IMS Resource Adapter: Web-Enabling IMS

By Dorothy Lin, Leilei Li, and Jack Yuan

As companies today build IT infrastructures that are oriented to Web-enabled applications, the ability to integrate with and access information associated with an Enterprise Information System (EIS) such as IMS becomes increasingly important. These legacy systems contain data critical to successful day-to-day operations; as the trend moves toward a Web Services world, it's necessary to Web-enable these systems and their data to integrate them with new, emerging technologies. The advent of business-to-business communication has the IMS resource adapter playing an integral role in achieving open integrated solutions for IMS.

IMS Resource Adapter

A resource adapter allows seamless integration and connectivity between an EIS such as IMS and Java 2 Enterprise Edition (J2EE) application components and application servers. The resource adapter lets the EIS system and J2EE application communicate. Specifically, the IMS resource adapter, also called IMS Connector for Java, which is based on J2EE Connector Architecture (JCA), is used by Java applications and application servers to access IMS transactions running on host IMS systems. The IMS resource adapter can plug into an application server that supports J2EE, such as WebSphere Application Server (WAS), and access an IMS system, reducing both development time and costs and providing customers with a scalable, available, interoperable solution.

The IMS resource adapter is stored in a Resource Adapter Archive (RAR) file. Similar to a .zip file, a RAR file is a compression of one or several files and is used to package resource adapter interfaces and implementation classes for use by J2EE applications in application servers.

Like all resource adapters, the IMS resource adapter abides by the JCA standard guidelines known as system-level contracts. These contracts determine how a system external to the J2EE platform (such as IMS) can integrate with the J2EE platform. The IMS resource adapter supports basic functions such as:

- Connection management, which lets application components and application servers pool connections to IMS
- Transaction management, which lets an application manage and perform transactional access across one or many IMS and DB2 subsystems
- Security, which provides support for secure access to IMS.

The IMS resource adapter provides a standard Application Program Interface (API) through which a client application accesses IMS. Java applications that submit transactions to IMS primarily use the IMS resource adapter. The IMS resource adapter can be used in a development environment and at run-time.

In the development process, C, COBOL, or MFS definitions of IMS transaction input and output messages are mapped to Java data structures and used to generate Java applications. These applications, which can be packaged as an Enterprise Archive (EAR) file and built in an application development environment, are then deployed to run on an application server. An EAR file is a zipped archive file that contains standard Java archive files such as J2EE application components and Enterprise JavaBeans (EJBs). The IMS resource adapter is included in WebSphere Studio Application Developer Integration Edition and in Rational Application Developer for use in the development of Java applications. Figure 1 illustrates the use of the IMS resource adapter during development.

At run-time, the IMS resource adapter can be deployed to a J2EE server (e.g., WAS) for use by J2EE applications. The WAS is available on distributed platforms and on a z/OS platform. When a Java application runs, it submits a transaction request to IMS through the IMS resource adapter to the IMS component, IMS Connect. The IMS resource adapter communicates with IMS Connect through TCP/IP. IMS Connect then sends the transaction request to IMS Open Transaction Manager Access (OTMA) using Cross-system Coupling Facility (XCF), and the transaction runs in IMS. The response is returned to the Java application using the same path.

The IMS resource adapter run-time component and IMS Connect are part of the IMS Version 9 product. Figure 2 illustrates the use of the IMS resource adapter at run-time.

IMS Connector for Java

The IMS resource adapter provides many key functions; IMS Connector for Java supports the following features:

- Connection factories for managed and non-managed applications (A connection factory is a uniquely named connection with predefined attributes and is used by application components to connect to an underlying EIS.)
- Connection pooling and reuse of connections
- Transparent connection retry for certain communication exceptions on TCP/IP connections
- Execution timeout, which is the maximum amount of time allowed for IMS Connect to send a message to IMS and receive a response from IMS; the value can be set on a per-interaction basis
- Global transaction support (two-phase commit), which allows applications to coordinate multiple resources such as IMS and DB2 into a single unit of work.
- Component-managed and container-managed EIS sign-on
- Runs thread identity for WAS on z/OS
- Secure Socket Layer (SSL) communication between IMS Connector for Java and IMS Connect
- Shareable and dedicated persistent socket connection with IMS Connect, including when execution timeout occurs
- Commit mode 1 and commit mode 0 IMS transactions, and the retrieval of queued output messages from undelivered commit mode 0 IMS transactions and output to alternate Pool Control Blocks (PCBs)
- Send-only interactions with IMS, where Java applications can run response mode or non-response mode IMS transactions using the value SYNC_SEND for the IMSInteractionVerb property
- Socket timeout, which is the maximum amount of time allowed for IMS Connector for Java to send a message to IMS through IMS Connect and receive a response from IMS Connect
- IMS conversational transaction processing
- Execution of Java applications built from MFS descriptions of input and output messages.

