Performance Enhancements In BBj 2.0

By Jeff Ash, Jim Douglas and Kshanti Greene

Since the release of BBj® 1.0 in March 2001, BASIS engineers have been working to improve performance in several key areas. GUIBuilder®, Print Preview and BBj's SQL engine are three areas in which we achieved significant improvements for BBj 2.0.

GUIBUILDER ENHANCEMENTS

The original version of GUIBuilder generates programs that analyze the resource file at run-time, using library routines in the _qres program. With BBj 2.0, this analysis is done as the program is being generated; information about the resource is written directly into the generated program. This change dramatically reduces the volume of code that must be executed when starting a GUIBuilder-generated program, reducing several thousand lines of code to about 100 to 200 lines of code. GUIBuilder-generated programs will now load much faster. The BASIS b-CommerceT program, for example, loads up to four seconds faster.

In BBj 2.0, the default for GUIBuilder-generated programs is the BBj-specific version of the framework code (gb_ini.cod). The functions for reading and writing to the screen are significantly faster than in the older version of gb_ini.cod, but they use features that aren't supported in Visual PRO/5®. If compatibility with Visual PRO/5 is important, gb_inibbx.cod is also included. To use this version, just copy it to gb_ini.cod.

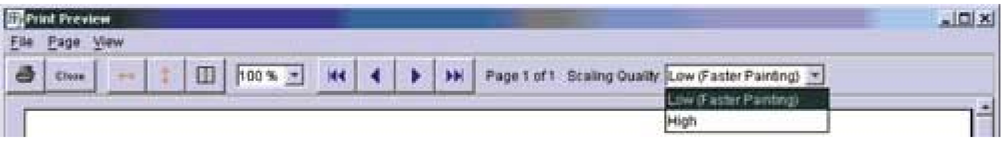
BBj PRINTING

We have made several significant optimizations and enhancements to BBj, which will speed printing to the print channel and Print Preview. These changes will greatly improve printing performance particularly when running a thin client application.

Print Preview

Improvements to Print Preview include optimizations and feature additions. We have considerably decreased the time to load Print Preview. Once the print channel is closed, Print Preview now loads in about 1 second. We have also added an option to change the scaling quality of the preview. Lower-quality scaling will load each page faster. This is generally sufficient to view print output. If you need a clearer image of the page, you can use higher quality. By default, Print Preview will scale the pages using low quality. You can change the quality using the drop-down option in Print Preview as shown in Figure 1. The quality option you chose will then become the default quality.

Figure 1. Select scaling quality from drop-down list in Print Preview.



Page 1 of 1 Scaling Quality Low (Faster Painting)

Printing to Print Channel

We have also made several optimizations and added new features to improve performance of applications that print to the print channel. Printing to the print channel is now five to 15 times faster than BBj 1.01.

We have also added the ability to do batch printing on the print channel. This enables the application to buffer writes to the print channel on the server side until the print channel is closed or until batching is ended. When batching has ended, all prints up until that point will be sent to the client side in one message. We recommend using batching for thin client implementations and for large print jobs.

Print Batching can be set up in one of two ways.

1. Include the BATCH mode in the printer modes. You can do this either through the config.bbx alias or through the open verb.

BATCH mode has two options: END and FF. The END option will buffer the writes until the print channel is closed with the CLOSE verb, END or the 'ENDSPOOL' mnemonic. The FF option will buffer the writes until an 'FF' mnemonic is written. Then it will send the batched writes to the client side. It will also send the writes when the print channel is closed. The FF option is useful for very large print jobs so that the buffer on the server side does not get too large. It will send the print job page by page.

Add the mode "BATCH" to get batching by default. If setting mode in config.bbx alias:

```
        ALIAS LP SYSPRINT "" BATCH=END
```

or

```
        ALIAS LP SYSPRINT "" BATCH=FF
```

If setting mode in OPEN verb:

```
        OPEN (SYSPRINT,MODE="BATCH=END")"LP"
```

or

OPEN(SYSPRINT,MODE="BATCH=FF")"LP"

2. Use the mnemonics 'STARTBATCH' and 'ENDBATCH'. These are dynamic and can be set at any time, once the print channel is opened, with or without the BATCH mode being set. If the 'STARTBATCH' mnemonic is used, prints will be stored on the server side until the 'ENDBATCH' (or 'ENDSPOOL') mnemonic or the CLOSE verb.

```
        SYSPRINT=UNT
        OPEN(SYSPRINT)"LP"
        PRINT(SYSPRINT)'STARTBATCH'
        PRINT(SYSPRINT)"batch printing"
        PRINT(SYSPRINT)"another line"
        PRINT(SYSPRINT)'ENDBATCH'
        CLOSE(SYSPRINT)
```

See the BBj documentation for a number of batch printing examples.

The optimizations and enhancements we have made to the BBj print channel and BBj Print Preview will improve the performance of your BBj applications considerably.

USING THE INTERNAL BBj SQL ENGINE

Another great improvement in BBj 2.0 is the increased performance when using the SQL verbs in BBj to access your BBj data. This enhancement is available when the BBj language interpreter and the SQL engine are running on the same machine within the same BBj services process. In this setting, the language interpreter no longer creates a socket connection to the SQL engine but can access the SQL engine directly. Elimination of the socket overhead has shown marked improvement in performance (especially in calls to

SQLFETCH). You can expect to see fetch times cut by as much as 50 percent.

Note, however, that when accessing third-party data via the SQL verbs, whether a socket connection is made to the database management system or not, is completely driver-dependent.

Our improvements in these three key areas-GUIBuilder, Print Preview and the BBj SQL engine-will dramatically increase the performance and efficiency of your applications.