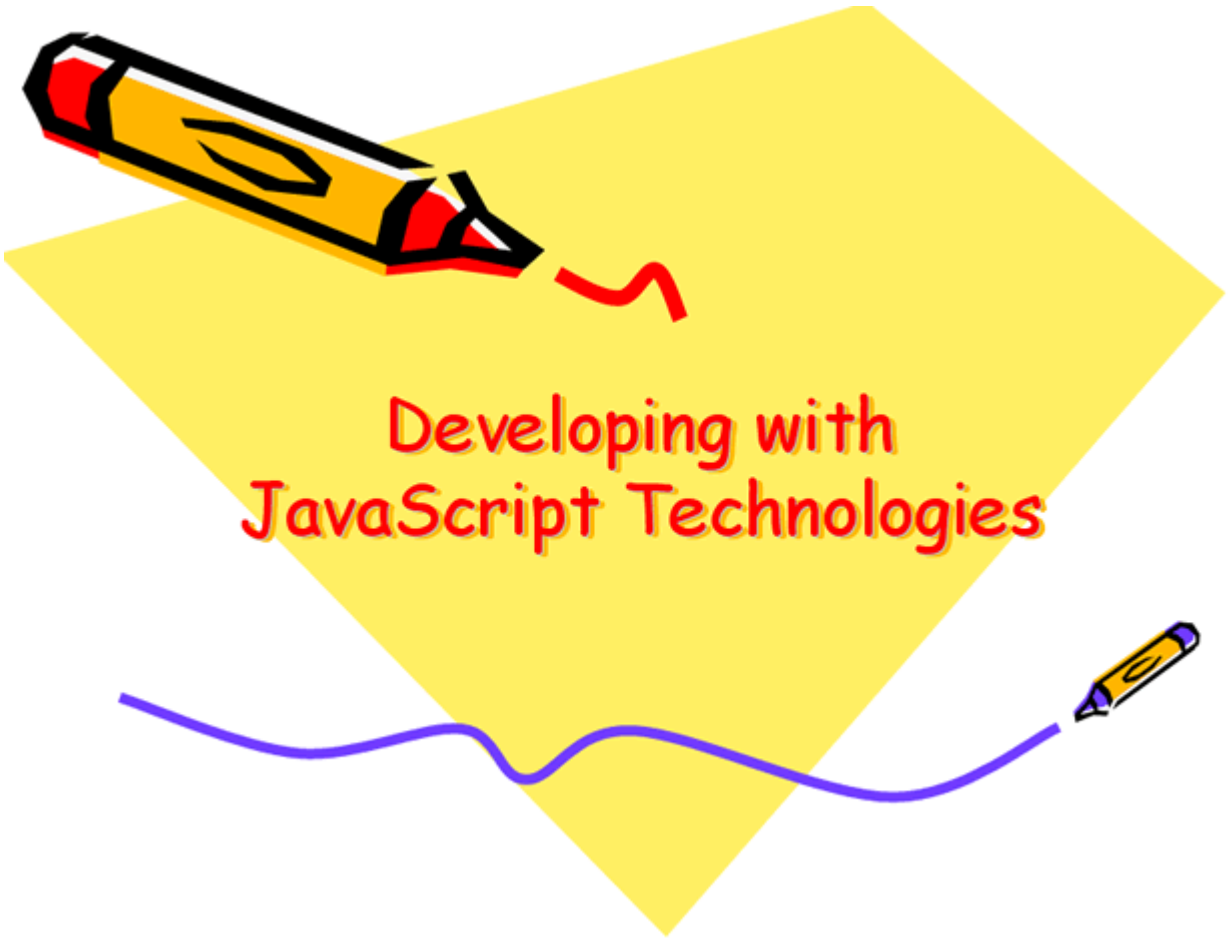
transentia

Exercise 8



XML in IE

Introduction

In this exercise you will look at some of the facilities built into Internet Explorer for manipulating XML-formatted data using JavaScript.

Setting Up

Open a new Command Prompt window and execute the following command:

```
C:\> xcopy /ieq "Z:\Exercises\8 ie\ie" C:\ie
```

Notes:

- This assumes that your CD-ROM device is Z:, you should use your real device letter instead
- Do not simply drag&drop from the CD-ROM; if you do, you will end up with a read-only directory structure and this will complicate things later on

All the files that you subsequently create or edit as you do this exercise should be contained in this directory.

The XML Data

In the eighteenth century, the French Astronomer Charles Messier produced a list of bright, non-stellar objects that is now called the Messier Catalogue. His real interest was searching for comets, but he was continually distracted by people asking him about various well-known objects that simply happened to resemble comets. He produced his catalogue to allow people to check their 'discoveries' for themselves.

