

Web Database Access *UPDATE* for Microsoft Internet Server Beta

Thank you for beta testing the Microsoft® Internet Server for the Windows NT™ Server version 3.51 operating system. Before installing this update to the beta 1 release, please read these release notes completely. They contain important information for ensuring proper installation and usage of Microsoft Internet Server with this update. This document should be used in addition to, and not in place of, the documentation and software you received with the Microsoft Internet Server for Windows NT Server.

This release contains an add-on for the beta software of the Internet Server for Windows NT that you received near the end of August. It contains the Open Data Base Connectivity (ODBC) components to enable access to a Microsoft® SQL Server database from the Web Server service of the Microsoft Internet Server. Although this release contains only the ODBC driver for Microsoft SQL Server, you can use other ODBC drivers in a similar manner by installing the driver and configuring it for use by the Web Server.

With the Web Server and ODBC, you can:

- Create Web pages with information contained in a database
- Insert, update, and delete information in the database based on user input from a Web page
- Perform other SQL commands.

1 Overview of the database access from the Internet Server

Conceptually, database access is performed by the Internet Server as shown in Figure 1:

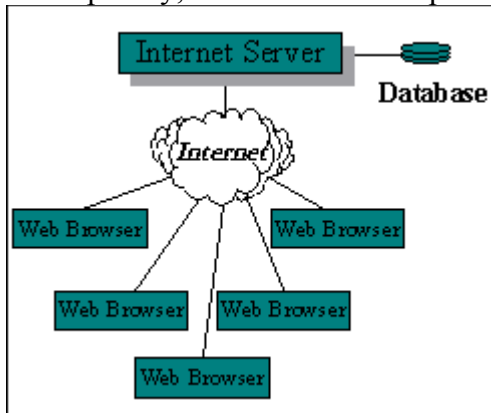


Figure 1: Conceptual view of database access from the World Wide Web

Web browsers (such as the one included in the Internet Server beta 1, or browsers from other companies such as Netscape) submit requests to the Internet server by using HyperText Transport Protocol (HTTP). The Internet Server responds with a document formatted in HyperText Markup Language (HTML). Access to databases is accomplished through a component of the Internet Server called *Internet Database Connector*, one of a family of *Internet Server Extensions*. The Database Connector is a Dynamic-Link Library (DLL) file, *Httpodbc.dll*, that uses ODBC to access databases.

Figure 2 shows the components for accessing databases from the Internet Server:

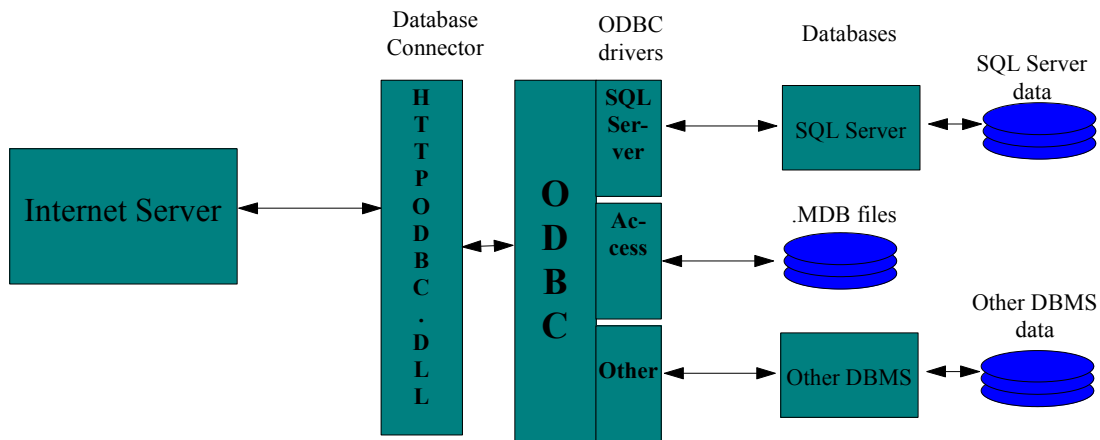


Figure 2: Database access components on the Internet Server

Httpodbc.dll uses two types of files to control how the database is accessed and how the output Web page is constructed. These files are Web Database Gateway (WDG) files and HTML extension (HTX) files.

The WDG files contain the necessary information to connect to the desired ODBC data source and execute the SQL statement. A WDG file also contains the name and location of the HTX file.

The HTX file is the template for the actual HTML document that will be returned to the Web browser after the database information has been merged into it by *Httpodbc.dll*.

The following example shows how a Web browser requests information from the Database Connector and how that information is returned.

2 An example database query

This example starts with a simple Web page called *Sample.htm*. The sample Web page will contain one hyperlink that will result in a query being executed, with the results returned as another Web page. (*Sample.htm* is located in the root directory of the

enclosed compact disc; installation instructions are at the end of this document.) Figure 3 shows Sample.htm as it is displayed by Microsoft Internet Explorer (assuming that the Internet Server has been installed on a computer called "webserver"):

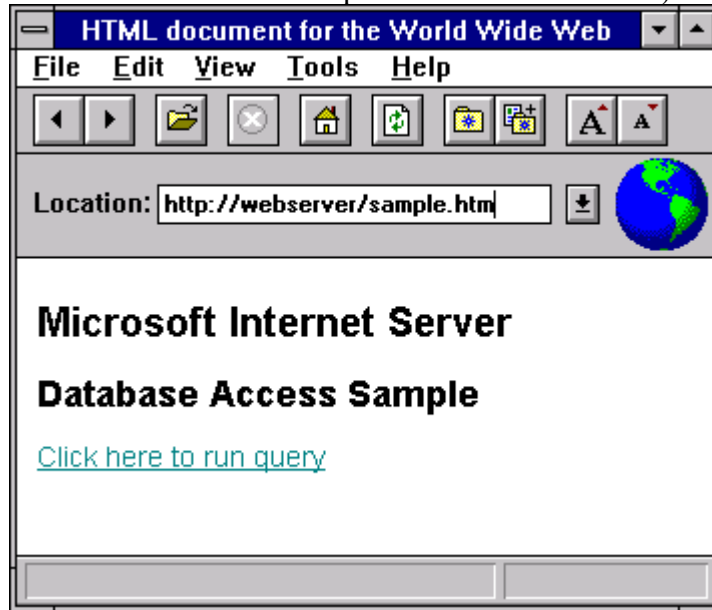


Figure 3: Sample.htm as seen in Internet Explorer

When the hyperlink *Click here to run query* is clicked, another Uniform Resource Locator (URL) is sent to the server. The URL precedes the hyperlink text (it is formatted as hidden text):

```
<A  
HREF="http://webserver/scripts/httpodbc.dll/scripts/sample.w  
dg?">Click here to run query</A>
```

As you can see, the Database Connector *Httpodbc.dll* is referenced in the URL, as is the WDG file for *Httpodbc.dll* to use (*Sample.wdg*). Both are located in a subdirectory called *\Scripts*, which is a subdirectory of the Web root directory you selected when you installed the Internet Server beta. (The default root directory for Web services is *C:\Wwwroot*. **Note:** Although it appears to be one long path name after *http://webserver*, in fact there are two distinct path names in the URL:

- /scripts/httpodbc.dll* specifies the path of the program for the Internet Server to run
- /scripts/sample.wdg* specifies the path of the file to be passed to *Httpodbc.dll*

On the Internet Server, the entire process of using the Database Connector is performed in six steps, as shown in Figure 4:

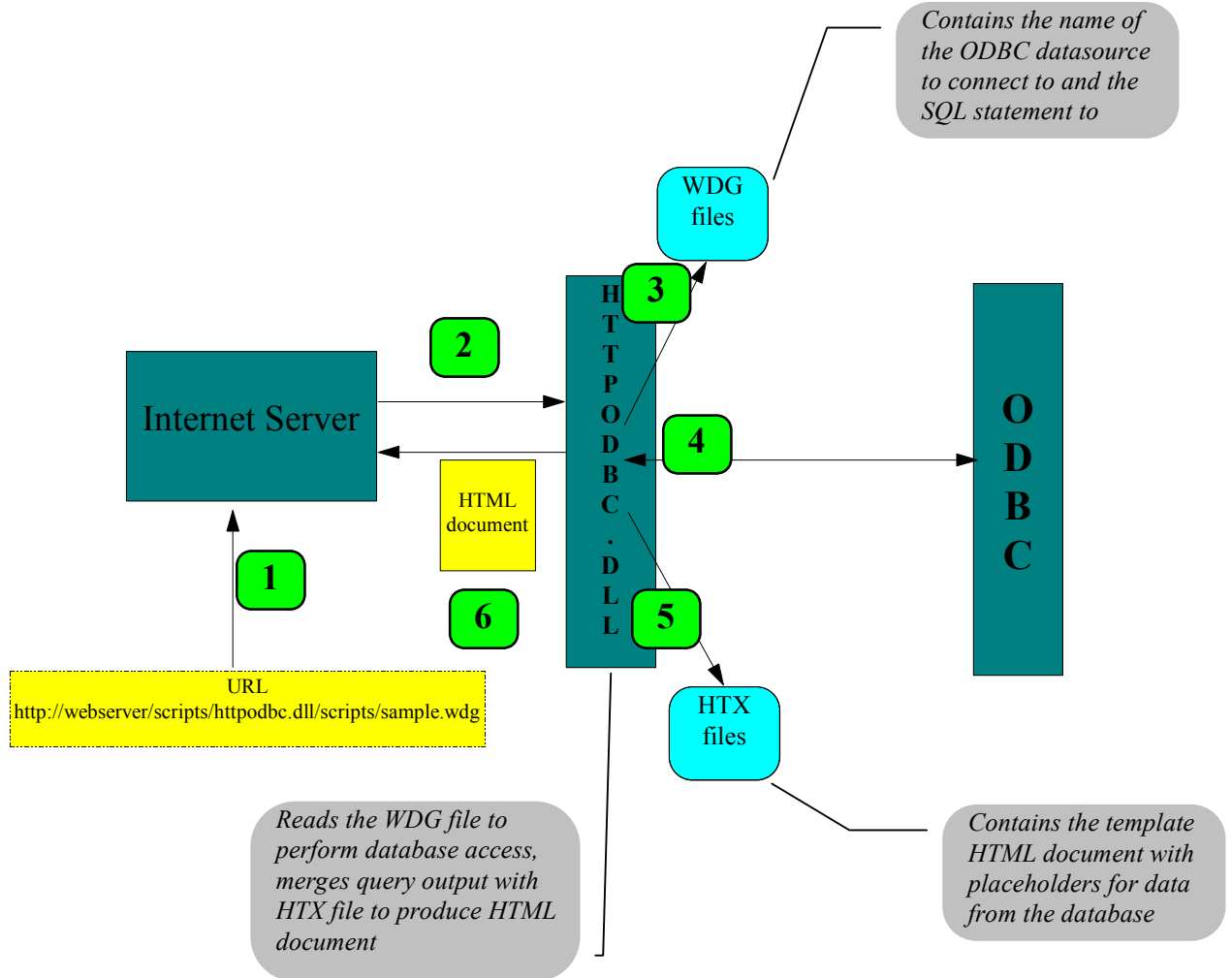


Figure 4: The six steps used to access a database in the Internet Server

Step 1: The URL is received by the Internet Server

The URL sent by the Web browser actually contains the name of the Database Connector, Httpodbc.dll, and the name of the WDG file to be used ("Sample.wdg" is shown here, but any name can be used). Step 1 is shown in Figure 5:

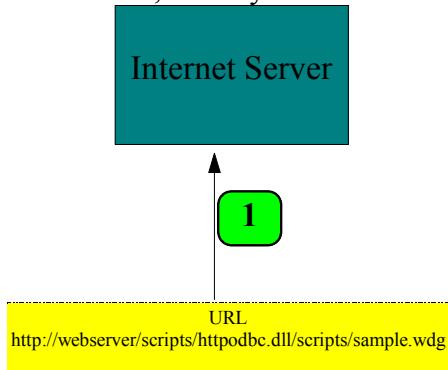


Figure 5: Step 1, the URL is received by the Internet Server

Step 2: The Internet Server loads Httpodbc.dll and provides it with the remaining information in the URL

Httpodbc.dll obtains the name of the WDG file (and other items) from the URL passed to the Internet Server, shown in Figure 6:

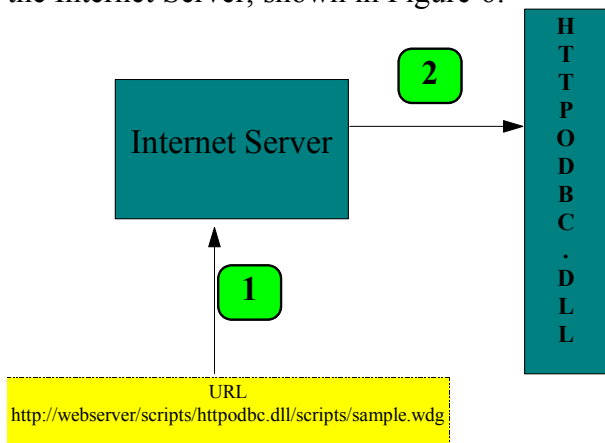


Figure 6: Steps 1 and 2, Internet Server calls Httpodbc.dll

Step 3: Httpodbc.dll reads the WDG file

The WDG File contains several entries in the format

field: value

In the Sample.wdg file, the ODBC data source is specified by:

```
Datasource: Web SQL
```

And the HTX file is specified by:

```
Template: sample.htx
```

Here is the file Sample.wdg referenced in the URL above:

```
Datasource: Web SQL
Username: sa
Template: sample.htx
SQLStatement:
+SELECT au_lname, ytd_sales
+ from pubs.dbo.titleview
+ where ytd_sales>5000
```

In the Sample.wdg file (also provided in the root directory of your compact disc), the data source name is Web SQL. The ODBC installation notes in the next section step through creating a data source called Web SQL that refers to a SQL Server.

The other items contained in Sample.wdg include:

- The Username, which must be a valid login to SQL Server
- Template, which specifies the file to use to merge the results
- SQLStatement, which contains the SQL statement to execute.

