

TechNote Parser Interface

From OpenDSSWiki

Parser Interface Added to COM Interface

In version 7.6.3.27 we have added an interface to the internal OpenDSS command line parser to the COM interface. This will help users who want to formulate their own version of a scripting language like the one in OpenDSS.

Also, it is better for parsing CSV files than the features in some languages. And it is more forgiving in many ways when there are inconsistencies in the data to be parsed.

Basic Usage

First, define a public variable, such as **DSSParser**, to retain the link to the Parser interface when you start the server. For example, the typical VBA code for starting the server is something like this:

```
Public DSSobj As OpenDSSengine.DSS
Public DSSText As OpenDSSengine.Text
...
Public DSSParser As OpenDSSengine.Parser
...

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

' Start the DSS
If Not DSSobj.Start(0) Then
    MsgBox "DSS Failed to Start"
Else
    ' MsgBox "DSS Started successfully"
    ' Assign a variable to each of the interfaces for easier access
    Set DSSText = DSSobj.Text
    Set DSSCircuit = DSSobj.ActiveCircuit
    Set DSSSolution = DSSCircuit.Solution
    ...
    Set DSSParser = DSSobj.Parser
```

Then load a string to be parsed into **DSSParser.CmdString**.

Once the parser is loaded, you can parse off the tokens one at a time. The tokens can have the same form as you would use in OpenDSS, including arrays and in-line math in RPN form.

If you wish, you can even change the hard delimiters and quote pairs.

You advance to the next token on the line by invoking the **NextParam** property. This returns the name of the field, if any (if an "=" was found). You can also check the string value of any token value at this time by invoking the **StrValue** property.

If you are expecting a double value, use the **DblValue** property.

If you are expecting an integer value, use the **IntValue** property.

When AutoIncrement is False, you must invoke the NextParam property to advance to the next token. If you are confident in the format of your data, you can set AutoIncrement to True and the parser will automatically advance to the next token for StrValue, DblValue, and IntValue properties.

The **Vector** and **Matrix** properties each return a variant array of doubles given input data in standard OpenDSS array and matrix format.

The Matrix properties returns the array in column order like Fortran would for two-dimensional arrays.

VBA Example

You can run this example in Excel to test the Parser Interface. You first define the DSSParser variable as above when OpenDSSEngine is started. It demonstrates parsing the same line of text with and without the AutoIncrement feature turned on.

```
Public Sub TestParser()

Dim Mydbl As Double
Dim Myint As Long
Dim MyArray As Variant
Dim MyMatrix As Variant
Dim ParamName As String, TokenValue As String
Dim i As Long

' VBA Example of using features of DSSParser interface to parse a text line

With DSSParser

    .AutoIncrement = False

    .CmdString = "cmd=ParseThiscmd dblvalue=1.234  intvalue=56  strvalue=teststring  Array=[10 11 1

' Open an output file
Open "Parsertest.txt" For Output As #1

    Print #1, "Command String"
    Print #1, .CmdString
    Print #1,

' cmd
    ParamName = .NextParam
    TokenValue = .StrValue
    Print #1, ParamName; "="; TokenValue

' dbl
```

```

    ParamName = .NextParam
    TokenValue = .StrValue
    Mydbl = .DblValue
    Print #1, ParamName; "="; TokenValue; ": "; Mydbl

' int
    ParamName = .NextParam
    TokenValue = .StrValue
    Myint = .IntValue
    Print #1, ParamName; "="; TokenValue; ": "; Myint

' strvalue
    ParamName = .NextParam
    TokenValue = .StrValue
    Print #1, ParamName; "="; TokenValue

' Array of dbls
    ParamName = .NextParam
    TokenValue = .StrValue
    MyArray = .Vector(10) ' specify expected size bigger than array; Parser will correct
    Print #1, ParamName; "="; TokenValue; ": ";
    For i = LBound(MyArray) To UBound(MyArray)
        Print #1, " ", MyArray(i);
    Next i
    Print #1,

' 2 x 2 Matrix of dbls
    ParamName = .NextParam
    TokenValue = .StrValue
    MyArray = .Matrix(2) ' return a 2x2 in column order
    Print #1, ParamName; "="; TokenValue; ": "
        Print #1, "Row 1: ", MyArray(0), " ", MyArray(2)
        Print #1, "Row 2: ", MyArray(1), " ", MyArray(3)

' Now repeat the same code with AutoIncrement ON
' NextParam is automatically called after the token is retrieved
' ParamName is not set and can't call strValue without advancing token pointer

Print #1,
Print #1, "----- with AutoIncrement -----"
Print #1,

.AutoIncrement = True
.CmdString = "cmd=ParseThiscmd dblvalue=1.234 intvalue=56 strvalue=teststring Array=[10 11 1
    Print #1, "Command String"
    Print #1, .CmdString
    Print #1,

' cmd
    TokenValue = .StrValue
    Print #1, TokenValue

' dbl
    Mydbl = .DblValue
    Print #1, Mydbl

' int
    Myint = .IntValue
    Print #1, Myint

' strvalue
    Print #1, .StrValue

```

```
.AutoIncrement = False ' can't use autoincrement with Vector and Matrix

' Array of dbls
ParamName = .NextParam
MyArray = .Vector(10) ' specify expected size bigger than array. Parser will correct.
Print #1, "Array: ";
For i = LBound(MyArray) To UBound(MyArray)
    Print #1, " ", MyArray(i);
Next i
Print #1,

' 2 x 2 Matrix of dbls
ParamName = .NextParam
MyArray = .Matrix(2) ' return a 2x2 in column order
Print #1, "2 x 2 Matrix"; ": "
    Print #1, "Row 1: ", MyArray(0), " ", MyArray(2)
    Print #1, "Row 2: ", MyArray(1), " ", MyArray(3)

Close #1

End With

Shell ("notepad Parsertest.txt")

End Sub
```

--Rdugan 19:45, 28 April 2014 (UTC)

Retrieved from "http://localhost/mediawiki/index.php?title=TechNote_Parser_Interface&oldid=685"

-
- This page was last modified on 28 April 2014, at 12:21.
 - This page has been accessed 74 times.