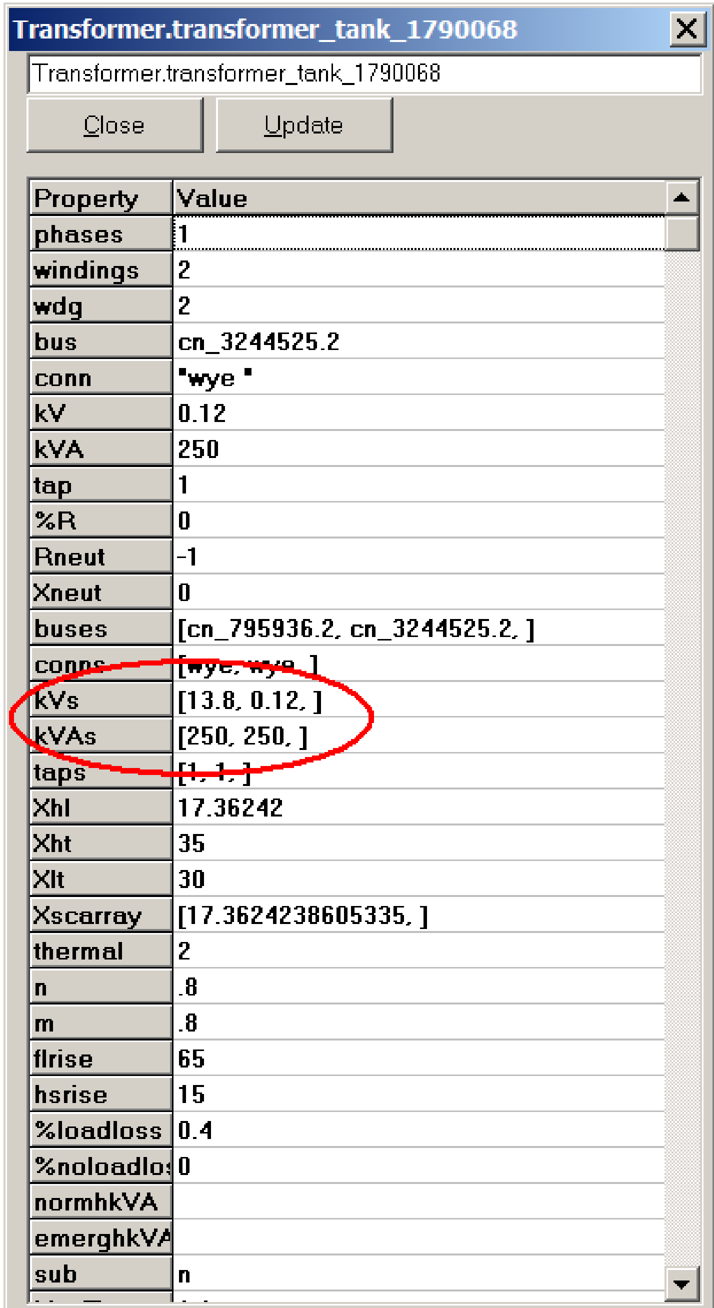


Figure 3-11
Transformer Rating Data Obtained from GE Circ Instance File

Incremental Power Flow Test

In this test, EdF made a change (switch opening) in the Aigue instance file, which would affect the power flow solution. OpenDSS passed this test. Comparing Figures 3-12 and 3-14, there is very little change in the solution summary statistics. However, comparing Figures 3-13 and 3-15 clearly shows that one additional branch has zero voltage, due to the switch opening. This test highlighted the advantage of OpenDSS graphical output options, as compared to the transmission style reports used by some other participants.

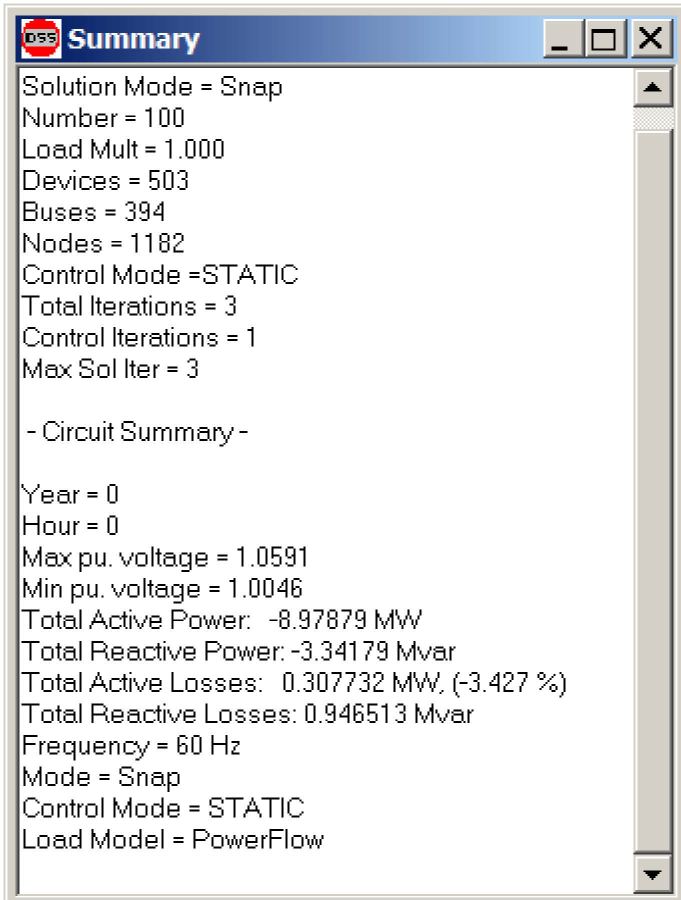


Figure 3-12
OpenDSS Solution Summary for Aigue Before Switch Opens

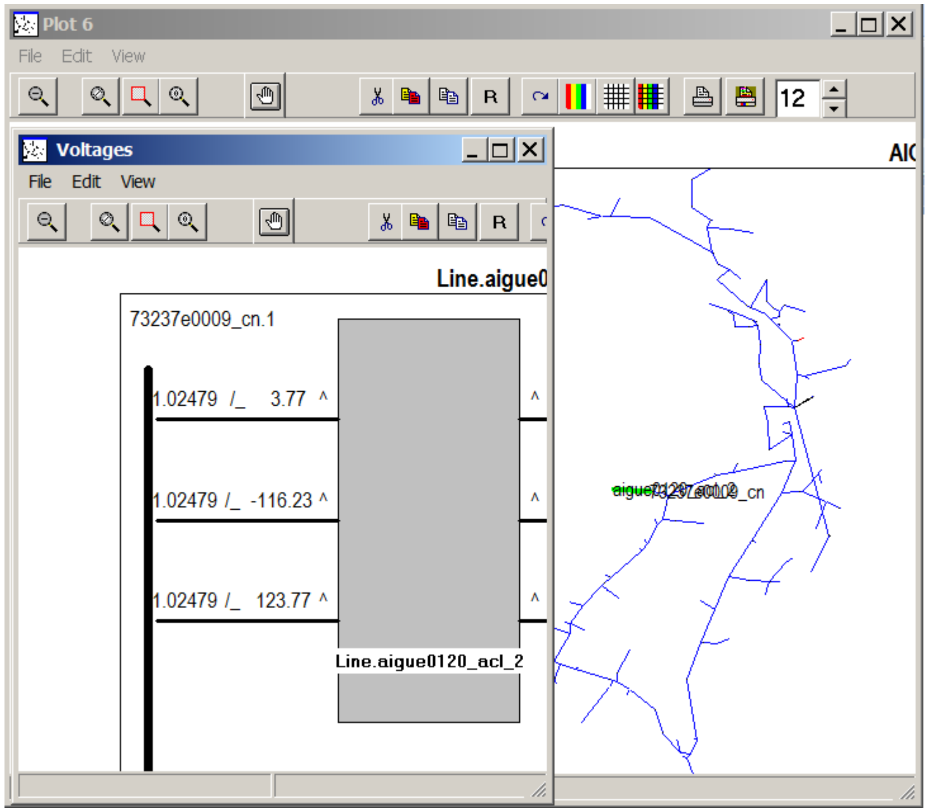
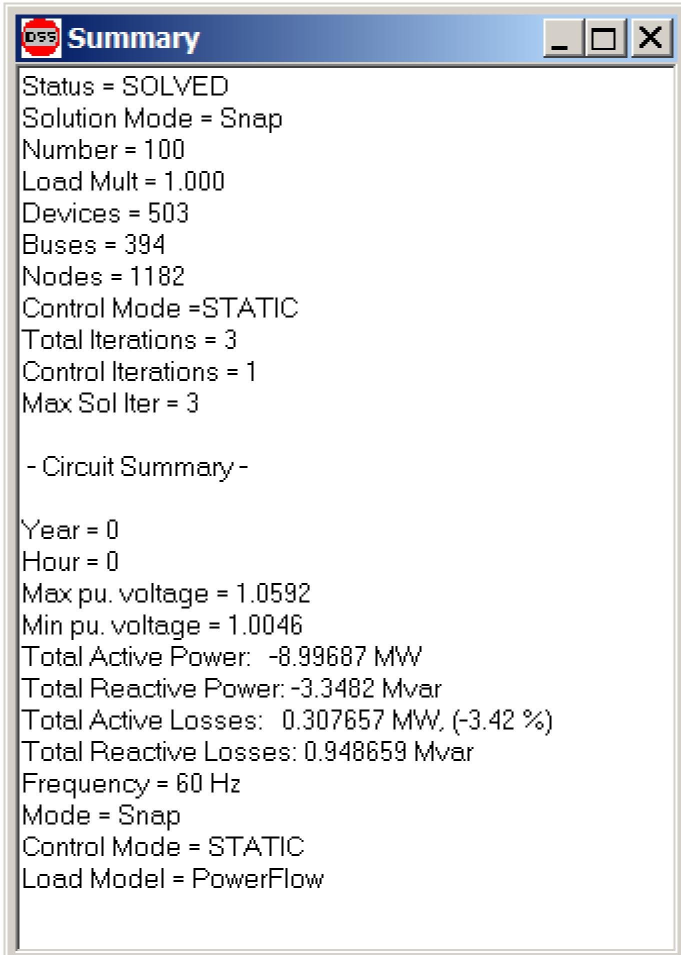


Figure 3-13
Aigue Circuit Plot and Branch Voltage Before Switch Opening

After Opening Switch – line segment aigue0120_acl_2 now has zero voltage


Summary

Status = SOLVED
Solution Mode = Snap
Number = 100
Load Mult = 1.000
Devices = 503
Buses = 394
Nodes = 1182
Control Mode =STATIC
Total Iterations = 3
Control Iterations = 1
Max Sol Iter = 3

