GET READY FOR BBj™ NOW

by Greg Grisham, Win Quigley, and Greg Smith

BASIS has been working very hard on developing BBj™ (patent pending), the next generation of our business application development environment. We are excited about the language, the new environment and what it means to the current application base as well as future development. We are working to make sure it is compatible with our existing products, which is a fundamental requirement for our development team.

As we are preparing BBj for our Customers, we urge you to start getting ready for BBj now because there are some changes in the environment model that affect the way that the applications and the data can be implemented. So what are the changes? Simply put, infrastructure.

BBj is going to utilize a three-tier architecture behind the scenes. Three-tier architecture is comprised of (1) the user interface, (2) the program or application, and (3) the data. You could, if you wish, run your application just the way it runs now. But, you probably don't want to because of all the advantages that three-tier architecture offers in scalability, marketability and flexibility. You should be looking at your application with a critical eye as to what you might want to change to optimize it to operate in a three-tier environment. What should you do now to prepare for BBj?

MOVE TO GUI

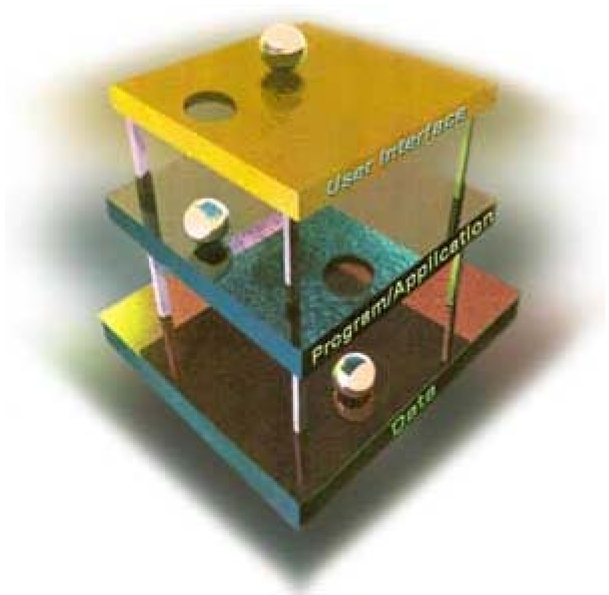
One of the first things that developers will want to review is the user interface of their applications. If they are operating in a character-based interface then moving to a graphical user interface (GUI) is an important step toward BBj. While character-based applications will work in BBj, they are not as marketable in today's software arena as GUI applications. Believe it or not, there are many businesses that have preliminary checklists for buying software, and GUI is one of the things most have on their checklists. If an application doesn't meet the checklist, then it doesn't even get in the front door.

One of most exciting features of BBj is that it will allow your applications to run as applets inside a browser window. Having a character-based application run inside a GUI application, while functional, is not very appealing. Having a GUI application that is integrated with the look and feel of the browser makes the application appear seamless, even more so if it can be enveloped with a company's own intranet/Internet look and feel.

Also, when the application is GUI, the application data doesn't have to be sent over the Internet. The data is accessed completely on the server and doesn't have to be sent to the client. This speeds the response time of the application, and the user perceives the application as if it were local rather than remote.

UNDERSTAND SCREEN ECONOMICS

It is not uncommon for a legacy order-entry application to have anywhere from four to eight different screens that the user must access in order to complete a single order process. If you examine your application as you would an e-Commerce site, you will want to become more demanding, and even stingy, with the number of screens displayed to perform a function. Once you put an application on the Web, anything more than two to three screens deep will become an issue for users.



Running a BBj application in a browser window also brings up screen sizing issues. You will have to define how big the applet window will be that will appear within the browser window. You won't be able to automatically use the whole screen or even to use the scale function. The browser window itself will take up some screen real estate as will any corporate graphics and navigation elements. So careful planning on window size is essential.

REDESIGN FOR PAGE VALIDATION

You'll need to consider "page entry" as opposed to field-entry validation. Legacy applications typically have validation code immediately following the input. Because a standard graphical user interface uses a resource file, a certain amount of field-level data validation can be accomplished within the resource. ResBuilder� provides for this type of development. But why do page-level instead of field-level validation?

Every time data entry is validated, it has to communicate back to the application. Legacy applications ask, "Is this correct?" for each and every field. This type of checking is no longer standard programming practice and, in the case of file validation, generates a great deal of network traffic. In addition to intranet traffic concerns, with BBj, your application can run inside a browser on the Internet. Communication with the application could be affected by Internet traffic also. Field-level data validation needs to be consolidated into a SAVE/CANCEL type of structure. Rather than sending a message back and forth to validate each field entry, a page-level dataset will be much more efficient.

CREATE YOUR DATA DICTIONARIES

This is something that we have been advising Customers to do for a number of years. Using ODBC is commonplace today as opposed to 10 years ago, and it is more pertinent than ever. You need to think about how you want to get data from other applications as well as how you might share your data with other applications. More and more packages are being developed that allow a single application to access multiple data sources. Normalization enhances the ability of other applications to make use of the valuable data contained in your files.

Businesses can now adopt systems that allow them to get data from any part of the organization and put those data together for many different kinds of analyses. This flexibility is critical for applications that will grow to enterprise-level proportions. DDBuilder® is an excellent tool to create your data dictionary identifying your data to the outside world. Furthermore, if you want to report your data in a Web environment, this is the shortest route. NOW is the time to prepare your application to work in a heterogeneous application environment. No longer is there just one set of data and one application.

