

# Determining the Native Type of an Element Parameter

## Summary

Each of the Element Parameters has a reference to an object called a ParameterDefinition. A ParameterDefinition is associated with the model for the object and describes the attributes of the parameters that will be associated with elements representing that model. So each model includes an array of parameter definitions that act like templates for creating the parameters associated with an Element representing the model. To get information about the type of the Parameter object we need to look at the ParameterDefinition or template from which it was created. For example lets create a script that adds a schematic, adds a MLIN element to that schematic and then prints some information about the parameters for that element.

## Code Snippets

```
' Code Module
Sub Main
    Dim schem As Schematic
    Dim elem As Element
    Dim param As Parameter
    Dim paramDef As ParameterDefinition

    Debug.Clear
    Set schem = Project.Schematics.Add("MySchematic")
    Set elem = schem.Elements.Add("MLIN", 0, 0)

    For Each param In elem.Parameters
        Debug.Print param.Name & " = " & param.ValueAsString;
        Debug.Print "  Data Type = " & param.ParameterDefinition.DataType
    Next param

End Sub
```

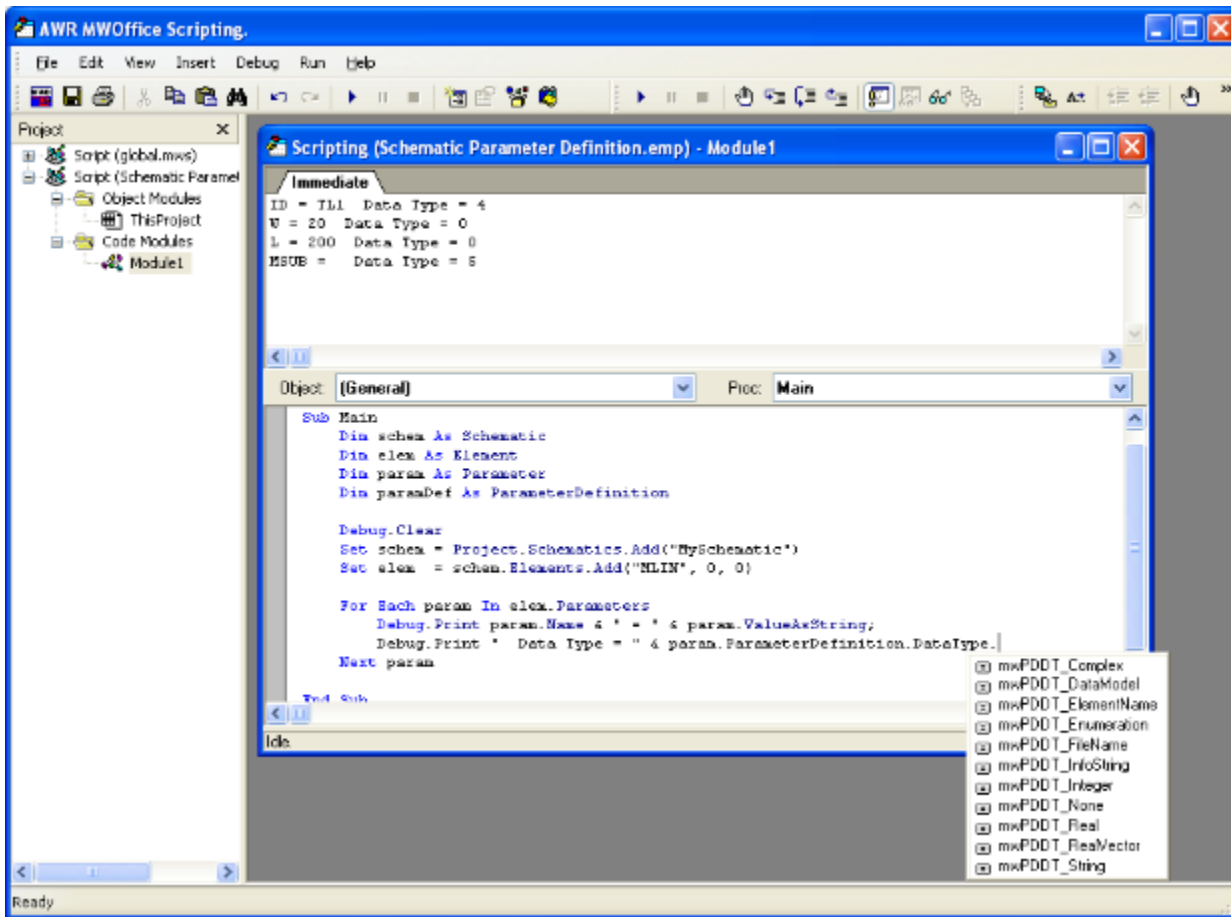
If we run this program we get the following output:

```
ID = TL1  Data Type = 4
W = 20  Data Type = 0
L = 200  Data Type = 0
MSUB =   Data Type = 5
```

