

IBM Toolbox for Java

What's New in V5R2?



Robb Wiedrich

New functionality gives users more power and flexibility

By Robb Wiedrich



As I mentioned in my article “What’s in the IBM Toolbox for Java V5R1?”, IBM Toolbox for Java is a set of Java classes that allow you to access iSeries data through a Java program. With these classes, you can write client/server applications, applets, and servlets that work with data on your iSeries. You can also run Java applications that use the IBM Toolbox for Java on the iSeries Java Virtual Machine (JVM).

In V5R2, the IBM Toolbox for Java has added functionality that gives users more power and flexibility. The new functions being delivered with V5R2 are designated as Modification 5. The major new IBM Toolbox for Java

components for V5R2 include the iSeries System Debugger (a debugging environment for ILE, Java, C, and C++ programs that run on an iSeries server) and the following micro classes (com.ibm.as400.micro package):

- MEServer
- AS400
- CommandCall
- DataQueue
- ProgramCall
- JdbcMe

The Toolbox also features these new classes:

- **ClusteredHashTable** – Share and replicate nonpersistent data among nodes in a cluster

- **CommandPrompter** – Prompt for parameters on a given command
- **RecordFormatDocument** – Use the new Record Format Markup Language (RFML) classes to specify record formats and create, read, and write data records

Along with new functions, the IBM Toolbox for Java now provides support for the following:

- The JDBC 3.0 specification, including savepoints, parameter meta data, and improved blob and clob support
- The Java Generic Security Service (JGSS) framework
- The Java Secure Socket Extension (JSSE) framework

Toolbox Modification	Ships with OS/400	LPP	Installs on OS/400	Connects Back to OS/400
Mod 0*	V4R2	5763-JC1 V3R2M0	V3R2 and above	V3R2 and above
Mod 1*	V4R3	5763-JC1 V3R2M1	V3R2 and above	V3R2 and above
Mod 2*	V4R4	5769-JC1 V4R2M0	V4R2 and above	V4R2 and above
Mod 3	V4R5	5769-JC1 V4R5M0	V4R3 and above	V4R2 and above
Mod 4	V5R1	5722-JC1 V5R1M0	V4R4 and above	V4R3 and above
Mod 5	V5R2	5722-JC1 V5R2M0	V4R5 and above	V4R3 and above

* These releases are no longer supported.

Figure 1. Be sure the versions and release levels of the IBM Toolbox for Java and OS/400 are compatible.



Figure 2. Screen shots of an iSeries application rendered on a wireless device emulator.

If you intend to run a Java program that uses IBM Toolbox for Java classes on the iSeries JVM, you must run the IBM Toolbox for Java at a version and release level that's compatible with the version and release of OS/400 that is running on your system. OS/400 ships with the parts of IBM Toolbox for Java that are needed to improve performance when your application is running on the iSeries JVM. Use the chart in **Figure 1** to ensure compatibility.

Micro Classes

The micro classes that are provided as part of IBM Toolbox for Java 2 Micro Edition (ToolboxME) allow you to access iSeries data from a wireless device like a cell phone or PDA. ToolboxME focuses its attention on iSeries access to commands, programs, data queues,

and databases. The ToolboxME classes provide only a subset of the function that is available in the IBM Toolbox for Java because the wireless devices are very small and resource constrained.

(From the TUG Web site, you can download a sample ToolboxME application using the AS400, CommandCall, ProgramCall, and DataQueue classes.) **Figure 2** shows the screen shots of what the application looks like in a typical wireless device emulator.

You can also download a sample ToolboxME application using the JdbcMe classes. **Figure 3** shows the resulting screen shots of what the application looks like in different wireless device emulators.

iSeries System Debugger

The iSeries System Debugger consists of the following components:

- A client-based Debug Manager
- A client-based System Debugger
- A host-based Debug Hub
- A host-based Debug Server

Debug Manager

The Debug Manager registers the client with the Debug Hub, which enables the graphical debugging mode for the selected systems. Once the client is registered, a user can issue the Start Debug (STRDBG) CL command from an emulation session to start the System Debugger.

