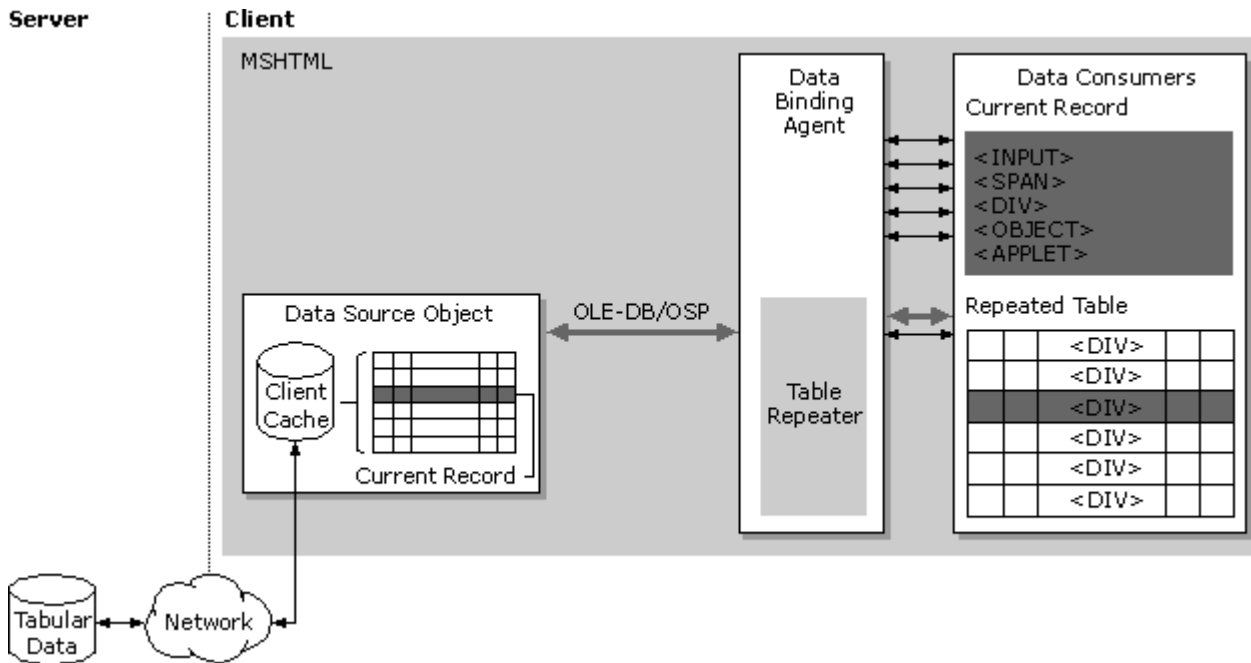


Data binding is based on a component architecture that consists of four major pieces—the data source object (DSO), data consumers, the binding agent, and the table repetition agent. Data source objects provide the data to a page, data-consuming HTML elements display the data, and the agents ensure that both the provider and the consumer are synchronized.

The following diagram illustrates how the components fit together.



This article covers the following topics.

- [Data Source Objects](#)
- [Data Consumers](#)
- [Binding Agents](#)
- [Table Repetition Agent](#)
- [HTML Extensions for Data Binding](#)
- [Specifying Data Rendering Using the DATAFORMATAS Attribute](#)
- [Optimizing Table Viewing Using the DATAPAGESIZE Attribute](#)

Data Source Objects

To bind data to the elements of an HTML page in Windows Internet Explorer, a DSO must be present on that page. DSOs implement an open specification that leaves it up to the DSO developer to determine the following:

- How the data is transmitted to the page. A DSO can use any transport protocol it chooses. This might be a standard Internet protocol, such as HTTP or simple file I/O. A DSO also determines whether the transmission occurs synchronously or asynchronously. Asynchronous transmission is preferred, because it provides the most immediate interactivity to the user.
- How the data set is specified. A DSO might require an Open Database Connectivity (ODBC) connection string and an Structured Query Language (SQL) statement, or it might accept a simple URL.
- How the data is manipulated through scripts. Since the DSO maintains the data on the client, it also manages how the data is sorted and filtered.
- Whether updates are allowed.

A DSO typically exposes this functionality through an object model that is accessible to scripts.

The only strict requirement imposed on a DSO is that it must provide access to its data through the [Microsoft OLE DB](#) or [OLE DB Simple Provider](#) APIs. Because all DSOs provide data access through a standard set of interfaces, MSHTML extends the object model of the DSO with a standard set of properties, methods, and events. It is through this object model that scripts can manipulate the data.

For specifics on adding a DSO to a page, see [Adding a Data Source Object to a Page](#). For information about implementing custom DSOs, see [Data Binding Interfaces](#).

Data Consumers

Data consumers are elements on the HTML page that are capable of rendering the data supplied by a DSO. Elements include many of those intrinsic to HTML, as well as custom objects implemented as Java applets or Microsoft ActiveX Controls. Internet Explorer supports data binding architecture to enable authors to bind an element to a specific column of data in a data set exposed by a DSO. Applets and ActiveX Controls support additional binding semantics described below.

Elements support the consumption of either single-valued or tabular data. Single-valued consumers, such as [input](#) elements, consume a single value from the current record provided by a data source. Tabular data consumers, such as the [table](#) element, repeat an entire set of records from a data provider. This is called set binding.

The [HTML Extensions for Data Binding](#) section of this article explains the extended attributes that allow Web authors to bind elements to the data supplied by a DSO. For more information about the individual elements that support data binding, see [Binding HTML Elements to Data](#).

