



The Messier Objects

Introduction

£

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.

The Exercise

In this exercise, you will learn how to create a small XML document using PERL (the Practical Extraction and Report Language), how to define and use an external XML Schema, and how to use tables in conjunction with IE5's Data Islands and Data Source Object features. You will also learn how to use JavaScript to work with the IE5/HTML Document Object Model.

You will take a simple columnar text file listing of the Messier objects and 'massage' it into a well-formatted, searchable XML-based catalogue.

Notes:

- Make sure that PERL is installed on your system before starting this exercise.
- Your Instructor will provide the file *MessierObjects.txt*.

Setting Up

Make a new directory for this exercise. Call this directory *Messier*. For example:

```
C:\> mkdir Messier
C:\> cd Messier
```

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

Creating the XML File using PERL

The source data file provided to you is a table containing rows resembling the following:

```
1 Taurus Diffuse_Nebula
```

You can see three columns: the catalogue entry number, the constellation which contains the object and the type of the object.

For this exercise, you will need to convert it to XML resembling the following:

```
<?xml version="1.0"?>
<MESSIER>
  <M INDEX="1">
    <CONSTELLATION>Taurus</CONSTELLATION>
    <DESCRIPTION>Diffuse Nebula</DESCRIPTION>
  </M>
  ...
</MESSIER>
```

To convert this to XML you will use the popular text-processing language PERL.

Use a text editor to create the following (rather terse!) PERL script in a file called *ProcessMessier.pl*:

```
use strict;

sub i
{
  my $s = " ";
  $s x=$_[0];
};
```



£

```
sub XMLBegin ()
{
    print "<?xml version=\"1.0\"?>\n";
    print "<MESSIER>\n";
};

sub XMLEnd ()
{ print "</MESSIER>\n"; };

sub MObject ()
{
    print &i (1), "<M INDEX=\"", $_[0], "\">\n";
    print &i (2), "<CONSTELLATION>";
    $_[1] =~ tr/_/ /;
    print $_[1];
    print "</CONSTELLATION>\n";
    print &i (2), "<DESCRIPTION>";
    $_[2] =~ tr/_/ /;
    $_[2] =~ s/&/&amp;/;
    print $_[2];
    print "</DESCRIPTION>\n";
    print &i (1), "</M>\n";
}

&XMLBegin ();

my $in;
while ($in = <STDIN>)
{
    chomp ($in);
    my ($index, $constell, $description) = split (' ', $in);
    &MObject ($index, $constell, $description);
}

&XMLEnd ();
```

PERL can be very obscure! To help you decipher this code, here are some hints:

- & prefixes a subroutine invocation
- \$ prefixes a variable name
- parameters are passed to a subroutine in the array called \$_[0]
- =~ performs a pattern match and modification all in one go
- tr/_/ / transforms underscore characters to spaces
- chomp removes leading and trailing spaces from a string

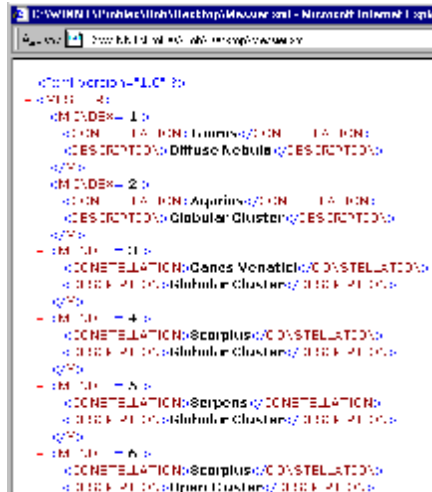
When you have worked out what this code is doing, use the following command to create an XML-formatted file:

```
C:\Messier> perl ProcessMessier.pl < MessierObjects.txt > Messier.xml
```

Once the *Messier.xml* file has been created, open it with IE5. You should see the following:



£



If IE5 complains, remove any errors from the PERL code and recreate the Messier.xml file.

Adding an External Schema

Although the file you have created is well-formed, it is still not valid. You need to add a schema to the file.

Enter this schema into a file named *MessierSchema.xml*:

```
<?xml version="1.0"?>
<Schema xmlns="urn:schemas-microsoft-com:xml-data"
  xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <AttributeType name="INDEX" dt:type="ui1"/>
  <ElementType name="CONSTELLATION" dt:type="string"/>
  <ElementType name="DESCRIPTION" dt:type="string"/>
  <ElementType name="M" content="eltOnly">
    <attribute type="INDEX"/>
    <element type="CONSTELLATION" minOccurs="0" maxOccurs="1"/>
    <element type="DESCRIPTION" minOccurs="1" maxOccurs="1"/>
  </ElementType>
  <ElementType name="MESSIER" content="eltOnly">
    <element type="M" minOccurs="1" maxOccurs="1"/>
  </ElementType>
</Schema>
```

To associate this schema with the data, you should also edit the *Messier.xml* file. Edit the opening MESSIER tag so that it becomes:

```
<MESSIER xmlns="x-schema:MessierSchema.xml">
```

Once you have made the changes, you can view the file in IE5—you should see almost the same display as before. This version is preferable since it allows a processor (not IE5, which contains a non-validating parser, unfortunately) to be a lot stricter when deciding whether or not the datatypes contained within the XML document adhere to their specifications and hence whether or not the file is valid.