Sysplex Distributor Environment

Today, IBM z/OS operating system customers increasingly depend on the z/OS Sysplex Distributor (SD) to connect front-end client applications to IMS using multiple TCP/IP gateways on multiple servers in a Sysplex environment. SD is a sophisticated connection dispatching technology among the z/OS IP servers. With SD, Java applications using the IMS resource adapter receive the benefits of high availability and workload distribution. The IMS resource adapter can take advantage of the SD environment and use the Workload Manager to access IMS and increase the availability and scalability of transactions. Figure 3 illustrates a client ATM application in a Sysplex environment where there are multiple WAS and each WAS has multiple Servant Regions (SR). Inside each SR, there's an IMS resource adapter that processes transactions in parallel.

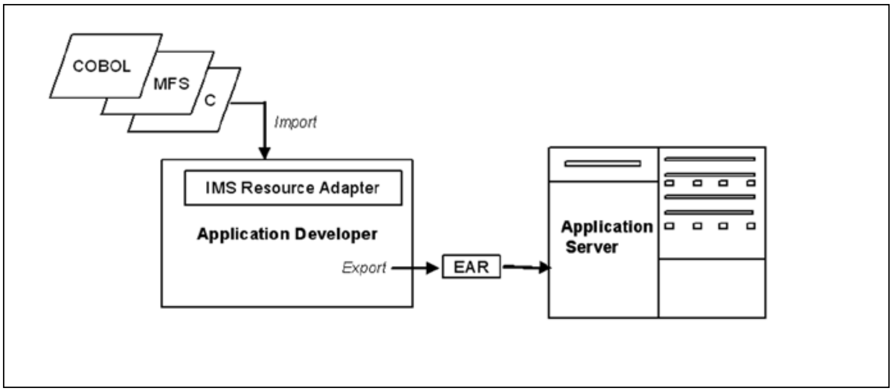

Figure 1: The IMS Resource Adapter During Development

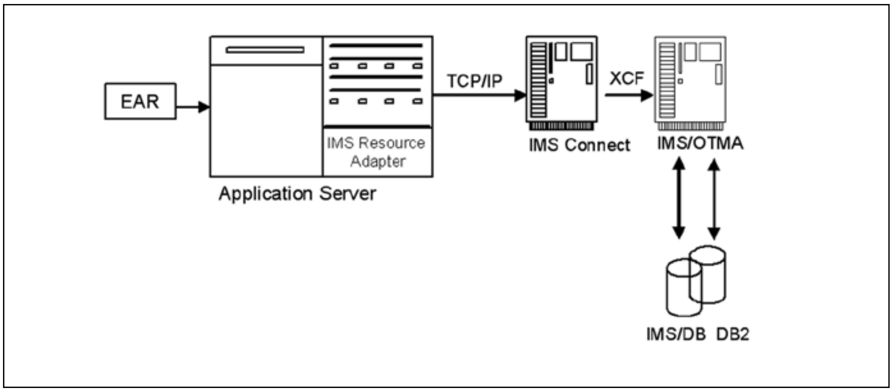

Figure 2: The IMS Resource Adapter at Run-Time

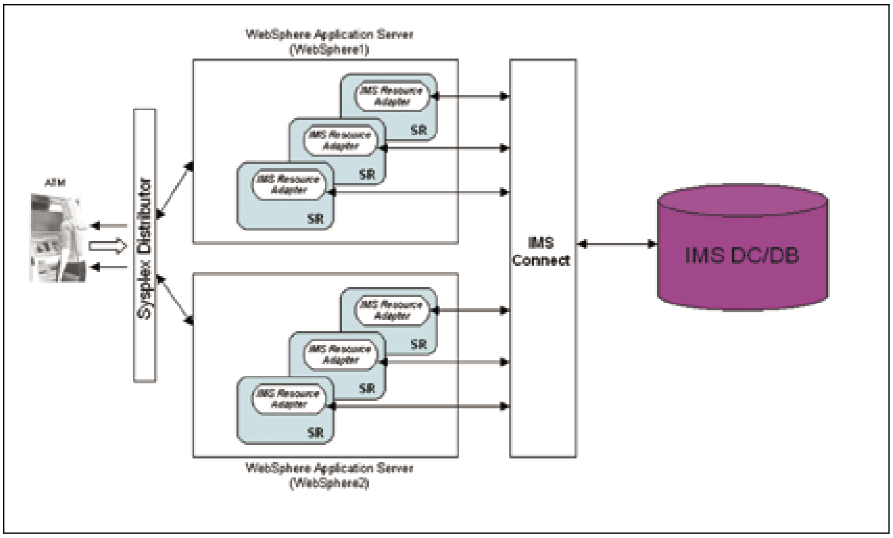
WebSphere Application Server (WebSphere1)
IMS Resource Adapter
SR
ATM
Sysplex Distributor
IMS Connect
IMS DC/DB
WebSphere Application Server (WebSphere2)

