

Creating a Windows Vista Sidebar Gadget Using SQL Anywhere 10 Web Services

A whitepaper from Sybase iAnywhere

CONTENTS

Introduction	3
Accompanying resources	3
Creating the web service for a gadget	3
Creating the sidebar gadget	11
Deploying the demo gadget	12
Summary	13

INTRODUCTION

A Windows Vista Sidebar gadget is a user-friendly, highly graphical utility or widget that allows users to see key information about running applications. As a developer, gadgets add value to your solution because your end-users can quickly view key pieces of information relating to your application. Database-driven solutions can use gadgets to display important application information stored inside the database. For example, you can create a small notification gadget to notify users of certain database changes or gather database statistics for performance monitoring.

This whitepaper describes the details of creating a Windows Vista Sidebar gadget that displays information stored inside a database. To accomplish this, Sidebar gadgets make use of HTML and web technologies and because SQL Anywhere 10 features easy-to-setup web services, the creation of such a gadget is fairly straightforward. The sample gadget directly queries a SQL Anywhere 10 web service for its content. It uses the sample database installed with the product and displays a few live database statistics, as well as a list of the sample customer names. To download a free copy of SQL Anywhere 10 Developer Edition, visit <http://www.sybase.com/detail?id=1016644>.

SQL Anywhere 10 has the advantage of integrated web services, so there is no need to use a third-party component to render the HTML for the gadget. Non-web-enabled databases must have a local service to retrieve the data and then format it. This has been combined into a single package, so that the gadget can query the SQL Anywhere 10 database server directly through a URL.

ACCOMPANYING RESOURCES

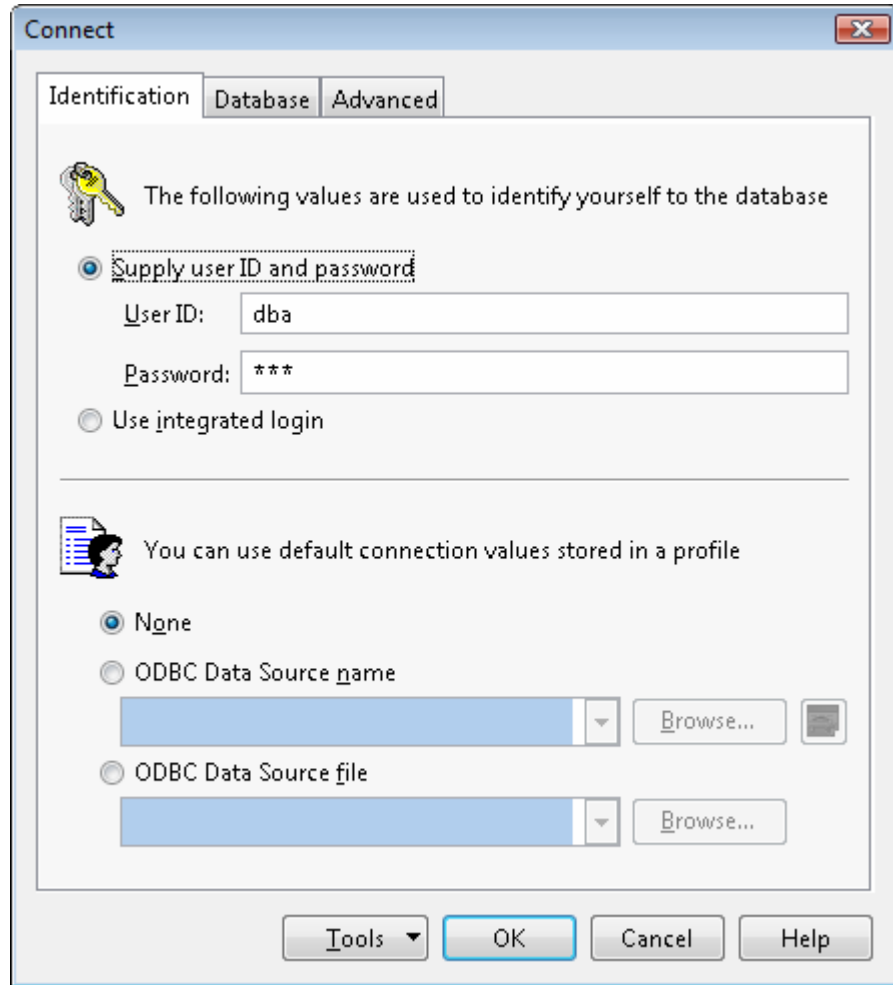
Distributed with this whitepaper is the sample gadget that can be deployed using the instructions below. Additionally, all the source code is available, as well as further background information since the steps below only show snippets of the gadget code. Only code that is relevant to using the database as a web service is shown.

