

Solutions for IMS Connectivity

Abridged from the "IMS Connectivity in an On Demand Environment: A Practical Guide to IMS Connectivity" RedBook (SG246794) by lead author Jouko Jäntti



IMS provides a variety of solutions for providing access to IMS applications and data as Web services. Figure 1-1 shows examples of IMS connectivity solutions for Web services.

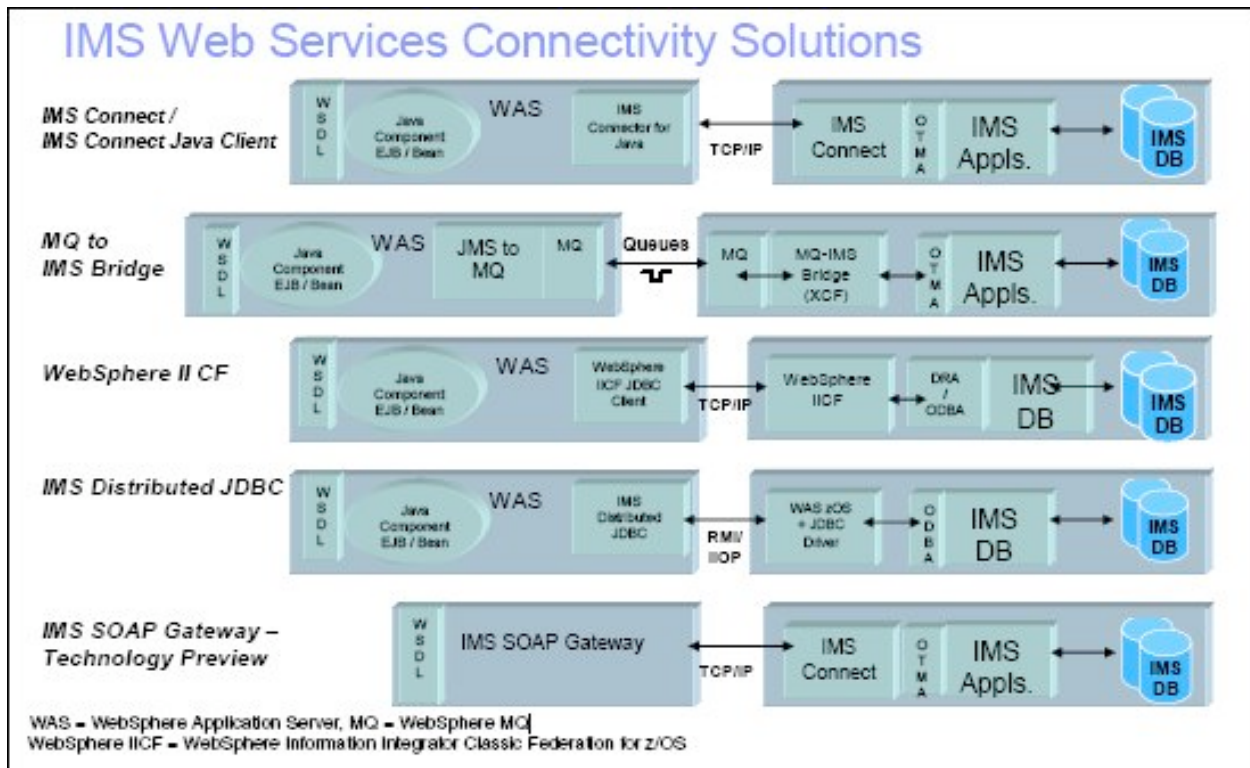


Figure 1-1 IMS connectivity solutions

When customers need rapid response for business transactions and inquiries from many locations using numerous different types of devices, the IMS Connect function provides easy-to-install, easy-to-use, high-performance transparent access to IMS applications and data from practically any application environment. Customers can use their storehouse of IMS applications from IMS Connect client applications to access their IMS and IBM DB2® data from the Internet. When customers need high-speed, transparent universal interchange of information throughout the enterprise, and with partners and customers, IMS provides the ability to store and retrieve XML data natively. IMS also converts non-XML data to XML for interchange and converts it back or stores it natively.