- Circuit Summary -

Year = 0
Hour = 0
Max pu. voltage = 1.0592
Min pu. voltage = 1.0046
Total Active Power: -8.99687 MW
Total Reactive Power: -3.3482 Mvar
Total Active Losses: 0.307657 MW, (-3.42 %)
Total Reactive Losses: 0.948659 Mvar
Frequency = 60 Hz
Mode = Snap
Control Mode = STATIC
Load Model = PowerFlow

Figure 3-14
Aigue Solution Summary after Switch Opening

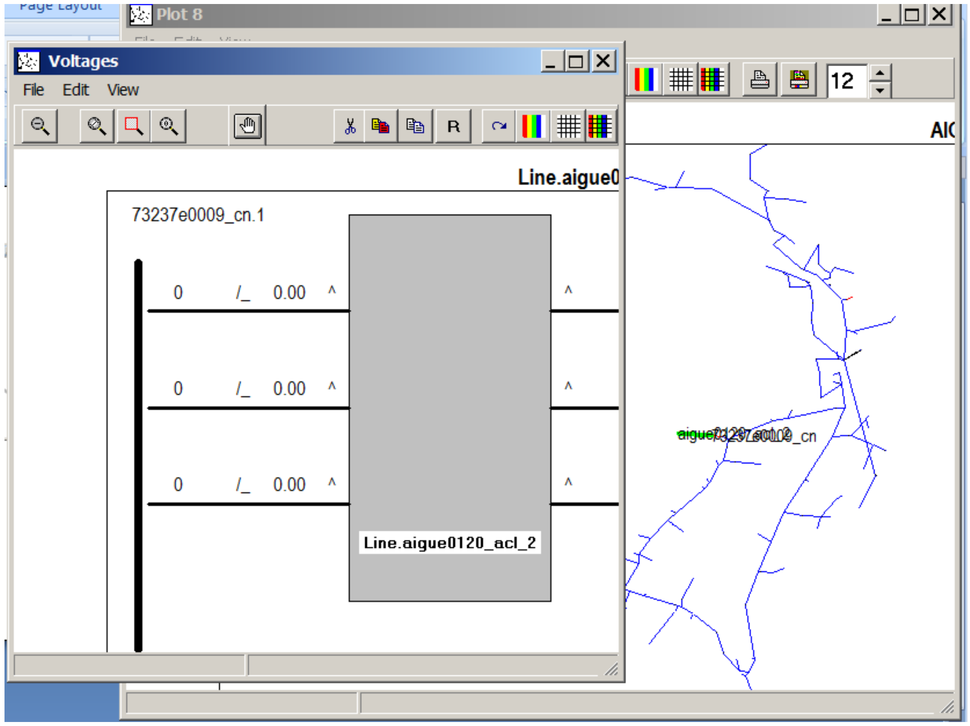


Figure 3-15
Aigue Circuit Plot and Branch Voltage After Switch Opening

Interoperability Test with Open Grid Systems

Open Grid Systems (OGS), using the CIMPHONY product, was able to import and re-export all of the instance files, including the two IEEE 13-bus test files. OpenDSS was then able to import the four EdF instance files from OGS, and generate the matching power flow solutions in Figures 3-16 through 3-19. This is more of a credit to OGS than to OpenDSS, but it does demonstrate interoperability between two vendors, which is a key focus point of these tests.

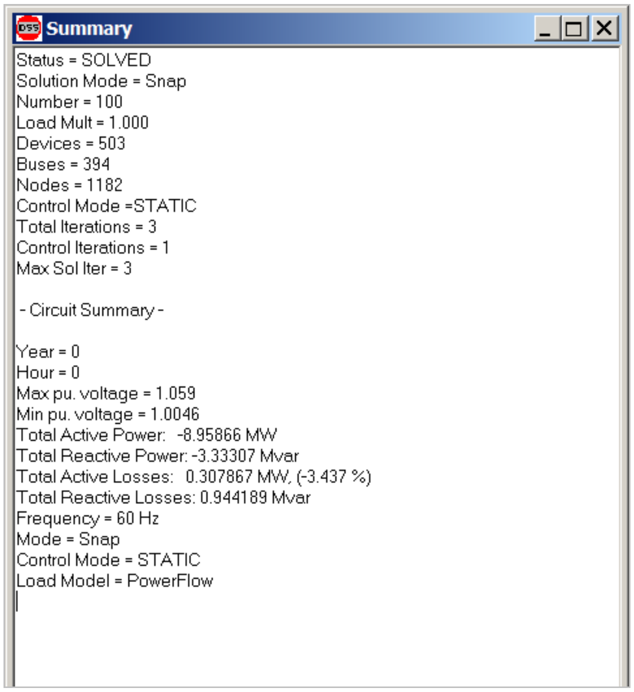


Figure 3-16
OpenDSS Solution for OGS Export of EdF Aigue

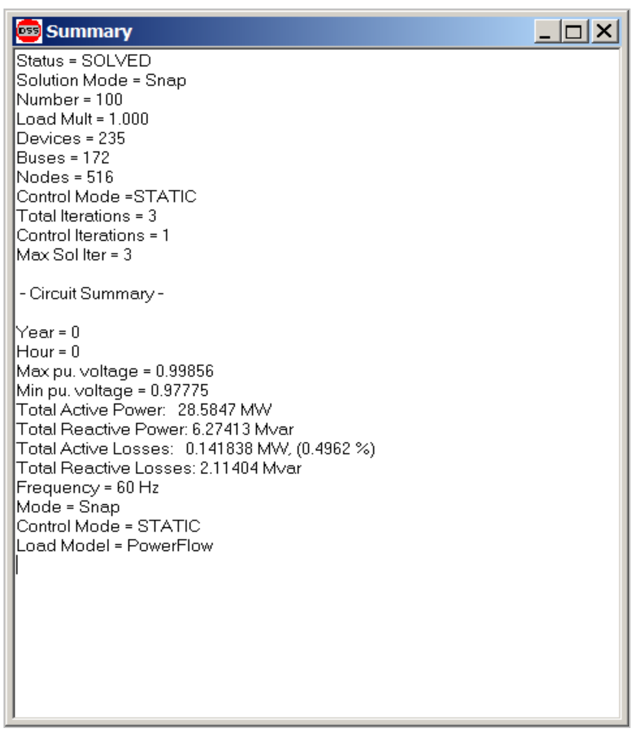


Figure 3-17
OpenDSS Solution for OGS Export of EdF C10

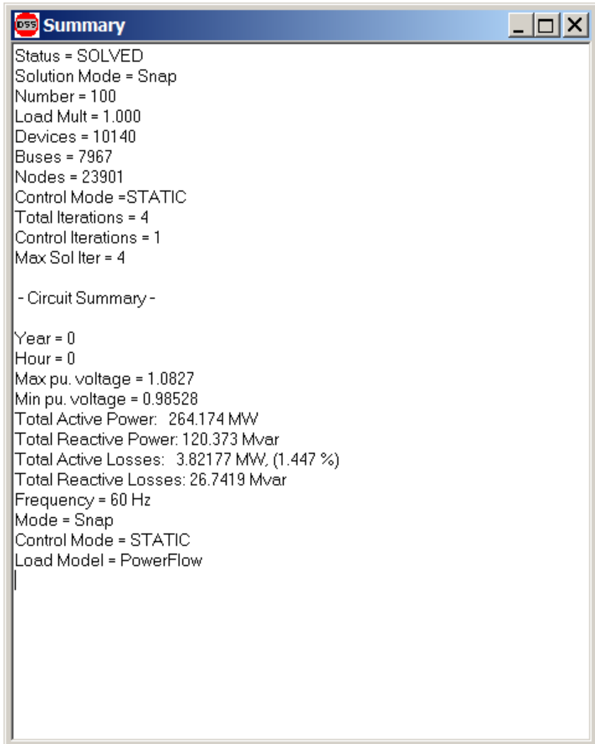


Figure 3-18
OpenDSS Solution for OGS Export of EdF Urban

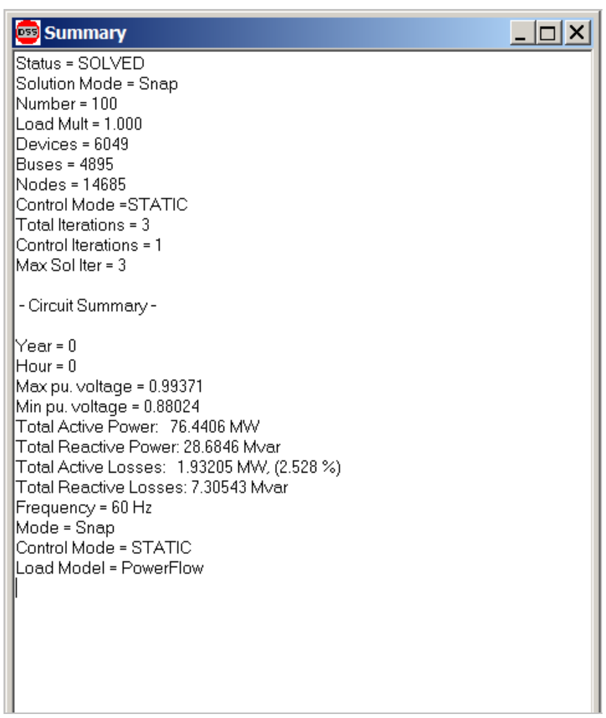


Figure 3-19
OpenDSS Solution for OGS Export of EdF Rural

Section 4: Model and Profile Issues

This section describes issues identified in the 2011 interoperability tests, relating to the basic distribution CIM model, and also relating to the classes and attributes selected for specific profiles. Section 5 of this report documents most of the classes and attributes mentioned in these issues.

Model Issues

Table 4-1 and Figure 4-1 document five distribution CIM issues submitted to the modeling team after the 2011 interoperability tests. The “sketch” mentioned in the fourth issue appears in Figure 4-1. The last column in Table 4-1 proposes a solution for each issue, but these won’t all necessarily be adopted by the modeling team. For example, Section 5 documents workarounds for the first issue on delta-connected single-phase loads, and for the last two issues on line conductor phasing.

