Features And Enhancements In BBj 2.02

By Greg Grisham

Software developers all over the world are fascinated with Java. A growing number have taken the plunge and are developing or enhancing their applications with the Java language. No matter the level of interest, there is still an open question of performance and memory requirements. BASIS has been aggressive in addressing these questions and working to provide solutions to these issues.

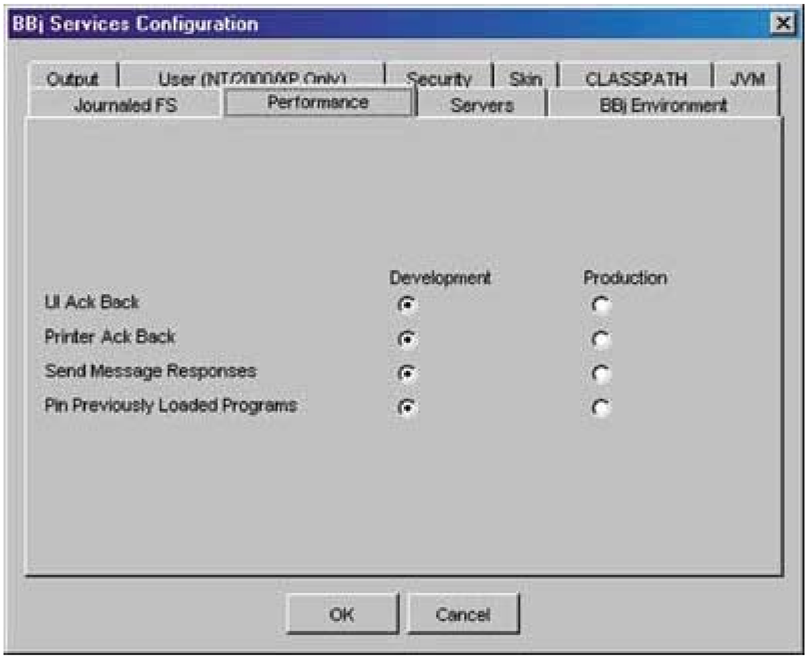
In October, we released BBj® 2.02. The ".02" denotes a maintenance release, but a number of the features in this release are directed at performance and memory management.

Features in 2.02

Pinning

No, this does not refer to a fraternity ritual. this type of pinning has to do with program execution logic that allows the interpreter to assume the program it knows and remembers (or has cached) hasn't changed. When this assumption can be made, the interpreter no longer has to compare its cached copy of the program with the version on the machine's hard disk. This verification is expensive timewise and can often be skipped in a production environment, resulting in much faster program loading and execution.

Notice the bottom selection on the **Performance** tab in the Enterprise Manager.

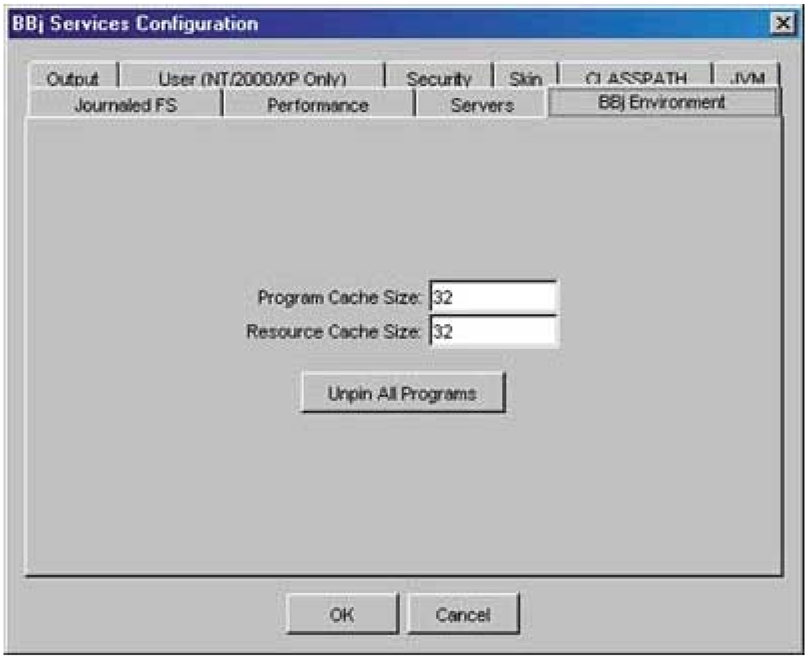


	Development	Production
UI Ack Back	●	○
Printer Ack Back	●	○
Send Message Responses	●	○
Pin Previously Loaded Programs	●	○

While pinning can enhance the performance of an application, the legacy Business BASIC programmer should handle it with a certain amount of caution. A common activity in supporting a legacy Business

BASIC application is to dial in, load and modify a program, test it, and announce to the end user that the system is "fixed." In a pinned environment, an additional step must be taken to "unpin" the program modified, and then allow it to "repin." The code can be tested in the interpreter session that is currently invoked, but the next time it is loaded, the cached version will be called unless it gets repinned.

