

## OpenDSS Tech Note: XfmrCode and LineSpacing

New options for specifying transformer and line parameters have been recently added to the OpenDSS. These may be useful when converting data from other programs that use transformer library data, or that use libraries of “line spacings”.

A transformer “bank” has also been added, as a property on existing transformer objects. At present, this is used only for Common Information Model (CIM) export, but the concept may be extended as needed for MultiSpeak interfaces, and for output reports. Switches, capacitors, and even loads are often grouped into “banks” for identification.

### XfmrCode and Bank

The new XfmrCode object has every property of the Transformer object, except for **bank**, **bus**, **buses**, and **xfmrcode**. You can specify transformer parameters on the XfmrCode object, and then re-use them through the new **xfmrcode** property on the Transformer. (It was previously possible to mimic a transformer library using the **like** property, but this had the disadvantage of creating “phantom buses” for the library transformer objects.)

The following example from the IEEE 13-bus test case illustrates both **xfmrcode** and **bank**. At present, the only effect of **bank** is that, when executing the “export cdpwm” command, the three regulator legs will be grouped into one bank as required by the CIM.

```
New XfmrCode.RegLeg phases=1 xhl=0.01 kvas=[1666 1666] kvs=[2.4 2.4] %LoadLoss=0.01

New Transformer.Reg1 XfmrCode=RegLeg Bank=Reg Buses=[ 650.1 RG60.1]
New Transformer.Reg2 XfmrCode=RegLeg Bank=Reg Buses=[ 650.2 RG60.2]
New Transformer.Reg3 XfmrCode=RegLeg Bank=Reg Buses=[ 650.3 RG60.3]
```

### LineSpacing

The new LineSpacing object expands the options for defining a library of overhead line data. This object comprises two arrays for X and H (or Y) coordinates, along with a unit specification. This is a subset of the exiting LineGeometry property set. In addition, a string array of **wires** has been added to the property set for both LineGeometry and Line itself.

LineSpacing Properties (in addition to **like**):

Property	Description
Nconds	Number of wires in this geometry. Default is 3. Triggers memory allocations. Define first!
Nphases	Number of retained phase conductors. If less than the number of wires, list the retained phase coordinates first.
Units	Units for x and h: {mi kft km m Ft in cm } Initial default is "ft", but defaults to last unit defined
X	Array of wire X coordinates.
H	Array of wire Heights.

Any time a LineSpacing is employed, the program constructs a temporary LineGeometry object to calculate the line parameters. The results would be equivalent; the only difference is in how the inputs are specified.

The current options for line parameters are:

1. Define impedances directly on the Line object (no changes).
2. Use the **linecode** and **length** properties on the Line object (no changes).
3. Use the **geometry** and **length** properties on the Line object.
  - a. Previously define a LineGeometry, with wire, x, and y specification by conductor number. All WireData objects must be previously defined (no changes).
  - b. **NEW** – Previously define a LineGeometry, with **spacing** and **wires** properties. You still specify **nconds** first, and may also specify **nphases** and **reduce** independently. This closely matches the IEEE test feeder definitions, and the data structures of FeederAll, CIMDIST, and some other programs.
4. **NEW** – use the **spacing**, **wires**, and **length** properties on the Line object. This closely matches the “by-phase” wire modeling in CIMDIST and some other programs. The LineSpacing object must be previously defined, and it determines the number of phases, and whether or not neutral reduction will be used. The number of wire names should match the LineSpacing’s number of conductors (i.e., include the neutral), and the matching WireData objects must have been previously defined.

Here is an example, from the IEEE 13-Bus test case, using **spacing** on the Line object. Note that on the two-phase line from bus 632 to bus 645, phase C is in the left-most position (x=-4), while phase B is in the right-most position (x=3).

```
new WireData.ACSR_556_5  DIAM=0.927  GMRac=0.37320  Rdc=0.035227273
~ Runits=kft Radunits=in gmrunits=in
new WireData.ACSR_4/0    DIAM=0.563  GMRac=0.09768  Rdc=0.112121212
~ Runits=kft Radunits=in gmrunits=in
new WireData.ACSR_1/0    DIAM=0.398  GMRac=0.05352  Rdc=0.212121212
~ Runits=kft Radunits=in gmrunits=in

new LineSpacing.500 nconds=4 nphases=3 units=ft x=[-4 -1 3 0] h=[28 28 28 24]
new LineSpacing.505 nconds=3 nphases=2 units=ft x=[-4 3 0] h=[28 28 24]
new LineSpacing.510 nconds=2 nphases=1 units=ft x=[0.5 0] h=[29 24]

New Line.671680 Bus1=671.1.2.3  Bus2=680.1.2.3
~ Spacing=500 Wires=[ACSR_556_5 ACSR_556_5 ACSR_556_5 ACSR_4/0] Length=1000 units=ft
New Line.632633 Bus1=632.1.2.3  Bus2=633.1.2.3
~ Spacing=500 Wires=[ACSR_4/0 ACSR_4/0 ACSR_4/0 ACSR_4/0] Length=500 units=ft
New Line.632645 Bus1=632.3.2    Bus2=645.3.2
~ Spacing=505 Wires=[ACSR_1/0 ACSR_1/0 ACSR_1/0] Length=500 units=ft
```

Here is the same example, using **spacing** on the LineGeometry object:

```
// include the same WireData and LineSpacing definitions from above
new WireData.ACSR_556_5 DIAM=0.927 GMRac=0.37320 Rdc=0.035227273
~ Runits=kft Radunits=in gmrunits=in
new WireData.ACSR_4/0 DIAM=0.563 GMRac=0.09768 Rdc=0.112121212
~ Runits=kft Radunits=in gmrunits=in
new WireData.ACSR_1/0 DIAM=0.398 GMRac=0.05352 Rdc=0.212121212
~ Runits=kft Radunits=in gmrunits=in

new LineSpacing.500 nconds=4 nphases=3 units=ft x=[-4 -1 3 0] h=[28 28 28 24]
new LineSpacing.505 nconds=3 nphases=2 units=ft x=[-4 3 0] h=[28 28 24]
new LineSpacing.510 nconds=2 nphases=1 units=ft x=[0.5 0] h=[29 24]

new LineGeometry.601 nconds=4 nphases=3 reduce=y spacing=500
~ wires=[ACSR_556_5 ACSR_556_5 ACSR_556_5 ACSR_4/0]
new LineGeometry.602 nconds=4 nphases=3 reduce=y spacing=500
~ wires=[ACSR_4/0 ACSR_4/0 ACSR_4/0 ACSR_4/0]
new LineGeometry.603 nconds=3 nphases=2 reduce=y spacing=505
~ wires=[ACSR_1/0 ACSR_1/0 ACSR_1/0]

New Line.671680 Bus1=671.1.2.3 Bus2=680.1.2.3 Geometry=601 Length=1000 units=ft
New Line.632633 Bus1=632.1.2.3 Bus2=633.1.2.3 Geometry=602 Length=500 units=ft
New Line.632645 Bus1=632.3.2 Bus2=645.3.2 Geometry=603 Length=500 units=ft
```