When customers need to deploy new Web-based applications, at the lowest possible cost, to maintain competitive advantage, IMS has state-of-the-art development tools available today to Web-enable and thereby protect customers' long-standing investment in IMS applications, data, and skills. Additionally, IMS customers can now build new applications using Java programming, with XML for universal data interchange, write and test them on workstations and integrate them with existing applications on the host. When customers need to develop a service-oriented architecture to better integrate business processes end-to-end, their IMS applications and IMS transactions can be published on the Internet as Web services, connecting to IMS through Simple Object Access Protocol (SOAP) and Enterprise JavaBeans™ (EJB™) bindings, within customers' service-oriented architecture (SOA), thereby assisting business-to-business application integration. IMS Version 9 is the IMS offering for the On Demand Operating Environment.

The integrated IMS Connect function in IMS Version 9 provides easy-to-install, easy to use, high-performance, high-volume, and secure transparent access to IMS applications and operations from any TCP/IP-supported environment. It provides commands to manage the network environment and assist with workload balancing, resulting in better resource utilization. It reduces the design and coding effort for client applications and provides easier access to IMS applications and operations, thereby improving programmer productivity. It can be used with IBM WebSphere and Rational® development tools to quickly transform static Web sites into sources of dynamic Web content, improving marketing effectiveness and customer service, and to transform IMS transactions into Web services for service-oriented architectures (SOAs), enabling quick response to new customer requirements, business opportunities, and competitive threats. It can be used with DB2 and the IMS Control Center to control both IMS and DB2 operations, improving system availability and operator productivity.

The integrated IMS Connect function of IMS Version 9 can be used to replace the separately priced IMS Connect product offered for earlier IMS versions, simplifying administration and reducing cost. IMS Connect provides one-to-any and any-to-one connectivity. Performance measurements have demonstrated more than 6000 transactions per second with a single IMS Connect instance to a single IMS. This can be greatly increased using parallel IMS Connect instances. Performance measurements have demonstrated more than 22,000 transactions per second (2 billion per day) with IMS on a zSeries Model 990 processor.

OTMA

Open Transaction Manager Access (OTMA) is a function of IMS. OTMA is a transaction-based, connectionless client/server protocol that provides an access path and an interface specification for sending and receiving transactions and data from IMS. In a Parallel Sysplex environment, different members can be on different z/OS images.

OTMA is designed to be a high-performance protocol that allows z/OS programs to access IMS applications. These z/OS applications are called OTMA clients. IMS Connect is probably the most commonly used OTMA client. Various IBM-supplied

connectivity solutions, such as IMS Connector for Java or IMS SOAP Gateway, are based on the use of IMS Connect. The IBM WebSphere MQ product also has the OTMA interface implemented, which is called MQ-IMS Bridge, and is another example of OTMA client. For those who are going to write an OTMA client of their own, there is an application programming interface called the OTMA callable interface (OTMA C/I).

ODBA

IBM IMS Open Database Access (ODBA) is a callable interface to access databases managed by the IMS Database (DB) Manager. ODBA can be used by any z/OS application program that uses the Recovery Resource Services (RRS) of z/OS as a sync point manager to access the IMS full function databases and data entry databases (DEDBs). This allows IMS DB and z/OS application programs to be developed, installed, and maintained independently of each other. ODBA also provides for failure isolation and independent resource recoverability.

Accessing IMS transaction and commands using OTMA and IMS Connect

IBM IMS Connect provides connectivity to IMS from any TCP/IP client, enabling you to access IMS resources from a TCP/IP network. IMS Connect supports TCP/IP sockets access to IMS transactions and commands.

IMS Connect provides a general purpose and structured interface for:

- _ IMS connectors
- _ User-written clients
- _ IMS Control Center

IMS Connect is an integrated feature of IMS Version 9 that provides high-performance communications for IMS between one or more TCP/IP or local z/OS clients and one or more IMS systems. IMS Connect has the following features:

- _ Provides commands to manage the communication environment
- _ Assists with workload balancing
- _ Supports multiple TCP/IP clients accessing multiple data store resources
- _ Reduces design and coding efforts for client applications
- _ Offers easier e-business access to IMS applications and operations with advanced security and transactional integrity.

IMS Connect is included as part of IMS Version 9 in the IMS system services function modification identifier (FMID) HMK9900. To provide the same function between one or more TCP/IP clients and one or more IMS Version 7 or IMS Version 8 clients, IMS Connect is also available as one of the IBM DB2 UDB and IMS Tools products, IMS Connect Version 2, Release 2 (program number 5655-K52). IMS Connect Version 2.2 will continue to be supported for IMS Version 7 and IMS Version 8 clients, but future

enhancements to the IMS Connect functionality will be made available only with IMS Version 9 or later.

