



Initiate XSLT in a Script

Currently, the default XML parser for supported Internet Explorer versions is MSXML 3.0.

Until Internet Explorer ships with MSXML 4.0 or later as its default XML parser, the features in MSXML 4.0 and later are available in Internet Explorer only via scripting, when an XML DOM object is instantiated using the MSXML4 (or later)-specific ProgID. This means that you cannot use

```
<?xml-stylesheet type="text/xsl" href="Products.xsl"?>
```

in an XML document to invoke MSXML 4.0 or later functionality.

However, you can write either server-side or client-side script to use MSXML 6.0 instead. In this SDK, we provide two ways to do this:

- Initiate XSLT from an HTML Page
- Initiate XSLT from a Command Prompt

The following examples illustrate each method. You can use the code in these examples to test other samples in this SDK. Just change the XML and XSLT file names as appropriate.

Initiate XSLT from an HTML Page

The following is an example.

Example

When the following HTML page is loaded, the

```
init()
```

function is invoked. This function loads the **hello.xml** and **hello.xsl** files into XML DOM objects:

```
srcTree
```

and

```
xsltTree
```

, respectively. It then applies the XSLT transformation by calling the

```
transformNode
```

method on the

```
scrTree
```

XML DOM object.

You can use this HTML page to test many of the other XSLT examples in this SDK, as well. You might want to use this approach if the output of the transformation is HTML. Otherwise, you should initiate XSLT from a command prompt, as described later in this topic.

For **hello.xml** and **hello.xsl**, see [Hello, World! \(XSLT\)](http://msdn.microsoft.com/en-us/library/ms765388(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms765388\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms765388(VS.85).aspx)]. The following is the code for the HTML page, **hello.htm**.

hello.htm

Html

[Copy Code](#)

```
<HTML>
<HEAD>
  <TITLE>sample</TITLE>
  <SCRIPT language = "javascript">
    function init()
```

```
{
    var srcTree = new ActiveXObject("Msxml2.DOMDocument.6.0");
    srcTree.async=false;
    // You can substitute other XML file names here.
    srcTree.load("hello.xml");

    var xsltTree= new ActiveXObject("Msxml2.DOMDocument.6.0");
    xsltTree.async = false;
    // You can substitute other XSLT file names here.
    xsltTree.load("hello.xsl");

    resTree.innerHTML = srcTree.transformNode(xsltTree);
}
</SCRIPT>
</HEAD>

<BODY onload = "init()" >
    <div id="resTree"></div>
</BODY>

</HTML>
```

 Note

MSXML 6.0 is used here, with the help of the version dependent ProgID of the DOM,

`Msxml2.DOMDocument.6.0`

. When you use this method to start the transformation, any XSLT style sheet that is already embedded in the source document is ignored. For more information about GUIDs and ProgIDs, see [GUID and ProgID Information](http://msdn.microsoft.com/en-us/library/ms757837(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms757837\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms757837(VS.85).aspx)] .

Try It!

1. Copy the code above, and paste it into a text file. Save the file as **hello.htm**.
2. Copy the XML source file, **hello.xml**, from the [Hello, World!](http://msdn.microsoft.com/en-us/library/ms765388(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms765388\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms765388(VS.85).aspx)] topic in the XSLT Starter Kit. Paste it into a text file, and save the file as **hello.xml**, in the same directory where you saved **hello.htm**.
3. Copy the XSLT style sheet, **hello.xsl**, from [Hello, World!](http://msdn.microsoft.com/en-us/library/ms765388(VS.85).aspx) [[http://msdn.microsoft.com/en-us/library/ms765388\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms765388(VS.85).aspx)] . Paste it into a text file, and save the file as **hello.xsl**, in the same directory where you saved **hello.htm** and **hello.xml**.
4. Double-click **hello.htm**. An Internet Explorer window should appear, containing the text "Hello, World! from an XSLT Programmer", formatted as specified in the XSLT style sheet.

[Initiate XSLT from a Command Prompt](#)

You can use the following JScript file to test XSLT samples quickly from a command prompt. This approach will work for transformations with any type of output.

xsltTest.js

J#

[Copy Code](#)

```
var oArgs = WScript.Arguments;

if (oArgs.length == 0)
{
    WScript.Echo ("Usage : cscript xslt.js xml xsl");
    WScript.Quit();
}
xmlFile = oArgs(0) + ".xml";
xslFile = oArgs(1) + ".xsl";
```

```
var xsl = new ActiveXObject("MSXML2.DOMDOCUMENT.6.0");
var xml = new ActiveXObject("MSXML2.DOMDocument.6.0");
xml.validateOnParse = false;
xml.async = false;
xml.load(xmlFile);

if (xml.parseError.errorCode != 0)
    WScript.Echo ("XML Parse Error : " + xml.parseError.reason);

xsl.async = false;
xsl.load(xslFile);

if (xsl.parseError.errorCode != 0)
    WScript.Echo ("XSL Parse Error : " + xsl.parseError.reason);

try
{
    WScript.Echo (xml.transformNode(xsl.documentElement));
}
catch(err)
{
    WScript.Echo ("Transformation Error : " + err.number + "*" + err.description);
}
```

To test samples with xsltTest.js

1. Copy the JScript code above, and paste it into a text file. Save the file as xsltTest.js in your system directory, such as c:\Windows. This allows you to invoke the script from any directory. If you save xsltTest.js to any other directory, you can either invoke it from that directory or invoke it using its absolute path.
2. Copy the XML and XSLT files you want to test. Paste them into text files, and save them to a test directory, such as c:\test. Name the files as appropriate—for example, sampleXML.xml and sampleXSLT.xsl.
3. Open a command prompt, and navigate to your test directory. Enter the following command:

[Copy Code](#)

```
xsltTest sampleXML sampleXSLT
```

Notice the absence of file extensions on the arguments.