XPath Tutorial Application

To help you get started working with XPath, this section shows you how to quickly build an interactive tutorial application that visually highlights the results of various XPath patterns as you selectively use them to query the data in a brief XML document file.

To run the XPath tutorial, you will need to make five files:

- Sample XML File (authors.xml) — A short XML sample data file showing book author information.
- HTML File (author-patterns.htm) — The HTML application form used to run the application.
- XSLT File (hilite-xml.xsl) — The primary style sheet used by the HTML file to highlight the XPath expression you specify.
- XSLT File (makeID.xsl) — Another secondary style sheet called by **hilite-xml.xsl** and used to generate IDs.
- Error Reporting Script (reportErrors.js) — A JScript file used as a helper file for handling and reporting errors when running and debugging the sample application.

To create and run the XPath tutorial application

1. Open Notepad.
2. Create the five files: **authors.xml**, **author-patterns.htm**, **hilite-xml.xsl**, **makeID.xsl**, and **reportErrors.js**.

   To create the files, copy and paste each of the following samples separately to the Notepad window. Save them all in the same folder, using the names provided.
3. Open the **author-patterns.htm** file.
4. Try experimenting with various XPath patterns and observing the result in the highlighted document.

   To update the highlighted selection, you can either click an XPath pattern in the list of linked patterns on the left. Alternately, you can type a different pattern in the **XPath Expression** text box, and then click **Select Nodes** to try out other XPath patterns.

Sample XML File (authors.xml)

The following sample code creates the short XML sample data file containing book author information used by the XPath tutorial application.

Copy Code

```
<?xml version='1.0'?>
<authors>
  <author>
    <name>Mike Galos</name>
    <nationality>French</nationality>
  </author>
  <author period="modern">
    <name>Eva Corets</name>
    <nationality>English</nationality>
  </author>
```

```
  <author>
    <name>Cynthia Randall</name>
    <nationality>Canadian</nationality>
  </author>
  <author>
    <name>Stefan Knorr</name>
    <nationality>Canadian</nationality>
  </author>
  <author period="modern">
    <name>Paula Thurman</name>
    <nationality>English</nationality>
  </author>
</authors>
```

HTML File (author-patterns.htm)

The following sample code creates the HTML application form used to run the XPath tutorial application.

Html Copy Code

```
<HTML>
<HEAD>
  <TITLE>XPath Tutorial Application</TITLE>
  <STYLE>
    BODY {font-family:Arial}
    .heading {font-family:"Arial Black"}
    .sample {cursor:hand; font:9pt Courier; text-decoration:underline; text-indent:-
1em; margin-left:1em}
    .note {font-size:x-small; font-style:italic}
  </STYLE>
<!--TOOLBAR_START-->
<!--TOOLBAR_EXEMPT-->
<!--TOOLBAR_END-->
</HEAD>

<SCRIPT src="reportErrors.js"></SCRIPT>

<SCRIPT>
  var makeID, source, stylesheet;

  function doQuery(theQuery)
  {
     xmlSrc.innerHTML = result;
     var q = source.selectNodes(theQuery);

    for (var e = q.nextNode(); e != null; e = q.nextNode())
    {
      try
      {
         eID = e.transformNode(makeID);
         document.all.item(eID).style.backgroundColor = "yellow";
      }
      catch (exception)
      {
         result = reportRuntimeError(exception);
         alert(result);
      }
    }
  }

  function loadQuery()
  {
     var q = window.event.srcElement.innerText;
     qryIn.value = q;
```

```
     doQuery(q);
  }

</SCRIPT>

<SCRIPT FOR="window" EVENT="onload">
   makeID = new ActiveXObject("Msxml2.DOMDocument.6.0");
   makeID.async = false;
   makeID.load("makeID.xsl");
   if (makeID.parseError.errorCode != 0)
   {
      result = reportParseError(makeID.parseError);
      alert(result);
   }

   source = new ActiveXObject("Msxml2.DOMDocument.6.0");
   source.async=false;
   source.load("authors.xml");
   source.setProperty("SelectionLanguage", "XPath");
   if (source.parseError.errorCode != 0)
   {
      result = reportParseError(source.parseError);
      alert(result);
   }

   stylesheet = new ActiveXObject("Msxml2.DOMDocument.6.0");
   stylesheet.async = false;
   stylesheet.load("hilite-xml.xsl");
   if (source.parseError.errorCode != 0)
   {
      result = reportParseError(source.parseError);
      alert(result);
   }

  try
  {
    result = source.transformNode(stylesheet);
  }
  catch (e)
  {
    result = reportRuntimeError(e);
  }
   xmlSrc.innerHTML = result;
</SCRIPT>

<BODY>
<DIV class=note>Demonstration of quering against an XML document using XPath
expressions.</DIV>

<DIV class=heading>XPath Expression:</DIV>
<INPUT ID=qryIn SIZE=60 TYPE=TEXT>
<INPUT TYPE="BUTTON" VALUE="Select Nodes" onClick="doQuery(qryIn.value)" id=BUTTON1
name=BUTTON1>
<BR>

<TABLE>
<TR>
<TD VALIGN="top">
<DIV class=heading>Some Sample Queries:</DIV>
<DIV class=sample onclick="loadQuery()">authors</DIV>
<DIV class=sample onclick="loadQuery()">authors/author</DIV>
<DIV class=sample onclick="loadQuery()">authors/author/name</DIV>
<DIV class=sample onclick="loadQuery()">authors/*/name</DIV>
<DIV class=sample onclick="loadQuery()">authors/author/*</DIV>
<DIV class=sample onclick="loadQuery()">authors/author[nationality]/name</DIV>
<DIV class=sample onclick="loadQuery()">authors/author
```

```
[nationality='Canadian']/name</DIV>
<DIV class=sample onclick="loadQuery()">authors/author[@period="modern"]</DIV>
<DIV class=sample onclick="loadQuery()">authors/author/@period</DIV>

</TD>
<TD VALIGN="top">
<DIV class=heading>XML source document</DIV>
<SPAN id=xmlSrc></SPAN>
</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

When this page is opened in the browser, it enables you to type or click an XPath expression and see the result automatically highlighted in the displayed copy of the XML source document, a short XML sample data file containing book author information.
XSLT File (hilite-xml.xsl)

The following sample code creates the primary XSLT style sheet used to run the XPath tutorial application.

Xml Copy Code