The IMS Connect architecture is designed to support any TCP/IP clients communicating with socket calls. IMS Connect supports TCP/IP sockets access to IMS transactions and commands. It does not require modifications to the existing IMS transactions. A roll your own (RYO) client can be written in any language that supports the TCP/IP interface. You need to know the protocol provided by IMS Connect and OTMA to develop this TCP/IP socket application. The advantage of a RYO client compared with IMS Connector for Java client is that the RYO client has similar functionality to and flexibility of the WebSphere MQ IMS bridge client applications, which support the OTMA protocol and the message routing.

IMS Connect also supports the TCP/IP clients using the IMS Connector for Java. The IMS Connector for Java is a collection of Java beans that enable a Java application to communicate data requests through TCP/IP and the IMS Connect to IMS. IMS Connect with WebSphere Development Tooling and the IMS Connector for Java can significantly ease the development of on demand business solutions that access IMS transactions. These solutions can be deployed in IBM WebSphere Application Server, allowing you to use Web applications, Java 2 Platform, Enterprise Edition (J2EE) applications, or Web services to quickly transform static Web sites into sources of dynamic Web content. IMS Connector for Java development environment is included in IBM WebSphere Application Developer Integrated Edition and in Rational Application Developer Version 6. The IMS Connector for Java client might have a little less flexibility than the RYO client, depending on your requirements. But you can build the IMS Connector for Java client without in-depth knowledge of the protocol of IMS Connect and OTMA, and you can use utilities that aid programmer productivity.

IMS Connect with IBM DB2 Version 8, the Universal Database Control Center provides a single graphical user interface to control both IMS and DB2, easing IMS operations. You can manage your IMS systems using a graphical interface from a workstation using the IMS Control Center. The IMS Control Center uses the IMS single point of control (SPOC) functions. The IMS Control Center is part of the IBM DB2 Universal Database™ Version 8 Control Center. The DB2 Control Center is available with the IBM DB2 Universal Database (DB2 UDB) Administration Client, Version 8.2.

Using the IMS Control Center, you can view the members of the IMSplex and define groups of members. The Control Center supports both IMS Version 8 and later versions. From the Control Center, you can issue IMS type-2 commands using command windows or wizards, depending on how much assurance you want. Wizards help you build and issue the commands. The results are displayed in the IMS Results window. In addition, using the Command Editor, you can issue both type-1 and type-2 IMS commands.

The IMS Control Center uses IMS Connect as a communication vehicle between the client and IMS Operations manager (OM). Before you can use the IMS Control Center, make sure that you have the correct software at the correct levels installed.

Accessing IMS databases through the ODBA

ODBA uses the IMS database resource adapter (DRA) function to establish a connection to the IMS subsystem specified by IMSID or DBCTLID coded in the DRA startup table. ODBA consists of a startup table for each IMS system and a group of modules that make up the ODBA callable interface. The DRA startup table specifies how to connect to a particular IMS subsystem, such as the subsystem name and the number of threads allowed.

The ODBA callable interface modules manage threads connecting an ODBA application to an IMS system and also provide some recovery services. They provide a set of Data Language/I (DL/I) database calls and system services calls. Application programs establish connections to IMS systems and make IMS database and system function calls using AERTDLI. The syntax is very similar to AIBTDLI.

ODBA requires the use of the application interface block (AIB). The AIB is a storage area obtained by the application program and is passed to IMS as a parameter in the AERTDLI call list when the application program issues DL/I calls. For ODBA, the AIB also provides the area the application program uses to keep track of the thread token. ODBA applications use z/OS Resource Recovery Services (RRS) and its application programming interfaces (APIs) to invoke commit processing. Figure 19-1 shows application access to IMS using the ODBA interface.