Use the Debug Manager to manage your debugging operations and connections:

- Add and remove systems



Figure 3. The way the application appears depends upon the wireless device emulator.

- Add and remove users
- Start debugging operations
- Launch the System Debugger

System Debugger

The System Debugger is used to debug programs that run on the iSeries server. You can debug programs that are running in existing jobs on the system or use the System Debugger to launch and then debug programs in a system batch job. You can set up the System Debugger to start automatically, start manually from a workstation command prompt, or start by using the Debug Manager interface.

Use the System Debugger to perform debugging activities, including the following:

- Set breakpoints
- Step through programs
- Inspect variables
- Examine the call stack
- Examine thread activity

Debug Hub

The Debug Hub serves as a registry for clients that want to use the System Debugger. It also handles incoming requests to start debug servers.

Use the Debug Manager interface to register your client with the Debug Hub. Registering a client stores both the user information and the TCP/IP address of the client in the registry. Using the STRDBG command from an emulation session contacts the Debug Hub to see if the user executing the command is registered with the Debug Manager. It also checks to see if the command being executed is from the same TCP/IP address as the Debug Manager. When these qualifications are met, the graphical iSeries System Debug application is started instead of the traditional debug environment.

The Debug Hub also serves as a single point of contact for all iSeries System Debugging applications. When an iSeries System Debugger performs a start debug operation, the Debug Hub submits a Debug Server job on the user's behalf and passes it the associated TCP/IP connection.

Debug Server

The Debug Server is a TCP/IP server that is started by the Debug Hub when the System Debugger issues a request to start debugging. The server job then services the job that is being debugged and issues the appropriate debugger APIs and commands.

ClusteredHashTable Classes

The ClusteredHashTable classes enable your Java programs to use highly available clustered hash tables to share and replicate data to non-persistent storage among the nodes in a cluster. They provide methods that enable you to perform the following actions:

- Open a connection to the clustered hash table server job
- Generate a unique key to store data into the clustered hash table
- Close the active connection to the clustered hash table server job
- Get an entry from the clustered hash table
- Store an entry in the clustered hash table
- Get a list of entries from the clustered hash table for all user profiles

The example in **Figure 6** operates on clustered hash table server named CHTSVR01.

CommandPrompter Classes

The CommandPrompter class prompts for the parameter on a given command. The CommandPrompter offers functionality that is similar to the iSeries CL command prompt (pressing F4).

RecordFormatDocument Classes

The Record Format Markup Language (RFML) is an XML extension for specifying record formats. The IBM Toolbox for Java RFML component enables your Java applications to use RFML documents to specify and manipulate fields within certain kinds of records.

RFML documents, called RFML source files, represent a useful subset of the data description specifications (DDS) data types defined for physical and logical files on iSeries systems. You can use RFML documents to manage the information in file records, data queue entries, user spaces, and arbitrary data buffers.



JTOpen--Open-Source Toolbox

The IBM Toolbox for Java has joined the ranks of the open-source community with

JTOpen, the open-source Toolbox for Java and IBM's first iSeries product to go open source. JTOpen currently uses the Mod 5 (V5R2) Toolbox code base, which runs on a V4R5 or higher iSeries and any client or server with a JVM. The IBM Toolbox for Java will continue to ship with each release of the iSeries, but with the JTOpen version, users can get up-to-the minute improvements.

