

# BBj JDBC/ODBC For The Web

By Greg Smith

**W**ith BBj there are at least four ways to get data to the web. One way is to use the BBj® JDBC/ODBC capability. BASIS has prepared several demos that use JDBC/ODBC, which are available on our web site at: [www.basis.com/devtools/coolstuff/index.html](http://www.basis.com/devtools/coolstuff/index.html)

Using the JDBC/ODBC driver for placing data on the web requires a CGI process on the web server. This can be any CGI process that can access either the ODBC data source on a Windows platform or the JDBC server on the UNIX platform. Let's begin with the Windows ODBC platform.

The BBj ODBC capability is included as a part of any BBj edition. A current version of BBj needs to be installed on the web server machine in order to provide the ODBC functionality. This does not mean, however, that the data to be accessed needs to be on this machine. Installation on the Windows web server machine is only necessary to provide the ODBC driver.

Within the Windows settings is the ability to configure a System DSN to use the BBj ODBC Driver. Selecting Add will bring up a list of the ODBC drivers currently installed on the machine. From this list, select the BBj ODBC Driver. The next dialog box will allow you to configure the ODBC driver.



Sample Dialog Box configures the ODBC Driver on Windows 2000.

A screenshot of the 'BBj ODBC Driver Configuration' dialog box. It has a title bar with the text 'BBj ODBC Driver Configuration' and a close button. The dialog contains several input fields: 'Name' with the value 'chile\_bbj', 'Description' with 'Chile Company through BBj', 'Server' with 'servername', 'Port' with '2001', 'User Name' with 'guest', 'Password' (empty), 'Database' with 'ChileCompany', and 'Prefetch Size' with '20'. There are three buttons on the right: 'OK', 'Cancel', and 'About'. At the bottom, there is a 'Character Set Translation' section with a dropdown menu and a 'Select...' button.

The "Name" you enter is what you'll refer to when you want to access the data from the CGI process you are using. The "Description" is optional but gives you the ability to determine which DSN goes where, when you have several defined. The "Server" is the machine where the data you want to access resides. The default "Port" for BBj ODBC access is 2001. The "User Name" and "Password," if any, required to access the database, can be entered here. Finally, the name of the database, as it is set up in the BBj Enterprise Manager, must be entered.

Once the DSN is set up, you can refer to it in the CGI process by using the name you provided. In the example above, the DSN is chile\_bbj. You access the data by using the required SQL syntax in your CGI process commands for selecting, updating and inserting data.

Using ActiveState Perl with the Win32 ODBC module installed on the Windows web server, the following code shows how to define the data source and retrieve a record set containing Customer number and name from the Chile Company database that is installed with BBj.

```
use Win32::ODBC;

$page =<<"EOL";
<html>
<head><title>Chile Company Customers</title></head>
<body>
EOL

$DSN = "chile_bbj";
$O = new Win32::ODBC($DSN);
if (! $O->Sql("SELECT CUST_NUM, FIRST_NAME, LAST_NAME FROM customer")){
    while($O->FetchRow()){
        $cust_num = $O->Data('CUST_NUM');
        $first_name = $O->Data('FIRST_NAME');
        $last_name = $O->Data('LAST_NAME');

$page .=<<"EOL";
$cust_num $first_name $last_name
<br>
EOL
    }
}

$page .=<<"EOL";
</body>
</html>
EOL

$O->Close();
print $page;
```

The "if" statement may appear odd, but the "SQL" method returns an error number for a problem and "false" for success. That's all there is to it. Every time the above CGI process opens the connection, a BBj license is checked out. Once the CGI process has closed the connection, the license is released. In most cases, the time involved will be fractions of a second, and the license is available for another instance.