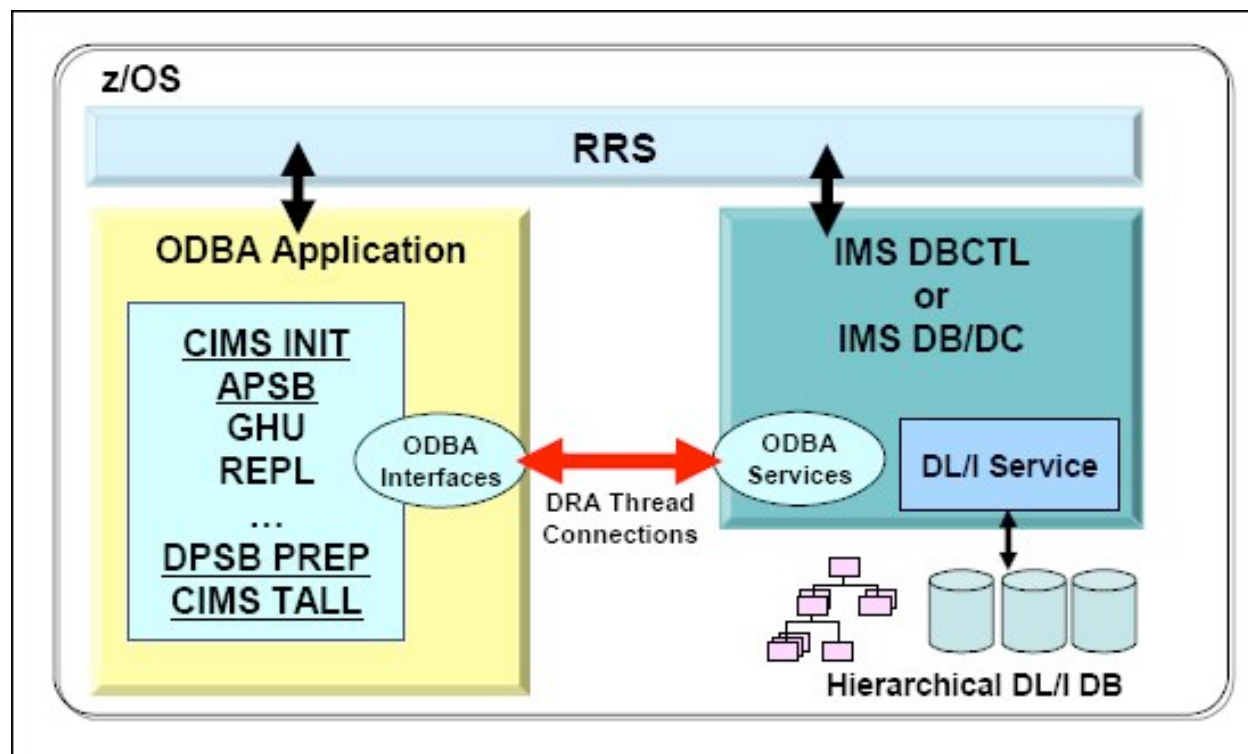


Figure 19-1 Application access to the IMS ODBA interface

Writing ODBA application programs

The z/OS ODBA application programs run in a separate z/OS address space that IMS regards as a separate region from the control region called the z/OS application region. The ODBA interface gains access to the IMS DB through the database resource adapter (DRA). The ODBA application programs (which can access to any address space within the z/OS in which they are running) gain access to IMS DB databases through the ODBA interface. z/OS ODBA application programs issue DL/I calls using an application interface block (AIB). Figure 19-3 shows a simple example of the program flow of a z/OS application program.