See the **Features of the Database Connector** section of this document for definitions for all the fields that can be specified in WDG files.

The SQLStatement in Sample.wdg returns all the author last names and year-to-date sales amounts from the “pubs” sample database in SQL Server for authors whose books have year-to-date sales of more than 5000 copies.

Figure 7 shows the process through Step 3.

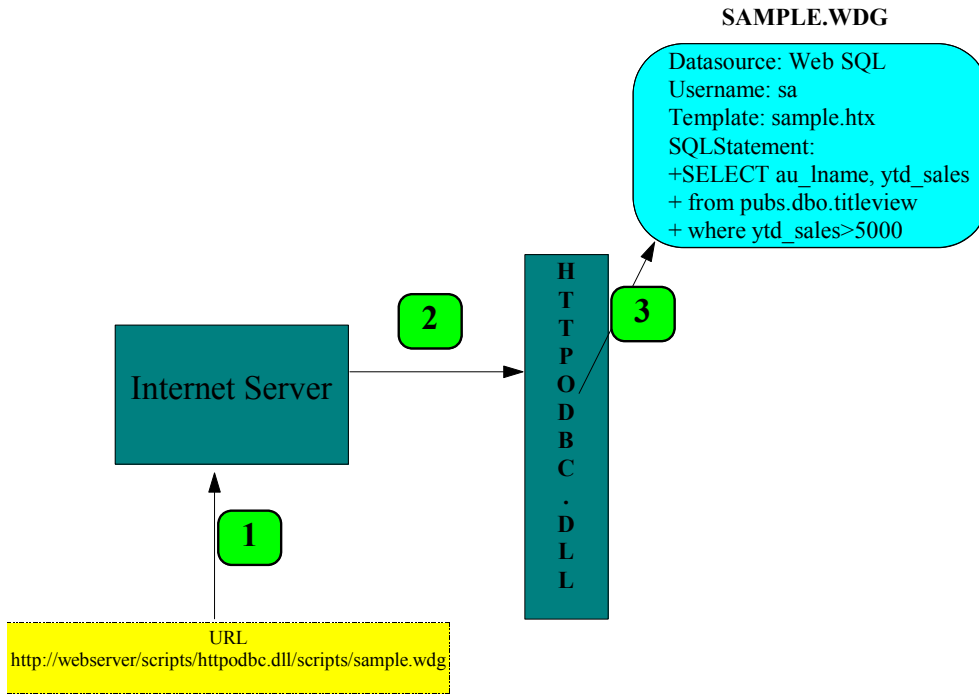


Figure 7: Steps 1 through 3, Httpodbc.dll reads Sample.wdg

Step 4: Httpodbc.dll connects to the ODBC data source, and executes the SQL statement contained within the WDG file

The connection is made to the ODBC data source by Httpodbc.dll, which in this example loads the ODBC driver for SQL Server and connects to the server specified in the definition of the data source. Once the connection is made, the SQLStatement in the WDG file is sent to the SQL Server ODBC driver, which in turn sends it to SQL Server. The interaction with ODBC is shown in Figure 8:

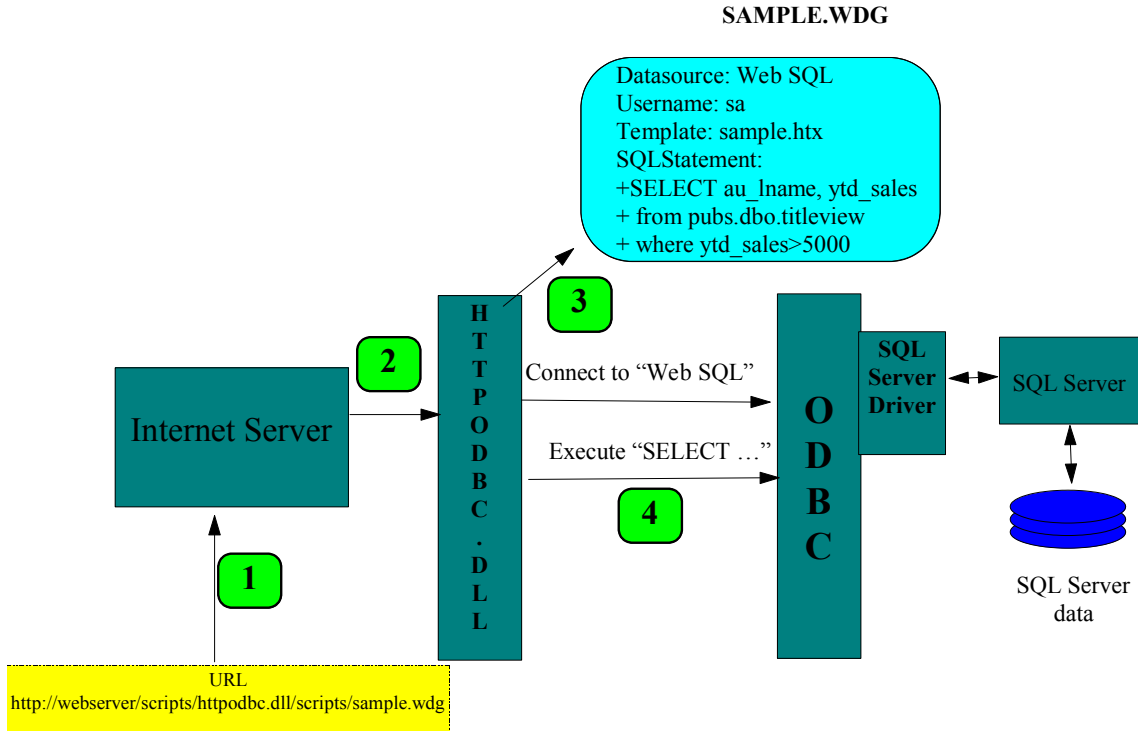


Figure 8: Steps 1 through 4, Connecting to the ODBC data source and executing the query

Step 5: Httpodbc.dll fetches the data from the database, and merges it into the HTX file

After the SQL statement has been executed, Httpodbc.dll reads the HTX file specified in Sample.wdg (Sample.htx) and replaces every placeholder with the actual data returned by the query.

The Sample.htx file is an HTML document than contains placeholders for data returned from the database (the placeholders are shown in bold for clarity):

```
<HTML>
<HEAD>
<TITLE>HTML document for the World Wide Web</TITLE>
<META NAME="GENERATOR" CONTENT="Internet Assistant for Word 2.0 Beta">
</HEAD>
<BODY>
<CENTER>
<H1>
Microsoft Internet Server
</H1>
<H2>Database Access Sample </H2>
</CENTER>
Here are the results:
<P>
<PRE>
<B>Author : YTD Sales</B>
<%begindetail%>
<%au_lname%>: <%ytd_sales%>
<%enddetail%>
</PRE>
<P>
The web page you see here was created by merging the results of the query specified in
Sample.wdg with the template file Sample.htx.
<P>
```

The merge was done by the Database Connector, Httpodbc.dll and the results were returned to this Web browser by the Microsoft Internet Server.

```
<P>  
<P>  
</BODY>  
</HTML>
```

The `<%begindetail%>` and `<%enddetail%>` sections delimit where rows returned from the database will appear in the document. Columns returned from the query are surrounded by `<%%>`, such as `<%au_lname%>` and `<%ytd_sales%>` in the preceding example.