Binding Agents

The binding and repetition agents are implemented by MSHTML.dll, the HTML viewer for Internet Explorer, and they work completely behind the scenes. When a page is first loaded, the binding agent finds the DSOs and the data consumers among those elements on the page. Once the binding agent recognizes all DSOs and data consumers, it maintains the synchronization of the data that flows between them. For example, when the DSO receives more data from its source, the binding agent transmits the new data to the consumers. Conversely, when a user updates a databound element on the page, the binding agent notifies the DSO.

To signal the Web author of important changes in the state of the data caused by providers and consumers, the binding agent fires a variety of [scriptable events](#).

Table Repetition Agent

The repetition agent works with tabular data consumers, such as the HTML **table** element, to repeat the entire data set supplied by a DSO. Individual elements in the table are synchronized through interaction with the binding agent.

HTML Extensions for Data Binding

To enable Web authors to add data binding to their pages, Microsoft Internet Explorer 4.0 and later provides support for several HTML attributes. All data binding-related HTML attributes correspond to a similarly named property in the [About the DHTML Object Model](#). This section introduces these attributes and describes their use.

The key to data binding is to connect a data source object to a data consumer. A data consumer can be any HTML element that supports the extended data-binding attributes. The [dataSrc](#) attribute specifies the DSO being used for data binding. If a DSO is used to provide a recordset containing several fields of data, which is generally the case, then the [dataFld](#) attribute is used to specify which field of data to bind to a given element. Since a data source can contain raw data, HTML data, or data that should be rendered in a specific format based on localization issues, the [dataFormatAs](#) attribute is used to specify how the data in a databound field is rendered. The [dataPageSize](#) attribute is used to specify the maximum number of records that are displayed at once in a data bound table.

Bound elements fall into two specific categories.

- Single-valued consumers
- Tabular data consumers

The binding agent takes a single value from the current record of a data source and passes it to a single-valued consumer. The repetition agent works with the binding agent to pass the entire set of records to a tabular data consumer.

By specifying the **dataSrc** and **dataFld** attributes on a single-valued data consuming element of a Web page, the Web author fully specifies the binding of an element to data. By specifying **dataFormatAs** on a single-valued element, the Web author indicates how the data should be interpreted. The **dataPageSize** element allows the author to restrict the number of records displayed by a tabular consumer. All these properties can be set at run time through the Dynamic HTML (DHTML) Object Model.

Here's a sample declaration for a single-valued, data-consuming element that might appear on an HTML page.

```
<SPAN DATASRC=#dsoComposer DATAFLD=compsr_first></SPAN>
```

The **dataSrc** attribute in the preceding example is set to the ID (dsoComposer) of a DSO embedded on the page. The **dataFld** is set to the name of a column in the data set provided by the DSO. For this binding to succeed, the data set retrieved by the DSO must contain a column named "compsr_first".

Click the Show Me button to see a working example.

Code example: <http://samples.msdn.microsoft.com/workshop/samples/author/databind/dbsingle.htm>

The following example shows how a table, a tabular data consumer, can be bound to a DSO. The repetition agent uses the table row (**tr**) in the table body as a template. As the DSO, identified in the preceding example as dsoComposer, supplies each record in the data set, the template including the **div** is repeated, until all composers' first names are displayed.

```
<TABLE DATASRC=#dsoComposer>
<TR>
<TD><DIV DATAFLD=compsr_first></DIV></TD>
</TR>
</TABLE>
```

Click the Show Me button to see a working example that uses the **TDC** to display a table of composers, including each one's first and last names, dates of birth and death, and country/region of origin.

Code example: <http://samples.msdn.microsoft.com/workshop/samples/author/databind/dbtable.htm>

Specifying Data Rendering Using the DATAFORMATAS Attribute

The data supplied by a DSO can be in a variety of formats, and the page author can use the **dataFormatAs** attribute on single-valued consumer elements that support the attribute to indicate how the data should be rendered. In Internet Explorer, HTML and TEXT are the valid values for this attribute, as shown in the following example.

```
<MARQUEE DATASRC=#dsoAdvertisement DATAFLD=banner
DATAFORMATAS=html>
</MARQUEE>
```

The preceding example describes a **marquee** that calls on a DSO that supplies various banners for display in a page.

Click the Show Me button to see how to use a timer to randomly select and display an advertisement supplied by a data source.

Code example: <http://samples.msdn.microsoft.com/workshop/samples/author/databind/dbadvert.htm>

Optimizing Table Viewing Using the DATAPAGESIZE Attribute

When binding a **table** to a data source, all the records in the data set are displayed by default. If the data set is large, the page might grow beyond what is visible to the user. In addition, any content positioned below the table is pushed off the screen. The **dataPageSize** attribute can be applied to a **table** to specify the maximum number of records to display at any one time. To enable the user to move to the next and previous pages of records viewed in the table, Web authors can use the [nextPage](#) and [previousPage](#) methods exposed by the **table** object in the DHTML Object Model. Authors can also change the page size of the table at run time by setting the **dataPageSize** property.

Click the Show Me button to see a working example.

Code example: <http://samples.msdn.microsoft.com/workshop/samples/author/databind/dbpage.htm>

In addition to limiting table size, Web authors should investigate the specific functionality exposed by the DSO they have chosen for use on their HTML page. Both the **TDC** and [Microsoft Remote Data Service \(RDS\)](#) provide filtering capabilities.