

## OpenDSS COM Documentation

# Bus Interface

Sept 2013

This interface is used to extract the listed values from a Bus object in the active circuit after the appropriate solution is performed. Most of the properties are readonly as it is unsafe to define these properties separate from the terminal connection definitions for circuit elements.

The interface operates on the *Active Bus*. You can set a bus active using the Buses(i) property or the SetActiveBus or SetActiveBusi functions in the Circuit interface. Also, when you set an active circuit element and terminal, the bus connected to that terminal becomes active.

### ***Properties***

Property	Type	Description
Coorddefined:	WordBool	readonly Returns TRUE if the X-Y coordinates are defined for the active bus. Else returns FALSE.
CplxSeqVoltages:	OleVariant	readonly Returns a variant array of doubles representing the sequence voltages in complex phasor form. Sequence order is 0, 1, 2
Cust_Duration:	Double	readonly Returns customer duration for active bus after reliability calcs
Cust_Interrupts:	Double	readonly Returns customer interruptions for active bus after reliability calcs
Distance:	Double	readonly Returns distance in km that this bus is from the parent EnergyMeter (typically at head of the feeder.)
Int_Duration:	Double	readonly Returns average interruption durations for active bus after reliability calcs
Isc:	OleVariant	readonly
Lambda:	Double	readonly Returns total annual failure rate for active bus after reliability calcs
N_Customers:	Integer	readonly Returns total number of customers downline from the active bus after reliability calcs
N_interrupts:	Double	readonly Returns total number of annual interruptions for the active bus after reliability calcs
Name:	WideString	readonly Name of active bus. Set the active bus in the Circuit interface (SetActiveBus function or Buses(i) property).

Nodes:	OleVariant	readonly A variant array of integers containing the order of the nodes at this bus. This is the order of the Voltages and puVoltage arrays.
NumNodes:	Integer	readonly Number of nodes at the active bus.
SeqVoltages:	OleVariant	readonly Similar to CplxSeqVoltage except voltage magnitudes only. Volts. Order is 0, 1, 2.
Voc:	OleVariant	readonly Open-circuit voltages after fault study. Variant array of complex voltages.
Voltages:	OleVariant	readonly Variant array of doubles representing complex voltage phasors, volts, (a + jb) form, for the most recent solution.
YscMatrix:	OleVariant	readonly Variant array of doubles containing complex short circuit Admittance matrix, column by column.
Zsc0:	OleVariant	readonly Zero-sequence short-circuit impedance looking into the bus. Variant array of doubles (2 values).
Zsc1:	OleVariant	readonly Positive-sequence short-circuit impedance looking into the bus. Variant array of doubles (2 values).
ZscMatrix:	OleVariant	readonly Variant array of doubles containing complex short circuit Impedance matrix, column by column.
kVBase:	Double	readonly Base kV for the bus.
puVoltages:	OleVariant	readonly Same as Voltage property except values in per unit on base kV of the bus.
x:	Double	X coordinate (for circuit plot)
y:	Double	Y coordinate

### ***Functions***

GetUniqueNodeNumber (StartNumber:Integer):	Integer	To help avoid collisions of neutral node numbers for specifying connections of circuit elements, this function returns a node number that is not being used, Starting at the StartNode value
ZscRefresh	WordBool	Refresh Zsc, Ysc values. Execute after a major change in the circuit. Not necessary after fault study.

### Example

This VBA sub loads the present solution's node voltages in complex form. If voltage bases are defined for the buses, the voltages are in per unit. One unique feature of this sub is that it tacks the node order onto the bus name in column 1. Then it places the voltage(s) in the cells according to the phase, or node, it corresponds to.

```
Public Sub LoadVoltages()

' This Sub loads the per unit complex voltages onto Sheet3 starting in Row 2

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

    Set WorkingSheet = Worksheets("VoltSheet") ' VoltSheet = Sheet3

    WorkingSheet.Rows("2:" & Rows.Count).ClearContents

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

        Set DSSBus = DSSCircuit.Buses(i) ' Set i-th bus active using Buses

        ' Loads pu voltages (complex) at active bus as variant array of doubles
        V = DSSBus.puVoltages
        NodeOrder = DSSBus.Nodes

        ' Construct full bus name
        Str = DSSCircuit.ActiveBus.Name
        For j = LBound(NodeOrder) To UBound(NodeOrder)
            Str = Str + "." & CStr(NodeOrder(j))
        Next j

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

        ' Put values in Variant array into cells in sequence provided by DSS
        iCol = 2
        With WorkingSheet
            For j = LBound(V) To UBound(V) Step 2
                iCol = NodeOrder(j / 2) * 2
                .Cells(iRow, iCol).Value = V(j)
                .Cells(iRow, iCol + 1).Value = V(j + 1)
            Next j
        End With

        iRow = iRow + 1
    Next i

End Sub
```

## Result in worksheet:

This is the first few rows of the solution for the IEEE 8500-node test feeder.

Bus	Re(1)	Imag(1)	Re(2)	Imag(2)	Re(3)	Imag(3)
_hvmv_sub_lsb.1.2.3	0.865398871	-0.593267853	-0.943739082	-0.455649678	0.073946307	1.04775717
hvmv_sub_48332.1.2.3	0.865398339	-0.593268409	-0.943739279	-0.455648964	0.073946998	1.047757071
m1009763.1.2.3	0.707352037	-0.705500302	-0.987319644	-0.227647038	0.185599725	1.013020715
l2673322.2			-0.987304697	-0.227639354		
m1069148.3					0.170798901	1.016329408
l2673309.3					0.17080227	1.016288397
Etc. etc.						