BUILD YOUR TOP TIER FIRST

Operating in a three-tier environment offers the flexibility of controlling any of the three tiers – the user interface, the program or application, and the data – independently. The user interface sits at the top of the pyramid and is the easiest to tier separate. Using BASIS GUI tools now to define, design and implement this first tier will net you a tremendous advantage when it comes time to integrate the application into a Web browser or thin client operation using BBj. Then you can put your resources into investigating a plan for data normalization. And later still, you can evaluate how your application validates data entry.

Keep in mind that all of this is a starting point, which is why you need to start now. Once you've begun your GUI migration and data normalization, there are still other considerations.

GATHER RESOURCES FOR THE NEXT PHASE

Taking your application to the next level requires a bigger toolbox and expertise you might not have. Start now to develop a network of resources – vendors who can offer security, networks and hardware. Down the line, other resources you may need to complete the solution for your Customer include hardware vendors, specialized software vendors, systems integrators, telephone companies and Internet service providers.

Start looking for ways to gain expertise in these areas:

Networks. TCP/IP is the standard protocol most computing networks follow. For BBj, it is critical. To implement BBj three-tiered architecture, you must have it. To use GLOBEtrotter licensing, you must have it. Networking is complicated. The time to study TCP/IP is now. Consider taking a course, if you intend to help your Customers build networks. Or find a business ally who can bring that expertise to the opportunity.

Security. Web transactions require serious consideration of security issues. You will want to establish a firewall. A firewall is software that allows users entering the system from the Web to access only certain ports and certain data. Firewalls often come with operating systems. Some routers have them. Several vendors sell the code you'll need. You will also need to get to know Secured Sockets Layer (SSL), a protocol used to transmit private documents via the Internet. Netscape and Microsoft browsers both support it. SSL is usually included with your Web server.

Web Server. This is a UNIX or Windows computer that serves up Web pages. Any computer can be a Web server. You install server software, available from many sources, and connect to the Internet. Some software is public domain and free, such as Apache. Vendors such as IBM and Sun sell hardware/software combinations.

HTML. If you use a Web server, you'll need to know HTML (HyperText Markup Language). This is the language used to create documents for the Web. HTML defines the structure and layout of a Web document by using a variety of tags and attributes. There are software packages designed to help you build Web pages, and you can create HTML code using any text editor.

Network Hardware. The building in which your Customer is located may already be wired for the Internet. The good news is the infrastructure is there. The bad news is you'll have to understand how the infrastructure is set up in tremendous detail. Today's standard is T1 wire. However, if you wish to develop expertise of your own, you can install any number of Ethernet schemes, routing systems, concentrators and switches.

All of this is dependent on what you want to do: the more you want to offer, the more you need to know. Modern enterprise computing is complicated and can be intimidating. But what you don't know you can learn or subcontract out. BASIS has many of the tools and resources you need to get started. BBj will take you farther. BASIS understands that the real value is in the application you made. We also know that the return on taking that application to the next level will be well worth the investment you make today.

PROJECT	RESOURCES
Move to GUI Understand Screen Economics Redesign for Page Validation	• BASIS GUI Tools: Visual PRO/5®, GUIBuilder™, ResBuilder®, GUI Configurator, Grid Management Library • www.basis.com/gui/index.html • SCS Consulting Inc., 800.423.1394 (in U.S.) or 505.345.5232 (international) • Books: *GUI The Windows Interface Guidelines for Software Design* by Microsoft Press. *The Essential Guide to User Interface Design* by Wilbert O. Galitz. *Bloopers Don'ts and Do's for Software Developers and Web Designers* by Jeff Johnson. *The Humane Interface: New Directions for Designing Interactive Systems* by Jef Raskin (ACM Press).
Create Your Data Dictionaries	• www.basis.com/onlinedocs/documentation/index.htm (DDBuilder chapter) • www.basis.com/whitepapers/odbc_nonnormal.html
Build Your Top Tier First	• www.sei.cmu.edu/str/descriptions/clientserver_body.html • www.pineapplesoft.com/newsletter/archive/19980101_3tier.html • Books: *About Face: The essentials of User Interface Design* by Alan Cooper.
Networks	• Books: *TCP/IP Illustrated, Volume 1: The Protocols* by W. Richard Stevens (Addison-Wesley Professional Comuting Series).
Security	• www.ntresearch.com/firewall.htm • home.earthlink.net/~michaelburns/fire.html • Books: *Building Internet Firewalls* by D. Brent Chapman, Elizabeth D. Zwicky and Deborah Russel (Editor). *Firewalls and Internet Security: Repelling the Wily Hacker* by William R. Cheswick and Steven M. Bellovin (Contributor) (Addison-Wesley Professional Computing Series).

Web Server	- www.apache.org - www.c2.net/products/sh2/ - Books: *Administrating Web Servers, Security and Maintenance* by Eric Larson and Brian Stephens. *Web Proxy Servers* (Web Infrastructure Series).
HTML	- www.w3.org/MarkUp/ - Books: *HTML Complete* by Sybex Inc.
Network Hardware	- Books: *The Architecture of Computer Hardware and System Software: An Information Technology Approach, 2nd Edition* by Irv Englander.