CREATING THE WEB SERVICE FOR A GADGET

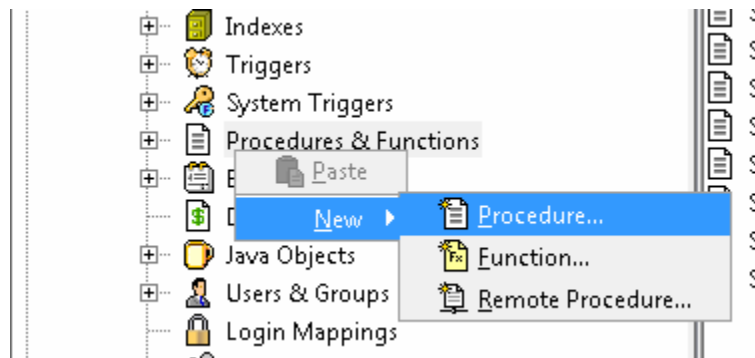
To have the gadget pull data from the database, you expose the data through a web service. This allows data requests to be received through an HTTP connection, as well as have the results returned via the same communication layer. It is simplest to create a stored procedure to generate and format the data set, and then expose this stored procedure through a web service.

1. Start Sybase Central by choosing **Start > All Programs > SQL Anywhere 10 > Sybase Central**.

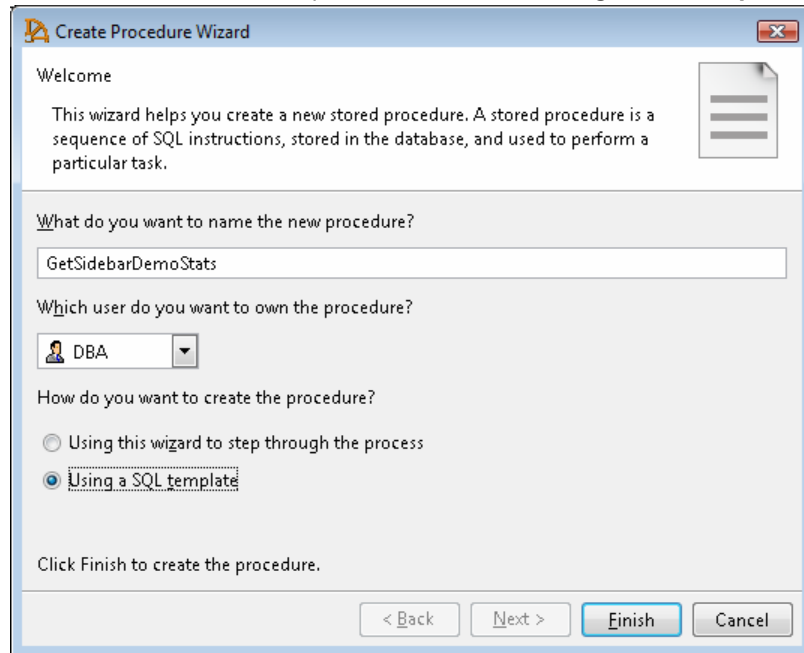
2. Connect to your database. The default location of the SQL Anywhere sample database is *C:\Users\Public\Documents\SQL Anywhere 10\Samples*.



