# Adding a Data Source Object to a Page

When a Web author has identified the data that she wants to display on a page, the next step is to choose the data source object (DSO) that will supply the data, and to add a reference to the DSO to that page. Microsoft Internet Explorer 4.0 and later ships with a number of data source objects, including the following:

- Tabular Data Control (TDC)
- Remote Data Service (RDS)
- XML Data Source
  - Using XML in Internet Explorer 5
  - Additional XML Resources
- MSHTML Data Source
- Creating Custom Data Source Objects

In addition to these data providers, another useful resource is the MSDN Web and Internet Samples site, which has numerous sample DSOs available for download. Check this site to learn about other data providers as they become available.

A DSO can be implemented as an Microsoft ActiveX control or as a Java applet. The object element is used to embed an ActiveX control on the page; the applet element is used to embed a Java applet on the page. In general, Web authors can copy and paste the appropriate **OBJECT** or **APPLET** declaration corresponding to the ActiveX of their choosing and modify the param tags appropriately. So that elements on the page can bind to the data source, the declaration should include the id attribute.

**Note** As of December 31, 2007, the Microsoft Java Virtual Machine has reached end of life. To use executable content specified by the applet element, a user's computer must have a Java Runtime Environment (JRE) solution installed. For more information, see Microsoft Virtual Machine Overview.

Because the data source object specification imposes no requirements on the DSO regarding the object model it exposes, Web authors should familiarize themselves with the documentation associated with the DSO they select to provide data to their page.

## Tabular Data Control (TDC)

The Tabular Data Control (TDC) is a simple DSO that provides access to delimited text files. This is the DSO used in the majority of the samples provided with this section. Consider using the TDC if:

- You have a simple data set to display. TDC supports some simple data types.
- You need to browse your data offline. The file transmitted by the TDC can be cached on the client and read offline.
- You want to prevent direct access to your Database Management System (DBMS). A server script can be written to dump data from your database into a delimited text file. In addition, most DBMSes support this feature.

The following example represents a declaration specific to the TDC.

Copy

```
<OBJECT CLASSID="clsid:333C7BC4-460F-11D0-B
C04-0080C7055A83"
    ID=dsoComposer WIDTH=0 HEIGHT=0>
    <PARAM NAME="DataURL" VALUE="composer.c
sv">
</OBJECT>
```

DataURL is a property specific to the TDC. The TDC uses this property to allow an author to specify the data set that should get loaded along with the page. For more information about the TDC, see the TDC Reference.

## Remote Data Service (RDS)

Microsoft Remote Data Service (RDS) is a more sophisticated DSO that ships with Internet Explorer 4.0 and later. Remote Data Service (RDS) obtains its data from a database using OLE DB or Open Database Connectivity (ODBC). Consider using RDS if:

- You have existing data in an OLE DB or ODBC-compliant DBMS such as Microsoft SQL Server, Microsoft Access, or Oracle.
- You want to specify the data using an Structured Query Language (SQL) statement.
- You need to provide update, insert, and delete capabilities.
- You want direct, real-time access to the data.

A declaration specific to RDS is as follows:

Copy

```
<OBJECT classid="clsid:BD96C556-65A3-11D0-9
83A-00C04FC29E33"
    ID=dsoComposer HEIGHT=0 WIDTH=0>
    <PARAM NAME="Server"  VALUE="http://mus
icserver">
    <PARAM NAME="Connect" VALUE="dsn=music;
uid=guest;pwd=">
    <PARAM NAME="SQL"     VALUE="select com
psr_name from composer">
</OBJECT>
```

First, note the class identifier (CLSID) specific to RDS. Every ActiveX component requires a CLSID to differentiate it from other objects registered on the system. Next, note that the base properties exposed by RDS differ significantly from those exposed by the TDC. That's because the TDC derives its data from a flat text file, while the RDS is capable of retrieving and updating data from any OLE DB or ODBC-compliant database. The following are the properties specified in the example declaration above.

| | |
|---|---|
| Server | String identifying the protocol and the server that supplies the data. |
| Connect | Standard ODBC connection string identifying the data source name configured on the server. |
| SQL | SQL query identifying the table and columns to be selected from the database. |

For the specifics on how to use RDS, see Microsoft Remote Data Service (RDS).

## XML Data Source

XML describes data and structured text on the Web in a standard way. Internet Explorer 4.0 shipped with a Java applet that serves as an XML data provider. While the XML DSO is a read-only data provider, Web authors should consider using it to display hierarchical data.

To use the XML data source object, add an **applet** element to your page, as in the following:

Copy

```
<APPLET
    CODE="com.ms.xml.dso.XMLDSO.class"
    ID="xmldso"
    WIDTH="0"
    HEIGHT="0"
    MAYSCRIPT="true">
    <PARAM NAME="URL" VALUE="composer.xml">
</APPLET>
```

Since the DSO is implemented in Java, embedding it on the page requires the use of the **applet** element. The className attribute specifies the package in which the code is implemented. The **param** tag specifies the location of the data. The XML DSO retrieves the XML from this location, parses it, and provides the data to bound elements on the page. The data-consuming elements are isolated from the details of XML.