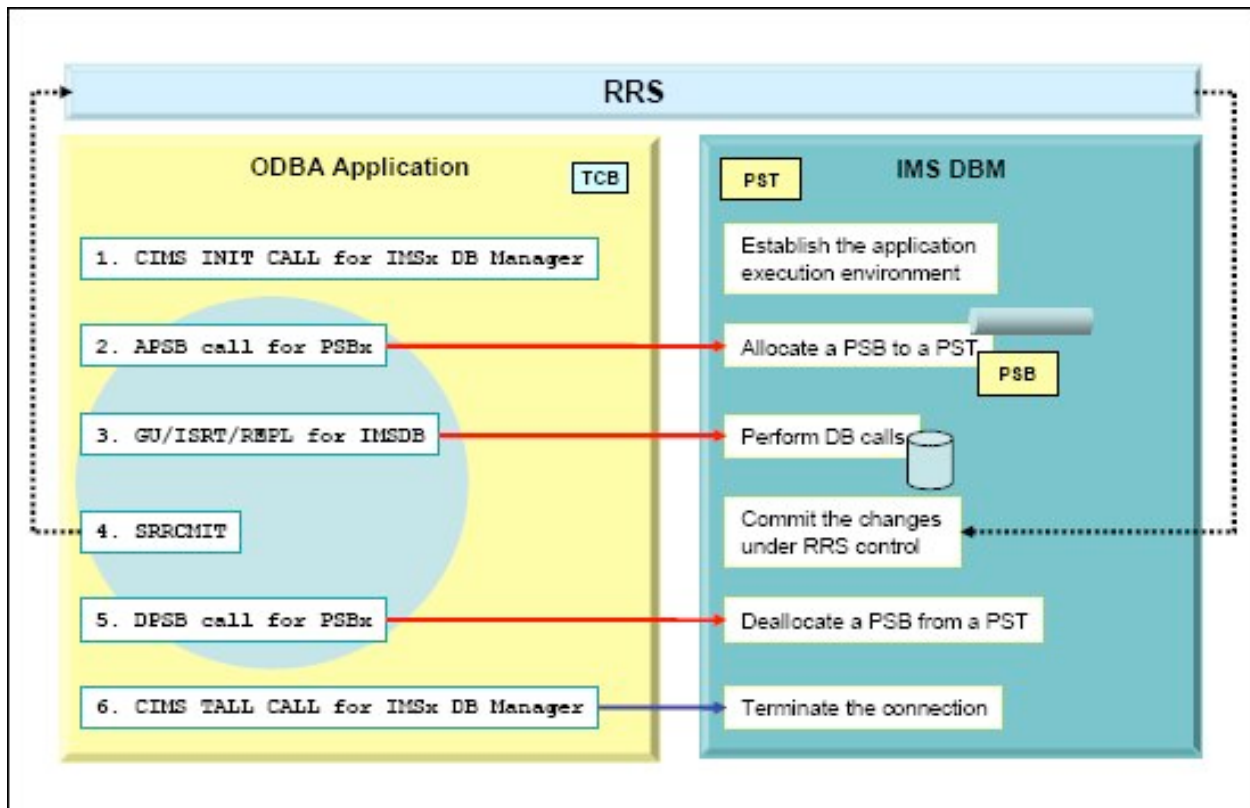


Figure 19-3 An example of the program flow of an ODBA application

IBM-supplied ODBA infrastructures

You do not have to code an ODBA application by yourself, because IBM provides several ODBA applications to meet your requirements. These ODBA applications work as infrastructure to execute your business logic that access to IMS databases. You can code the business logic by using various API such as DL/I or SQL through ODBC and JDBC interfaces, and you can run it on any platforms. In this section, we introduce the following IBM-supplied ODBA infrastructures:

- _ DB2 stored procedure
- _ IMS Remote Database Access from WebSphere Application Server for z/OS
- _ WebSphere Information Integrator Classic Federation for z/OS

DB2 stored procedure

The ODBA interface allows a DB2 stored procedure to directly connect to an system and issue DL/I calls to access IMS databases. A stored procedure can issue database DL/I requests through an ODBA callable interface. You need to update the startup procedure for the WLM-established stored procedure address space to add the ODBA data set names to the STEPLIB and DFSRESLB concatenations. Figure 19-5 shows the general structure of ODBA from a DB2 stored procedure. After you have coded your stored procedure containing the logic for IMS database access, you can call this stored procedure from any platform that can access DB2 for z/OS stored procedure environment.

