ASP Data Access for Beginners
By Jason Butler

print this article

email this article to a colleague

Introduction

The driving force behind Active Server technologies is data access. In order to develop a truly dynamic web site developers need to allow users to access data on demand. Luckily, with Microsoft's Active Server Pages, this is exceedingly easy to do. This article for ASP beginners details how to connect to a SQL Server 7.0 database using ActiveX Data Object (ADO) and Open Database Connectivity (ODBC).

In order to illustrate the process of connecting to a data source with ASP, we will need to do three things:

- Create a database
- Create an ODBC data source name (DSN)
- Create an ASP page

Creating an SQL Server Database

First we need a database. Since this article isn't about database design, we will create a very simple SQL Server database -- one table! We'll name this database 15Seconds, and we will name our table t_articles.

To create the database:

1. Open SQL Server 7.0 Enterprise Manager (Start (Programs (SQL Server 7.0 (Enterprise Manager).
2. Expand the Enterprise Manager tree, selecting the SQL Server to which you would like to add the database, until you see the "Database" node.

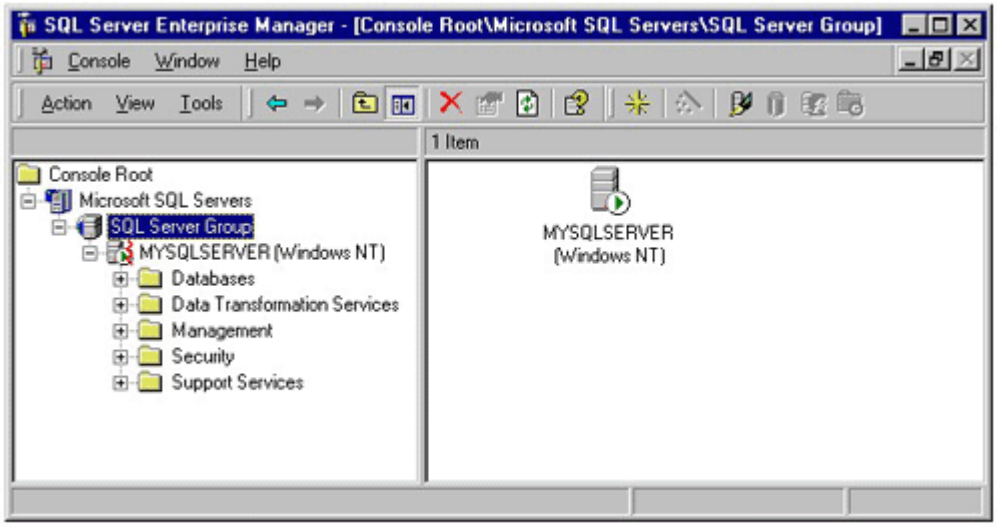

Figure 1

3. Right click on the "Database" node and select "New Database. . ."
4. On the "Database Properties" dialog box, enter "15Seconds" in the "Name" field.

