

IMS and XML: The Perfect Couple

IMS™ On Demand solutions protect and extend customer investments by enabling access to IMS transactions and data. These solutions are designed to support open, integrated technologies and modernize use of IMS transactions and data, while fully integrating with WebSphere® and other industry tools. Using a common programming model for a service oriented architecture (SOA), IMS solutions are based on standards, such as XML, SOAP, Java™ and JDBC as well as new ones as they are developed. Support of this evolving technology encourages new application development and new application developers.

For the past seven years, IMS customers have been taking advantage of XML with IMS and it's now considered critical for the transparent integration of applications and data now and in the future.

Processing XML documents in new IMS applications

Today, XML documents can be processed in new IMS C or C++ applications using the XML toolkit for z/OS®, or in new IMS Java applications using the SDK for z/S, Java 2 Technology Edition. The new IMS COBOL and PL/I XML application capability can also process XML documents using the IBM Enterprise COBOL and PL/I compilers. This allows you to develop new or modify existing IMS applications using XML support for COBOL and PL/I. By receiving and sending XML documents, this can be used to enhance your existing high-performance IMS transactions written in COBOL and PL/I in a business-to-business environment.

XML documents are transmitted in the data portion of the IMS message. The messages can be placed and retrieved for the IMS messages queue for all message regions for IMS Message Processing Programs, Fast Path Programs and Batch Message Processing Programs. You can also use the XML Converters for COBOL in WebSphere Developer for z/Series to generate and parse XML input and output from COBOL applications.

Bridging XML and existing IMS applications

IMS application support is provided for publishing as Web Services using existing IMS C, COBOL, MFS applications with connections thru SOAP and EJB bindings. You can Web-enable your MFS applications using XML, and even transform your MFS-based IMS applications into Web Services.

Existing IMS applications can also bridge to XML using MQSeries® Integrator to support IMS COBOL and C applications.

IMS now provides direct SOAP support through the IMS SOAP Gateway. This enables you to reuse IMS applications as Web Services. The WebSphere Developer for z/Series® can be used with this IMS SOAP support.

The IMS SOAP Gateway maximizes reuse of enterprise assets via standard interfaces. It provides and requests services independently of platform, environment, application language or programming model. You can use SOAP messages to support end-to-end integration between IMS transactions and Microsoft .Net & Java applications, or any third-party applications, such as SAP XI RYO applications. The IMS SOAP Gateway includes HTTP/SOAP transport and processing, using SOAP envelope and headers handled by the gateway. It uses WebSphere Developer for z/Series tooling to create converters for transforming XML messages to COBOL data and vice versa without changes to existing IMS applications. It can run on any Java supported platform, but is initially supporting Windows environments.

The IMS Soap Gateway is now generally available (GA) and can be downloaded at www.ibm.com/ims. We are also providing IMS XML adapter support soon. This will enable use of the WebSphere Developer for z generated converters for transforming XML inside of IMS itself without IMS application changes.

IMS XML database port

Another key element is XML support for storage and retrieval of XML data in IMS databases. Since XML and IMS are hierarchical, IMS is a natural fit for XML data. IMS introduced a way to view IMS data as collections of XML documents -- aligning the stored IMS records with the valid XML documents for retrieval and automated conversion between these. Now, you can use IMS to store, retrieve and query XML in IMS databases for universal information interchange. IMS also supports generating XML schema for existing IMS databases. The implementation is native IMS, not merely a mapping which is done with other technologies, giving you more efficient use of resources and top overall performance.

Existing IMS data can be easily converted to XML to facilitate integration with business processes, improve programmer productivity and reduce development lead times. XML data can be decomposed for use by applications that are not XML-enabled, thereby preserving and extending past investments and enhancing programmer productivity. Data descriptions can be the same for distributed and host environments to reduce overhead and improve data consistency and integrity.

To store and retrieve XML at runtime, DL/I model utility enhancements generate XML schema from existing IMS database definitions (DBDs) and program status blocks (PSBs). This improves application development time, reduces errors and makes it possible to consolidate skills by allowing programmers to code in an industry standard interface. Early on, IMS introduced a way to view IMS data with the DL/I model utility for IMS Java tooling support. More recent DL/I model utility enhancements allow inclusion of COBOL descriptors of IMS database segments for more granularity, and support XML in IMS databases, which includes the generation of validating XML schemas from new or existing IMS PCBs for use in XMS runtime support. This utility eases Java application development by mapping XML metadata to create Java classes, allowing parsing and creating of metadata.

This was subsequently enhanced with the DL/I model utility GUI. It provides a graphical version of the DL/I model utility, built as an Eclipse plug-in. This GUI simplifies IMS metadata generation, eases IMS Java and XML database application development, and offers a visual representation of IMS databases. Initial support of the GUI is provided as a "technology preview" on the IMS Web site. This is currently being enhanced with even more capability, to be made available very soon from the IMS Web site at ibm.com/ims.

The emerging XQuery standard is a powerful query language to search, aggregate, evaluate, pick and choose parts of the XML collection and then convert the resulting data into XML. XQuery is an SQL-like query language for hierarchically-structured data. It uses the emerging W3C standard created by IBM, Oracle, Microsoft™ and others. IBM development and research have teamed up to build a fully functional, performance-oriented XQuery implementation atop IMS. An IMS XQuery technology preview is now available on alphasworks through the IMS Web site www.ibm.com/IMS.

XML and IMS follow-on requirements

Future requirements are also planned to address enhanced support for industry tooling, additional languages, transformation and improving IMS XML database support.

With new WebSphere tooling, you would be able to generate XML documents for output from PL/I applications. You would also be able to enable IMS PL/I applications as Web Services.

Enhanced XML database support in IMS would include the ability to create new IMS XML databases. Support would also provide dynamic definition of IMS applications and databases, enabling even more rapid deployment of XML in IMS databases.

IMS continues to advance the XML and Web Services technology

IBM's IMS Development Team at the Silicon Valley Laboratory developed the Common Application Metamodel (CAM). CAM is an IBM open standard initiative for enterprise application integration (EAI). It was submitted as a proposal to the Object Management Group (OMG), the world's largest software consortium with a membership of over 800 vendors, developers, and end users. See <http://www.omg.org>. A good description of CAM is part of a document, with diagrams and illustrations, at the Web site for the Instituto Tecnologic de Informatica in Spain. Sections 6 (page 13) and 9 (page 79) are of special interest to those who have labored in the Open Transaction Management Access (OTMA) vineyards.

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, as I have, knows there has 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.

IMS is enabled for On Demand computing with XML. IMS continues to provide enhancements to enable users to take advantage of the vast amount of applications and data on IMS.

Klein is the senior IMS strategic plan and brand manager at IBM's Silicon Valley Laboratory in San Jose, California. Involved with IMS, DB2, and CICS for more than 30 years, she has held a variety of positions in planning, design, development, systems engineering and management. She is currently responsible for IMS family strategy, planning and the introduction of new IMS versions and functions.



© International Business Machines Corporation 2006.

CICS, DB2, IBM, the IBM logo, IMS, MQSeries, Tivoli, WebSphere, z/OS and zSeries are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product and service names may be trademarks or service marks of others.

All rights reserved. IBM Corporation, 100 North Sepulveda Boulevard, El Segundo, California, 90245 U.S.A.