There are a number of reasons why open sourcing the Toolbox is beneficial:

- Ability to obtain new functions and features from the Toolbox user community
- Ability to respond to customer and Business Partner requirements rapidly
- Improve the ability for our customers to build and debug their own applications when using the Toolbox functions

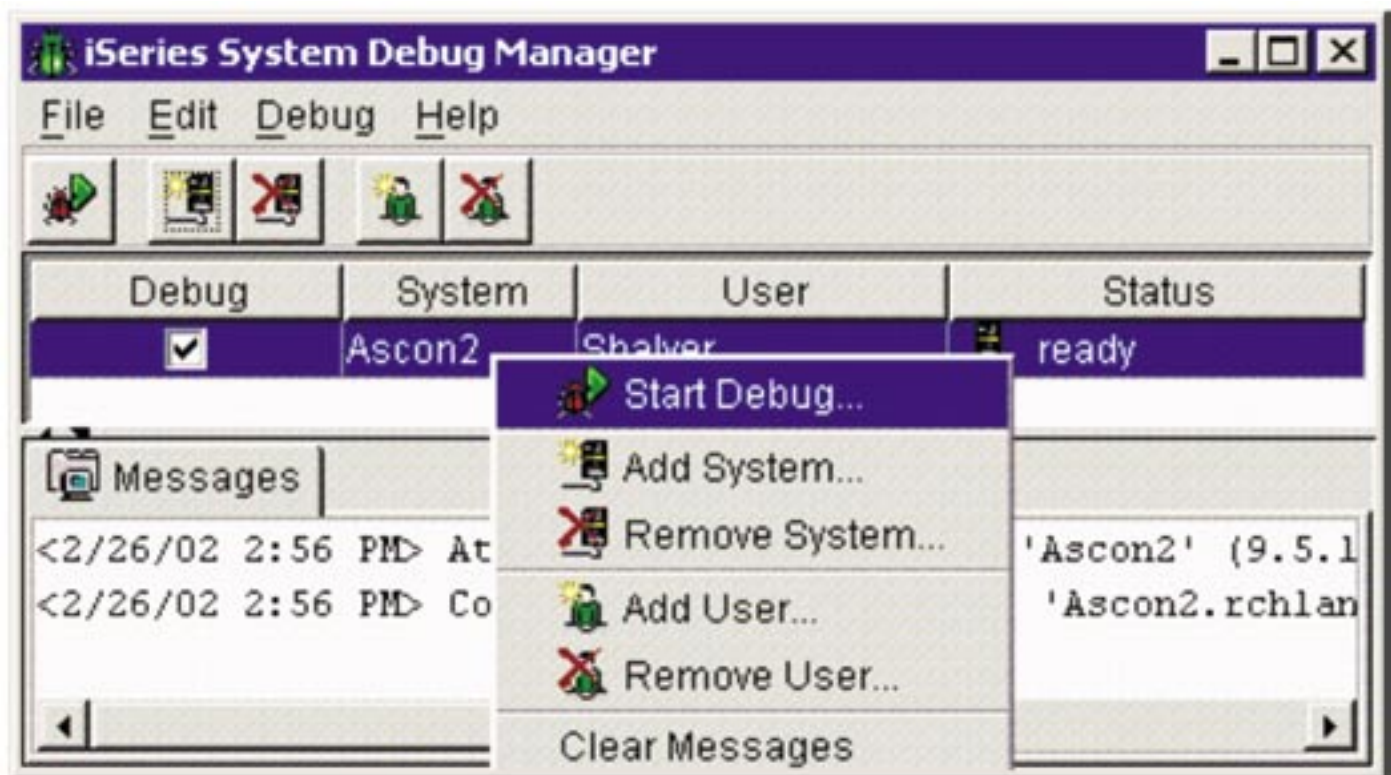


Figure 4. The graphical Debug Manager manages your debugging operations.

- Continue the drive to keep iSeries a leader in Java technology and application development

JTOpen is available from the Concurrent Versioning System (CVS) repository, a system that allows anybody to work simultaneously on groups of files. A list of guidelines and contacts is posted for those who want to participate.

The JTOpen Core Team manages the project and currently consists of three IBM employees and two non-IBM members (including MC Press author Joe Pluta) who ensure that developer contributions are of good quality and will be useful to the community. New functionality that is committed to JTOpen will not necessarily be added into the Toolbox LPP. However, new function that exists in future releases of

the Toolbox LPP may also get submitted to the JTOpen repository, at IBM's discretion. The result is that JTOpen will become a superset of the Toolbox LPP.