Figure 9 shows the process up to the data retrieval from the data source and merge with the Sample.htx file:

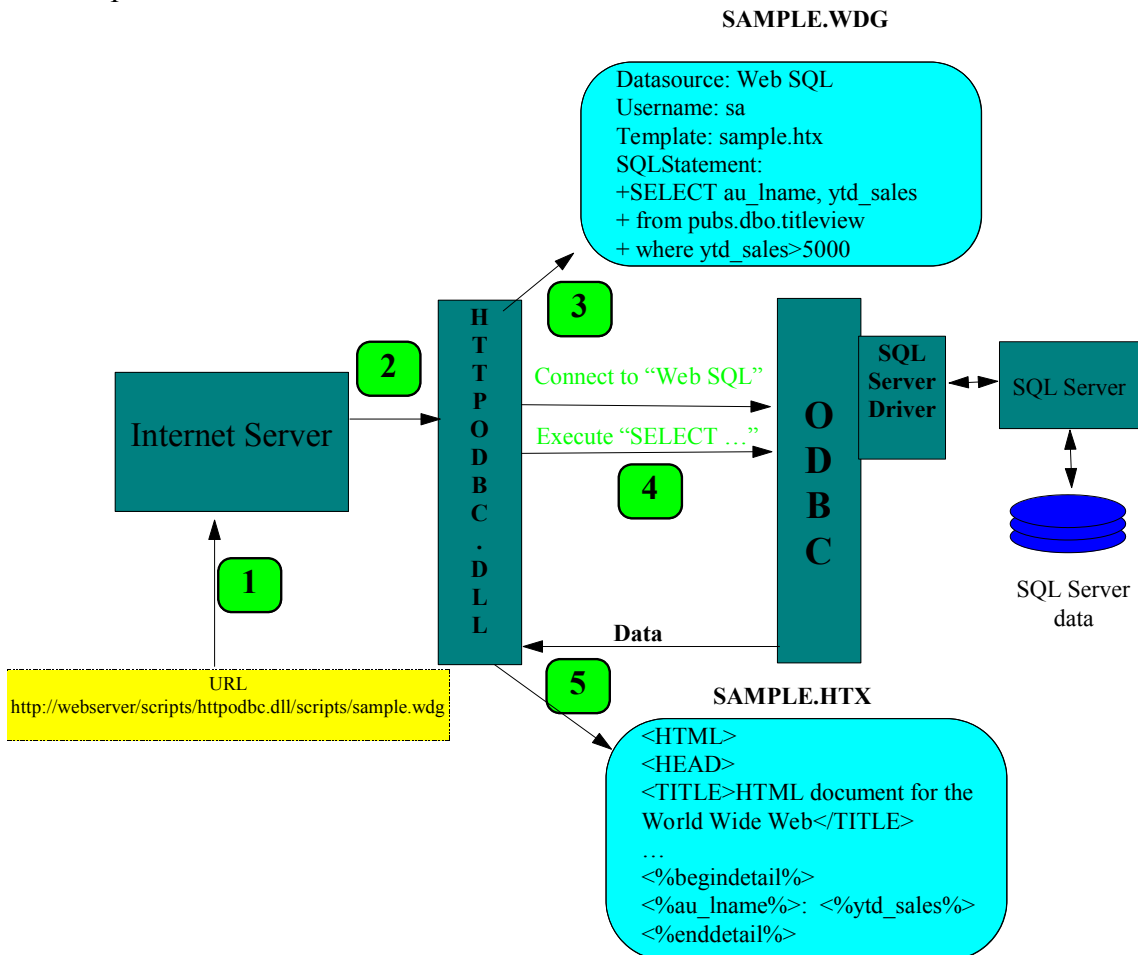


Figure 9: Steps 1-5, data retrieval and merge with Sample.htx file

Step 6: Httpodbc.dll sends the merged document back to the Internet Server, which returns it to the client.

After all the data has been merged into the placeholders in Sample.htx, the complete document is sent back to the client. Figure 10 shows the process through step 6, returning the merged document back to the client Web browser (Step 1 has been omitted from the diagram for clarity):

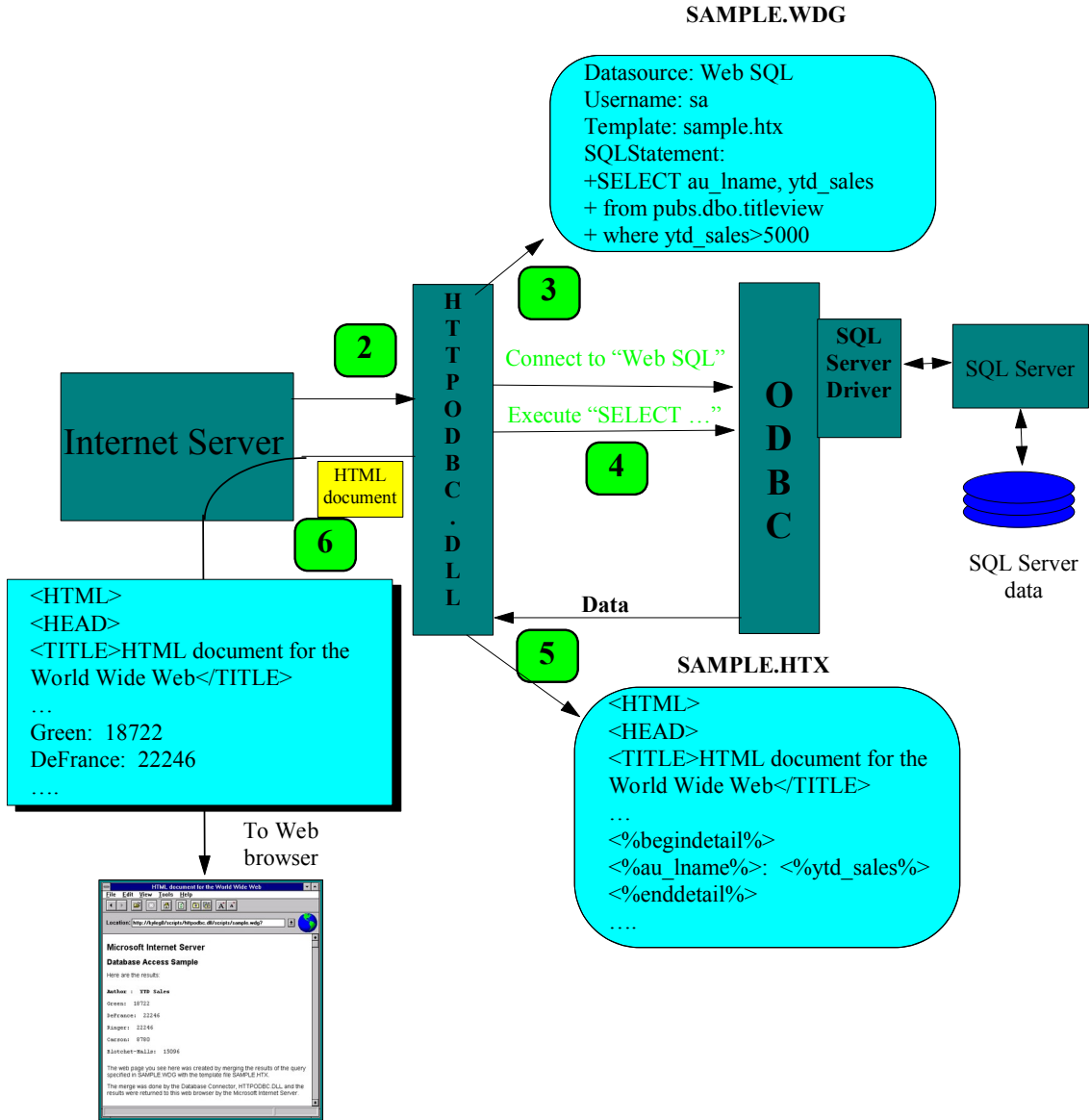


Figure 10: Returning HTML document to the client Web browser

And the resulting Web page is displayed in the Microsoft NT Internet Explorer as shown in Figure 11:

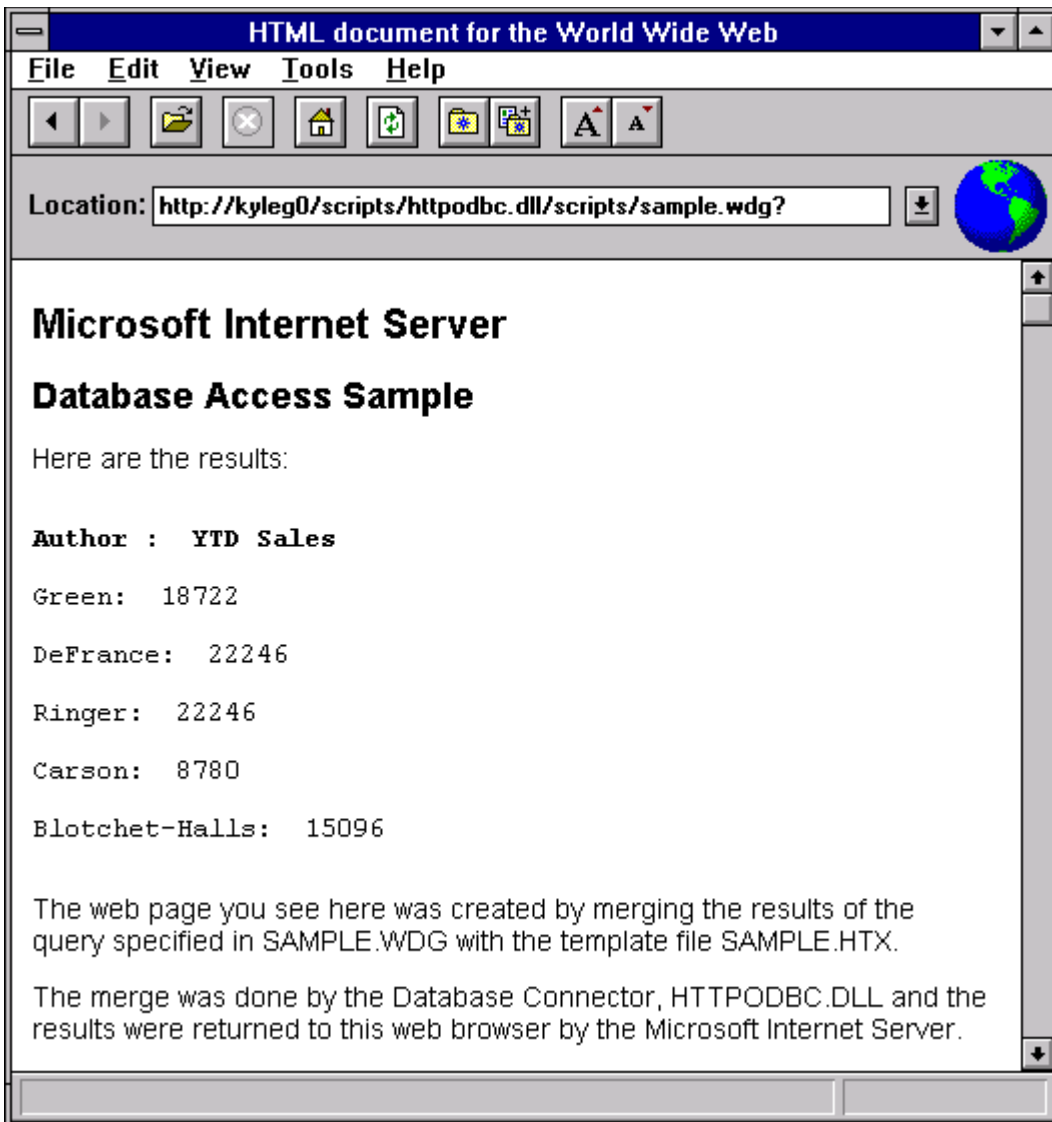


Figure 11: Output Web page containing the results of the query

3 Features of the Database Connector

The Database Connector has several features to provide additional functionality for creating Web pages that contain data from a database.

3.1 WDG files

As seen the examples above, WDG files contain the information used to access the database. The following section describes the features of WDG files.

3.1.1 Parameters

The example in the preceding section shows only the simplest kind of query, a query which was defined completely in the WDG file on the Internet server. Although this type of query is useful, even more powerful Web pages can be constructed through the use of

parameters. Parameters the names and values of HTML form controls such as <INPUT ...> and names specified directly in URLs. These are sent by Web browsers and can be used in SQL statements on the server.

For example, in the last section the query in Sample.wdg returned only the authors whose year-to-date sales exceeded 5000. By using a parameter, we could build a Web page that asks the user to decide what number to use instead of 5000.

The Web page needs to prompt the user for the year-to-date sales figure and then name the associated variable to *sales*. Sample2.htm (included on the compact disc) shows a form with an input field used to obtain the number:

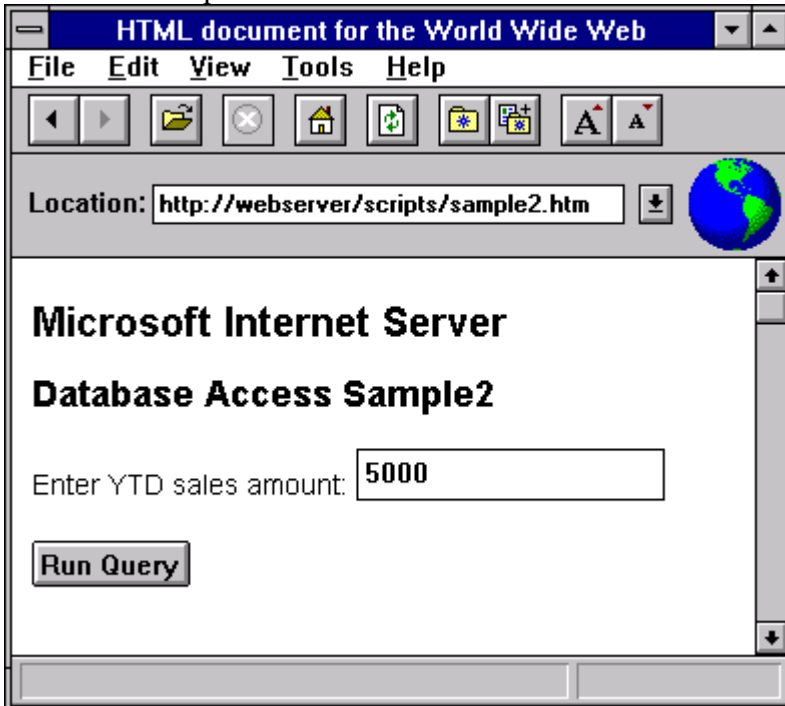


Figure 12: Sample2.htm

The HTML syntax for the input field and button in Sample2.htm is:

```
<FORM METHOD="POST" ACTION="/scripts/httpodbc.dll/scripts/sample2.wdg">  
<P>  
Enter YTD sales amount: <INPUT NAME="sales" VALUE="5000" >  
<P>  
<INPUT TYPE="SUBMIT" VALUE="Run Query">  
</FORM>
```

