SOLUTIONS SHOWCASE

Scaling The Internet Everest With The Basic Web Utility

In this Solutions Showcase, Stéphane, a developer based in Montréal, and the Canadian office-supply buying company Novexco, for whom he works, wanted to develop a Web-based ordering system for their customers. Using the Basic Web Utility included with PRO/5®, Stéphane and his development group were able to build a working system. What follows is the story of how they did it.

By Stéphane Lagacé

Up until a few years ago, our world was pretty simple. If you are like me--a programmer in the PC world with some years of experience (I mean you were there before they released the mouse and button)--you probably started working on a single computer, moved to a network, which then was upgraded with even more powerful machines. At the same time, you had to adapt to multiple users, multitasking, graphical environments, open systems, open databases, modem communications and then faster modem communications. Probably with each of these improvements, you strove to keep up with the pace of progress.

Then came the Internet! Suddenly, your steadily ascending path became a cliff. So now, business moves more and more to the Internet, and e-commerce is where you need to go!

Our Problem

Working for an office supply distributor, my job is to give merchants from our group the tools they need to face the competition. We were frequently asked by our merchants to develop an application in which their customers could place orders by connecting directly to their data. The temptation was strong to just take our regular order entry, remove inappropriate functions and set up modem access for each merchant to receive their customers' online orders. I've been around technical support long enough to know that with a number of sites hooked up that way, I would spend the rest of my career on the phone trying to resolve their customers' communication problems.

The Net, But How?

Internet was the solution, but we had no expertise other than being able to browse, search and send e-mail. It seemed like scaling Everest. We thought about hiring a firm specializing in e-commerce to develop and maintain an online ordering site for us. That was very expensive, but it seemed we didn't really have a choice. Using the Internet would solve our problems with having to support all kinds of connections. If the customers had a problem connecting, it would be their Internet providers' problem.

Then we heard about Synergetic Data Systems' BB-Web product, which enabled an HTML document to interact with BBX® data. HTML, as you already know, is the basic standard used on the World Wide Web to display and get information. There was hope for us to ascend the Everest, but it still seemed like a daunting task. We would need to learn HTML, CGI, BB-Web, remote access to data and making all of that work together. Who were we to even think we could master all of that? So, we remained undecided.

Customers can search the company's inventory through Novexco's Web-based order system.

But in September 1998 when BASIS bought BB-Web and included it free in PRO/5®, we had to try it. (For you potential climbers, it comes with real online HTML documentation.)

Starting With HTML

We started by taking a look at HTML. HTML is a language in the sense that it has commands and syntax rules, but it stops there. In HTML, there are no "if" statements or "for...next" loops. The language gives instructions on how to display information and connect to other HTML documents. That was something we could handle.

Forms are a special type of HTML document. A form has text entry fields, check boxes, radio buttons and other controls that allow user input. It has also at least one submit button to send the information entered and a call to a CGI script. A common example of an HTML form is a search engine where you enter the keywords you're looking for and then click on the search button to submit the request. It's easy to design such a page, but what in the world was a CGI script and what was happening when Mr. Customer clicked on that search button after typing a keyword in the field?

In most cases, a CGI script is nothing more than a batch or script file. But CGI could start something more familiar to us. What could it start? A PRO/5 session!

Adding PRO/5

There is a little game played between your Internet browser and the Web server where the HTML pages reside. And when you understand it, anything is possible. Your browser displays a form on your screen and lets you fill in the blanks, but when you click on the submit button, it follows a very simple set of steps. It executes the script or batch file specified in the form and passes along all the named variables in the page (just go take a look at the source code of any HTML form on the Internet). In return, it expects another HTML page to show.

So each PRO/5 program involved only needs to look at the data entered by the end user (possibly a very far end user), process that data and send a response in the form of an HTML page or even another HTML form. That's where the Basic Web Utility (the new name of BB-Web) becomes very useful!

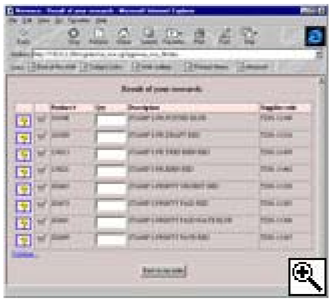
The Basic Web Utility is actually a set of utilities designed to help PRO/5 programs produce HTML output. The Basic Web Utility, through a simple call, gives you access to the user input in the form of a template filled with the data. If you had a field named **`USER_ID`** in the HTML form, you would find a variable named **`CGI.USER_ID`** in your PRO/5 program.

