roXMLElement

roXMLElement is used to contain an XML tree.

Supported Interfaces

- ifXMLElement

Description

Example:

```
    <tag1>this is some text</tag1>
```

Would parse such that:

```
   Name = "tag1"
   Attributes = invalid
   Body = roString with "this is some text"
```

Example:

```
<emptytag caveman="barney" />
```

Would parse such that:

```
   Name = "emptytag"
   Attributes = roAssociativeArray, with one entry { caveman: "barney" }
   Body = invalid
```

If the tag contains other tags, body will be of type roXMLList.

To generate XML, create an roXMLElement, then use functions like SetName(), AddAttribute(), SetBody(), AddElementWithBody(), AddElement(), AddBodyElement(), and AddText() functions to build the XML object hierarchy.

Then call GenXML() to return the XML as a string.

GenXML() takes one parameter (boolean) that indicates whether the generated xml should have the <?xml …> tag at the top.

Example subroutine to print out the contents of an roXMLElement tree

```
PrintXML(root, 0)

Sub PrintXML(element As Object, depth As Integer)
    print tab(depth*3);"Name: ";element.GetName()
    if not element.GetAttributes().IsEmpty() then
        print tab(depth*3);"Attributes: ";
        for each a in element.GetAttributes()
            print a;"=";left(element.GetAttributes()[a], 20);
            if element.GetAttributes().IsNext() then print ", ";
        end for
        print
    end if
    if element.GetText()<>invalid then
        print tab(depth*3);"Contains Text: ";left(element.GetText(), 40)
    end if
    if element.GetChildElements()<>invalid
        print tab(depth*3);"Contains roXMLList:"
        for each e in element.GetChildElements()
            PrintXML(e, depth+1)
        end for
    end if
    print
end sub
```

Example generating XML

```
root.SetName("myroot")
root.AddAttribute("key1", "value1")
root.AddAttribute("key2", "value2")
ne = root.AddBodyElement()
ne.SetName("sub")
ne.SetBody("this is the sub1 text")
ne = root.AddBodyElement()
ne.SetName("subelement2")
ne.SetBody("more sub text")
ne.AddAttribute("k", "v")
ne = root.AddElement("subelement3")
ne.SetBody("more sub text 3")
root.AddElementWithBody("sub", "another sub (#4)")
PrintXML(root, 0)
print root.GenXML(false)
```

Another Example

```
xml = CreateObject("roXMLElement")
xml.SetName("root")
subel1 = xml.AddBodyElement()
subel1.SetName("subelement1")
subel2 = xml.AddBodyElement()
subel2.SetName("subelement2")
```

Is the same as:

```
xml = CreateObject("roXMLElement")
xml.SetName("root")
subel1 = xml.AddElement("subelement1")
subel2 = xml.AddElement("subelement2")
```