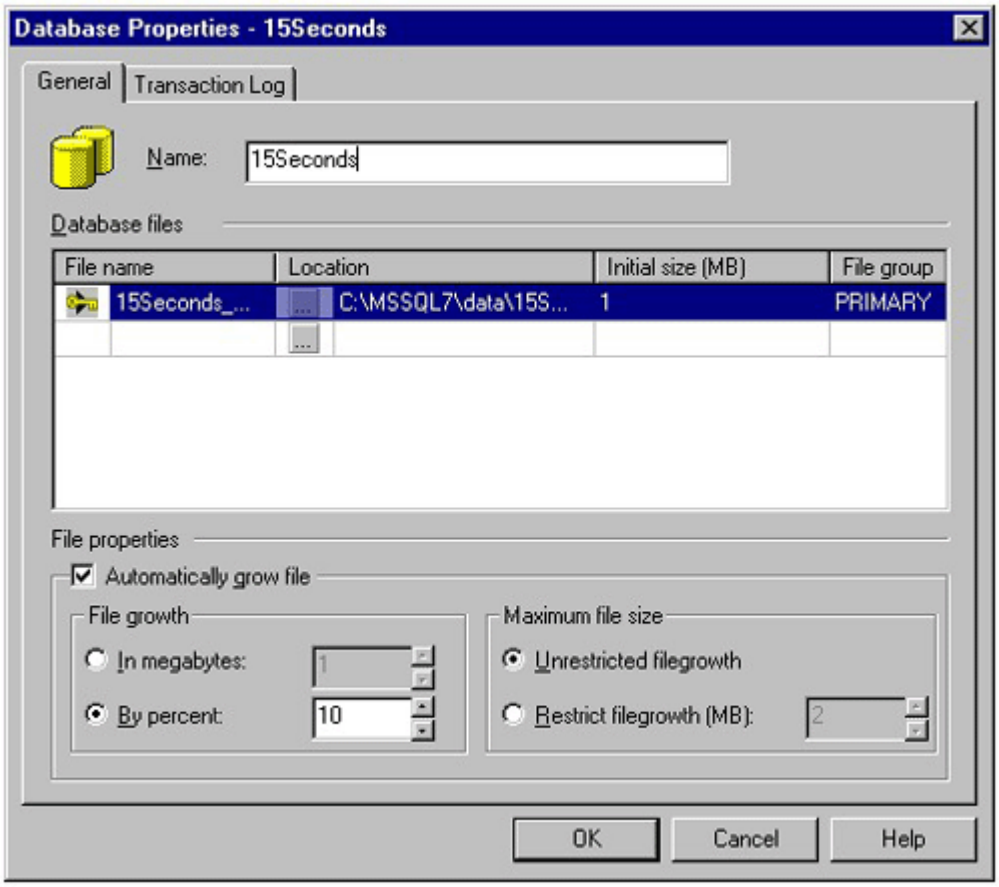

Figure 2

5. Click the "OK" button.

We now have a database named "15Seconds" to which we can add our table.

To create the table, t_articles, perform the following:

1. In Enterprise Manager, expand the "Database" node.
2. Right click on the "15Seconds" node.
3. Select "New."
4. Select "Table. . . ."
5. On the "Choose Name" dialog, Enter "t_articles" in the "Enter a name for the table:" textbox.
6. Click the "OK" button.
7. On the "Add Table" dialog box, enter the following information (see Figure 3):

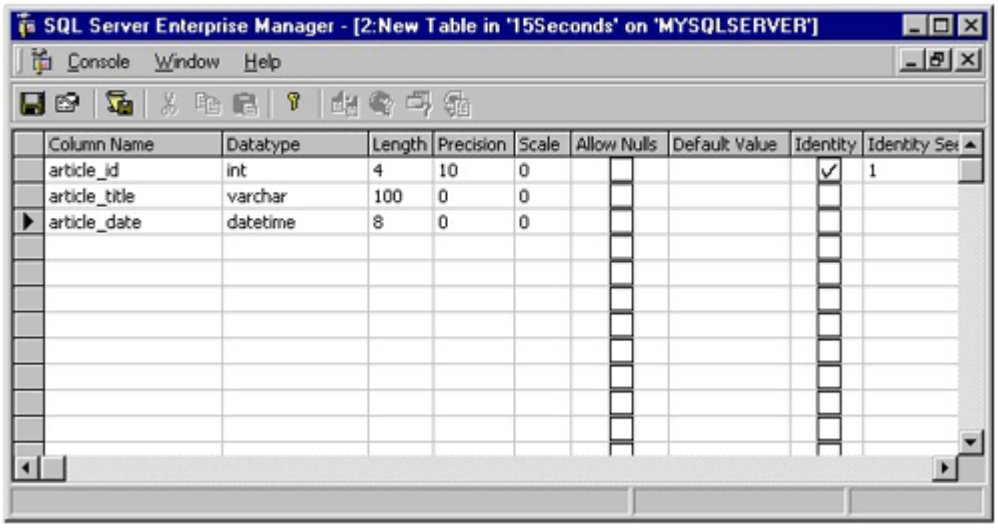

Figure 3

1. Close the "New Table" dialog box.

Now we have a database and table. Let's add some sample data. We'll add one record. To add data to t_articles, perform the following:

1. In Enterprise Manager, expand the "15Seconds" database node.
2. Double click on "Tables."
3. On the right side of Enterprise Manager, right click on "t_articles."
4. Select "Open Table."
5. Select "Return all rows."
6. On the "Data in Table 't_articles'" enter the following data (see Figure 4):

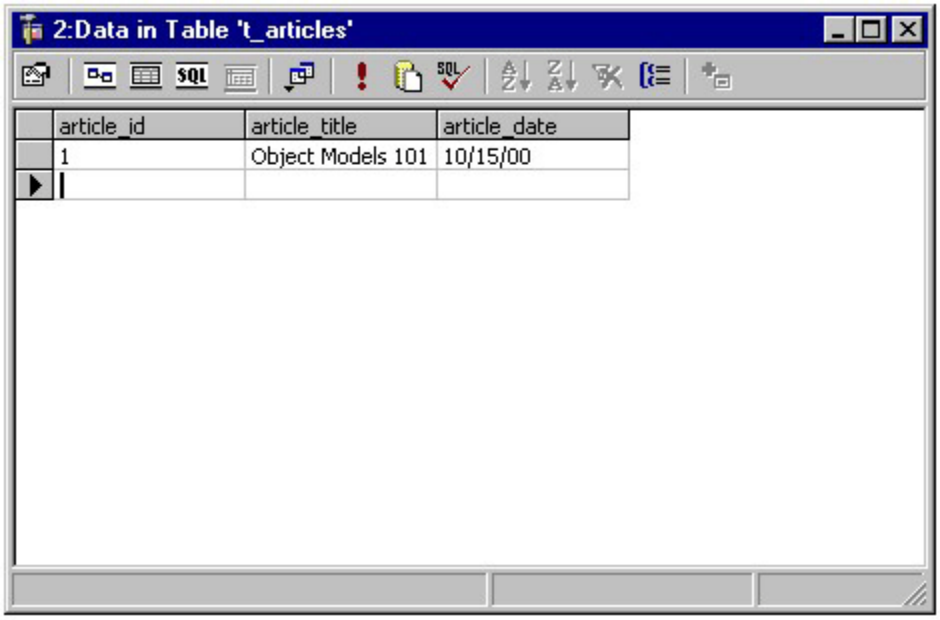

Figure 4

So, there is the entire database. Pretty impressive, huh? Now that we have a database and a table, we need to

create an ODBC connection to our database.

Creating an ODBC Connection

To create an ODBC connection, perform the following tasks:

1. Open Control Panel (Start (Setting (Control Panel).
2. Double click on the "Data Sources (ODBC)" icon in the Control Panel.
3. Select the "System DSN" tab on the "ODBC Data Source Administrator" dialog box.
4. Click the "Add" button.
5. On the "Create New Data Source" dialog box (see Figure 5), highlight "SQL Server" and click "Finish."