3. Create a new stored procedure by right-clicking **Procedures & Functions** and choosing **New > Procedure**.



4. Enter a name for the stored procedure and select **Using a SQL template**. Click **Finish**.



Your stored procedure should return a result set, XML, or HTML. If you want to format your HTML using any styles or scripts, you must return HTML. The following procedure returns live database statistics:

```
ALTER PROCEDURE "DBA"."GetSidebarDemoStats"( )
RESULT ( html_doc XML )
BEGIN
CALL dbo.sa_set_http_header( 'Content-Type', 'text/html' );
SELECT HTML_DECODE
(
XMLCONCAT
(
'<tr>',
'<td><b>',
'ENG:',
'</b></td>',
'<td>',
( SELECT TOP 1 "Value" FROM sa_eng_properties( ) WHERE
"PropName" = 'ServerName' ),
'</td>',
'</tr>',
'<tr>',
'<td><b>',
'DB:',
'</b></td>',
'<td>',
( SELECT TOP 1 "Value" FROM sa_db_properties( ) WHERE
PropName = 'Name' ),
'</td>',
'</tr>',
'<tr>',
'<td><b>',
'CPU:',
```

```

        '</b></td>',
        '<td>',
            ( SELECT TOP 1 "Value" + ' s' FROM sa_eng_properties( )
              WHERE "PropName" = 'ProcessCPU' ),
        '</td>',
        '</tr>'
    )
);
END

```

This procedure invokes the sa_eng_properties system stored procedure that returns many different database server statistics. Other vital statistics include information about amount of data sent and received and server process statistics. Database-level statistics can be obtained by using the sa_db_properties system procedure and connection-level statistics can be obtained by using the sa_conn_properties system procedure.

One method of returning multiple rows formatted in HTML is by using a cursor as shown in the second stored procedure that retrieves the customer names:

```

ALTER PROCEDURE "DBA"."GetSidebarDemoData"( )
RESULT ( html_doc XML )
BEGIN
    DECLARE datacursor CURSOR FOR SELECT "CompanyName" FROM
        GROUPO"."Customers" ORDER BY "CompanyName";
    DECLARE html LONG VARCHAR;
    DECLARE company LONG VARCHAR;

    SET html = '';

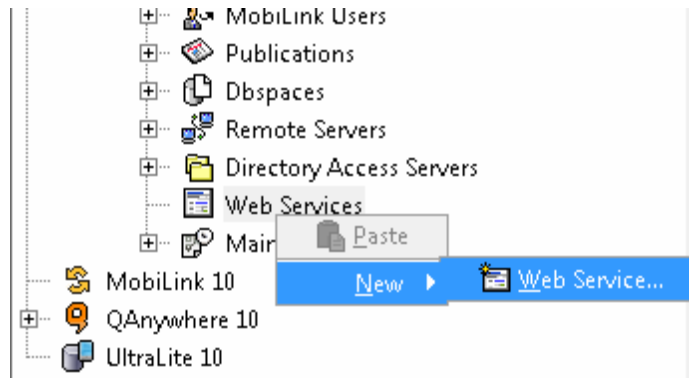
    CALL dbo.sa_set_http_header( 'Content-Type', 'text/html' );

    OPEN datacursor;
lp: LOOP
    IF SQLCODE <> 0 THEN LEAVE lp END IF;
    FETCH NEXT datacursor INTO company;
    SET html = HTML_DECODE( XMLCONCAT( html, '<tr><td>' + company +
        </tr></td>' ) );
    END LOOP;
    CLOSE datacursor;

    SELECT HTML_DECODE( XMLCONCAT( html ) );
END

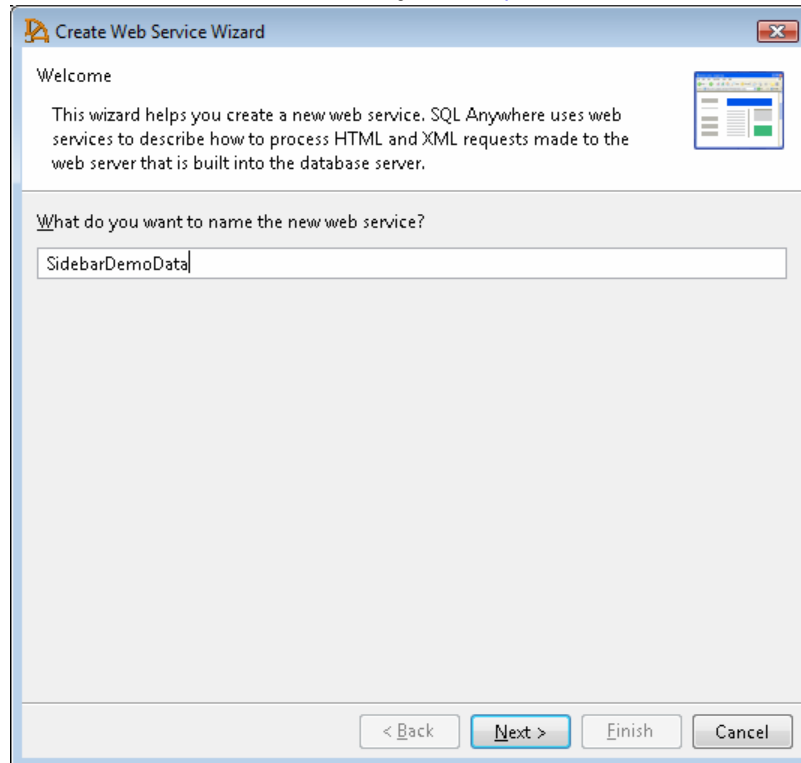
```