### Using XML in Internet Explorer 5

Internet Explorer 5 further integrates XML support directly into the browser through the use of XML data islands. XML data islands can be used to embed data in XML format directly into an HTML page using the xml element. For example, the following XML element might be embedded directly into an HTML page that describes the accessKey property.

Copy

```
<!--[if gte IE 5]>
<XML ID="xml1">
<topic-info>
    <page-type>reference</page-type>
    <member-type>property</member-type>
    <persistent-name>ACCESSKEY</persistent-
name>
    <runtime-name readable="1" writeable="1
">accessKey</runtime-name>
    <abstract>Sets or retrieves the acceler
ator key for the object.</abstract>
</topic-info>
</XML>
<![endif]-->
```

The use of conditional comments in the previous example prevents downlevel browsers from rendering the data contained within the XML data island as text.

Data islands can also be used to refer to an external source of XML data by specifying the src attribute in association with the **xml** element. For complete details, see the article on XML Data Islands.

### Additional XML Resources

For additional information on XML, see the following resources:

- XML Core Working Group at the W3C
- W3C XML Resources

## MSHTML Data Source

In addition to using external components as data source objects, Web authors can define their data sets within an HTML document and use MSHTML itself to provide read-only data to

a page. For the purposes of this section, the page supplying the data is called the data page.

When MSHTML is used as a DSO, it parses through the data page in search of elements with an **id** attribute. The set of unique IDs defines the columns in the data set, and Web authors use these **id** attributes as the value for the dataFld attribute on elements of the bound page. For example, given the following span property from a data page, MSHTML interprets compsr_last as the name of a column and Mozart as data in that column.

Copy

```
<SPAN ID=compsr_last>Mozart</SPAN>
```

Multiple elements on the page sharing the same **id** identify additional records in the data set. A column that is not represented is given the null value in the corresponding data set. Only elements with an opening and closing tag—for example, div, span, or hn—can be used to supply data. The following example defines a two-column table consisting of three records.

Copy

```
<H1 ID=COMPSR_FIRST>Hector</H1>
<MARQUEE ID=COMPSR_LAST>Berlioz</MARQUEE>
<DIV ID=COMPSR_BIRTH>1803</DIV>
<H2 ID=COMPSR_FIRST>Modest</H2>
<H3 ID=COMPSR_LAST>Moussorgsky</H3>
<BUTTON ID=COMPSR_BIRTH>1839</BUTTON>
<TEXTAREA ID=COMPSR_FIRST>Franz</TEXTAREA>
<XMP ID=COMPSR_LAST>Liszt</XMP>
<SPAN ID=COMPSR_BIRTH>1811</SPAN>
```

Observe that the MSHTML DSO disregards the lack of consistency in the elements used. In the example above, the **H1**, **H2**, and textArea elements represent the data for the first-name column in the data set; the marquee, **H3**, and xmp elements represent the last-name column; the **div**, button, and **span** elements represent the date-of-birth column.

Once the data page is defined, use an **object** tag on the databound page to supply the data, as follows:

Copy

```
<OBJECT ID=htmlComposer DATA="compdata.htm"
 HEIGHT=0 WIDTH=0>
</OBJECT>
```

The data attribute points to the data page and can specify a complete or relative URL.

Elements on the page used to present the data can be bound using the About Data Binding Architecture.

Click the Show Me button to see how the MSHTML DSO works.

**Code example**:
http://samples.msdn.microsoft.com/workshop/samples/author/databind/dbmshtml.htm

## Creating Custom Data Source Objects

While the components described above are implementations of DSOs provided with Windows Internet Explorer and available through the DSO gallery, the specification is completely open and language independent so that independent software vendors (ISVs) can create their own data providers. For more information on creating a DSO, see the COM Objects as Data Providers document.

## Community Content

Add FAQ

### The Connect parameter typically uses Data Source instead of DSN

Vlastivosť `RDS.DataControl.Connect` odpoviada vlastivosti `ADO.Connection.ConnectString` dla dostavtzi `MS Remote`, który očekuie określenia nazvy zbioru danych pod hasłem `Data Source`. Vyvołyvana usługa `RDS.DataFactory` překazuie opis połączenia do činnika bezpiečeństva `MASDFMAP.handler`, który připořądkovuie podanemu hasłu odpoviedni tiąg ustavień, opisany v iei pliku přigotovavčim `MSDFMAP.INI`. Tylko administrator może zmieniť zavartość pliku `MSDFMAP.INI`.

History

11/20/2008
yecril