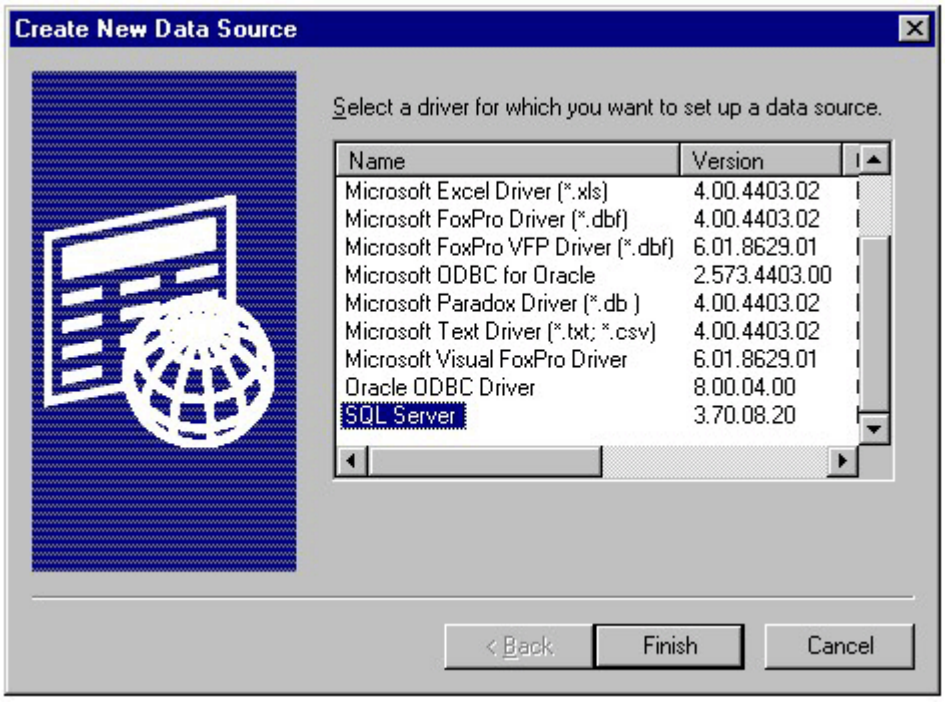

Figure 5

6. On the "Create a New Data Source to SQL Server" dialog box (see Figure 6):
   • Enter "15Seconds" in the "Name" field. This is not the name of the database, but the name for the DSN. I kept it the same just for simplicity, however, this is not good practice for security reasons.
   • In the "Description" field, enter a brief description for the DSN. I entered "15Seconds Sample DSN."
   • From the "Server" drop-down box, select the SQL Server to which you would like to connect. Since, my instance of SQL Server resides on the same machine where I am creating the DSN, I selected "(local)."
   • Click the "Next" button.

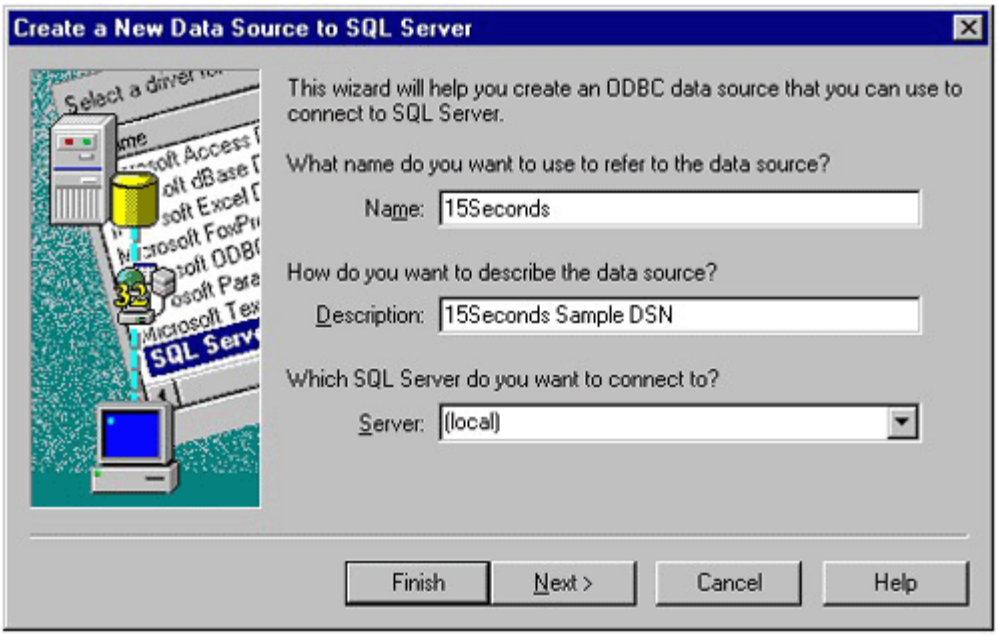

Figure 6

7. On the second "Create a New Data Source to SQL Server" (see Figure 7) dialog box:
   • Select the "With SQL Server authentication using a login ID and password entered by the user" radio button to indicate that database security with be implemented by SQL Server rather than Windows NT.
   • Select the "Connect to SQL Server to obtain default settings for the additional configuration options" checkbox.
   • In the Login ID textbox, enter "sa."
   • Leave the "Password" textbox empty.
   • Click the "Next" button.

