BBj Components

BBj Provides Components for End Users, Developers and Third-Party Access

By Brian Hipple

By now you've probably heard a lot about BBj®, but maybe you're just a little unsure about the many different components of BBj that are available. Whether you're an end user, developer or third-party application provider, BBj probably offers a component that your company can't live without.

END-USER COMPONENTS

BBj Services: BBj Services is the heart and soul component of BBj. It provides for central control and access to all data and BBj programs. A lot of first-time BBj users might try to equate BBj Services with the PRO/5 Data Server®, which is really like comparing my bank account to Bill Gates' account. Whereas the PRO/5 Data Server only provided for remote file access, BBj Services provides local and remote file access, SQL processing, administration, user access, interpreter access and thin client support. In order to accomplish these tasks, BBj Services is broken down into two servers: the BBj Data Server and the BBj AppServer.

BBj Data Server: The BBj Data Server knows and coordinates who and what can connect to data and program files. We combined three servers to create the BBj Data Server: the file system server, SQL server and the admin server. The file system server handles all file access, both local and remote. The SQL server does all the SQL processing, which is far more efficient than PRO/5 Data Server where the client, either ODBC or interpreter, does the processing. In client-side SQL processing, the server must transfer all data records to the client first. But in server-side SQL processing, the server sends only the records that meet the query criteria to the client, which is much more efficient. The admin server is the server that (1) interacts with the BBj Enterprise Manager™; (2) allows for configuration of BBj Services, such as ports to use and logging information; and (3) provides database configuration and user administration.

BBj AppServer: The BBj AppServer knows what and how application code can execute. The BBj AppServer is comprised of four servers: Thin Client server, Thin Client proxy server, terminal server and interpreter server. All of these servers have two basic purposes:

1. Start an interpreter session on a client's behalf
2. Run all of the interpreter sessions in one Java Virtual Machine (JVM)

In Thin Client mode, the Thin Client server runs the application specified by the client and keeps in communication with the thin client for user activity and additional processing. Speed is the primary goal in running all of the interpreter sessions in one JVM. It would be very inefficient to run a JVM for each session because the majority of the speed loss in Java occurs in the loading of the JVM.

BBj Enterprise Manager: The task of managing the number of servers that make up BBj Services could make any system administrator cringe. But to make it easy for the system administrator, BBj has a new tool called the BBj Enterprise Manager. A system administrator can use the Enterprise Manager to configure and manage any of the BBj Services running on a network from any location.

BBj Interpreter: The BBj Interpreter is the parser and interpreter of the BBj language. The BBj Interpreter improves on the previous releases of BBx interpreters by allowing multiple-line IF statements, non-line number programs, embedded Java code, reserved words, register/callbacks, dynamic limits and memory allocation, and ASCII programs.

BBj Thin Client: BBj Thin Client connects to BBj Services and requests a BBj program be run on its behalf. BBj Thin Clients keeps in communication with BBj Services and displays the requested program's interface, either CUI or GUI, on the client machine. With BBj Thin Client, BBj users can now run any BBj program in thin client mode! You can also use this interface in a Web browser. You can use it on any machine that has a Web browser, a Java Runtime Environment (JRE) version 1.3 or higher and TCP/IP installed. In this type of configuration, you can run any BBj application over the network without even having BBj installed on the user's machine!

DEVELOPER COMPONENTS

BBjIDE: The BBjIDE is a set of tools for the BBj developer that allows for rapid development, debugging and maintenance of BBj applications. Because we wrote BBj in Java, the BBjIDE can run on any platform with a JRE 1.3 or higher, a GUI environment and TCP/IP support. You can view the display of the BBjIDE in Windows, Metal or Motif - your choice. The BBjIDE contains an editor/interpreter that allows for multiple-line editing. The BBjIDE allows for cut/copy/paste actions and has a find/replace utility for finding and/or replacing occurrences of a string in a BBj program. A developer does not have to put an ESCAPE into a program to debug a program. With the BBjIDE, setting a breakpoint is as easy as a click of a mouse. And for those developers that like the command line interface, you still have a command line processor in the BBjIDE. The History tab shows all commands executed at the command prompt, and we've added a Watch tab to allow for watching program variables.

ResBuilder®: ResBuilder allows developers to create and maintain ASCII (.arc) resource files for use with BBj GUI applications. In future releases of BBj, the BBj AppBuilder will replace ResBuilder, which we will incorporate into the BBjIDE.

Utilities: The ResConverter utility converts existing binary (.brc) resource files into ASCII (.arc) resource files for use with BBj GUI applications. Two other utilities, the BBj Compiler (BBjCPL) and the BBj Lister (BBjLST), respectively, convert BBX or ASCII programs into BBj tokenized code and convert BBj tokenized code into ASCII listings.

THIRD-PARTY COMPONENTS

BBj ODBC Driver: The BBj ODBC driver provides an SQL interface to the BBj file system for third-party applications running on Microsoft Windows. In previous releases of the ODBC driver, the SQL process took place on the client. With the BBj ODBC driver, the SQL engine is now part of BBj Services, which makes the driver very thin (small).

BBj JDBC Driver: The BBj JDBC driver provides an SQL interface to any third-party Java application running on any platform BBj supports.

CLIB: The CLib is C language API (Application Programming Interface) that enables a C program access to the BBj file system.

JLIB: The JLib is Java API that enables a Java program access to the BBj file system.

With components for developers, end users and third-party access, we created BBj to be a flexible, modular, computing solution. This overview gives you a good starting point in determining which components are right for you and your Customers.