While this is useful in that it provides values for the data type it would be nicer if we had some enumerations. In the object browser we can see that the ParameterDefinition.DataType is a property of type mwParamDefDataType which has the following definition.

```
enum mwParamDefDataType
{
    mwPDDT_Real,
    mwPDDT_Complex,
    mwPDDT_Integer,
    mwPDDT_String,
    mwPDDT_ElementName,
    mwPDDT_DataModel,
    mwPDDT_InfoString,
    mwPDDT_None,
    mwPDDT_Enumeration,
    mwPDDT_RealVector,
    mwPDDT_FileName
} mwParamDefDataType;
```

Also if we place a "." after the DataType property then it will give us a list of the possible values as shown in Figure 11.



This can be very useful if we are trying to control flow based on some specific data type, like printing only double values, however in our case we want to print the data type so I'll create a mapping function to map from the type value to a string. A quick mapping function might look like this:

```

Function ParameterDefTypeName(id As Integer) As String
    Dim ret As String

    Select Case id
    Case mwPDDT_Real
        ret = "mwPDDT_Real"
    Case mwPDDT_Complex
        ret = "mwPDDT_Complex"
    Case mwPDDT_Integer
        ret = "mwPDDT_Integer"
    Case mwPDDT_String
        ret = "mwPDDT_String"
    Case mwPDDT_ElementName
        ret = "mwPDDT_ElementName"
    Case mwPDDT_DataModel
        ret = "mwPDDT_DataModel"
    Case mwPDDT_InfoString
        ret = "mwPDDT_InfoString"
    Case mwPDDT_None
        ret = "mwPDDT_None"
    Case mwPDDT_Enumeration
        ret = "mwPDDT_Enumeration"
    Case mwPDDT_RealVector
        ret = "mwPDDT_RealVector"
    Case mwPDDT_FileName
        ret = "mwPDDT_FileName"
    Case Default
        ret = "Unkown"
    End Select

    ParameterDefTypeName = ret

End Function

```

And then we can use this in the main body of the code:

```

' Code Module
Sub Main
    Dim schem As Schematic
    Dim elem As Element
    Dim param As Parameter
    Dim paramDef As ParameterDefinition

    Debug.Clear
    Set schem = Project.Schematics.Add("MySchematic")
    Set elem = schem.Elements.Add("MLIN", 0, 0)

    For Each param In elem.Parameters
        Debug.Print param.Name & " = " & param.ValueAsString;
        Debug.Print "  Data Type = " & ParameterDefTypeName(param.ParameterDefinition.DataType)
    Next param

End Sub

```

Which now produces the following output:

```

ID = TL1  Data Type = mwPDDT_ElementName
W = 20  Data Type = mwPDDT_Real
L = 200  Data Type = mwPDDT_Real
MSUB =  Data Type = mwPDDT_DataModel

```

So the detailed type information about a parameter is available from the ParameterDefinition which acts as a template to create the parameter when the Element is created and the parameter object includes a reference back to it's ParameterDefinition object.