The CGI process can also be accomplished with the Microsoft VBScript language within the Microsoft IIS environment. The following code is an example of how that would be accomplished using the same BBj ODBC driver:

```
<%@ LANGUAGE=VBSCRIPT %>
<HTML>
<HEAD><TITLE>Chile Company Customers</TITLE></HEAD>
<BODY>

<%
strConnect = "SERVER=servername;DRIVER={BBj ODBC Driver};" _
    & "DATABASE=ChileCompany;UID=guest;pwd="

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

```

oConn.Open strConnect
strSQL = "SELECT CUST_NUM, FIRST_NAME, LAST_NAME FROM customer"
Set oRs = oConn.Execute(strSQL, lngRecs, 1) 'adCmdText
%>

<% Do While Not oRs.EOF %>
<% = oRs.Fields("CUST_NUM") %>
<% = oRs.Fields("FIRST_NAME") %>
<% = oRs.Fields("LAST_NAME") %>
<br>
<%
oRs.MoveNext
Loop
%>
<%
Set oRs = Nothing
Set oConn = Nothing
%>
</body>
</html>

```

Both of these code samples access the same BBJ ODBC Driver. The ActiveState Perl can be used in the Apache web server for Windows, while the VB Script is for the Microsoft IIS web server. Notice that the DSN, "chile\_bbj," is used in the Perl script while the database name, "ChileCompany," is used in the VB script. This is necessary because the Perl Script uses the BBJ ODBC manager to identify the BBJ ODBC Driver, while the BBJ ODBC Driver is explicitly called in the VB Script.

The BBJ JDBC Driver is used on UNIX platforms and requires an interface from the CGI process to BBJ. Using Perl, this interface is supplied with the Perl DBD-JDBC module. This module contains a daemon that acts as the interface between Perl and the BBJ JDBC Driver. Within the module documentation are examples of how to set up the daemon. The following Perl code shows how to use the BBJ JDBC Driver:

```

#!/usr/bin/perl

use DBD::JDBC;

$database = "ChileCompany";
$user = "guest";
$pwd = "";
$hostname = "servername";
$port = 9001;
$url = "jdbc:basis:servername?database=ChileCompany";
$url =~ s/([=;])/uc sprintf("%%02x",ord($1))/eg;
$dsn = "dbi:JDBC:hostname=$hostname;port=$port:url=$url";
$dbh = DBI->connect( $dsn, $user, $pwd );

$page =<<"EOL";
<html>
<head><title>Chile Company Customers</title></head>
<body>
EOL

```

This illustration demonstrates a live JDBC query from a cell phone accessing the Chile Company masterfile on the BASIS web site. You can too, if your phone has browser capabilities. Connect to [www.basis.com/cgi-bin/sm\\_data.cgi](http://www.basis.com/cgi-bin/sm_data.cgi)



```

$sql = ("select CUST_NUM, FIRST_NAME, LAST_NAME  from customer");
$sth = $dbh->prepare($sql);
$sth->execute;
$sth->bind_columns( undef, \$cust_num, \$first_name, \$last_name);

while ($sth->fetch()){

$page .=<<"EOL";
$cust_num $first_name $last_name

EOL

}

$sth->finish();
$dbh->disconnect();

$page .=<<"EOL";
</body>
</html>
EOL

print $page;

```

Once you have created whichever program you want to run, you should save it in the location and with the file name extension appropriate for your web server configuration. For the Perl programs, the location is usually in the cgi-bin directory identified in the httpd.conf file for the Apache web server. Usually, the file name extension is either .cgi or .pl.

For the Windows web server, the location is defined in the Internet Services Manager under the Scripts folders for the web site you will be using. The file name extension would be .asp for the Windows web server.

To execute the scripts you would use the appropriate URL to the file. Make sure that you have set the scripts permissions to be executable as well.

All of the methods presented here utilize the BBj ODBC/JDBC driver, which will allow you to get your data to the web using HTML presentations. This capability of BBj, as well as the BBj Thin Client and the BASIC Web Utility, provides several flexible alternatives for allowing access to data via a web browser.

To view the Perl modules contained in this article, visit:

www.

***www.cpan.org***

**The code in this article is available on our website at:**  
[www.basis.com/devtools/coolstuff/index.html](http://www.basis.com/devtools/coolstuff/index.html)