PUTTING YOUR DATA ON THE WEB

by Greg Smith

The first Web project I worked on some years ago involved taking data from a mainframe computer and putting it into a Web-based presentation. This Web application took about three weeks to put together. The interesting aspect about this project was that the Web-based application replaced a traditional client/server application that took three programmers one and a half years to complete.

Placing data and information onto the Web was the reason the Web was created in the first place. Many people have just begun to realize the capabilities of Web presentation of their data and how it can help them interact not only with customers but also suppliers, distributors and internal colleagues as well.

A company's data is normally maintained in some sort of database. Whether that database is an accounting system, an inventory system, a document-tracking system or a customer/client system, there is methodology for storing the data. There are also many different ways of connecting to a database. Many Microsoft Windows-based applications use ODBC drivers to allow applications access to databases. There are several other methods of database access that are used by applications on other platforms.

Accessing the data is relatively easy these days. The difficult part is in getting the right data slice the user wants and formatting it in an easy-to-understand manner. Getting just the right data slice means that the user needs to be able to specify what they want. This means designing some sort of selection mechanism for formulating the right data request. This can be done in several ways.

One way is through a "drilldown" or top-down approach in which the user selects options that are continually decreasing in scope. An advantage to this is that the data slices are usually small as the data returned for each request is a pointer category for the next level until the bottom level is reached. This is also a disadvantage in that the user has to possibly go through several iterations of the "drilldown" to get all the data he or she desires.

Another approach is to provide choices to the user on the various data elements available and then return the data that corresponds to those choices. This method involves building a dynamic query that can accommodate the various choices the user makes. SQL, structured query language, is a standardized query method that is supported by many databases. SQL makes requesting and even formatting data fairly straightforward once the syntax is understood.

The process that allows data from a database to be presented to a Web browser is not a complicated one. Web servers utilized a technology known as CGI (common gateway

interface) that allows the Web server to execute programs, receive the data returned from the programs and deliver it back to the user. Some common types of languages used in this manner are Perl, Java, C++, ASP (Active Server Pages) and yes, even BBx®.

These languages work like this. First, the Web server receives a request from the user. This request can be either a command to execute one of the programs or a command to execute a program with additional information provided that will be used by that program. The user provides the additional information through HTML forms on the Web browser. The user fills in information or selects it from drop-down boxes or uses radio buttons or check boxes to identify the information they want to send to the Web server, which will in turn send it to the program.

Once the program has the information, it will access the database through a variety of methods. The database is queried for data that is then returned to the program. Once the data has been returned to the program, it can be manipulated. The program can sort the data, expand on aspects of the data based on programmed or user-defined conditionals, and even perform additional queries based on the data returned.

Once the program has done whatever data manipulations are required, the data must be formatted for return back to the Web server, which will send it back to the user's Web browser. This means placing the data into HTML format. The HTML format is what the browser uses to construct the page for display.

The process of getting data to the Web then involves six steps.

1. The user makes a request for the data, possibly setting parameters for the data through the Web browser.
2. The Web server takes the request and executes the appropriate program, passing it any parameters that the user has specified.
3. The program accesses the database and retrieves the appropriate data, based on the user's request.
4. The program manipulates the data, possibly performing additional queries of the database.
5. The program places the data into HTML format and sends it back to the Web server.
6. The Web server sends the data back to the user's Web browser.

There are many uses for putting data from a database into a Web browser. Lately, this has been in the form of e-commerce in which a company places its goods for sale on the Web. Customers stop by, virtually of course, shop the store by viewing the products as they are displayed from the database and making their selections. Other types of uses are internal to a company. A customer service system, for example, or a technical support system can be run through a Web browser.

Some of the advantages to using a Web-based data delivery system are:

- A common client that is easy to maintain
- The same client can be used for many applications
- The ability to separate the data from the application and the client
- The ability to separate the application from the client
- Flexibility in the choice of languages for the application

While putting your data on the Web with BBx is now possible using the Basic Web Utility, which is distributed with our current PRO/5® and Visual PRO/5® products, our upcoming BBj™ product will make it even easier.