Figure 3: A Client ATM Application in a Sysplex Environment With Multiple WAS and SR

The IMS resource adapter supports shareable persistent socket connections. A shareable persistent socket connection can be used to support the Sysplex distributor configuration. The shareable persistent socket connection supports many types of interaction verbs, including:

- SYNC_SEND for sending a response mode or non-response mode interaction
- SYNC_SEND_RECEIVE for sending an input request and receiving an output response from IMS
- SYNC_RECEIVE_ASYNCOUTPUT_SINGLE_NOWAIT for retrieving one existing asynchronous output message
- SYNC_RECEIVE_ASYNCOUTPUT_SINGLE_WAIT for retrieving one asynchronous output message. If there's no output message at the moment of the request, wait for it until the execution timeout occurs.

Both the SYNC_RECEIVE_ASYNCOUTPUT_SINGLE_NOWAIT and SYNC_RECEIVE_ASYNCOUTPUT_SINGLE_WAIT verbs currently need to be run within the same application using the same socket connection as the commit mode 0 SYNC_SEND_RECEIVE interaction that queued the output.

Another new feature is the option to control whether undelivered output for IMS commit mode 0 interactions on shareable persistent socket connections is queued or discarded. The options to queue or discard the undelivered output are:

- **Purge not deliverable:** This option is controlled by the purgeAsyncOutput property of the IMS resource adapter. When this option is set, it informs IMS Connect to purge the IMS commit mode 0 I/O PCB output response if IMS Connect can't deliver the response to the IMS resource adapter. Also, this option informs IMS to purge the commit mode 0 I/O PCB output response if IMS cannot deliver it to IMS Connect. Purging the undeliverable response ensures that there are no queued messages in the IMS transaction pipes (TPIPE) so the TPIPE storage can be freed during IMS checkpoints and, ultimately save system resources. This option is especially useful for query types of IMS transactions.
- **Reroute not deliverable:** This option provides a destination name for unde-

livered output for commit mode 0 interactions on shareable persistent socket connections. This option is controlled by the reRoute flag and reRouteName properties on the IMS resource adapter. When this option is set, IMS Connect reroutes an undeliverable I/O PCB output response to a special IMS queue with the name as the value defined in the reRouteName property so the output can be retrieved later.

Summary

The IMS resource adapter provides many useful functions and features and is invaluable as it allows J2EE client application components and servers to seamlessly integrate and access IMS. With the latest features and additions such as support of shareable persistent socket connections, the IMS resource adapter has increased the availability and scalability of IMS. The modernization of an enterprise information system such as IMS on z/OS strategically relies on the IMS resource adapter.

About the Authors

Dorothy Lin was the lead technical writer for IMS Connector for Java and is currently a project manager for DB2 for z/OS at IBM in Silicon Valley. She holds a master's degree from the University of Washington, Seattle in Technical Communications and has written and contributed to numerous IMS manuals and IMS Connector for Java online documentation.
e-Mail: dclin@us.ibm.com

Leilei Li is a key developer of IMS Connector for Java. She holds a master's degree from New York Institute of Technology and has six years of software development experience with IBM in many areas including Lotus Domino and IMS. Originally from Beijing, China, she now resides in San Jose, CA.
e-Mail: leilei@us.ibm.com

Jack Yuan is a senior software engineer at IBM's Silicon Valley Laboratory in San Jose, CA. He holds a master's degree in Computer Science Engineering from San Jose State University and has nine years of experience in the IMS data communication field.
e-Mail: jackyuan@us.ibm.com
<<author photos>>

Callout: The IMS resource adapter supports shareable persistent socket connections.