Table 4-1
CIM Issues Identified During the 2011 Interoperability Tests

Tom McDermott	4/5/2011	61970-552	Support phase-to-phase (delta connected) unbalanced loads and capacitors.	Add AB, AC, BC, s1s2 to SinglePhaseKind.
Tom McDermott	4/5/2011	61970-552	Unbalanced transformer connections were implemented in 2009 with constructs like AB phasing in conjunction with I (independent) winding connection and the phase clock angles. Since then a different pattern has been adopted in DCIMPhaseModel for all other types of component that could be unbalanced. Two patterns must be learned and implemented.	Extend the DCIMPhaseModel with a new TransformerTankEndPhase, having two attributes phase:SinglePhaseKind and isReversedPolarity:Boolean.
Tom McDermott	4/5/2011	61970-552	Can't model single-phase sources or machines, nor single-phase PV inverters.	Add EnergySourcePhase, AsynchronousMachinePhase, SynchronousMachinePhase, StaticVarCompensatorPhase, SeriesCompensatorPhase, FrequencyConverterPhase.
Tom McDermott	4/5/2011	61970-552	Should not have to create artificial extra WireSpacingInfo to implement transposition. The sketches below show 3 typical WireSpacingInfo entries with numbered wire positions. In the UML each numbered WirePosition must have a phase:SinglePhaseKind (A,B,C, or N). To cover all phasing combinations, one must potentially create 15 WireSpacingInfo instances from the 3 real ones.	Add conductorNumber:Integer to both ACLineSegmentPhase and WirePosition. This corresponds to use of endNumber with TransformerEnds. WireInfo can then be associated with ACLineSegmentPhase, and mapped to the correct location on the pole/crossarm using conductorNumber.
Tom McDermott	4/5/2011	61970-552	Support transposition on PerLengthPhaseImpedance, without having to create artificial extra instances.	Would require new conductorNumber:Integer attribute on ACLineSegmentPhase to be mapped onto a row/column of the matrix.

Phasing Options:			
ABCN	ABN		AN
ACBN	ACN		BN
BACN	BAN		CN
BCAN	CAN		
CABN	BCN		
CBAN	CBN		

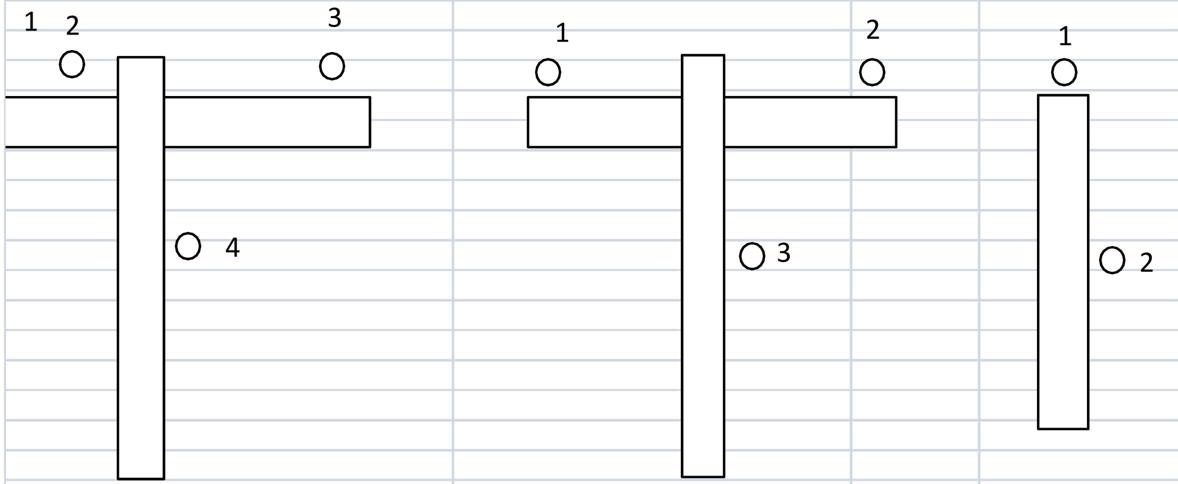


Figure 4-1
WireSpacingInfo Permutations for Phasing

Profile Issues

These are suggested changes to the Combined.owl profile, based on exporting two versions of the unbalanced IEEE 13-bus system from OpenDSS. One version uses PerLengthImpedance for ACLineSegments, along with TransformerCoreAdmittance and TransformerMeshImpedance for transformers. The other uses AssetInfo classes. Some of these changes would propagate to Assets, ElectricalProperties, and Functional profiles.

A CIM profile uses only selected classes and attributes from the complete model. The suggestions in this section describe changes to the selected classes and attributes for power flow model exchange.

1. ACLineSegment
   a. PerLengthImpedance should be allowed for use with AssetInfo classes. Even without AssetInfo, this association is needed to support PerLengthPhaseImpedance and PerLengthSequenceImpedance as electrical properties. Therefore, these classes should be available in the ElectricalProperties profile
   b. bch, r, x should not be required, since PerLengthImpedance may be used
2. ConcentricNeutralCableInfo – add the neutralStrandGmr, neutralStrandRDC20, and neutralStrandRadius attributes
3. ConnectivityNode – allow isolated nodes at circuit branch end points
4. EnergyConsumerPhase – need pfixed, phase, and qfixed to specify unbalanced loads
5. EnergySource
   a. nominalVoltage to allow specified operation at other than 1.0 per-unit voltage
   b. r0 and x0 needed for unbalanced load flow
6. PowerTransformer – add vectorGroup for descriptive summary of the connection
7. ProtectedSwitch – breakingCapacity should be optional; not needed for power flow
8. RegulatingControl – add this class to the profile
   a. Terminal association and monitoredPhase attribute to describe which voltage is controlled
   b. targetRange and targetValue for the voltage setpoint and bandwidth
9. ShuntCompensator – need nomU to specify rated kVAR along with bPerSection
10. TapChanger
    a. Add ltcFlag and regulationStatus for status information

    b. Add initialDelay and subsequentDelay settings

11. TapChangerControl – entire class needs to be added to the profile; it has the line drop compensator settings and the “first house” (limitVoltage) setting
12. Terminal – TopologicalNode association should not be required if using connectivity nodes
13. WireSpacingInfo – the Impedances association should not be required. While nice to have, it is often unavailable at the time of data exchange.

Section 5: IEC 61968-11 Contributions

This section includes clauses of IEC 61968-11 that MelTran originally drafted, related to unbalanced feeder modeling and the interoperability tests. The WG model manager maintains the entire document and the UML diagrams. Other parties have also contributed editorial changes and clarifications. The IEC will hold copyright on this material and eventually publish it in IEC standards. An earlier version was published in 2010, but there are major changes in the pipeline for 2011. The material is included in this report, so that EPRI distribution modeling experts have a chance to review and comment before final adoption in IEC standards.

The material does not stand alone, although it is useful to an understanding of the IEEE 13-bus instance files. Full use of the distribution CIM requires access to the Enterprise Architect model files, the current version of IEC 61970-301, and possibly other IEC standards.

This section uses IEC font conventions.

Single-Phase and Unbalanced Loads

Figure 5-1 shows the classes available to model distribution loads, which are often unbalanced among the three phases at a location. In some cases, single-phase and two-phase loads will occur. (TODO: figure: Add Terminal and SinglePhase enum).

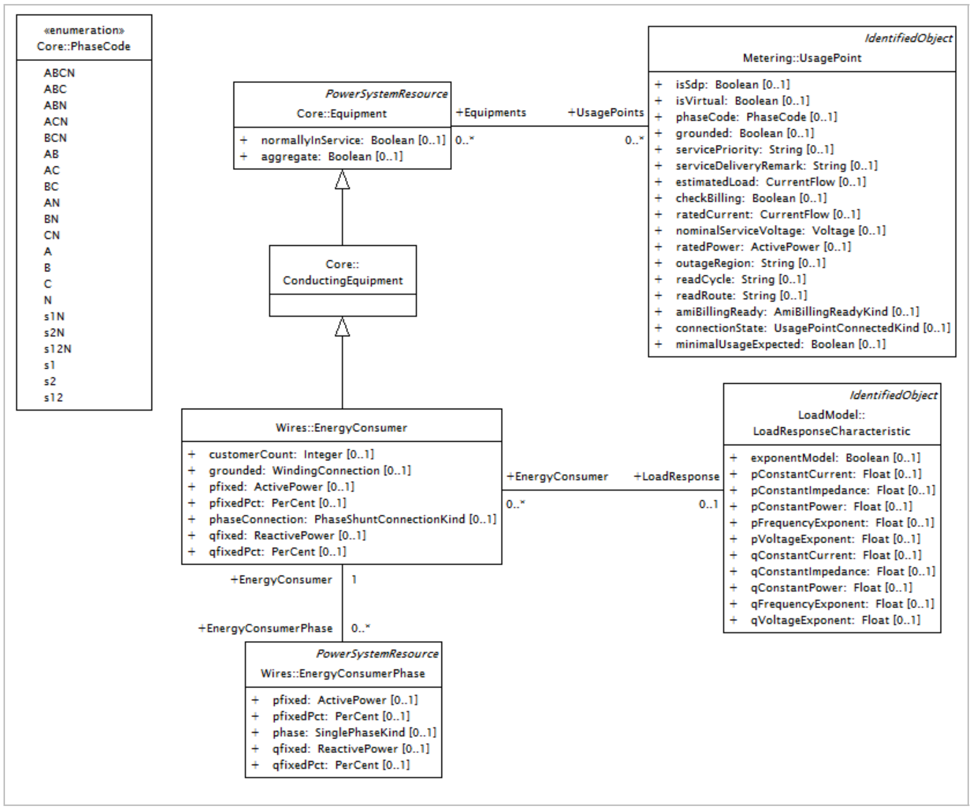


Figure 5-1
Load Model

EnergyConsumer class should be used to instantiate the load model, which can optionally be associated with a meter through **UsagePoint EnergyConsumer** inherits from **ConductingEquipment** which is associated with **Terminal. Terminal** has **phases** attribute, which may be assigned a **PhaseCode** enumeration literal. For example, **AN** describes a single-phase load from A to neutral, **BC** describes a single-phase load from B to C, **ABCN** describes a three-phase wye-grounded load, **ABC** describes a three-phase delta-connected load, etc.

The **phaseConnection** attribute of EnergyConsumer should be **Y** or **Yn** for a wye connected or phase-to-neutral load; it should be **D** for a delta or phase-to-phase load. For a balanced three-phase load, specify the total real and reactive power on **EnergyConsumer** attributes, for equal distribution among the three phases.

If a three-phase load is unbalanced, it can still be modelled with the same **EnergyConsumer** instance, now referring to three additional **EnergyConsumerPhase** instances, each representing one phase with its **phase** attribute. This makes it possible to use an **EnergyConsumer** with its identity for unbalanced modelling as well. With single-phase or two-phase loads, the model should also include correctly phased conductors or transformers that establish connectivity back to the source. The total real and reactive power need not be specified on **EnergyConsumer** attributes, because they can be derived from the distribution among phases specified on **EnergyConsumerPhase** attributes. Each **phase** can be **A, B, C, S1**, or **S2**, because **N** does not apply to this case. For a delta or phase-to-phase connected load, which is indicated with **D** for **phaseConnection**:

- use **A** for the load connected between phases **A** and **B**
- use **B** for the load connected between phases **B** and **C**
- use **C** for the load connected between phases **C** and **A**
- use **S1** for the load connected between phases **S1** and **S2**

The load model could be some combination of constant current, constant power, or constant impedance. In that case, an instance of **LoadResponseCharacteristic** should be associated with the **EnergyConsumer** instance.