5. Create a web service that listens on a port. Right-click **Web Services** and then choose **New > Web Service**.

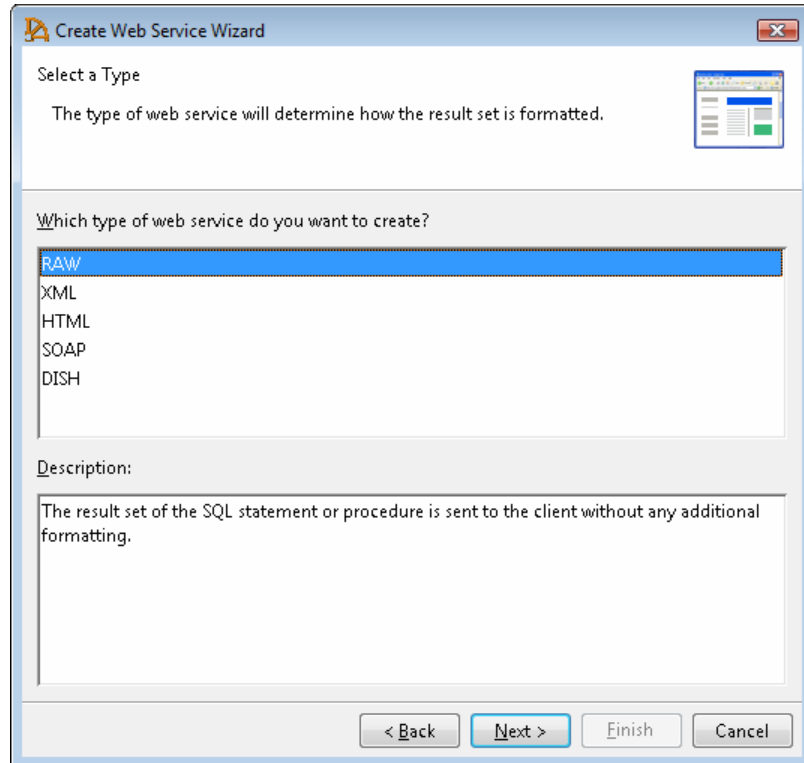


The stored procedure you defined earlier is executed by this web service.

6. Name the web service, and then click **Next**. If you specify the name **root**, then this will be the default web service (accessed directly from <http://localhost:8888>).