   Note: I used "sa" for login ID with a blank password for convenience. This is the default SQL Server administrator account. Again, this is not a good idea for security reasons.

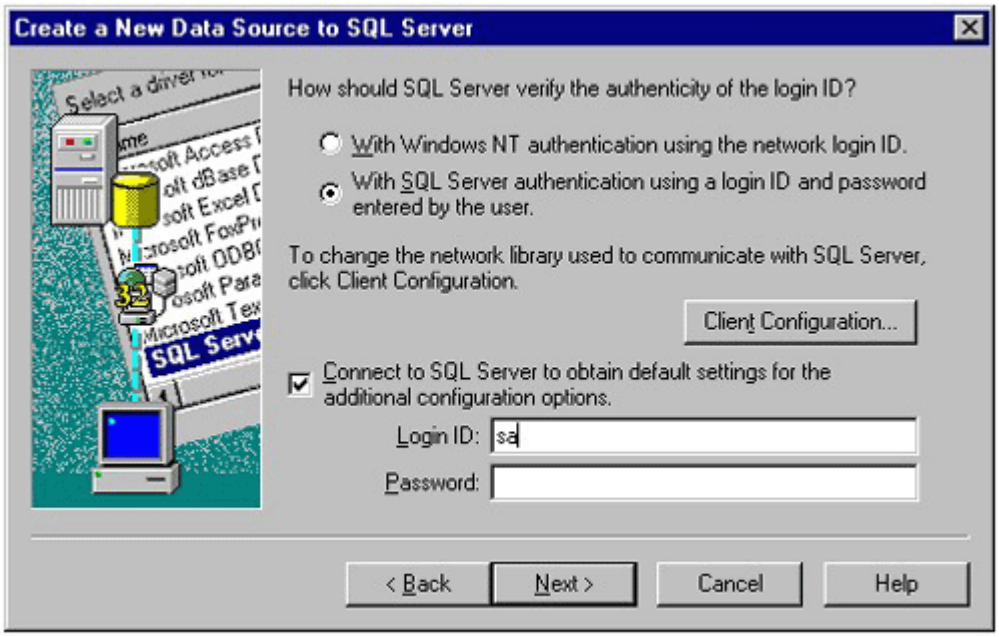

Figure 7

8. On the third "Create a New Data Source to SQL Server" dialog box (see Figure 8): • Select the "Change the default database to" checkbox and select "15Seconds" from the accompanying select box.
   • Click the "Next" button.