Distribution Line Segments

Figure 5-2 shows the classes available to model AC line segments (i.e., conductors). There are three ways to describe the impedance parameters of an **ACLineSegment** using only the **Wires** package, and a fourth way making use of the **AssetInfo** package.

In order to represent a 1-phase, 2-phase, or unbalanced 3-phase line using different wires, one **ACLineSegmentPhase** instance should be associated with each phase or neutral wire retained in the model. The applicable **phase** attribute values are **A, B, C** for three-phase lines, **S1** and **S2** for single-phase secondary circuits, and **N** for cases where the neutral wire is modelled. Each **ACLineSegmentPhase** has an optional association to **WireInfo**, described later in more detail, but all other characteristics come from its associated **ACLineSegment**.

For an unbalanced line, the calculated impedances reflect physical wire positions on the tower or pole, and these positions are typically ordered from left-to-right and top-to-bottom based on their horizontal and height coordinates. For example, consider a pole with crossarm having 3 wire positions numbered 1 (left-most), 2 (offset from center), and 3 (right-most), Impedances could be calculated with phase A's wire in position 1, phase B's in position 2, and phase C's in position 3. A different **ACLineSegment** might use the same pole type and wires, but with phase C's wire in position 1, phase B's in position 2, and phase A's in position 3. The calculated impedance parameters will be different than the first case. They are mathematically related by impedance matrix row and column operations, but such operations are not supported in the CIM, nor is there a concept of ordering phases in CIM. This means that each different physical phase ordering requires a different impedance description in the CIM. This consideration affects the use of **PerLengthPhaseImpedance** and **WireSpacingInfo** in Figure 5-2.

Transposed lines can be modelled with a different **ACLineSegment** for each section, and a different impedance description for each section as just discussed.

The impedance description should use either **PerLengthPhaseImpedance** or **WireSpacingInfo**, because sequence parameters assume continuous transposition. It's not necessary to use **ACLineSegmentPhase** instances for transposed 3-phase line sections, assuming all phase wire types are identical within a section.

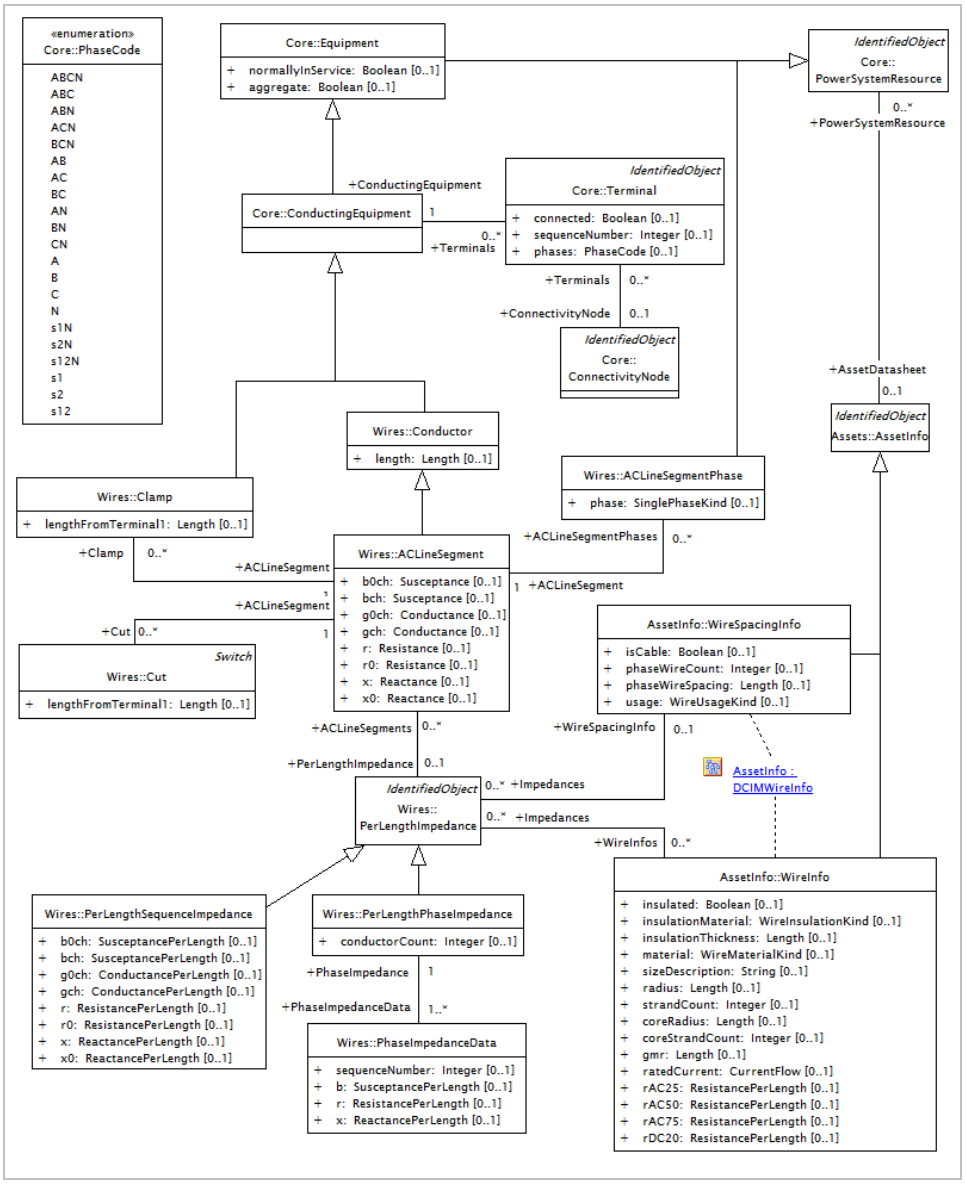


Figure 5-2
Line Connectivity Model Using Only the Wires Package

- The **ACLineSegment** attributes **r, r0, x, x0, bch, bch0, gch,** and **bch0** may be used for a balanced model. This is the option most compatible with CPSM. See 0 for usage with 1-phase and 2-phase line segments. **Conductor.length** is optional, and not used in the impedance calculation.

- Provide an association to **PerLengthPhaseImpedance**, which references pre-calculated NxN symmetric impedance and admittance matrices per unit length. The **conductorCount** (N) has to be at least equal to the number of phases, but it could be higher if the neutral (or other grounded conductor) is retained in the matrix. **PhaseImpedanceData** implements Z and Y matrix elements, stored in column order. The attributes **r, x**, and **sequenceNumber** are all required, while **b** is optional. Only the lower triangular elements are stored, so that a 3x3 matrix would have 6 elements (i.e., instances of **PhaseImpedanceData**). The matrix rows and columns have to be in phase order. A referencing **ACLineSegment** will assign row 1 to the first phase present from the ordered list (**A, B, C, S1, S2, N**), row 2 to the next phase present, and so on. **Conductor.length** is required for the **ACLineSegment**. See also 0 below.

- Provide an association to **PerLengthSequenceImpedance**. This class implements a library of "line codes" that have sequence impedances and line charging per unit length. **Conductor.length** is required. See 0 for usage of sequence parameters with 1-phase and 2-phase line segments.

To specify impedances using the **AssetInfo** package, the **Conductor.length** attribute is required, because the instance electrical parameters are defined as (per-unit length parameters) * (**Conductor.length**). See also 0 below.

A **WireSpacingInfo** should be associated to the **ACLineSegment** using either of two methods (note: a profile may select one method over the other).

- Use the **PowerSystemResource.AssetDatasheet** attribute of **ACLineSegment** to reference **WireSpacingInfo**. This method has also been used to associate **WireInfo** for the current rating.
- Instantiate the **Asset** class, in which **Asset.AssetInfo** references the **WireSpacingInfo**, and multiple **Asset.PowerSystemResources** attributes reference each **ACLineSegment** using that **WireSpacingInfo**.

For impedance calculations, it's also necessary to associate **WireInfos** to each referencing **ACLineSegment** or **ACLineSegmentPhase**. Either the **AssetDatasheet** or **Asset** class pattern accomplishes this. Whenever the **ACLineSegment** has associated **ACLineSegmentPhases**, each **ACLineSegmentPhase** should have its own **WireInfo**. If not:

- At least one **WireInfo** must be associated to the **ACLineSegment**, and it's assumed to apply for all phase wires. It also applies to any neutral wires, unless a second, identifiable **WireInfo** can be associated.
- An optional second **WireInfo** may be associated for the neutral wire, identified as the one with smallest **WireInfo.radius** attribute value. Whenever this assumption does not apply to the line, use by-phase modelling with **ACLineSegmentPhase**.

Figure 5-2 shows associations between **PerLengthImpedance**, **WireSpacingInfo**, and **WireInfo**. However, these are for asset library management and not for impedance calculations. Either the **AssetDatasheet** or **Asset** class pattern is required to use **WireSpacingInfo** and **WireInfo** for impedance calculations.

Figure 5-3 shows the **WireInfo** class hierarchy, used for defining line segment impedances from physical data (sometimes referred to as “line codes” and “cable codes”). **WireInfo** is a base class that should not be instantiated directly. Its attributes describe the physical data for an overhead wire (note that derived **OverheadWireInfo** currently has no attributes of its own), or a cable’s core phase conductor.