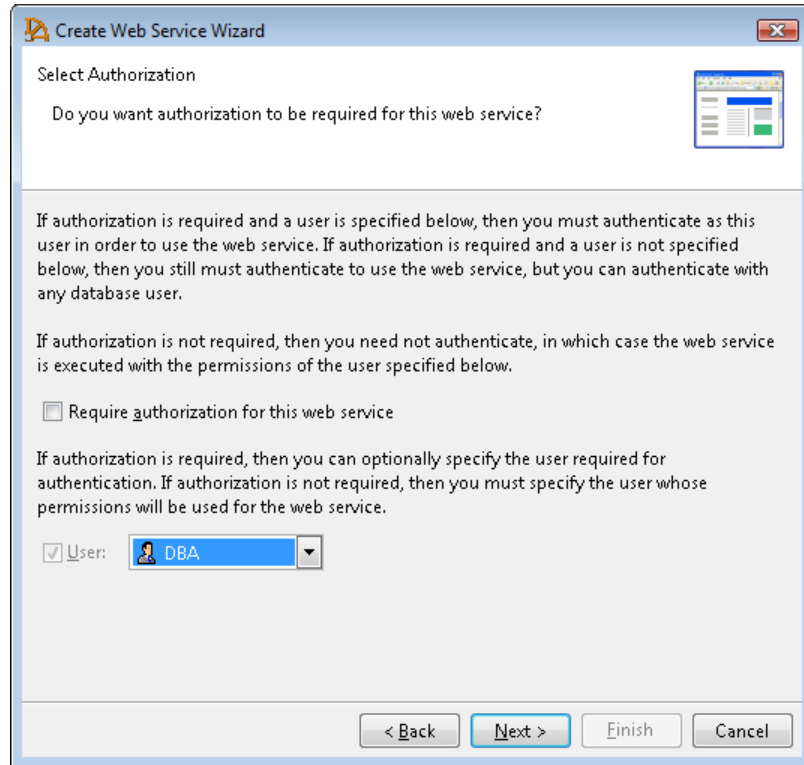
7. If your stored procedure returns HTML or XML directly, then choose the **RAW** output format. Otherwise, choose **HTML**. Click Next.



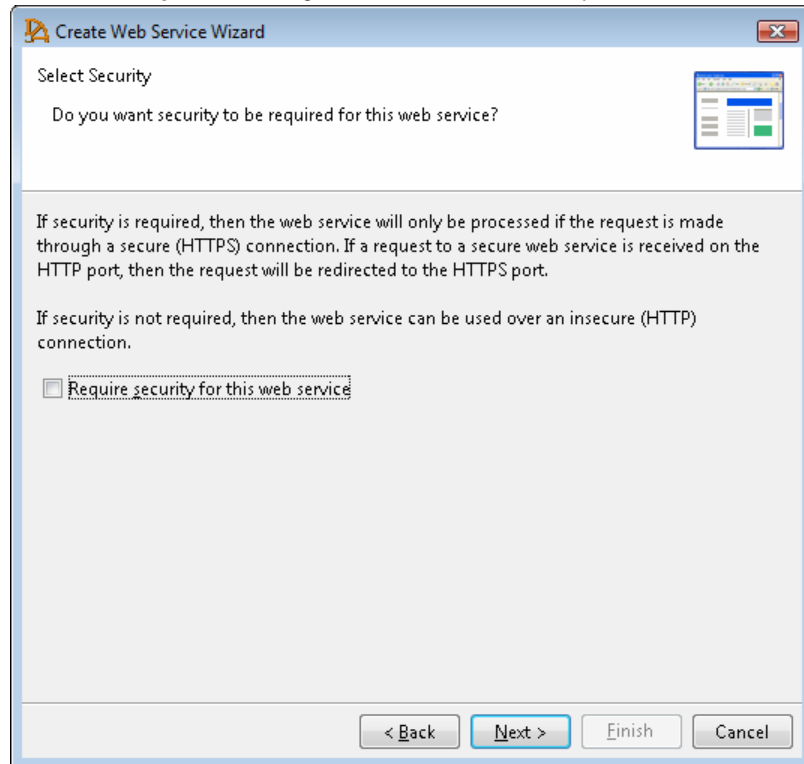
The following types of output format are available:

- Raw** The result set of the procedure is sent to the client without any additional formatting.
- XML** The result set of the procedure is assumed to be XML; if it is not, then the result set is converted to an XML RAW format.
- HTML** The result set of the procedure is formatted as an HTML document with a table containing the rows and columns.
- SOAP** The request must be a valid Simple Object Access Protocol (SOAP) request, and the result set is formatted as a SOAP response.
- DISH** A DISH service acts as a proxy for a group of SOAP services and generates a Web Services Description Language (WSDL) file for each of its SOAP services.