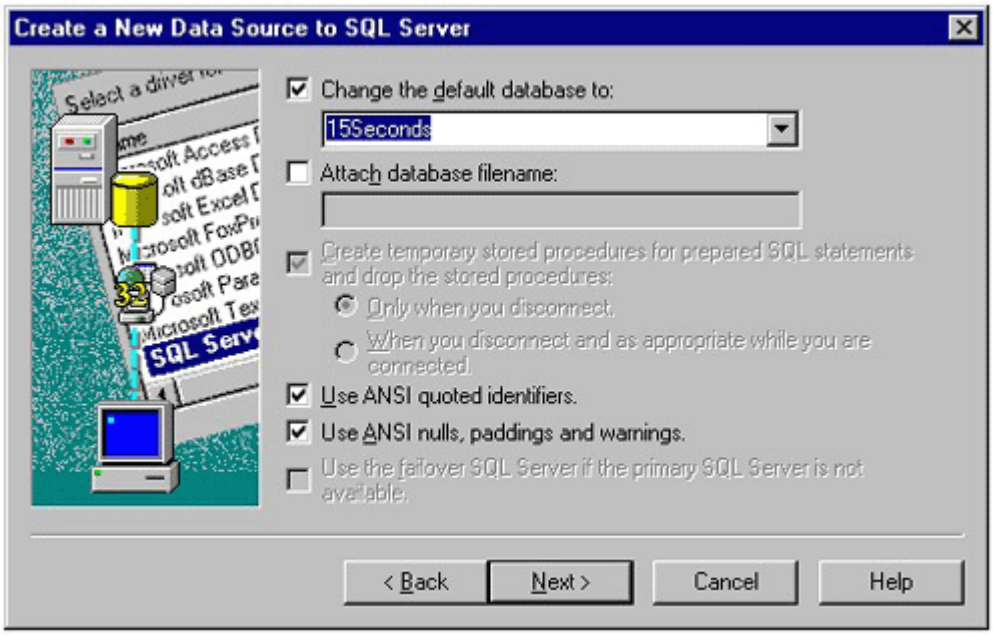

Figure 8

9. On the fourth "Create a New Data Source to SQL Server" dialog box (see Figure 9): • Click the "Finish" button.

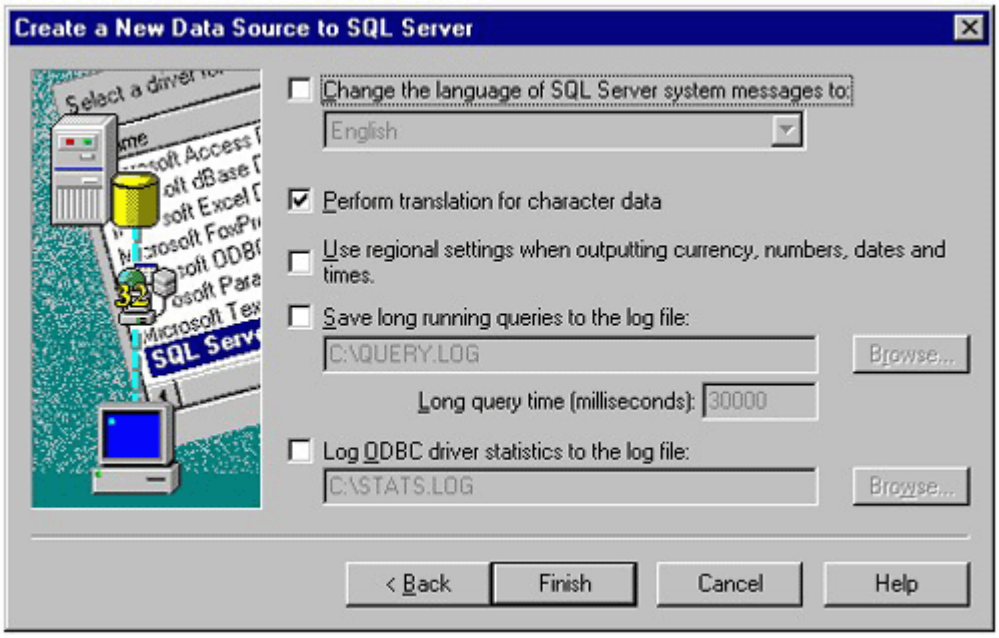

Figure 9

10. On the "ODBC Microsoft SQL Server Setup" dialog box (see Figure 10), do one of the following:
    • Click the "Test Data Source. . . " button to ensure that the ODBC connection has been created successfully.
    • Click the "OK" button to complete the ODBC DSN setup process.