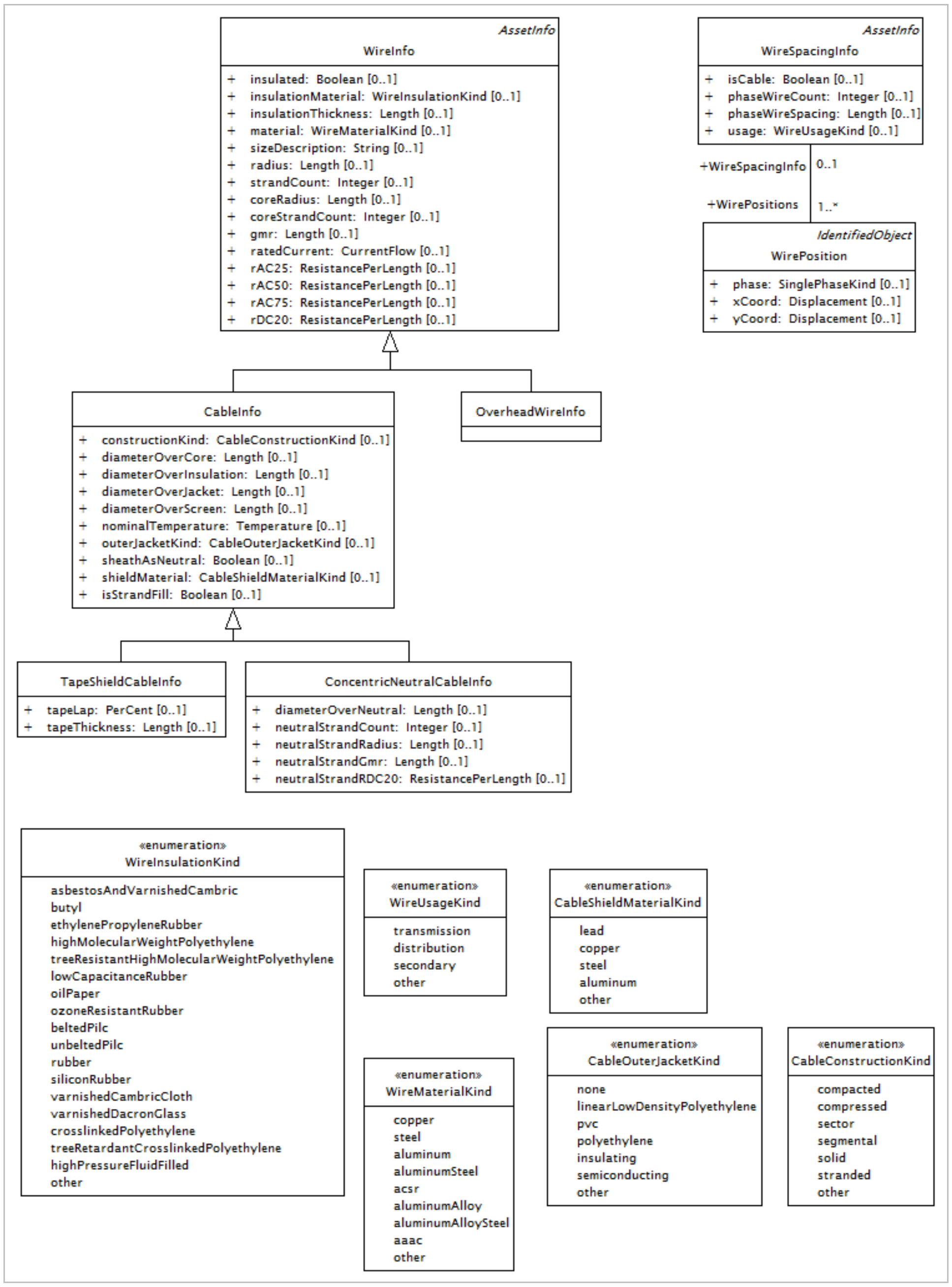


Figure 5-3
Conductor (Line and Cable Data Sheet) Model

WireInfo attributes **material, sizeDescription, strandCount**, and **coreStrandCount** (for ACSR, aluminium conductor steel reinforced wire) help to identify the wire, and are often coded into the instance local name. The minimum required electrical attributes are **gmr, radius**, and **rAC50**. A complete wire table would usually have resistances defined at other temperatures and frequencies, such as **rAC25, rAC75**, and **rDC20**. For ACSR wires, the **coreRadius** is optional for frequency-dependent calculation of the **gmr**. The **coreRadius** is zero for non-ACSR wires. The **ratedCurrent** is the ampacity at 50 °C. The **ratedCurrent** is necessary to interpret power flow output, but not for the power flow solution itself. The **insulated, insulationMaterial,** and **insulationThickness** attributes apply mainly to the derived cable classes, but they also support triplex secondary lines and overhead spacer cable.The **CableInfo** attributes describe additional properties of layers over the conducting core. The **diameterOverCore** includes both conductor and semi-conducting screen; it should be the insulation's inside diameter. The **diameterOverInsulation** is the insulating layer's outside diameter, not including any outside screens or semi-conducting layers. The **diameterOverScreen** includes the insulation plus screen or semi-conducting layer; it should be the shield's or sheath's inside diameter. The **diameterOverJacket** is the cable's outside diameter; it should be the largest diameter value specified except for concentric neutrals.

The **TapeShieldCableInfo** attributes **tapeLap** and **tapeThickness** describe a thin tape covering the cable, usually made of copper, in overlapping turns. The **ConcentricNeutralCableInfo** has attributes for the outer neutral conductors, which typically consist of several solid copper wires. The number of wires is **neutralStrandCount**, the radius is **neutralStrandRadius**, the geometric mean radius is **neutralStrandGmr**, and the resistance per wire is **neutralStrandRDC20**. These wires are so small that AC resistance is typically not used or specified. The **diameterOverNeutral** is the diameter over the concentric neutral strands, which may be the same as **diameterOverJacket**.

Figure 5-3 currently supports the two types of single-conductor cable most commonly found on distribution systems. Many other cable types found in transmission or industrial systems are not supported; these include 3-conductor, pipe-type, submarine, and others.

WireSpacingInfo identifies the line geometry data. Any conductors numbered above the **ConductorInfo.phaseCount** are assumed to be continuously grounded; the application may eliminate these conductors from the impedance and admittance matrices through Kron reduction. The phaseWireCount and **phaseWireSpacing** attributes refer to sub-conductor bundling, which is not common for distribution lines, but may appear on high-voltage transmission lines in the model.

WirePosition defines the horizontal (**xCoord**) and vertical (**yCoord**) coordinates of each wire on the pole or tower cross section, or in the cable trench/duct, and **phase** identifies which phase wire is mounted there. All three attributes are required. The overhead wire height above ground is **yCoord**, including any sag effects. For underground cables, **yCoord** is the average burial depth, entered as a negative number. The wire horizontal position, **xCoord**, is measured from an arbitrary but consistent reference line. Common choices for the horizontal reference are the pole centerline, and the left-most wire position.

Any wires at a **WirePosition** with **phase=N** are assumed to be continuously grounded. The application may eliminate these conductors from the impedance and admittance matrices through Kron reduction.

Many times, the **WirePosition.phase** attribute value could change depending on the field installation. For example, a 1-phase line can be used on phase **A, B,** or **C**. This requires three **WireSpacingInfo** instances, the only difference being three different values for one of the associated **WirePosition.phase** values, namely **A, B**, or **C**. The **WirePostion.phase** value for a neutral wire would not change; it should always be **N**. In summary, a 1-phase line type requires 3 **WireSpacingInfo** instances to cover all phasing possibilities. A 2-phase line type requires 6 instances, and a 3-phase line type also requires 6 instances.

Using Sequence Impedances (Balanced Case)

The positive and zero sequence impedances may be transferred through the **r, x, r0**, and **x0** attributes of **PerLengthSequenceImpedance** associated with an **ACLineSegment** instance. The **bch, b0ch, gch**, and **g0ch** attributes are not important for overhead distribution lines. For three phases, this describes a balanced three-phase, or perfectly transposed, line. The attributes in **PerLengthSequenceImpedance** are expressed in units per length, so it is necessary to multiply their values with the **length** attribute of **Conductor**.

NOTE This is equivalent to the attributes of **Wires::ACLineSegment**, which are pre-calculated for the whole length of the segment and thus must be defined on each instance of the segment. In contrast, **PerLengthSequenceImpedance** is referenceable, and as such can be used (through association) by several segment instances, thus decreasing the amount of data transferred in data exchanges.

If the **ACLineSegment** has only one or two phases, a balanced model can still be transferred through the **r, x, r0**, and **x0** attributes. This represents an impedance matrix with equal complex diagonal elements, Z_s, and equal complex off-diagonal elements, Z_m. For a single-phase line, the attributes to transfer are:

$$Z_1 = Z_0 = Z_s \qquad \textit{Equation 5-1}$$

For a two-phase or three-phase line, the attributes to transfer are:

$$Z_1 = Z_s - Z_m \qquad \textit{Equation 5-2}$$

$$Z_0 = Z_s + (n - 1)\, Z_m \qquad \textit{Equation 5-3}$$

where *n* is the number of phases. Upon receipt of **r, x, r0,** and **x0**, the balanced two-phase or three-phase impedance matrix is constructed from:

$Z_s = (Z_0 + (n - 1) Z_1) / n$ *Equation 5-4*

$Z_m = (Z_0 - Z_1) / n$ *Equation 5-5*

The referencing **ACLineSegment** should have associated **ACLineSegmentPhases**, assigned a appropriate **phase** values to show the phases actually present, such as **A, B, C, S1,** or **S2**. The neutral, **N**, should not appear because any neutral conductor must have been incorporated into the earth return when sequence impedances are used.

For underground distribution cables, the sequence impedances are also appropriate, including the **bch** and **b0ch** attributes.

Using Phase Impedances (Unbalanced Case)

Calculated matrix parameters may be transferred by referencing a **PerLengthPhaseImpedance** instance, in lieu of the physical model described in 0. A matrix model is useful when:

- the target application has no means of calculating parameters from the physical data.
- the underlying physical data is not readily available.
- it is necessary to match the unbalanced line parameters as closely as possible.

For a two-phase line, with the neutral already reduced, the Z and Y matrices will be 2x2, but due to symmetry, there will only be 3 unique matrix elements. That leads to three associated instances of **PhaseImpedanceData** with column-wise storage:

- **sequenceNumber** = 1 for row 1, column 1 of the matrix;
- **sequenceNumber** = 2 for row 2, column 1 of the matrix;
- **sequenceNumber** = 3 for row 2, column 2 of the matrix.

This instance of **PerLengthPhaseImpedance** could be referenced from a **ACLineSegment** having phases **AB, AC**, or **BC**. Row 1 always corresponds to the first phase present and row 2 always corresponds to the second phase present, following the order **A, B, C, S1, S2**, and **N**. If the phase wires are installed in different positions, such that **C** (for example) should be in row 1, a new instance of **PerLengthPhaseImpedance** is required.

Using Physical Parameters

For overhead lines, a physical model can be transferred through reference to a **WireSpacingInfo** instance, which is associated with further classes shown in Figure 5-3. This will support calculation of an unbalanced phase impedance matrix through the use of Carson's equations, or an equivalent method of handling the earth return. For example, suppose there are three phase wires, plus a different size neutral wire, on a pole with horizontal crossarm. This requires one **WireSpacingInfo** instance, four **WirePosition** instances that describe the four conductor positions, and two **WireInfo** instances describing the phase and neutral wire types. The **length** attribute of **Conductor** must be used, and many **ACLineSegments** will typically refer to the same **WireSpacingInfo** instance.

