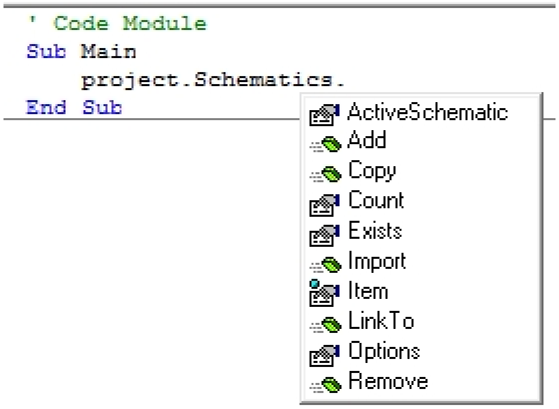
Scripting How-To: What is the Difference Between the Schematic and Schematics Objects

The AWRDE object hierarchy is structured so that there are single objects (Schematic) and collections of similar objects (Schematics). The programmer must understand whether he is trying to operate on the collection as a whole or an individual member of the collection. This discussion uses schematics as the example, but the same collection/member relationship applies throughout the object hierarchy. Other objects that share this type of organization include (but not limited to) graphs, EM structures, elements, parameters, etc.

An examination of the Intellisense listing for a collection (enter "project.schematics.") will show only a few methods/properties:



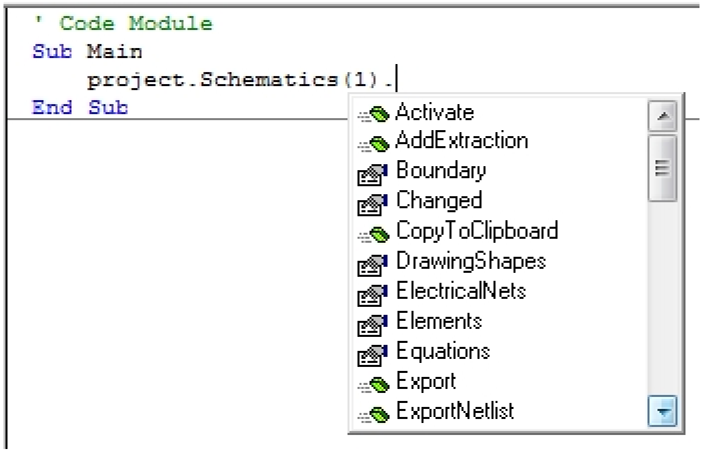
```
' Code Module
Sub Main
    project.Schematics.
End Sub
```

The methods (Add, Copy, Import, LinkTo, Remove) all operate on the collection, and this is the way to add/remove schematics from a project. The properties (ActiveSchematic, Count, Exists, Item, Options) also relate to the collection.

Members of the collection can be accessed several ways. When you type "Project.Schematics(", Intellisense says it is looking for a Variant. Assuming the project has one schematic, called "Amp", the following code accesses the schematic:

```
 Project.Schematics(1)
Project.Schematics("Amp")
```

Using either method, we now have a handle on a specific member of the collection and the Intellisense reflects the methods/properties available at that level:



```
' Code Module
Sub Main
    project.Schematics(1).
End Sub
```

One very convenient feature offered by this arrangement is the ability to assign a shortcut to an object as it is created. The following code:

```
Dim el  as Element
 Dim sc  As Schematic

 Set sc = Project.Schematics.Add( "mySch")
 Set el = sc.Elements.Add( "RES",0,0,0, False)
el.Parameters( "R").ValueAsDouble = 10
```

As the new schematic is created, it's assigned to sc (which must be dimensioned as an appropriate object), eliminating the need to type "Project. Schematics("mySch") from that point forward. Similarly, el is assigned to the new element. Without these shortcuts, the last line of code would look like this:

```
 Project.Schematics("mySch").Elements( "RES.R1").Parameters( "R").ValueAsDouble = 10
```

Beyond the likelihood of a typo, the first method produces code that is much easier to read.