JTOpen 3.1, which is the latest release, has the following enhancements:

- A new build process using Apache Ant makes it easier for developers to build the JTOpen jar files themselves.
- The utilities package is now included in the jt400.jar.
- New utilities are being packaged in the com.ibm.as400.util package, including the new AS400ClassPathOptimize, which is a tool used to run CRTJVAPGM on various jars and classes in your CLASSPATH when running on OS/400.

- The CommandHelpRetriever, which is available in the com.ibm.as400.util package, generates HTML help text for any specified CL command(s) on the fly.
- There are new access classes for CommandList, Command, PanelGroup, DirectoryEntryList, and DirectoryEntry.
- Performance improvements have been added to the list-processing classes (JobList, UserList, MessageQueue, JobLog).
- New CCSIDs are now supported, including the popular CCSID 1153 (Latin-2 with Euro) along with others in the 1100 range. Also, support for CCSID 4971 was added to accommodate the new GB18030 Chinese codepage.

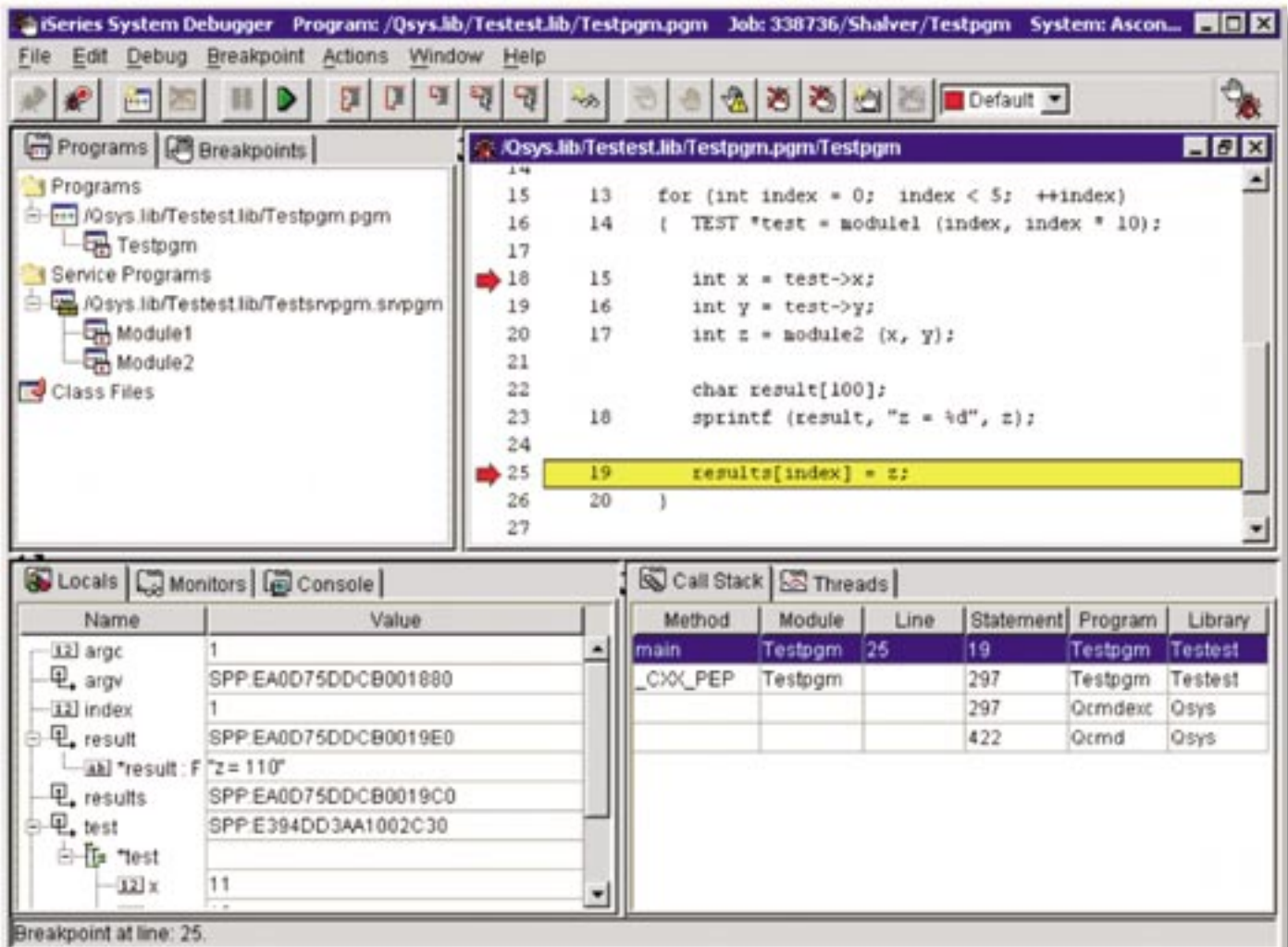


Figure 5. The System Debugger performs your debugging activities.

```

ClusteredHashTableEntry myEntry = null;

String myData = new String("This is my data.");
System.out.println("Data to be stored: " +
myData);

AS400 system = new AS400();

ClusteredHashTable cht = new ClusteredHashTable
(system,"CHTSVR01");

// Open a connection.
cht.open();

// Get a key to the hash table.
byte[] key = null;
key = cht.generateKey();

// Prepare some data that you want to store into
the hash table.
// ENTRY_AUTHORITY_ANY_USER means that any user
can access the
// entry in the clustered hash table.
// DUPLICATE_KEY_FAIL means that if the
specified key already exists,
// the ClusteredHashTable.put() request will
not succeed.
int timeToLive = 500;
myEntry = new ClusteredHashTableEntry(key,myDat