The **WirePosition** instances have a **phase** attribute that maps to **ACLineSegmentPhase.phase**. The resistance attribute of **WireInfo** should be supplied for fundamental power frequency, and at the wire's desired operating temperature for calculations.

A single-phase, concentric neutral cable requires one instance of **ConcentricNeutralCableInfo**, one **WireSpacingInfo** with attribute **isCable=true**, and one **WirePosition** instance to specify the burial depth. A three-phase tape shielded cable, with bare neutral conductor, would require one instance of **TapeShieldCableInfo**, one **WireSpacingInfo** with attribute **isCable=true**, and four **WirePositions** for the three phases and neutral. There would also be one **WireInfo** for the neutral.

Distribution Transformers

Electrical Model

Figure 5-4 shows the classes that model power transformer instances. They can use the **Wires** package exclusively to define impedance parameters, or make use of the **AssetInfo** classes detailed in Figure 5-5, to define a library of transformer types.

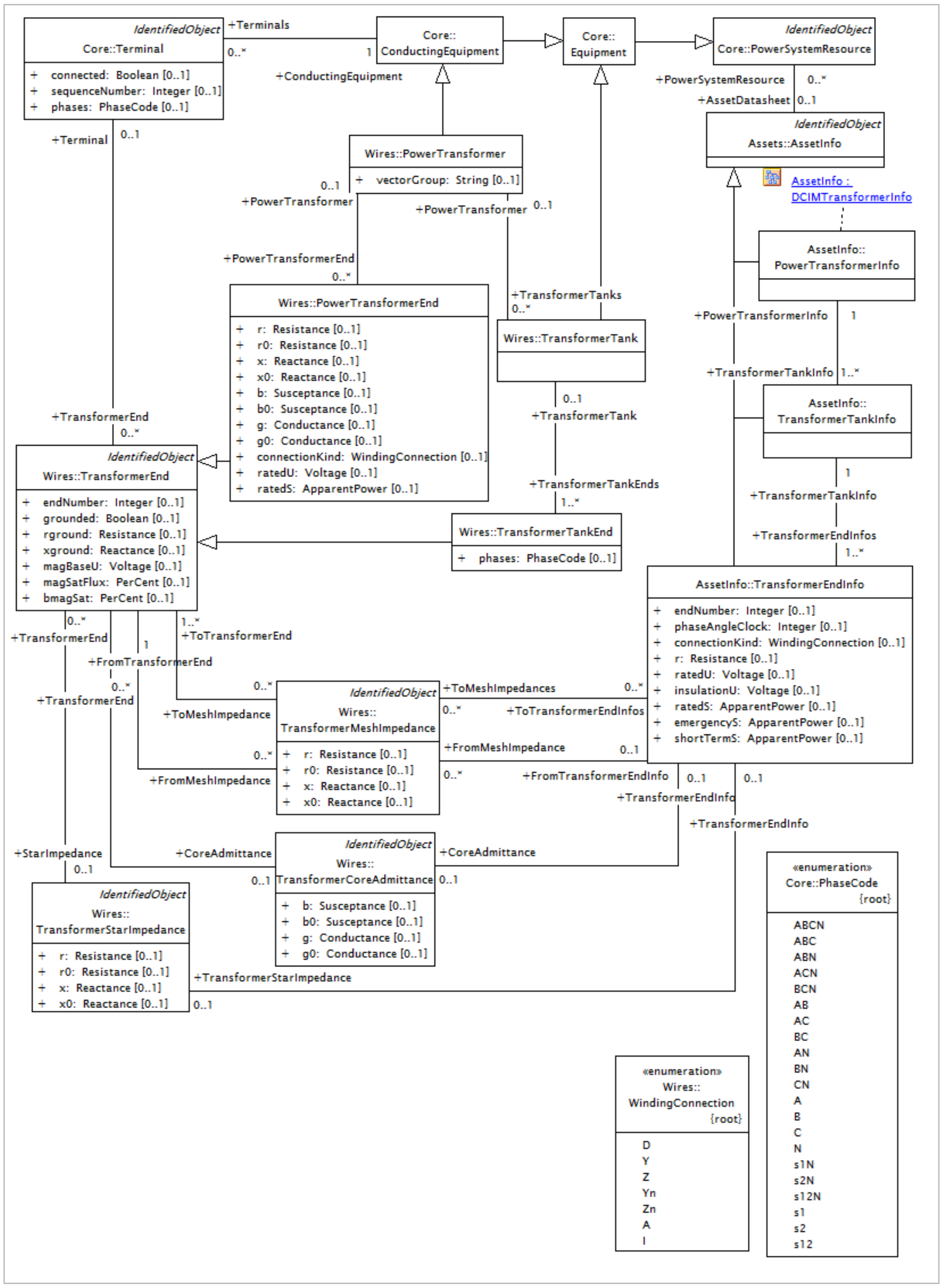


Figure 5-4
Transformer Connectivity Model

PowerTransformer is the top-most instance for a power transformer, whether composed of a three-phase tank, or possibly different single-phase tanks. It descends from **ConductingEquipment**, and therefore, has associated **Terminals** with **phases** attribute values at each winding connection point. In the CIM, a transformer winding is referred to as an "end".

When composed of different tanks, a **PowerTransformer** has often been called a "transformer bank", and the CIM supports modelling with or without tanks. In distribution systems, independent phase voltage regulators and open wye/open delta transformers provide two examples that require tank-level modelling. At the transmission level, EHV transformer banks may also contain single-phase transformers, which need not be identical, especially when a spare is in service. The **vectorGroup** attribute for protective relaying is derived from the internal winding connections and phase angles; it uses IEC 60076-1 nomenclature to describe any number of windings that may be included in the bank.

When used, the **TransformerTank** must be associated with a **PowerTransformer**. It inherits from **Equipment**, not **ConductingEquipment**. The tank may have associated **TransformerTankInfo** from the **AssetInfo** package, for asset datasheet modelling, described in more detail later with Figure 5-5. Because transformer testing is done on tanks, the datasheets are fundamentally associated with tanks. When not using tanks in the model, the **PowerTransformer** can still have an association to **PowerTransformerInfo**, for asset datasheet modelling. In both cases, the data actually resides in **TransformerEndInfo** instances, associated to a **TransformerTankInfo**.

There are two methods of referencing transformer asset datasheets, and a profile may prefer one over the other:

- Use the **PowerSystemResource.AssetDatasheet** attribute of either **PowerTransfomer** or **TransformerTank**, to reference either **PowerTransformerInfo** or **TransformerTankInfo**.
- Instantiate the **Asset** class, in which **Asset.AssetInfo** references either the **PowerTransformerInfo** or **TransfomerTankInfo**. Multiple **Asset.PowerSystemResources** attributes reference each **PowerTransformer** or **TransformerTank** using that datasheet.

TransformerEnd was called TransformerWinding in some earlier versions of CIM. It is not a **ConductingEquipment**, but it does have one associated **Terminal** with phasing information.The **magBaseU, magSatFlux**, and **bmagSat** attributes represent core saturation, typically modelled at no more than one of the ends. The other instance attributes define the grounding options:

- solidly grounded: **grounded** = true, **rground** = 0, **xground** = 0;
- impedance grounded: **grounded** = true, **rground** $\geq$ 0, **xground** $\geq$ 0;
- ungrounded: **grounded** = false.

TransformerEnd should not be instantiated directly; one of its two descendants should be instantiated:

- **PowerTransformerEnd**, if not using tank-level modelling. Specify the **ratedU** and **ratedS** attribute values for winding rating data, and **connectionKind** for the wye, delta, or other type of connection.
- **TransformerTankEnd**, if using tank-level modelling. The winding connection and rating values come from a **TransformerTankEndInfo**

instance, which means that the **AssetInfo** package is required for tank-level modelling.

With **PowerTransformer** and **PowerTransformerEnd**, there are three ways of specifying impedance parameters, using only the **Wires** package. They are all supported in current versions of IEC 61970-301.

- Use the **r, x, r0, x0, b, b0, g**, and **g0** attributes of **PowerTransformerEnd** to specify pi impedance parameters. This is the option most compatible with earlier versions of IEC 61970-301, and there are some important conventions described in that standard.
- Use one associated **TransformerStarImpedance** for each **PowerTransformerEnd**, comprising a star equivalent (also sometimes called tee or wye equivalent). This can be mathematically exact for up to three ends (windings), however, negative attribute values may occur in the case of three ends. Optionally, use one **TransformerCoreAdmittance** associated to one of the **PowerTransformerEnds**, representing the exciting current and core losses. This can be the lowest-voltage winding (i.e. closest to the core), or the winding that was actually subjected to a no-load test, if known. The reference voltage for all attribute values, which have units of Ohms or Siemens, must be **ratedU** for the end to which the impedance or admittance is connected.
- Use a **TransformerMeshImpedance** associated to each combination of **PowerTransformerEnd** pairs. There must be (#Ends-1) * #Ends div 2 of these; for example, one **TransformerMeshImpedance** between two ends, three of them between three ends, six of them between four ends, etc. The advantages of a mesh model are; mathematically exact for more than three ends, no negative attribute values, and more direct correspondence with transformer short-circuit test data. The reference voltage must be **ratedU** of the **FromTransformerEnd**; note the other end nearly always has a different rated voltage. Optionally, use one **TransformerCoreAdmittance** as described for the star equivalent.

Data Sheet Model

Figure 5-5 shows the classes that allow for exchange of transformer datasheet models, sometimes referred to as "transformer codes" in applications. The main class is **TransformerEndInfo**, which contains rating and connection data for the corresponding transformer winding. It has an **endNumber** attribute, used to map the data to **TransformerEnd.endNumber**. The associated **Terminal.sequenceNumber** could also use the same **TransformerEnd.endNumber**, but this is not required. By convention, the **endNumber** usually starts at 1 for the highest voltage winding, and all other windings are numbered in order of decreasing voltage rating. Some transformers have more than one winding with the same **ratedU**, split-secondary transformers providing just one example, and they must have different **endNumbers**.

The **endNumber** attribute allows the datasheet library to be used and updated with just one association to **PowerTransformerInfo** or **TransformerTankInfo**.

The **TransformerTankInfo** and **PowerTransformerInfo** provide two navigation paths to **TransformerEndInfo**. With **PowerTransformerInfo**, it may be necessary to create an artificial **TransformerTankInfo** for model transfer, because datasheets are at the tank level. A **PowerTransformerInfo** instance actually refers to a collection of one or more tanks.