In the WDG file, you use the parameter in place of the number 5000:

```
SQLStatement:  
+SELECT au_lname, ytd_sales  
+ from pubs.dbo.titleview  
+ where ytd_sales > %sales%
```

Here the parameter name must be *sales* so that it corresponds to the <INPUT NAME="sales" ...> on the Web page. Parameters must be enclosed with percent characters (%) to distinguish them from a normal identifier in SQL.

You can build powerful collections of Web pages by using the output of one query to provide links to other queries. For example, to show the titles for an individual author, instead of returning the author name as plain text, you can embed it with a link, and then use the link to do another query.

The final example included on the compact disc shows how to do this type of linkage. Sample3.htm is used to run the query in Sample3.wdg, which uses Sample3.htx for the output template. Sample3.htx will return author last names as links, which, when clicked on, will display the titles for each author using Sample3a.wdg and Sample3a.htx.

3.1.2 Fields in WDG files

The following table lists the fields that can be specified in a WDG file. Field names in bold are required.

Datasource	The name that corresponds to the data source name you created earlier by using the ODBC Administrator.
Username	A valid user name with access to the SQL server. (Note: If the SQL server is running integrated security, the username will be ignored.)
Password	The password that corresponds to the user name. If the password is null, this field can be left out. (Note: If the SQL server is running integrated security, the password will be ignored.)
MaxFieldSize	The maximum buffer space allocated by Httpodbc.dll per record. Any characters beyond this will be truncated.
MaxRecords	The maximum number of records that httpodbc.dll will return from any one query.
Template	The name of the .HTX file that will be used to format the data when the data from this query is returned.
RequiredParameters	The parameter names, if any, that httpodbc.dll will ensure are passed from the client or it will return an error.
DefaultParameters	The parameter values, if any, that will be used in the WDG file if a parameter is not specified by the client.
Expires	The number of seconds to wait before refreshing a cached output page. By default, Httpodbc.dll caches the Web page sent to the client. If a subsequent request is identical, the cached page will be returned without ever accessing the database. The Expires field is useful when you want to force a requery of the database after a certain period of time.
SQLStatement	The SQL statement to execute. The SQL statement can contain parameter values from the client, which must be enclosed with percent characters. The SQLStatement can occupy multiple lines in the WDG file. Following the SQLStatement field, each subsequent line beginning with a plus sign (+) will be considered part of the SQLStatement field.

3.2 HTX Files

HTX files contain a number of keywords that control how the output HTML document is constructed. These keywords are explained below.

3.2.1 <%begindetail%>, <%enddetail%>

The <%begindetail%>, <%enddetail%> keywords surround a section of the HTX document in which the data output from the database will be merged. Within the section, the column names delimited with <% and %> are used to mark the position of the returned data from the query. For example,

```
<%begindetail%>
<%au_lname%>: <%ytd_sales%>
<%enddetail%>
```

will list the columns au_lname and ytd_sales. Any column can be referred to in this way. Column names can also be referred to elsewhere in the HTX. Note: if there are no records returned from the query, the <%begindetail%> section will be skipped.

3.2.2 <%if%>..<%else%>..<%endif%>

HTX files can contain conditional logic with an if-then-else statement to control how the Web page is constructed. For example, one common usage is to insert a condition to display the column names from the query on the first row within a <%begindetail%> section, but if there are no records returned by the query display the text “No records found.” By using the <%if%> statement and a built-in variable called “CurrentRecord” you can tailor the output so that the headers are not printed when no records are returned, as they are in the example shown in the last section. Here is an example showing the use of the <%if%> statement.

```
<%begindetail%>
<%if CurrentRecord EQ 0 %>
<B>Author : YTD Sales</B>
<%endif%>
<A HREF=" /scripts/httpodbc.dll/scripts/sample3a.wdg?lname=<
%au_lname%>"><a u_lname%></A>: <%ytd_sales%>
<%enddetail%>
<%if CurrentRecord EQ 0 %>
No records selected!
<%endif%>
```

The general syntax is:

```
<%if condition %>
HTML text
[<%else%>
HTML text]
<%endif%>
```

Where *condition* is of the form:

value1 operator value2

and *operator* can be one of the following:

EQ	if <i>value1</i> equals <i>value2</i>
LT	if <i>value1</i> is less than <i>value2</i>
GT	if <i>value1</i> is greater than <i>value2</i>
CONTAINS	if any part of <i>value1</i> contains the string <i>value2</i>

The operands *value1* and *value2* can be column names, one of the built-in variables (CurrentRecord or MaxRecords, see below), an HTTP variable name (see below), or a constant. When used in an `<%if%>` statement, values are not delimited with `<%` and `%>`. For example, to do special processing on author name “Green”, use the condition:

```
<%begindetail%>  
<%if au_lname EQ "Green"%>  
this guy is green!  
<%endif%>  
<%enddetail%>
```

The `<%if%>` statement can also be used to do special processing based on information from HTTP variables. For example, to format a page differently based on the type of client Web browser you could include the following in the HTX file.

```
<%if HTTP_USER_AGENT contains "Mozilla"%>  
client is Netscape  
<%else%>  
client is <%HTTP_USER_AGENT%>  
<%endif%>
```

3.2.3 CurrentRecord, MaxRecords

The CurrentRecord built-in variable contains the number of times the `<%begindetail%>` section has been processed. The first time through the `<%begindetail%>` section, the value is zero. Subsequently, the value of CurrentRecord changes every time another record is fetched from the database.

The MaxRecords built-in variable contains the value of the MaxRecords field in the WDG file.

3.2.4 Parameters from WDG files

Parameters from WDG files may be accessed in the HTX file by prefixing the name of the parameter with “wdg” and a period. In Sample3.htx shown earlier, you could show the value of the parameter %sales% by including the line:

```
The value of the sales parameter is: <%wdg.sales%>
```

3.2.5 Client HTTP variables

Several variables can be used in HTX files which give a lot of information about the environment and Web client connected to the server. The table below gives a partial listing of the information available.