The way the Basic Web Utility lets you build the HTML responses you want to give to the user resembles what a word processing program does when you want to send the same letter personalized for each of your customers. It merges a sample HTML document with data from your BASIS database. So if you have an HTML document containing a string like **`[CUST_NAME]`** and you have a template in your program with a field like **`CUSTOMER.CUST_NAME`**, a call to the proper routine will merge those together to produce an HTML page showing your customer's name.

Among the many utilities in the Basic Web Utility set, you will find some that automatically create a complete input page based on a template and others that handle search lists, including a next-page button. But there's more.

Creative Adapting

Of course, working with the Basic Web Utility implies some adaptation (I guess you're used to it). One is finding a way to debug your program without being able to escape in the middle of a process and print the content of some variables. You can get around this problem by using log files and adding a debug flag to your programs. Then you can edit the log file to see what happened. We, however, developed a more unusual technique. We modified the standard error routine provided with the Basic Web Utility (which prints an error message to the browser's window in an HTML format) to also print the result of the **`DUMP`** instruction. Then, any time an error occurred, we got a complete picture of the data in the program. If we wanted to know about the content of some variables, we added an instruction like **`LET X=1/0`** to cause an intentional error so we could see the state of the program in the browser's window.



Customers can select an item to add to their order.

Because each PRO/5 session ends after producing its output, another concern we had was where to keep the data we needed on subsequent pages. Well, the Basic Web Utility includes a session-tracking mechanism that uses cookies to retrieve stored data and to automatically get rid of it when it's outdated.

We were able to create an order entry system protected by a user id and password scheme, in which our merchant's customer could search items and add them to his order at his regular price. The system includes a confirmation prompt before sending the order directly to our merchant's order database. We also added an online statement on the current customer's invoices.

A Short Time Well Spent

Potential climbers, realize that we only spent a week learning about HTTP servers, HTML, CGI and the Basic Web Utility, and testing the sample application. Also in that same week, we created a small excerpt of the project to verify we could make it work by ourselves. We did that locally on our own UNIX system, which included its own HTTP server.

It took us another two days of planning and reflection to map out all the site's pages and PRO/5 programs, along with the data that would be exchanged between them. Those two days were the best time-saving steps we took through the whole project. You really don't want to skip the planning and documentation because the whole project is a big bunch of really small pieces all put together. You will need a good map while you're doing it and also a month after it's done. In my opinion, the tradeoff in up-front planning will pay for itself 10 times over later on with less confusion and wasted time.

After that, we got outside help for the graphical design of the web pages. (There are still some limits to our skills).

It took us another five days to do the coding and testing, with the help of four new programmers freshly graduated (though they were very good).

That's about it, and we now have a complete Internet site that can feed any of our merchants' ordering systems.

More Than Just Everest

Put the Internet aside for a moment and try to think about how you could create an application that would run a common interface over an heterogeneous network. With the Basic Web Utility, you could write such an application and all you would need would be a browser on each station (they are not very expensive these days) and an HTTP server (Linux, Unix or even NT) somewhere on your network, with access to your data, to respond to requests.

This type of application frees your server because while your users are filling in information on their screens, your server is not being used. If we take this scenario a little further, how many users would you need to license to respond to 30 or even 75 stations if each PRO/5 connection is active only for the time the request is being processed? Hmmm!...

When the project was finished, there was a strange feeling in our department. We all felt like we had conquered the world. When you think about it, as far as computers are concerned, the Web is the world. And we now know how to make the world work for us. Professionally, perhaps your company could greatly benefit from achieving an Internet business solution. But more than that, personally, you will be back on top of the mountain.

Stéphane Lagacé is assistant supervisor for the external MIS department at Novexco Inc., one of the largest office-supply buying groups in eastern Canada, and was the project leader for the company's Internet business solution. Réjean Laframboise was the project director and Erick Laframboise was the Web designer. The programming team included Mélissa Boudreault, Annie Dargis, Sylvain Gendron and David Primeau. Stéphane may be contacted by email at apport@videotron.ca.

Solutions Showcase is a column appearing regularly in the Advantage that describes how BASIS customers are using BASIS products to make money. If you have an application or process you would like to see featured in this column, please contact Elizabeth Barnett at bethb@basis.com.