IMS Connect Strategy: Providing IMS Connectivity for a Millenium

Background

Access to IMS Applications

Traditionally messages have come into IMS through its SNA data communication protocol from VTAM. With the introduction of APPC/IMS support in IMS Version 4, the extended need to communicate with other software subsystems, and to be able to do so across the Sysplex, IMS took advantage of the new MVS Cross Coupling facility (XCF) for communications with APPC/MVS. XCF is a software facility that allows MVS subsystems to communicate more efficiently and to do so across the Sysplex. With the IMS Version 5 Open Transaction Management Access (OTMA) facility, IMS extended its use of XCF for use by other IBM subsystems, such as IMS Connect and MQSeries, providing them more efficient and richer capabilities in accessing IMS. OTMA allows access to existing, unchanged IMS applications on any IMS TM system on any MVS system of an MVS Sysplex. Using the OTMA interface, IMS Connect provides high performance access to IMS.

Access to IMS Databases

In addition to providing the IMS OTMA interface for IMS application access, IMS provided the Open Database Access (ODBA) facility, for easier IMS database access from other OS/390 or z/OS subsystems. IMS has also more recently provided Java application support with JDBC access to IMS DB. Building on the IMS ODBA interface is the industry standard Java Database Connectivity (JDBC) access to IMS DB. JDBC access is now provided from IMS TM Java applications, CICS Java applications, DB2 Java stored procedures, and WebSphere enterprise Java beans. Currently Java applications in these environments can access IMS DB data locally from within the S/390 or z/Series environment. IMS Connect requirements include providing a distributed IMS Database access facility.

Access to IMS Operations

IMS Version 8 Single-Image Operations Manager provides the framework for a single point of control for IMS, supporting IMS commands and responses and introducing some new IMS Sysplex commands. It provides an Application Programming Interface (API) which allows commands to be entered from a single point of control (SPOC) to one or more IMS systems and consolidated command responses to be returned. This API allows a user or vendor to write tools to automate IMS operations. It simplifies operations by supporting a single point of control to present a single system image for the IMS Sysplex by allowing the user to enter commands to all IMSs in the IMS Sysplex from a single console. A single point of control application program is also provided to help manage a group of IMSs in an IMS Sysplex from a TSO SPOC, issuing commands to all the IMS subsystems in an IMS Sysplex, displaying consolidated responses from these commands, and sending a message to an IMS terminal that is connected to any IMS in the IMS Sysplex. This would offer less complexity in managing multiple IMSs in a Sysplex environment. More recently a Control Center is being provided to offer IMS operations from a distributed DB2 environment through IMS Connect.

IMS Connect Strategy: Providing IMS Connectivity for a Millenium

IMS Connect V1R1

IMS Connect V1R1 was provided with IMS Version 7 General Availability October 2000, as a new separately priced product which runs in OS/390 or z/OS for IMS application access. Initial support, over the existing prototype IMS TCP/IP OTMA Connection (IMS TOC) solution, provided enhanced IMS TCP/IP support, SMP installation support, dump and trace formatting improvements for increased serviceability, user exits, asynchronous output support for enhanced usability, and persistent sockets and bottleneck relief for improved performance.

IMS Connect can be used by Java applications created by VisualAge for Java or WebSphere Studio Application Development Integration Edition IMS Connector for Java. These Java applications access IMS transactions through IMS Connect from WebSphere Application Servers for zOS and OS/390. WebSphere Studio Application Developer Integration Edition, which initially GA'd in March 2002, is IBM's strategic product for e-business application development, specifically targeted for a WebSphere Application Server (WAS) runtime environment.