HTTP Variable	Meaning
GATEWAY_INTERFACE	Type of interface available from the server, for example, "CGI/1.1"
HTTP_ACCEPT	List of file types acceptable. Example: "*/*, q=0.300, audio/x-aiff, audio/basic, image/jpeg, image/gif, text/plain, text/html"
HTTP_USER_AGENT	The name of the Web browser that submitted the request. For example, The Windows NT Internet Explorer returns "Microsoft Internet Explorer/0.1 (Win32)"
QUERY_STRING	The name and values of all variables specified in the Web page. For example, for the items <INPUT name=prompt1 value="hello"> <INPUT name=prompt2 value="there"> The QUERY_STRING would be prompt1=hello&prompt2=there
REMOTE_ADDR	The IP address of the client
SERVER_NAME	The server name on which the Internet Server is running
SERVER_PROTOCOL	The version number of HTTP on the server, for example, "HTTP/1.0"
SERVER_PORT	The port number of the server, for example: 80
SERVER_SOFTWARE	The server software, for example "HTTP-Srv-Beta1/1.1"

All environment variables can be accessed. There is also a special HTTP variable called ALL_HTTP that can be used to dump all the HTTP-specific information. For example, specifying

```
<%ALL_HTTP%>
```

in an HTX file used with the Microsoft Internet Explorer for Windows NT will result in the an output Web page similar to the following:

```
HTTP_ACCEPT: */*, q=0.300, audio/x-aiff, audio/basic, image/jpeg,  
image/gif, text/plain, text/html  
HTTP_USER_AGENT: Microsoft Internet Explorer/0.1 (Win32)  
HTTP_REFERER: http://webserver/scripts/sample3.htm  
HTTP_CONTENT_TYPE: application/x-www-form-urlencoded  
HTTP_CONTENT_LENGTH: 10  
HTTP_EXTENSION: Security/Digest
```

The complete list of HTTP variables will be provided with the final documentation for the Internet Server.

4 Installing ODBC and the SQL Server ODBC driver

1. Run Setup.exe from the directory on the enclosed compact disc named for your platform. For example, for Intel platforms, you can run Setup from the Program Manager by choosing File, Run, and entering the information shown in Figure 13 (assuming your CD-ROM drive is Drive D):



Figure 13: Running ODBC Setup

2. The ODBC Setup dialog box appears as shown in Figure 14:



Figure 14: The ODBC Setup dialog box

Click the Continue button. The next dialog box will present the ODBC driver list. In this case, only the SQL Server driver is available. Click the SQL Server entry in the list box as shown in Figure 15:

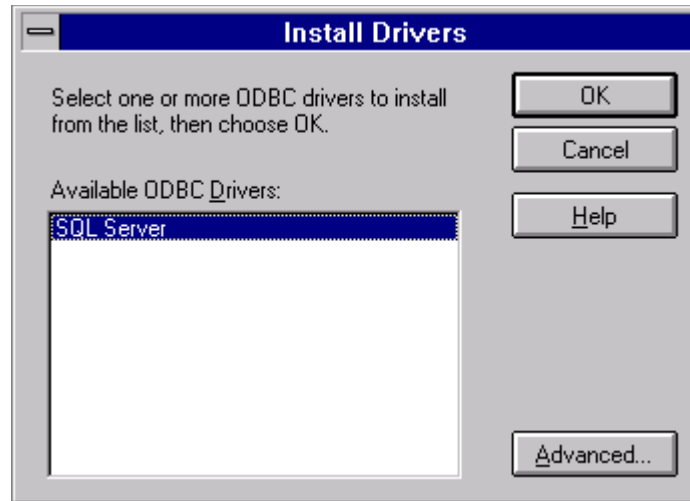


Figure 15: Select SQL Server driver

Click OK. The necessary files for ODBC and the SQL Server driver are copied from the CD to your Windows NT system. If you have an application running that uses ODBC, you may see an error message telling you that one or more components are in use. If this is the case, close all applications and services that use ODBC before continuing. Next, the ODBC Data Sources dialog box is displayed, as shown in Figure 16:

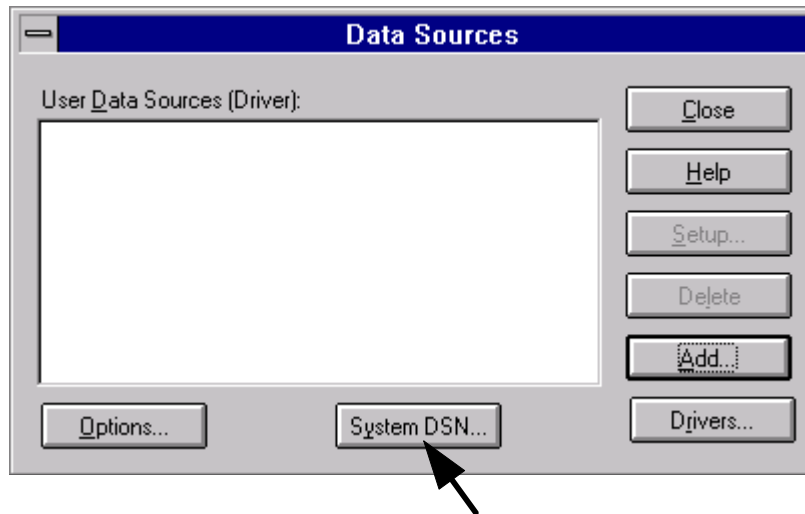


Figure 16: ODBC Data Sources dialog box

You may see other data sources in the list if you previously installed other ODBC drivers. **Important: Be sure to click the System DSN button** indicated in Figure 16 because the Database Connector will work only with System DSNs (Data Source Names). System DSNs were introduced in ODBC version 2.5 to allow Windows NT services to use ODBC. The System Data Sources dialog box is shown in Figure 17:

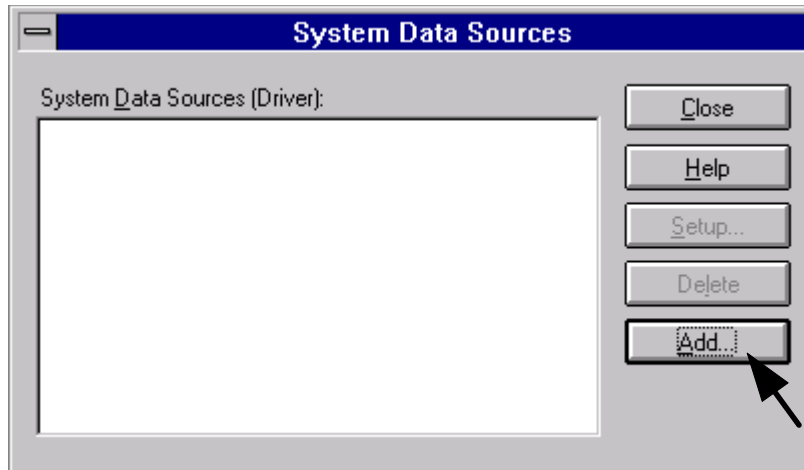


Figure 17: System Data Sources dialog box

Click the Add button. The Add Data Source dialog box appears, as shown in Figure 18:

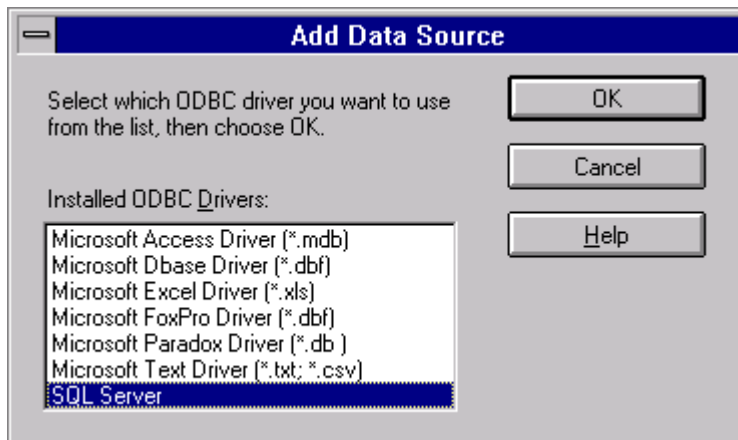


Figure 18: Add Data Source dialog box

You may have a different list of ODBC drivers than that shown in Figure 18. If you have never installed ODBC components on your computer before, only the SQL Server driver will appear in the list box.

Select SQL Server from the list box and click OK. The ODBC SQL Server Setup dialog box will appear, as shown in Figure 19:

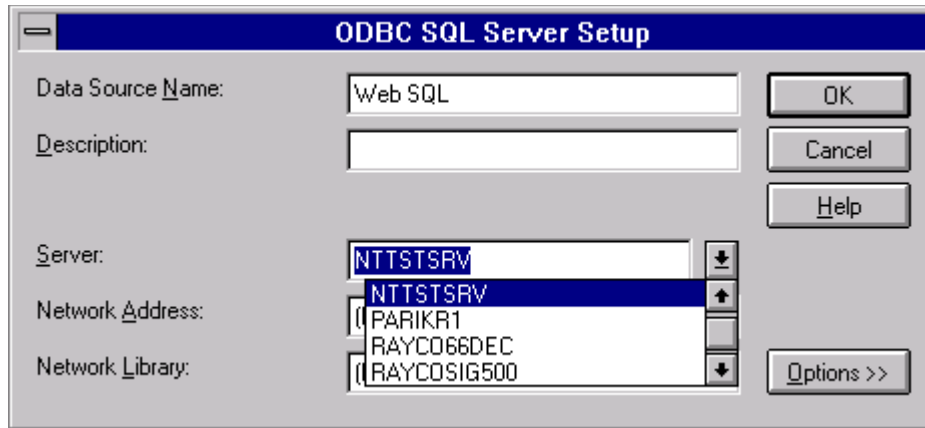


Figure 19: SQL Server Setup dialog box

Enter the name of the data source. The data source name is a logical name used by ODBC to refer to the SQL Server driver and the actual server name on which SQL Server is running. You can also use the server name “(local)” if SQL Server is running on this computer. The data source name is used in WDG files to tell the Web Server where to access the data.

In Figure 19, the data source name “Web SQL” was entered and the SQL Server “NTTSTSRV” is selected. “Web SQL” was used in a earlier examples, so you should use that name for now. Later, you can change it if you like. The server name, network address, and network library are specific to your installation. If you do not know what to enter in these controls accept the defaults. To find out the details, click the Help button and find the section that describes your network.

After the data source name and server name have been entered, click OK. The System Data Sources dialog box will be displayed again, but now will have the name of the data source displayed as shown in Figure 20:

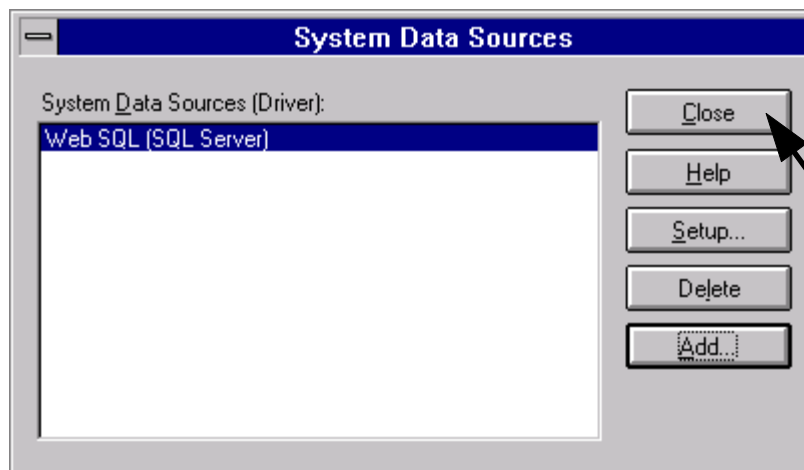


Figure 20: System Data Sources dialog box with new data source displayed

Click Close. The Data Sources dialog box is displayed again as shown in Figure 21:

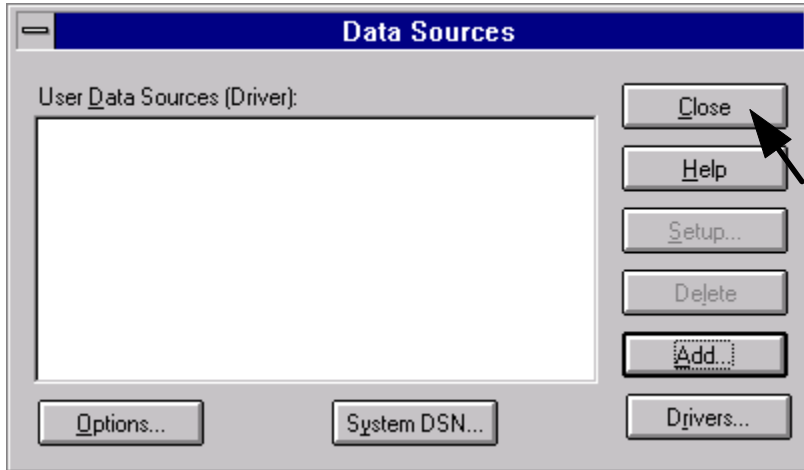


Figure 21: Data sources dialog box

Click Close. The final completion dialog box of ODBC Setup is displayed. Click OK, and the ODBC Setup is complete.

5 Installing the sample files used in this document

The files specified in section 3.1 are included on the compact disc in the root directory. There is also a batch file called Copysamp.bat that can be used to copy these files to the appropriate directory. To use this batch file, go to the command prompt on the computer running the Internet Server. At the command prompt, switch to the drive letter of your CD-ROM drive (shown as Drive D: here) and run Copysamp.bat.

C:\>D:

D:\>copysamp

The preceding directions assume that you accepted the default installation directory of C:\Wwwroot when you installed the Internet Server Beta. If not, you can specify an alternate directory on the command line, for example:

D:\>copysamp C:\Web\Scripts

In any case, the files must be in a directory called \Scripts, which must be located in a directory or subdirectory other than the root directory for the Web services.

Once the files are in the \Scripts directory and the ODBC data source “Web SQL” has been defined, you can use the Internet Explorer to specify the following URLs to run the examples shown in section 3 of this document.:

http://<your server name>/scripts/sample.htm
http://<your server name>/scripts/sample2.htm
http://<your server name>/scripts/sample3.htm