In a pinned environment, click on "unpin all programs" to clear the program cache. If the "pin" option is turned on, programs will get "re-pinned" as they are loaded.



Program Cache Size: 32

Resource Cache Size: 32

Print Preview

Now that your programs are loading faster and sharing memory, you'll also notice the **Print Preview** has been enhanced. The rendering is significantly sharper, the presentation is many times faster than before, and it consumes much less memory.

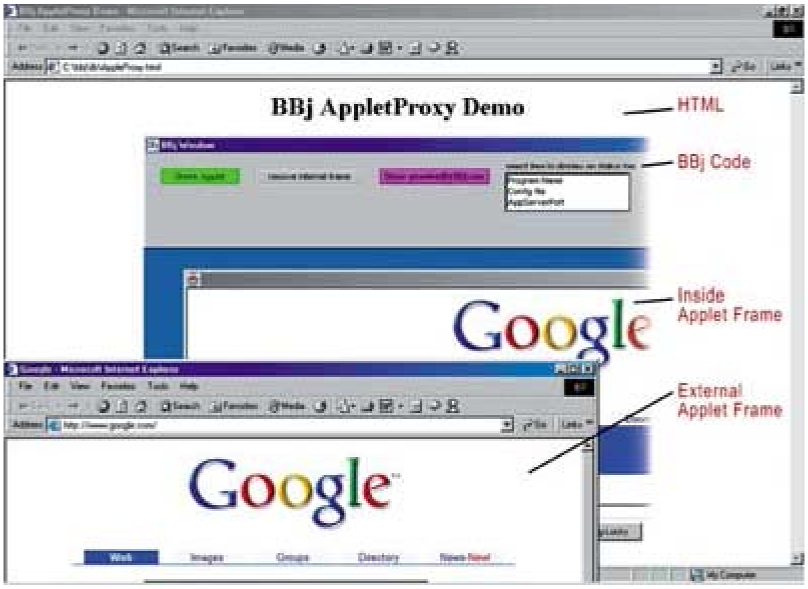
Enhanced Tokenizer

The BBjCPL program is also considerably faster than ever before. This is important to anyone maintaining code through a version control system, where they are archiving their programs in a text format. It is also very important to the developers using GUIBuilder to generate their applications. GUIBuilder uses the BBjCPL during its build process to generate the final, tokenized version of code. In recent tests of the 2.02 version of BBjCPL, this activity was more than ten times faster with a number of sample programs. Many of our beta testers reported even better performance.

BBjAppletProxy

If you read only one section of the documentation for this release, make sure it is the one on the BBj AppletProxy, found in BBj Objects. The BBj AppletProxy is a BBj 2.02 feature for web deployment. This AppletProxy allows for window control of your web application much like you would find in a Win32 desktop environment. You can launch and control subsequent external browser windows from a URL. You can also launch a URL within your current window frame.

The BBj program launched a **google.com** session in another frame in the same browser window. Then it launched another **google.com** session in a separate browser window.



In the screen shown on the left, which is code taken directly from the documentation, you'll notice that the BBj application window is in the frame from which it was launched, then, the code launched an internal browser window to *google.com* and an external browser window linking to *google.com*. Obviously, these could be different sites, or other applets.

The functionality of the AppletProxy is so dynamic, it allows the developer to bring together whatever other browser applications are needed to a single desktop. If you have developed some BASIC Web Utility programs, and maybe some Perl queries using ODBC, and possibly some unique HTML, you can now launch and control them all from one BBj program. Even more exciting, you can launch them in the same window (like a menu), or in an external window (lookups, external apps, whatever). This is a significant feature in Business BASIC control of a browser application.

BBjIDE

The BBjIDE has undergone a number of fixes and, most notably, it can now be run in a Linux environment. Since the BBj installation is handled by a shellscript, in order to take advantage of this new feature, you must be sure to download the JDK, and not just the JRE. You must also have the NetBeans.jar file supplied from BASIS in the same directory as the installation script. The script will automatically detect the presence of the jar file and install it. Once it is installed, you can invoke the IDE using the BBjIDE script in the **bin** directory. The screen below depicts the Chile Company Customer masterfile **chile_cust.bbj** code loaded into the IDE. BBj 2.02 may be just a maintenance release, but it stands as evidence of how BBj really *is* "Business BASIC getting better!"

This screen capture shows the Chile Company Customer masterfile code loaded into the BBj IDE from a SCO Linux 3.11 computer.