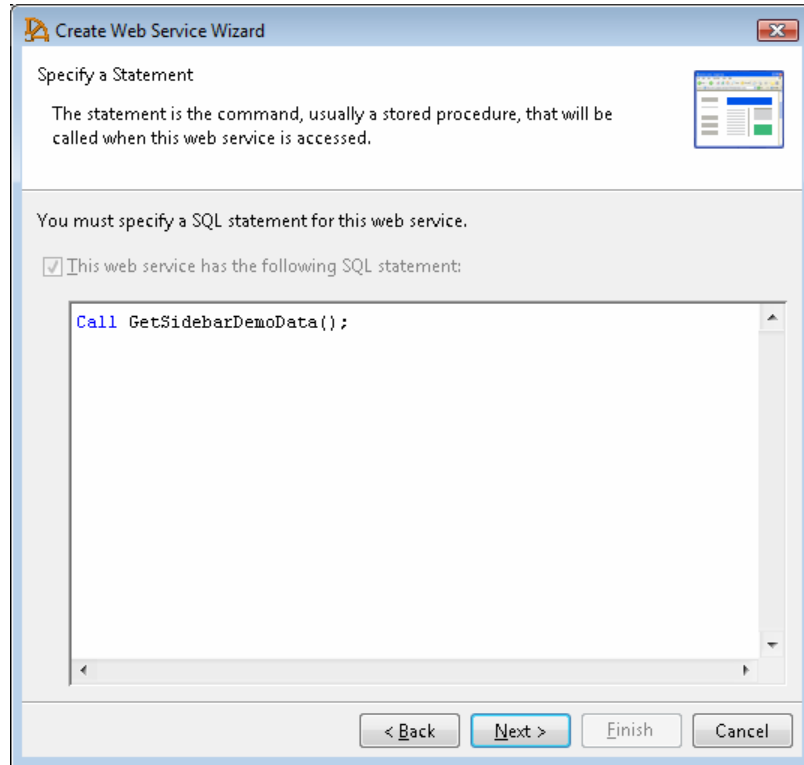
8. For simplicity, do not use authorization for this web service. Clear the **Require authorization for this web service** option. Select the **DBA** user, and then click **Next**.



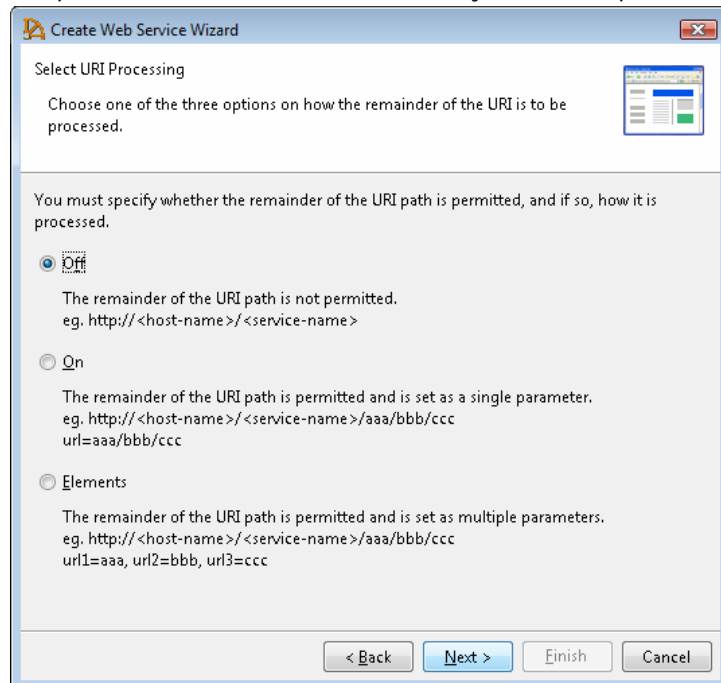
9. Leave the **Require security for this web service** option cleared, and then click **Next**.