IMS Connect V1R1 additional enhancements in 2001, provided through the service process. included:

- Local support, available via APAR PQ45057 4/01, for communication using Program Calls without requiring TCP/IP from a webserving application to IMS in a z/OS or OS/390 environment, easing the management in this environment.
- Unicode support, available via APAR PQ47906 5/01, for sending Unicode application data to an IMS host application capable of dealing with Unicode, such as a Java application running in IMS.
- ACK/NAK required notification support, available via APAR PQ46195 4/01, provides client notification that an ACK or NAK response is required by the client without additional testing of data received. This notification will be sent to the Client in the CSM and RSM.
- Output message structure change, available via APAR PQ48182 5/01, to include the full message length preceding the output message to the client, reducing the design and coding effort of a client application.

IMS has been exploiting the latest programming technologies for the Internet and Java. This includes enablement of interactive and multimedia applications in a simplified fashion. With Java, users can transparently download and seamlessly run applications. It is becoming widely used and is platform independent.

IMS Connect V1R2

IBM initially provided a common connector framework and subsequently, with IMS Connect V1R2 in November 2001, supported the industry standard J2EE with a set of common, consistent Java interfaces that connectors for any subsystem can and are implementing, making it easier for programmers not to worry about the differences between those subsystems. For such, IMS has provided IMS Connector for Java development code, initially in VisualAge for Java and subsequently in WebSphere Studio Application Developer Integration Edition, to develop Java applications running under WebSphere that access IMS applications through IMS Connect. IMS has provided IMS Connector for Java runtime code in IMS Connect for access to IMS

applications and data. By being compliant with the connector standards, IMS Connector for Java can be used in any component server environment that supports the standard. The IMS Connector for Java J2EE Runtime piece ships with IMS Connect for download to the WebSphere platform for deployment in connecting to IMS transactions via IMS Connect. The runtime environment of a Java Servlet Application can run under OS/390, z/OS, Linux or another environment. When a user executes the application and provides the appropriate input data, IMS Connector for Java within the Java Application establishes a connection with IMS Connect through a TCP/IP (or Local Option, if in the same LPAR as Connect) connection. IMS Connect, running under OS/390 or z/OS can run in the same or separate LPAR from IMS. It will forward the transaction request to the IMS application through the IMS OTMA (Open Transaction Manager Access) interface, using MVS Cross-system Coupling Facility (XCF). The IMS application could be written in Java or another language. IMS Connect would also send the output back to the IMS Connector for Java application in a similar manner. In the servlet runtime environment, the user can invoke the HTML page using the web browser and put in the input data. The request will be sent to webserver and the corresponding servlet will be invoked by the WebSphere application server. The servlet will then use the IMS Connector for Java to establish and connection with IMS and invoke the requested transaction with the input data through IMS Connect. The output result will be handed back to the IMS Connector for Java in the servlet via IMS Connect and send to the output HTML page by the web server. Development on NT can be deployed in any WebSphere environment.
In addition to the IMS Connector for Java, provided with IMS Connect for runtime access to Java applications, IMS Connect includes samples for other language access as well.

IMS Connect V1R2 provided IMS Connector for Java J2EE runtime support for these new applications accessing IMS transactions. Additional V1R2 enhancements are provided in 2002 through the service process:

- Two phase commit support from OS/390 and z/OS environments across the Local option. Initial support requires WebSphere, IMS Connect and IMS in a single LPAR.
- Support for the new IPV6 network protocol for scalability and simplification.
- Support for the new IMS V8 Operations Manager, through the IMS Structured Call interface, for distributed operations using the GUI frontend of DB2 V8 Control Center to greatly ease IMS operations . The IMS Control Center capability, integrated into DB2 UDB V8, allows a single user interface to control both IMS and DB2. The DB2 administration tools include a Control Center for navigating IMS Sysplex systems, wizards for creating the new IMS V8 Operations Manager commands, a results window for sorting and filtering single-system image command results, and a Command Center for typing and executing both the new commands and the old operations commands. IMS Connect provides for communication to the IMS V8 Operations Manager from this distributed Control Center.
- Additional Security Items:
  - Passticket support
  - Need for time out granularity at a message
  - Relief of limitations on the use of the User Message exits with the limit moving from 15 to 254 user exits

IMS Connect Strategy: Providing IMS Connectivity for a Millenium

- The IMS Connector for Java has also been providing security enhancements with Container-managed sign-on support, initially for local option environments, and later for distributed environments

Supported Environments

IMS Connect runs under OS/390 V2R10 or later, and z/OS environments. IPv6 support, however, requires z/OS 1.4.