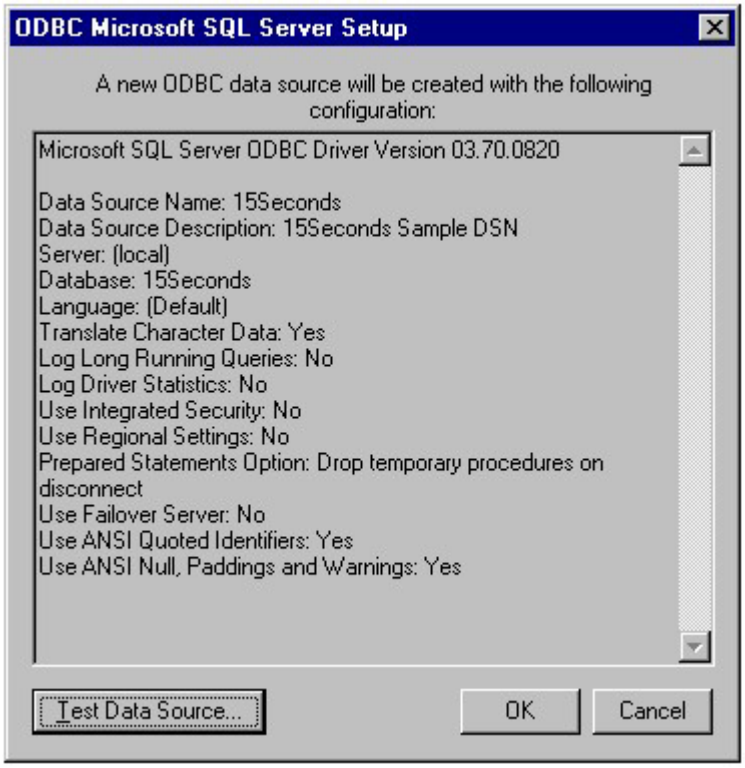

Figure 10

We have a database, a table, some data and an ODBC DSN. The next step is to create an ASP page to access the data.

Creating the ASP Page

1. Begin the ASP script with standard code (or at least my standard):

```
<% @LANGUAGE="VBSCRIPT" %>
<%
Option Explicit
Response.Buffer = True
On Error Resume Next
```

2. Dimension all variables we'll be using in the scripts. I like to dimension my variables in blocks for ease of readability.

```
Dim oConn, oRS, oFld
```

3. Create the Connection and Recordset objects:

```
Set oConn = Server.CreateObject("ADODB.Connection")
Set oRS = Server.CreateObject("ADODB.Recordset")
```

4. Set the connection objects ConnectionString property. Notice in the ConnectionString, we are specifying "15Seconds" as the DSN, "sa" as the User ID (UID). This should look familiar because it is the same information we provided while creating the ODBC DSN.

```
oConn.ConnectionString = "DSN=15Seconds;UID=sa"
```

5. Open the Connection:

```
OConn.Open
```

6. Open the Recordset:

```
oRS.Open "SELECT * FROM t_articles", oConn
```

7. Start writing the HTML to output:

```
Response.Write("<HTML>" & chr(13))
```

```
Response.Write("<HEAD>" & chr(13))
Response.Write("<TITLE>ASP Database - For
Beginners</TITLE>" & chr(13))
Response.Write("</HEAD>" & chr(13))
Response.Write("<BODY>" & chr(13))
```

8. Now we will start an HTML table and create an HTML table row by iterating through the Recordset's Field collection to create column headers for our HTML table.

```
Response.Write("<TABLE BORDER=1 CELLPAADING=0 CELLSPACING=0>" &
chr(13))
Response.Write("  <TR>" & chr(13))
For Each oFld in oRS.Fields
  Response.Write("    <TD>" & oFld.Name & "</TD>"
& chr(13))
Next
Response.Write("  </TR>" & chr(13))
```

9. Using the Record object's GetString method, we will create an HTML table row for each record in t_articles:

```
Response.Write("  <TR>" & chr(13))
Response.Write("    <TD>")
Response.Write(oRS.GetString(,,"</TD>" & chr(13) & "
<TD>","</TD>" & chr(13) & "  </TR>" & chr(13)
& "  <TR>" & chr(13) & "    <TD>","--null--"))
Response.Write("    </TD>" & chr(13))
Response.Write("  </TR>" & chr(13))
```

10. Close all of the object references:

```
oRS.Close
Set oRs = Nothing
oConn.Close
Set oConn = Nothing
```

11. Complete the HTML output:

```
Response.Write("</TABLE>" & chr(13))
Response.Write("</BODY>" & chr(13))
Response.Write("</HTML>" & chr(13))
%>
```

Conclusion

This article details how to connect to the data source using ASP. Data access with ASP, at its core, is very simple. However, there are many ways to improve and build upon the provided example. Try using OLE DB rather than ODBC, or incorporating eXtensible Markup Language (XML) into your data access processes.

About the Author

Jason Butler, a Virginia Tech graduate, is a technical manager for a Big 5 consulting firm. He has built numerous Microsoft-centric Web/e-commerce applications for Fortune 500 and dot-com clients. He can be reached at Jason_m_butler@hotmail.com.