Displaying in IE5

I am sure that you would agree that IE5's default display is neither pretty nor particularly useful.

To improve on this, create the following HTML file. Call it *Messier.html*:

```
<!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;
      xmlDso.load("Messier.xml");
      countMess.innerHTML = " " +
```



£

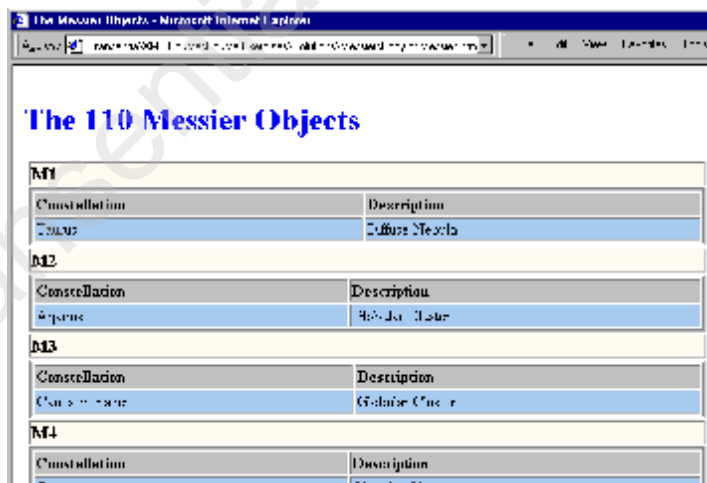
```
xmlDso.documentElement.childNodes.length +
" ";

-->
</SCRIPT>
</HEAD>
<BODY>
  <OBJECT WIDTH="0" HEIGHT="0" ID="xmlDso"
    CLASSID="clsid:550dda30-0541-11d2-9ca9-0060b0ec3d39">
  </OBJECT>
  <H1 STYLE="color:blue">The <SPAN ID="countMess"></SPAN> Messier Objects</H1>
  <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>
        </TABLE>
      </TD>
      <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"></TD>
            <TD><DIV DATAFLD="DESCRIPTION"></TD>
          </TR>
        </TABLE>
      </TD>
    </TR>
  </TABLE>
</BODY>
</HTML>
```

This file is using a number of technologies:

- tables
- IE5's ability to execute embedded JavaScript programs
- IE5's Data Source Object which allows you to associate an XML element with an HTML element via the DATAFLD attribute

Once you have created this file, you can open it directly in IE5. You should see the following:



IE5 provides another way of achieving the same effect: using XML Data Islands. Modify the start of the Messier.html file to resemble the following (note that the OBJECT tag has been replaced by the XML tag, and the script is shortened):



£

```
<!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>
  </HEAD>
  <BODY>
    <XML SRC="Messier.xml" ID="xmldso"></XML>
```

When you open this file in IE5, you should see the same effect but this is a 'cleaner' solution.

Processing with JavaScript and IE5

A nice facility to add to this page is to allow the user to search for which Messier Objects occur in a given constellation. You will modify this page to provide this facility via an external JavaScript function. You will also supply the code for that function.

Add a second SCRIPT code block to the HEAD of the file:

```
<SCRIPT LANGUAGE="JavaScript" SRC="showAccordingTo.js"></SCRIPT>
```

Modify the start of the BODY of the file to resemble the following:

```
<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">
      &nbsp;
      <INPUT TYPE="BUTTON" VALUE="Show"
        ONCLICK="showAccordingTo(document,xmldso)">
    </P>
    <SELECT NAME="selListBox" SIZE="5">
      <OPTION>Enter a Constellation name...
    </SELECT>
  </FORM>
  <TABLE WIDTH="100%">
```

Create a separate text file with the name *showAccordingTo.js*. This file should contain the following JavaScript code:

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

Once you have done all this, open the new *Messier.html* file. You will see the a page like the following:



£

The Messier Objects - Microsoft Internet Explorer

http://www.transentia.com.au/messier/objects/objects.html

The 110 Messier Objects

Enter a Messier object number or a name in the Messier Object search facility below

Constellation

☐ All Messier Objects
☐ All Open Clusters

M11	
Constellation	Description
Scorpio	Open Cluster

M12	
Constellation	Description
Sagittarius	Globular Cluster

M13	
-----	--

Give the search facility a try!