a.getBytes(),timeToLive,
ClusteredHashTableEntry.ENTRY_AUTHORITY_
ANY_USER,
ClusteredHashTableEntry.DUPLICATE_KEY_
FAIL);

// Store (or put) the entry into the hash table.
cht.put(myEntry);

// Get an entry from the hash table.
ClusteredHashTableEntry output = cht.get(key);

// Close the connection.
cht.close();

```



Figure 6. Sample code referencing clustered hash table.

The IBM Toolbox for Java in V5R2 has given developers unprecedented flexibility for building iSeries Java applications, applets, or servlets. And now, with the advent of JTOpen, the Toolbox looks to draw on the knowledge of the Toolbox user community. Visit the JTOpen site and join other developers in making the Toolbox the most powerful client/server Java package available.



Robb Wiedrich is the Toolbox team leader at IBM Rochester. He has spent the last five years working on the IBM Toolbox for Java development team. Robb can be reached at wiedrich@us.ibm.com.

Upcoming 2003 IBM Technical Conferences

Content Management & WebSphere Portal Technical Conference
June 9-11, 2003 *
Orlando, Florida

IBM eServer iSeries Technical Conferences
June 23-27, 2003 *
Las Vegas, Nevada
October 13-17, 2003 *
Orlando, Florida

Storage and Storage Networking Symposium
July 28-August 1, 2003 *
Chicago, Illinois

Networking Solutions Technical Conference
August 25-29, 2003 *
Miami Beach, Florida

SecureWorld: The IBM End-to-End Security Conference
August 25-29, 2003 *
Miami Beach, Florida

IBM eServer xSeries Technical Conference
September 8-12, 2003 *
Orlando, Florida

IMS Technical Conference
September 15-18, 2003 *
Las Vegas, Nevada

IBM eServer pSeries (RS/6000) and Linux Technical University
September 22-26, 2003 *
Miami Beach, Florida

IBM Data Management Technical Conference
October 27-31, 2003 *
Las Vegas, Nevada

Check the IBM Training Web site for more info:
www.ibm.com/training/conf