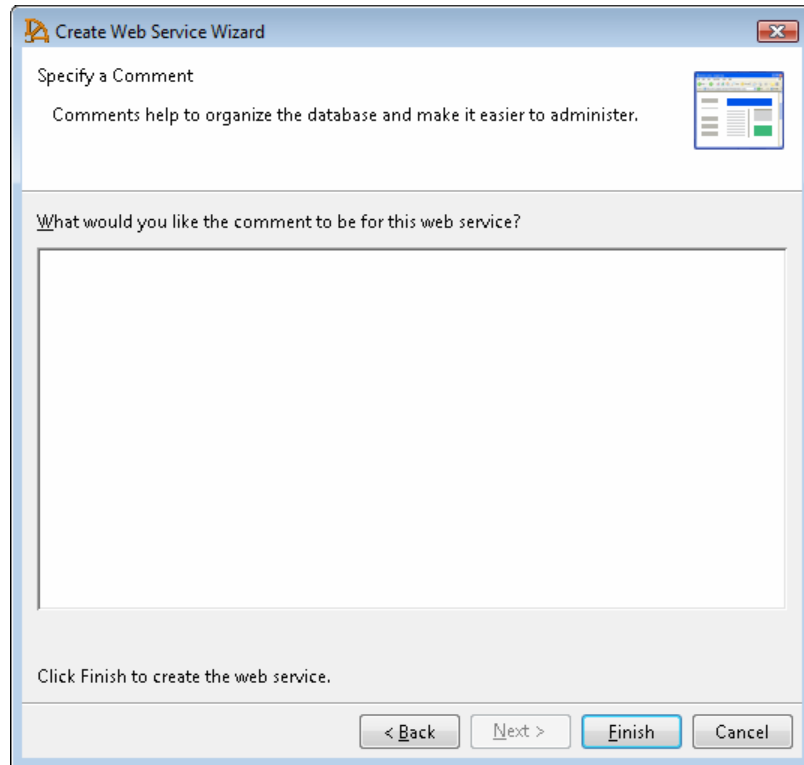
10. Enter a call to your stored procedure. This is how you instruct the web service to run the stored procedure. Click **Next**.



11. If your stored procedure does not take any in parameters, then leave the remainder of the URI path set to **Off**. Otherwise, select how you want the parameters to be parsed. Click **Next**.



12. Enter a comment if desired, and then click **Finish**.



13. Close Sybase Central and shut down the database server (if necessary).
14. Restart the database server by running the following command:

```
dbeng10 -n SidebarDemo demo.db -xs http(port=8888)
```

CREATING THE SIDEBAR GADGET

1. Refer to Microsoft's development guide for the development of your gadget. The documentation can be found at <http://msdn2.microsoft.com/en-us/library/bb456468.aspx>.
2. Sidebar gadgets are essentially just a micro HTML document. Because the SQL Anywhere 10 web service provides HTML, the data can be displayed directly in the gadget. One way to do this is to use a JavaScript XMLHttpRequest to retrieve the data. There are also many other ways of incorporating your HTML or XML into the gadget. You may want to refer to the full source code since only the relevant sections are shown below.

```
function displayData()
{
    var url = "http://localhost:8888/SidebarDemoData";

    dataRequest = new ActiveXObject("Microsoft.XMLHTTP");
    dataRequest.onreadystatechange = writeData;
    dataRequest.open("GET", url, true);
    dataRequest.send(null);
}

function writeData()
```

```

{
    if(dataRequest.readyState == 4)
    {
        if(dataRequest.status == 200)
        {
            document.getElementById('data').innerHTML =
                "<tableid='dataTable' class='data'>" +
                dataRequest.responseText + "</table>";
        }
    }
}