There are two ways of specifying impedance parameters in a datasheet model:

- Associate **TransformerEndInfo** to the **TransformerMeshImpedance**, **TransformerStarImpedance**, and **TransformerCoreAdmittance** classes from the Wires package. These associations were shown in Figure 5-4.
- Use **TransformerTest** and its three descendant classes, shown in Figure 5-5. Each application is responsible for converting the test data to mesh equivalent, star equivalent, or some other electrical impedance model.

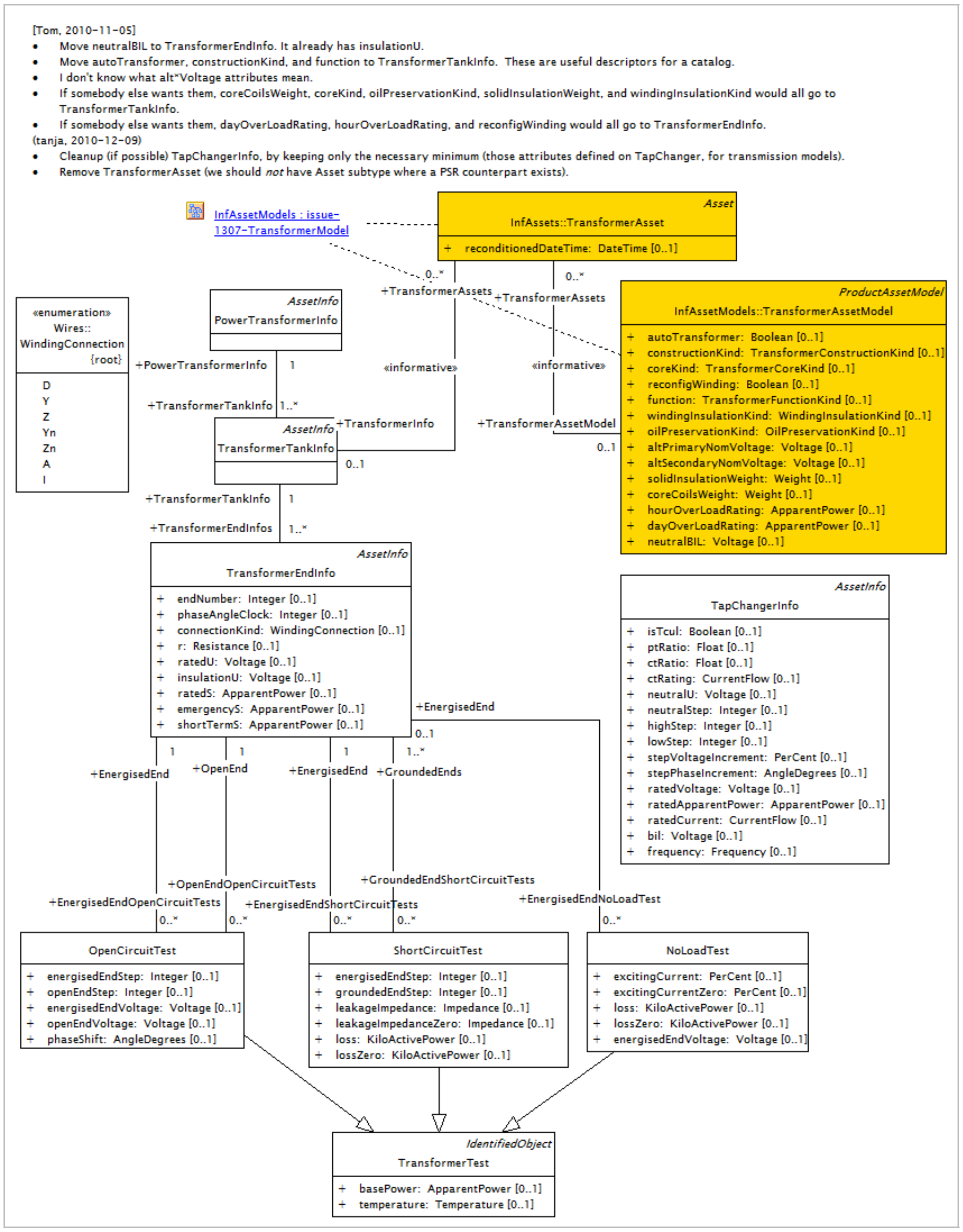


Figure 5-5
Transformer Data Sheet Model

WindingConnection enumeration includes the standard **D, Y, Z, Yn**, and **Zn** nomenclature to describe delta, wye, zig-zag, and neutral connections in three-phase transformer vector groups. **A** is used for a common autotransformer winding, and **I** for a single-phase transformer winding.

TransformerTankInfo is referenced by the **TransformerTank** and **PowerTransformerInfo**. It serves mainly to organize the data into a library with one entry point. **TransformerTankInfo** collects associations to **TrnasformerEndInfo**, which generalizes to any number of windings.

TransformerEndInfo includes all of the winding rating information, including **ratedU**, **ratedS**, **connectionKind**, **phaseAngleClock**, **shortTermS**, **emergencyS**, and **insulationU**. **r** is the winding's DC resistance. **endNumber** is the winding's order in the **vectorGroup**; the high-voltage winding is always 1. The **endNumber** usually increases as the **ratedU** decreases, but some transformers have two windings with the same **ratedU**, in which case the **endNumber** distinguishes between them.

TransformerTest is the parent class for all of the transformer test classes, with **basePower** and **temperature** attributes applicable to all tests. **basePower** is essential for converting datasheet values to impedance or admittance at the correct reference voltage.

NoLoadTest's attributes **excitingCurrent**, **excitingCurrentZero**, **loss**, and **lossZero** are all measured on the **EnergisedEnd** winding. It provides the basic data for a core admittance branch. Positive and zero sequence test data can be reported in the same instance of **NoLoadTest**. This test is generally done with rated voltage applied to the energised winding, but different values can be specified in **energisedEndVoltage**, and several tests may be provided to define core saturation parameters.

ShortCircuitTest is done by circulating rated current through the **EnergisedEnd** winding, with one or more **GroundedEnds** windings short circuited. It provides the basic data for the mesh or star equivalent circuit. If tests were done at different tap settings, the tap values are specified in **energisedEndStep** and **groundedEndStep**. Positive and zero sequence test data can be reported in the same instance of **ShortCircuitTest**. The AC resistances derived from **loss** and **lossZero** are likely to differ from the winding DC resistances, **TransformerEndInfo.r**, which were obtained from a separate test.

OpenCircuitTest's attributes **openEndVoltage** and **phaseShift** are measured on a single open winding, when the **EnergisedEnd** has **energisedEndVoltage** applied to it. Tap settings for both windings are specified in **energisedEndStep** and **openEndStep**. The tests are done to verify the transformer turns ratio, winding connections, and winding polarity. They are usually nt required to determine electrical impedance parameters.

Tap Changer Model

Figure 5-6 shows the classes used to model an unbalanced voltage regulator. A **RatioTapChanger** is associated to a **TransformerEnd**. Each regulator uses autonomous local control, so that **RegulationSchedule** and **TapSchedule** (present in IEC 61970-301) are not used. Phase angle regulators and variation curves are also not generally used on distribution systems.

A three-phase line voltage regulator usually has three independent regulators to help correct voltage unbalance. The regulators are usually connected in wye. The model starts with a **PowerTransformer** containing three **TransformerTanks**, and a total of six **TransformerTankEnds**. There will be three instances of

RatioTapChanger, each associated to a different **TransformerTank**. The **RegulatingControl.monitoredPhase** attribute should be include among the associated **Terminal.phases**. The tap positions, and sometimes the other attributes, will not be the same in each phase of the regulator. An open-delta regulator is also fairly common; this consists of two single-phase regulators connected line-to-line in a bank, with partial capability to correct voltage unbalance.

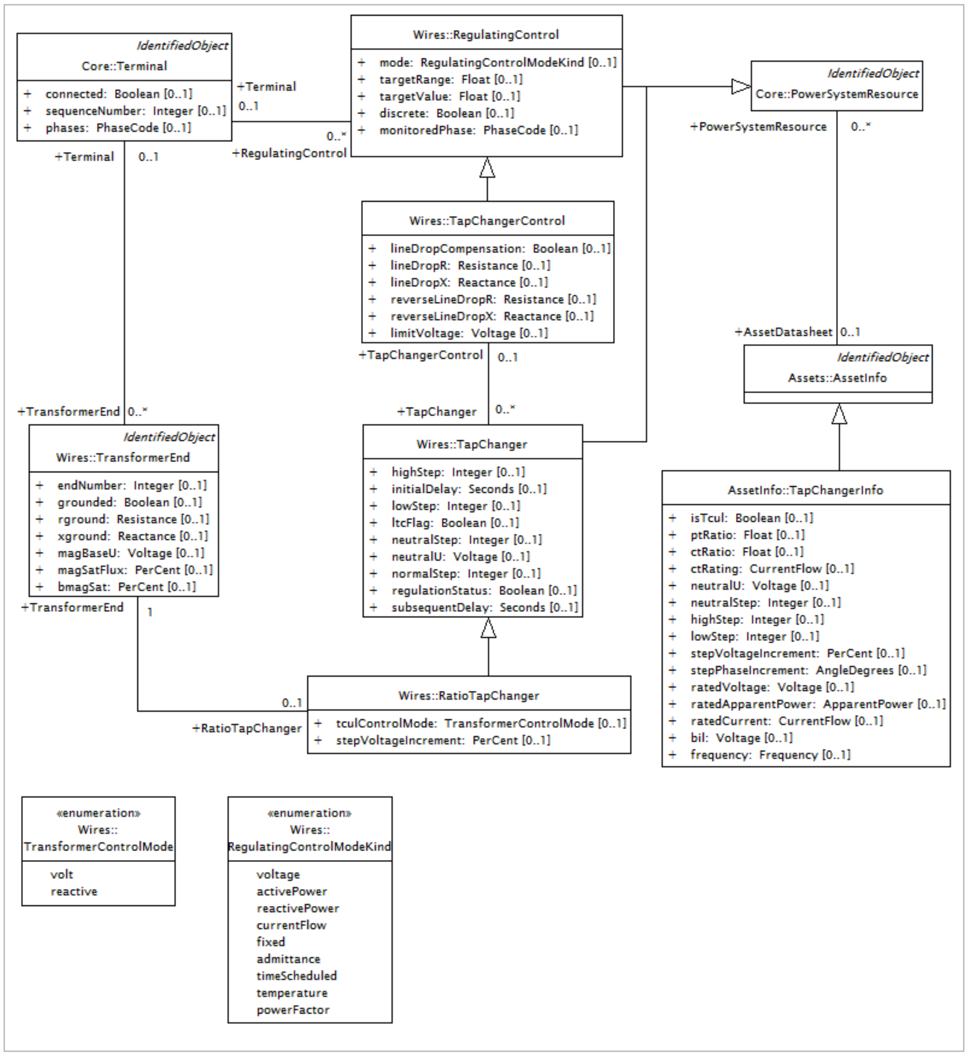


Figure 5-6
Tap Changer Model

A three-phase substation voltage regulator usually changes all three taps together, with no ability to correct voltage unbalance. In this case, tank-level modelling is not required and the model might consist of one **PowerTransformer** with two **PowerTransformerEnds**. There would be just one **RatioTapChanger** associated to one **PowerTransformerEnd**. The **RegulatingControl.monitoredPhase** attribute can be **A**, **B**, or **C** if the potential transformer is connected line-to-ground. It can also be **AB**, **AC**, or **BC** for line-to-line potential transformers. Typically, only one potential transformer controls this type of regulator.