Base IMS Connect TCP/IP, or Local, support requires the IMS Version 6 Transaction Manager, allowing customers to begin to take advantage of IMS Connect before migrating their networks, applications, and/or databases to IMS Version 7. IMS V7 Transaction Manager is, however, required for use of asynchronous output and the additional enhancements to IMS Connect. Distributed IMS Operations Control using IMS Connect requires either the IMS V8 Database Manager or the IMS V8 Transaction Manager, thus making IMS Connect for this function useful to IMS DB customers.

Performance

In web computing the system must match capacity to business requirements on an as-needed basis and provide an easy growth path, minimize downtime and provide quick return on investment. These are available with IMS and IMS Connect along with z/OS and the S/390. Recent enhancements have included improvements to the processing of requests, yielding overall increases in throughput. Performance improvements, higher bandwidth networking, and numerous other enhancements continue to make IMS and the S/390 a powerful, flexible system for growth in web serving as well as the rest of mission critical work. IMS solutions exploit the security, performance, and other facilities of the S/390 to optimize performance. Testing with IMS Connect has demonstrated very high transactions rates. Early IMS Connect performance data resulted in nearly 6000 trans/second with a single IMS and single IMS Connect. Early performance work on IMS Connect shows potential for even more growth in this transaction rate as well. In addition one IMS Connect can talk to many IMSs across the Sysplex and any number of IMS Connects can talk to each IMS, thus providing significant potential for even greater growth and availability considerations.

What's next:

IMS Connect and other IBM Tools are taking advantage of XML, critical for future transparent application integration. Today, IMS documents can be processed in new IMS Java and C++ Applications, through use of the XML Parser, and/or access existing IMS applications using MQSeries. We are also making available IMS COBOL and PL/I XML Application Capability for parsing, using the IBM Enterprise COBOL and PL/I compilers, which allows you to develop new or modify existing IMS applications using XML support for COBOL and PL/I. This can be used to enhance your existing high performance IMS transactions written in COBOL and PL/I in a Business-to-Business environment by receiving and sending XML documents. IMS supports the transmission of XML documents in the data portion of the IMS message. The messages can be placed and retrieved for the IMS messages queue for all messages regions for

IMS Connect Strategy: Providing IMS Connectivity for a Millenium

IMS Message Processing Programs, Fast Path Programs and Batch Message Processing Programs. With WebSphere Application Developer Integration Edition, you can also enable existing IMS COBOL and C applications as Web Services by connecting SOAP and EJBs to IMS. XML requirements are also being addressed with enhanced support for industry tooling. With the new WebSphere Enterprise Developer tooling you would be able to generate XML documents for outputs from new COBOL applications. With WebSphere Application Server, you would also be able to web enable your MFS applications using XML and you would be able to transform your MFS based IMS applications into web services.

IBM's IMS e-business team is leading the IBM cross-divisional Common Application Metamodel (CAM) project. In January 2002, the Unified Modeling Language (UML) for Enterprise Application Integration (EAI), which includes CAM, was officially adopted by the Object Management Group (OMG) board of directors as an industry standard for data exchange. OMG is the world's largest software consortium with a membership of over 800 vendors, developers, and end users. A good description of CAM is provided in the Spring 2002 IMS newsletter. CAM defines and publishes a metadata exchange standard for information about accessing enterprise applications such as CICS and IMS. Anyone who has written COBOL COPYBOOK to XML translators or who has tried to make IMS message contents discernible to Java code, knows there has just got to be a better way. CAM is that better way! Because CAM provides physical representation of data types and storage mapping to support data transformation in an EAI environment, it enables Web Services for enterprise applications. Web Services offers the next step in the evolution of the web, creating the building blocks toward a more integrated web business. Using industry standard XML protocols and packets we can evolve to common information distribution and easier access. This not only provides for Business to Consumer but also Business to Business.

