

Connection Pooling – Part 2

License Pooling and Lingering

By Mark Schnedar

License pooling and lingering, added in BBj® 4.01, reduced the number of socket connections between BBjServices and the BASIS License Manager (BLM) and decreased the time needed for some license checkouts. This article explains the features and benefits of both.

Licensing 101

The BASIS License Manager (BLM) issues different kinds of licenses for different tasks. A user working on a display device checks out a “user-based” license, while the ODBC/JDBC driver checks out a “connection-based” license.



When BBjServices requires a user-based license, it looks to see if that user is already holding a license. If that user’s application invokes a secondary BBj session using SCALL (“bbj”), then the new bbj session uses the same license as the main session used. If the user is starting their first BBj session, then BBjServices requests a new user-based license from the BLM.

When BBjServices needs a new connection-based license for an ODBC/JDBC connection, for example, it must *always* request the new license from the BLM.

License Pooling and Lingering

BBj 4.01 introduced two enhancements to reduce the overhead of checking out licenses:

- License lingering for all licenses
- License pooling for connection-based licenses

License Lingering

License lingering applies to all licenses. It speeds up subsequent checkouts by keeping the socket connections to the BLM open for a short period after BBjServices checks in the licenses. If BBjServices checks out the same license again during this time, it reuses the same socket connection. Once the linger time has expired, the connection closes. The license lingering feature speeds up the check out process of recently used licenses.

Connection-Based License Pooling

License pooling means that BBjServices maintains a single socket connection to the BLM for all connection-based licenses. Checking out a new connection-based license is much faster because it is not necessary for BBjServices to open a new socket to the BLM for each new license.

Pooling and lingering can improve performance significantly in many environments. One example is a Web server that issues many ODBC commands. Prior to BBj 4.01, BBjServices required a new socket connection to the BLM for each command. Each socket remained open until BBjServices checked in the license. If the Web server made 50 ODBC requests in a short amount of time, BBjServices needed to open and close 50 BLM connections.

In BBj 4.01 and above, BBjServices pools these requests through a single socket connection. The first request establishes a new connection and all subsequent requests re-use that same BLM connection. When BBjServices checks in all connection-based licenses and license lingering expires, it closes the pooled socket.

How Do I Use These Features?

The BLM always pools connection-based licenses; the application does not have to do anything special to implement this feature. The duration of a lingering license is configurable in BBjServices. If the `basis.license.linger=120` property exists in the `BBj.properties` file, BBjServices sets the license lingering duration to 120 seconds. If the property is not present in the file, the lingering duration defaults to 60 seconds.

Summary

Lingering and pooling can improve overall performance significantly in many BBj environments. The best news is that these BBj features provide a great impact on improving application performance with no user intervention. 



Mark Schnedar
Software Engineer