You have been given the list of the 110 Messier objects as an XML-formatted data file in **C:\ie\Messier.xml**. You should take a quick look at this file (simply double-click on its icon: it will open for browsing in IE).

Using the IE Data Source Object

Internet Explorer incorporates a facility known as the Data Source Object (DSO). Among other things, the DSO allows us to treat an XML document as a simple 'database' that can be queried, iterated across, etc. under the control of JavaScript.

This is a very sophisticated facility and this exercise only scratches its surface. For more, see: http://msdn.microsoft.com/workshop/author/databind/data_binding_node_entry.asp.

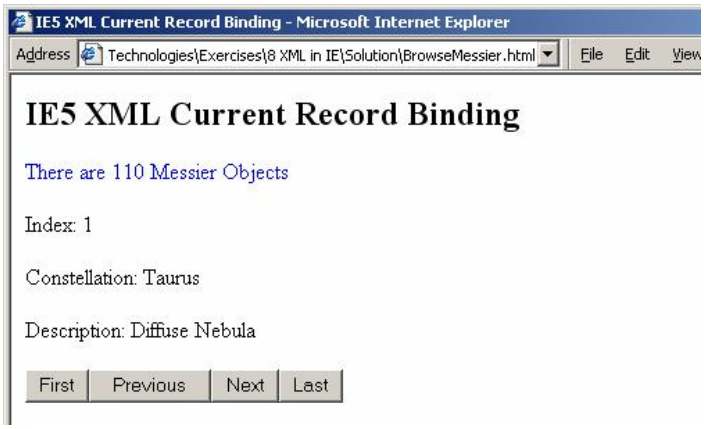
Edit the file **C:\ie\BrowseMessier.html** to become:

```
<html>
<head>
<title>IE5 XML Current Record Binding</title>
    <script for="window" event="onload">
      <!--
      var xmlDso = dsoMessierList.XMLDocument;
      countMess.innerHTML = " " +
                            xmlDso.documentElement.childNodes.length +
                            " ";
      -->
    </script>
</head>
<body>
<h2>IE5 XML Current Record Binding</h2>

<xml id="dsoMessierList" src="Messier.xml"></xml>

<p style="color:blue">
  There are <span id="countMess"></span> Messier Objects</p>
<p>Index:
  <span datasrc="#dsoMessierList" datafld="INDEX" size="5"></span><p>
<p>Constellation:
  <span datasrc="#dsoMessierList" datafld="CONSTELLATION"></span><p>
<p>Description:
  <span datasrc="#dsoMessierList" datafld="DESCRIPTION"></span><p>

<button onClick="dsoMessierList.recordset.MoveFirst()"
        title="First Record" id="cmdFirst"> First </button>
<button onClick="if (! dsoMessierList.recordset.BOF)
dsoMessierList.recordset.MovePrevious()"
        title="Previous Record"> Previous </button>
<button onClick="if (! dsoMessierList.recordset.EOF)
dsoMessierList.recordset.MoveNext()"
        title="Next Record"> Next </button>
<button onClick="dsoMessierList.recordset.MoveLast()"
        title="Last Record"> Last </button> 
</body>
</html>
```

Points of Interest

- The <xml> tag, which defines an "XML Data Island" which is associated with the external Messier.xml file. This file is loaded into memory alongside the enclosing HTML file.
- The way that the XML Data Island is referenced as a DSO
- Navigation through the DSO recordset by JavaScript event handlers
- A window event handler that navigates the XML DOM once the page has fully loaded

Once you have created this file, you can open it directly in IE5 (simply double-click on the file). You should see the following:



You should be able to use the buttons on the page to browse through the list of Messier objects.

Searching the XML Data/Using Tabular Data Binding

This section shows two useful features of IE and JavaScript.

It is possible for an HTML document to reference an external JavaScript document. Edit the supplied ***C:\ie\showAccordingTo.js*** file to become as follows:

```
function showAccordingTo(doc,dsoM0)
  {
  var form = doc.forms[0];
  var strCriteria = form.txtCriteria.value.toLowerCase();
  var objListBox = form.selListBox;
  objListBox.options.length = 0;
  var messierRS = dsoM0.recordset;
  messierRS.MoveFirst();
  while (!messierRS.E0F)
    {
    var StrConstell = ("" + messierRS("C0NSTELLATI0N")).toLowerCase();
    if (StrConstell.index0f(strCriteria) >= 0)
      objListBox.options[objListBox.options.length++].text =
        "M" + messierRS("INDEX") + " : " + messierRS("DESCRIPTI0N");
    messierRS.MoveNext();
    }
  }
```

Point of Interest

- The use of the recordset associated with the DSO to sequentially iterate through the DSO