CAM-based tools will open up the application development story to even more programming languages and standardize the existing ones. Enterprise accounts seeking e-business connectivity have been important adopters of WebSphere Application Server. With the CAM-based tools, the WebSphere Application Server story gets even better for IBM. Bolstering the business results are numerous upcoming projects based upon CAM technology in WebSphere Application Developer Integration Edition, WebSphere MQ Integrator, WebSphere Studio Enterprise Developer, IMS DL/I Model Utility (IMS Java tooling), and VSAM Data Model Utility.

Because of CAM and its implementation of WebSphere Studio tools for Importers, Adapters and Web Services, IMS is able to meet B2B challenges to provide a standard interface between dissimilar systems for MFS and non-MFS IMS applications.

IMS Connect Follow-on

IMS Connect Follow-on plans would include providing:

- Two-phase commit support for distributed environments across TCP/IP, providing for a WebSphere Application Server in a non-OS/390, non-z/OS environment, to connect to IMS through IMS Connect (with IMS and IMS Connect initially in the same LPAR).
- Enhanced Security Items
  - Secure Sockets Layer (SSL)
  - Capability to control the security environment
  - "Trusted user" support

IMS Connect Strategy: Providing IMS Connectivity for a Millenium

    - Auto reconnect support
    - RACF= and RACFID= keywords on the DATASTORE statement
- Message Format Services (MFS) Web Services support to enable IMS customers to publish existing MFS-based IMS applications on the Internet as Web Services.

Architecture

IMS Connect is part of the overall restructure of IMS for the 21st Century and is architected as the base for all future IMS Connectivity. The structure of IMS Connect is designed such that drivers can be interchangeable. That is, alternatives for the TCP/IP frontend or OTMA back end interfaces are already being provided (eg. the Local front end and the Operations Management back end). These are allowing IMS to exploit newer, additional, and enhanced protocols and/or interfaces.

Futures

With this structure IMS Connect could evolve to address other connectivity requirements, such as enhanced two phase commit and distributed database access to IMS DB data, and enhanced SOAP/WebServices.

For this, distributed data access to IMS DB data through ODBA, using SQL and JDBC, would be provided in a subsequent IMS Connectivity follow-on . Also provided would be SQL access to IMS data using Web Services, similar to the support provided for DB2 using its Web Services Object Framework (WORF).

The next steps for enhanced two-phase commit connectivity would be to provide distributed two-phase commit support for environments where a) IMS and IMS Connect are in separate LPARs from each other (using z/OS V1R2 Multi-cascaded Transactions support and IMS V8 Transaction Manager OTMA Synchronous Shared Queues support, enhanced through the service stream) and b) WAS on z/OS in a separate LPAR (or separate z/OS system) from IMS Connect and IMS, providing enhanced failure isolation between WebSphere on z/OS applications and IMS applications.

The next step of supporting IMS XML based on CAM would be in providing non-Java XML and Web services solutions in IMS Connect without requiring the WebSphere Application Server. Future XML requirements could address XML transformation processing within IMS Connect, support PL/I and additional languages, and possibly use XML as an IMS Data Definition language.

Summary

IMS Connect has been designed to be the distributed communications front-end to IMS for the 21st Century. It is provided as a separate entity, while being developed alongside of IMS DB and TM. This ensures that it can maintain consistency with IMS and exploit new IMS functions in an optimized, timely manner. At the same time it can provide more frequent, independent deliveries of new enhancements. And it can run independently from IMS in another LPAR, ensuring ultra-high performance and availability.