```
<?xml version="1.0"?>
<!-- Generic stylesheet for viewing XML -->
<xsl:stylesheet version="1.0"
                xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                xmlns:msxsl="urn:schemas-microsoft-com:xslt">

  <xsl:template match="/">
    <DIV STYLE="font-family:Tahoma; font-size:11pt; margin-bottom:2em">
      <xsl:apply-templates/>
    </DIV>
  </xsl:template>

  <xsl:template match="*">
    <DIV STYLE="margin-left:1em; color:red">
      <xsl:attribute name="id">
            <xsl:value-of select="generate-id()"/>
      </xsl:attribute>
      &lt;<xsl:value-of select="name()"/><xsl:apply-templates select="@*"/>/&gt;
    </DIV>
  </xsl:template>

  <xsl:template match="*[node()]">
    <DIV STYLE="margin-left:1em">
      <SPAN STYLE="color:olive">
         <xsl:attribute name="id">
             <xsl:value-of select="generate-id()"/>
          </xsl:attribute>&lt;<xsl:value-of select="name()"/><xsl:apply-templates
select="@*"/>&gt;</SPAN>
      <xsl:apply-templates select="node()"/>
      <SPAN STYLE="color:olive">&lt;/<xsl:value-of select="name()"/>&gt;</SPAN>
    </DIV>
  </xsl:template>

  <xsl:template match="@*" xml:space="preserve">
    <SPAN STYLE="color:navy"><xsl:attribute name="ID"><xsl:value-of select="generate-
id()"/></xsl:attribute>
       <xsl:value-of select="name()"/>="<SPAN STYLE="color:black"><xsl:value-of
select="."/></SPAN>"
    </SPAN>
  </xsl:template>
```

```
  <xsl:template match="processing-instruction()">
    <DIV STYLE="margin-left:1em; color:maroon">
      <xsl:attribute name="id">
            <xsl:value-of select="generate-id()"/>
      </xsl:attribute>
      &lt;?<xsl:value-of select="name()"/><xsl:apply-templates select="@*"/>?&gt;
    </DIV>
  </xsl:template>

  <xsl:template match="node()[nodeTypeString=cdatasection]">
    <pre>
       <xsl:attribute name="id">
            <xsl:value-of select="generate-id()"/>
       </xsl:attribute>cdata
       &lt;![CDATA[<xsl:value-of select="."/>]]&gt;
    </pre>
  </xsl:template>

  <xsl:template match="text()">
    <SPAN>
      <xsl:attribute name="id">
            <xsl:value-of select="generate-id()"/>
      </xsl:attribute><xsl:value-of select="."/>
    </SPAN>
  </xsl:template>

</xsl:stylesheet>
```

XSLT File (makeID.xsl)

The following sample code creates the secondary XSLT style sheet used to generate IDs, which are called and used by the primary XSLT file (hilite-xml.xsl) listed above.

Xml Copy Code

```
<xsl:stylesheet version="1.0"
                xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                xmlns:msxsl="urn:schemas-microsoft-com:xslt">
  <xsl:output omit-xml-declaration="yes"/>

  <xsl:template match="/">
      <xsl:apply-templates/>
  </xsl:template>

  <xsl:template match="*">
      <xsl:value-of select="generate-id()"/>
  </xsl:template>

  <xsl:template match="*[node()]">
      <xsl:value-of select="generate-id()"/>
  </xsl:template>

  <xsl:template match="@*">
      <xsl:value-of select="generate-id()"/>
  </xsl:template>

  <xsl:template match="processing-instruction()">
      <xsl:value-of select="generate-id()"/>
  </xsl:template>

  <xsl:template match="node()[nodeTypeString=cdatasection]">
      <xsl:value-of select="generate-id()"/>
```

```
  </xsl:template>

  <xsl:template match="text()">
      <xsl:value-of select="generate-id()"/>
  </xsl:template>

</xsl:stylesheet>
```

Error Reporting Script (reportErrors.js)

The following sample code creates a helper file for debugging purposes when running the XPath tutorial application.

J# Copy Code

```
// JScript File (reportErrors.js)

// Parse error formatting function
function reportParseError(error)
{
  var s = "";
  for (var i=1; i<error.linepos; i++) {
    s += " ";
  }
  r = "<font face=Verdana size=2><font size=4>XML Error loading '" +
      error.url + "'</font>" +
      "<P><B>" + error.reason +
      "</B></P></font>";
  if (error.line > 0)
    r += "<font size=3><XMP>" +
    "at line " + error.line + ", character " + error.linepos +
    "\n" + error.srcText +
    "\n" + s + "^" +
    "</XMP></font>";
  return r;
}


// Runtime error formatting function.
function reportRuntimeError(exception)
{
  return "<font face=Verdana size=2><font size=4>XSL Runtime Error</font>" +
      "<P><B>" + exception.description + "</B></P></font>"
      "<P>" + exception + "</P>";
}
```