Now edit the file **C:\ie\FindMessier.html** as follows:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<html>
  <head>
    <title>The Messier Objects</title>
    <script for="window" event="onload">
      <!--
      var xmlDso = xmldso.XMLDocument;
      countMess.innerHTML = " " +
                            xmlDso.documentElement.childNodes.length +
                            " ";
      -->
    </script>
    <script language="JavaScript" src="showAccordingTo.js">
    </script>
  </head>
  <body>
    <xml src="Messier.xml" id="xmldso"></xml>
    <h1 style="color:blue">The
      <span id="countMess"></span> Messier Objects</h1>
    Enter the name of a constellation to search for all the Messier
    Objects in that constellation:
    <form>
    <p>
       Constellation:
       <input type="TEXT" name="txtCriteria" value="Enter name...">
        
       <input type="BUTTON" value="Show"
              onClick="showAccordingTo(document,xmldso)">
     </p>
     <select name="selListBox" size="5">
       <option>Results show here...
     </select>
    </form>
    <table width="100%">
      <tr>
        <td>
          <table border="1" width=75% datasrc=#xmldso>
            <tr>
              <td bgcolor="lightYellow"
                  style="font-size:18; font-weight:bold">
                M<span datafld="INDEX"></span>
              </td>
            </tr>
            <tr>
              <td>
                <table border="1" width=100%>
                  <thead align="left" bgcolor="silver">
                    <th>Constellation</th>
                    <th>Description</th>
                  </thead>
                  <tr align="left" bgcolor="lightBlue">
```

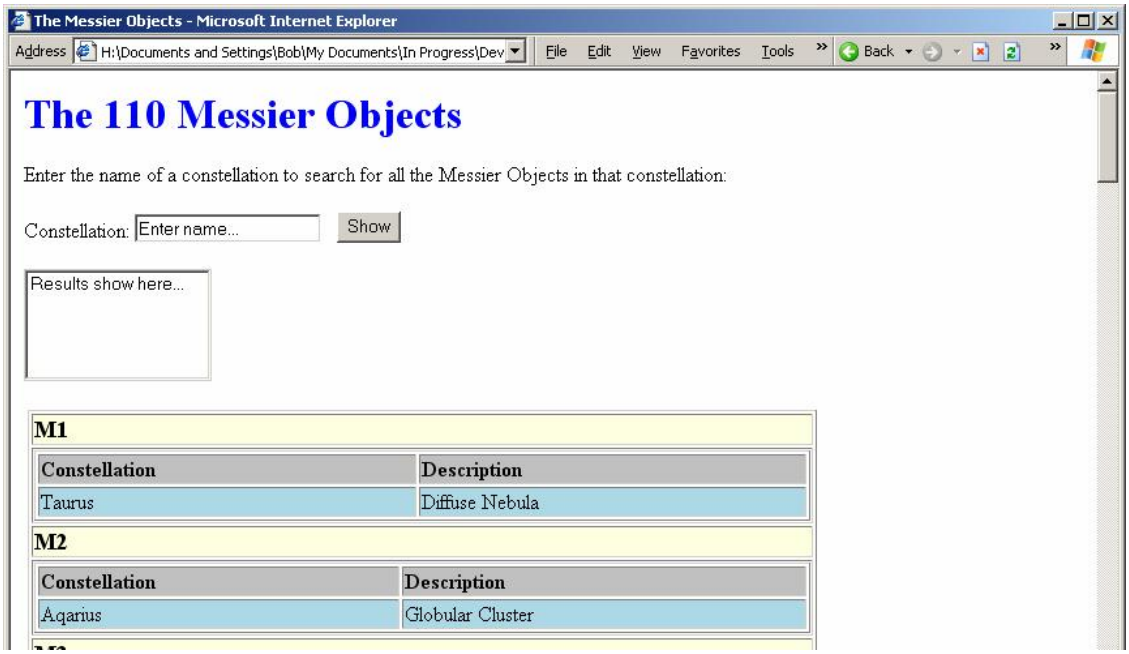
```
                    <td><div datafld="CONSTELLATION"></div></td>
                    <td><div datafld="DESCRIPTION"></div></td>
                  </tr>
                </table>
              </td>
            </tr>
          </table>
        </td>
      </tr>
    </table>
  </body>
</html>
```

Points of Interest

- The way the KISS[1] rule is applied…why program when a better (declarative) approach is possible via IE5's Data Binding facility, which allows you to associate data contained in an XML Data Island with an HTML element via the DATAFLD attribute; in this way, IE creates the table with the minimum of programmer effort.

- The ability to reference and execute external JavaScript programs

Once you have created this file, you can open it directly in IE5 (simply double-click the file). You should see the following:



[1] "Keep It Simple, Stupid"

You can enter a constellation name to see a list of all the objects that can be found in that constellation.