```

3. Ensure that the address and port used above match those of your SQL Anywhere 10 web service.

DEPLOYING THE DEMO GADGET

1. Start the sample database by running the following command:

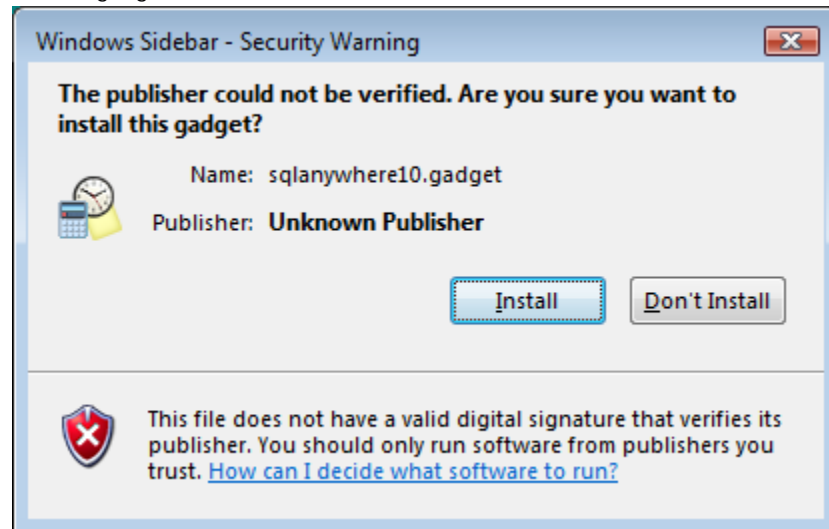
```
C:\Users\Public\Documents\SQL Anywhere 10\Samples> dbeng10 -n SidebarDemo
demo.db -xs http(port=8888)
```

- **-n SidebarDemo** specifies that database server is named **SidebarDemo**.
- **demo.db** the name of the database file that is loaded as the default database.
- **-xs http(port=8888)** tells the database server to add an HTTP listener on port 8888. This exposes all web services, allowing requests to be received.

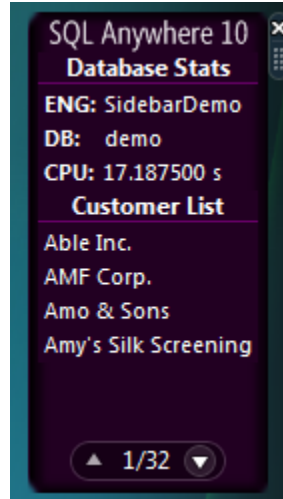
2. Using Interactive SQL, connect to the database and execute the *SetupDemo.sql* script. To do this, run the following command (assuming the location of the script is *C:\SetupDemo.sql*):

```
dbisql -c "ENG=SidebarDemo;UID=DBA;PWD=sql" C:\SetupDemo.sql
```

3. Ensure that Windows Sidebar is running.
4. Double-click the *sqlanywhere10.gadget* file. You are prompted to confirm installation of the sidebar gadget.



5. Click **Install** to have Windows install the gadget.
The gadget shows up automatically in Windows Sidebar.



The gadget shows the web service database server name (ENG), as well as the database name (DB). The third parameter is the CPU usage time for the database server, and is a cumulative timer of how long the process has been granted CPU time, represented in seconds. These statistical values update roughly every 500 ms, depending on system load. Below this information is a list of companies from the Customers table of the SQL Anywhere 10 sample database. This data is refreshed on changing of the page only. Use the Up and Down arrow buttons to scroll through the different pages of data.

SUMMARY

Upon completion of this demonstration, you have successfully created a Microsoft Windows Vista Sidebar gadget, powered by a SQL Anywhere 10 database using web services. The following tasks were described:

1. Creating a stored procedure.
2. Creating a web service.
3. Starting a web service.
4. Creating a sidebar gadget.

COPYRIGHT © 2007 IANYWHERE SOLUTIONS, INC. ALL RIGHTS RESERVED. SYBASE, AFARIA, SQL ANYWHERE, ADAPTIVE SERVER ANYWHERE, MOBILINK, ULTRALITE, AND M-BUSINESS ANYWHERE ARE TRADEMARKS OF SYBASE, INC. ALL OTHER TRADEMARKS ARE PROPERTY OF THEIR RESPECTIVE OWNERS.