Example: Open Wye / Open Delta Transformer

The transformer in Figure 5-7 shows an open wye/open delta bank, which is used to supply inexpensive three-phase service to smaller customers.

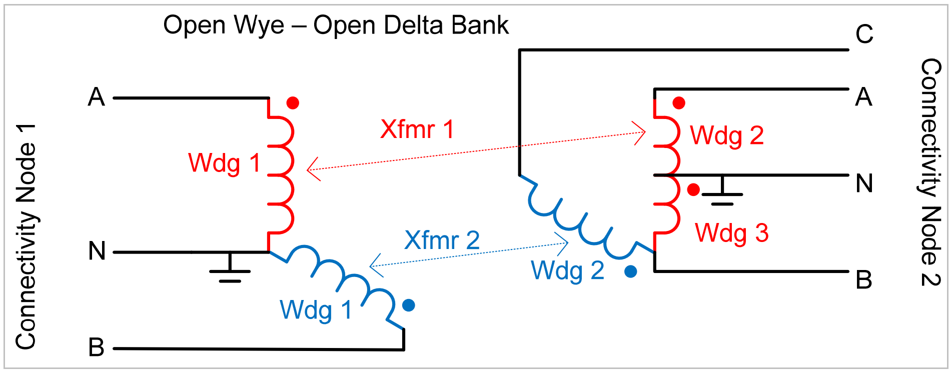


Figure 5-7
Open Wye/Delta Transformer Example

Table 5-1 shows some of the important attribute values for this example. It requires tank-level modelling.

Table 5-1
Open Wye/Open Delta Transformer Tank Connections

Transformer Tank	Transformer TankEnd	ratedU	ratedS	connection Type	phaseAngle Clock	TransformerTankEnd. phases
Xfmr 1	Wdg 1	7200	100e3	I	0	AN
	Wdg 2	120	50e3	I	0	AN
	Wdg 3	120	50e3	I	6	BN
Xfmr 2	Wdg 1	7200	50e3	I	0	BN
	Wdg 2	240	50e3	I	0	BC

A phase angle clock value of "6" indicates that Wdg 3 is actually from N to B, rather than B to N. Through the **Terminals**, **ConnectivityNode** 1 will have phases **ABN** present from this transformer bank. Other connected equipment, such as a line segment, could add phase **C.** **ConnectivityNode** 2 will have phases **ABCN** present from this transformer bank. The "lighting leg" (Xfmr 1) usually has a different rating than the "power leg" (Xfmr 2). This means that phase and rating assignments to the bank might be ambiguous, and thus need to be specified on **TransformerTankEnds**.

Autotransformers for CIM

An autotransformer is made by connecting two transformer windings in series, so there is a metallic connection between the two voltage levels. The main advantage is a cost savings, because the MVA rating is higher than for the same two windings connected as a conventional transformer. Autotransformers are also more efficient and have less voltage drop. The main disadvantage is probably higher short-circuit current in fully developed systems, because autotransformers have lower impedance. Two common applications are:

1. Transformations between two extra-high voltage (EHV) levels in a substation, where the cost savings are important for turns ratios up to approximately 2:1.
2. Line voltage regulators on distribution feeders, where the regulating (buck/boost) winding is connected in auto.

Figure 5-8 shows a two-winding transformer to the left, with a short-circuit test connection on the L winding terminal. All of the current is transformed magnetically, and the turns ratio is n1:n2. To the right, Figure 1 shows the same two windings connected as an autotransformer. The **red** current flows directly to the short-circuit test connection on the L winding terminal. In addition, the magnetically transformed **blue** current also contributes to the **short-circuit** current, which is higher than in the two-winding case. The autotransformer turns ratio is (n1+n2):n2. The H winding is sometimes called the series (S) winding in an autotransformer, and the L winding is sometimes called the common (C) winding in an autotransformer.

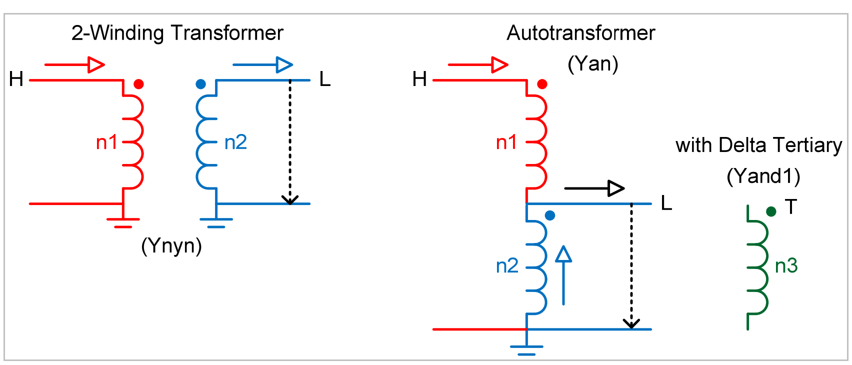


Figure 5-8
Two-Winding Transformer Connected as an Autotransformer

The IEC vector groups define grounding and phase shift characteristics of each winding, which are important for protective relaying, paralleling, and other applications. Figure 1 shows these in parentheses. Ynyn denotes a transformer with two windings, both wye grounded. (An external neutral impedance may still be added, but is not listed in the vector group). The autotransformer could also be denoted Ynyn, because it has the same neutral connection and phase shift characteristics. However, some transformer vendors use "A" or "a" to denote an autotransformer winding. Figure 1 shows the Yan vector group for an autotransformer. Even though the H terminal has a conducting path to neutral through the L winding, the H winding itself is not connected to the neutral. Per IEC 60076-1, the highest voltage winding is capitalized in the vector group, while all other windings are lower-case. In practice, this means the "a" for an autotransformer would always be lower case.

Many autotransformers have a delta tertiary, shown to the right of Figure 1. The vector group would be Yand1, where the 1 refers to a 30-degree lag (1 o'clock) with respect to the H winding. The Z_{HT} mesh impedance value is affected by the auto connection, but not the Z_{LT} mesh impedance value. That adjustment is not presented here, but may be found in several technical references.

The conversions from two-winding data (left side of Figure 5-8) to autotransformer turns ratio (N), volt-ampere rating (S), and mesh impedance (Z) on the right side of Figure 5-8 are:

$$N = 1 + n_1/n_2 \qquad \textit{Equation 5-6}$$

$$S_{auto} = S_{2\text{-}wdg}\,[N/(N-1)] \qquad \textit{Equation 5-7}$$

$$Z_{auto} = Z_{2\text{-}wdg}\,[(N-1)/N]^2 \qquad \textit{Equation 5-8}$$

In per-unit, the autotransformer impedance is:1

$$Z_{pu\text{-}auto} = Z_{pu\text{-}2\text{-}wdg} / N \qquad \textit{Equation 5-9}$$

For example, suppose the 2-winding transformer is 115/115 kV, rated 100 MVA, with 10% impedance on 100 MVA. The turns ratio is 1:1. Viewed from either winding, the short-circuit impedance is 13.225 ohms. Connected as an autotransformer, 230/115 kV, the turns ratio is 2:1 (N=2) and the rating is 200 MVA. Viewed from the 115-kV terminal, the short-circuit impedance is 3.30625 ohms, which is 5% on the new rating of 200 MVA. The iron core and copper winding costs are half of what they would be for a conventional 2-winding transformer of the same rating. The total cost is somewhat more than 50%, because the L winding leads must carry more current, and the H winding must be insulated for a higher voltage. The test sheet for this autotransformer would show a voltage ratio of 230 / 115 kV, a rating of 200 MVA, and a short-circuit impedance of 5%.

It is common to model an autotransformer as a conventional two-winding transformer, using data from the test sheet, ignoring the fact that the windings are actually connected in series. However, there are times when the difference is important, such as more accurate core modeling, or more accurate modeling of the impedance vs. tap characteristic. It is possible to derive the physical autotransformer model in Figure 1, if the series and common windings have been identified.

In the CIM, an autotransformer should be modeled with converted two-winding data for impedances, admittances, and ratings, as typically found on autotransformer test reports. Each physical winding will have a corresponding **PowerTransformerEnd** or **TransformerTankEnd** in the CIM. It is optional to specify the autotransformer connection with an attribute value **A** for **connectionKind** on the common end, and with "**an**" appearing as part of **PowerTransformer.vectorGroup** for that end. The series end must then have attribute value **Y** for **connectionKind**. The series and common **endNumber**s should be 1 and 2, respectively. The receiving application may then derive the physical autotransformer model if needed. To ignore autotransformer connections in the model, specify **Yn** for **connectionKind** on both series and common ends; there is no restriction on the **endNumber**s. The **connectionKind** attribute appears on **TransformerEndInfo** if using tank-level modeling and on **PowerTransformerEnd** if not using tank-level modeling. Note that **A**, **Y**, **D**, and **Z** are always capitalized in **connectionKind**, but whenever the **endNumber** is greater than 1, they should be lower-cased in the **vectorGroup**.

Export Control Restrictions

The Electric Power Research Institute Inc., (EPRI, www.epri.com) conducts research and development relating to the generation, delivery and use of electricity for the benefit of the public. An independent, nonprofit organization, EPRI brings together its scientists and engineers as well as experts from academia and industry to help address challenges in electricity, including reliability, efficiency, health, safety and the environment. EPRI also provides technology, policy and economic analyses to drive long-range research and development planning, and supports research in emerging technologies. EPRI's members represent more than 90 percent of the electricity generated and delivered in the United States, and international participation extends to 40 countries. EPRI's principal offices and laboratories are located in Palo Alto, Calif.; Charlotte, N.C.; Knoxville, Tenn.; and Lenox, Mass.

Together...Shaping the Future of Electricity

Program:

IntelliGrid

1023218

Electric Power Research Institute
3420 Hillview Avenue, Palo Alto, California 94304-1338 • PO Box 10412, Palo Alto, California 94303-0813 USA
800.313.3774 • 650.855.2121 • askepri@epri.com • www.epri.com