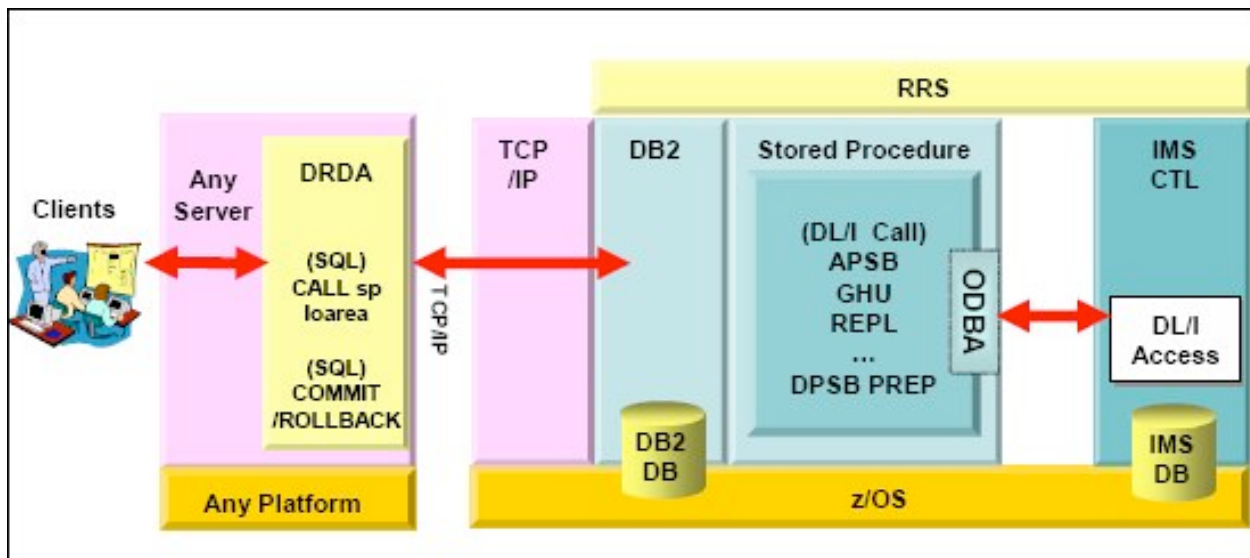


Figure 19-5 The general structure of ODBA from a DB2 stored procedure

WebSphere Application Server for z/OS and IMS Remote Data Access

WebSphere Application Server for z/OS Java applications can access IMS database directly through the IMS ODBA interface. You can use the classes in the IMS Java library to build a WebSphere Application Server for z/OS that accesses IMS data when WebSphere Application Server for z/OS and IMS are running on the same z/OS image. The WebSphere Application Server for z/OS application you build can access both the IMS full function and data entry databases using the IMS ODBA interface.

To provide this capability in WebSphere Application Server for z/OS, the IMS Java class library implements the J2EE Connector architecture resource adapter interfaces known as the IMS JDBC resource adapter.

With IMS Version 9, IMS Java Remote Database Services (RDS) support provides architected components on both the client and server side that allow a Java application deployed on a distributed WebSphere Application Server platform to issue JDBC calls

for IMS data. These requests are sent (transparently to the application) across the network and processed in IMS.

For the client-side of the connection, IMS Java RDS provides an IMS distributed JDBC resource adapter. To condition the distributed WebSphere Application Server for JDBC access to IMS, the IMS JDBC resource adapter must first be installed. For the server-side (IMS side) of the connection, a server EJB is delivered as part of the IMS Java RDS support. This EJB must reside on a WebSphere Application Server for z/OS environment because it uses the IMS JDBC resource adapter. Figure 19-6 shows the ODBA from WebSphere Application Server for distributed platforms.

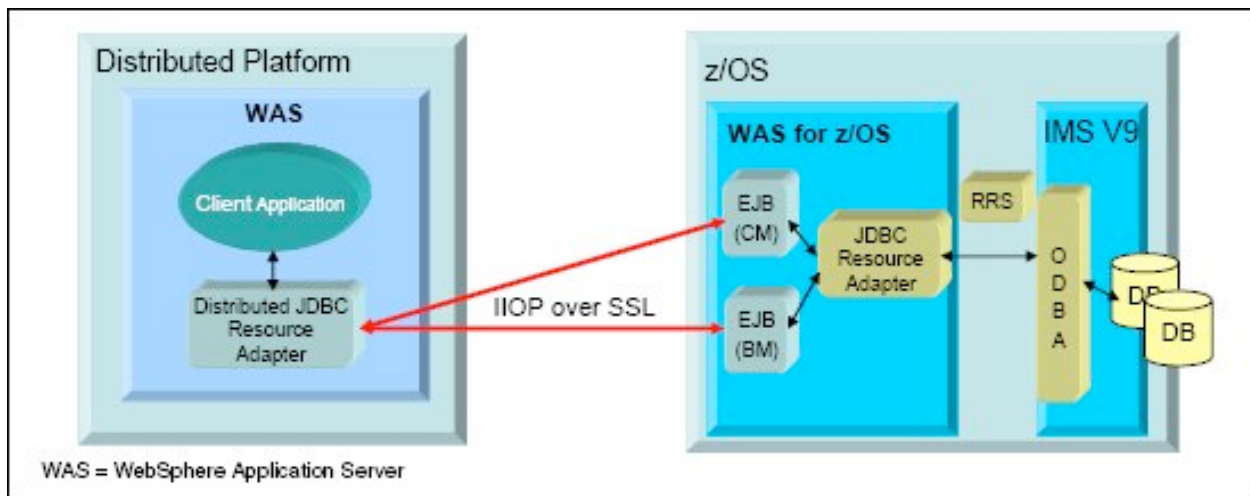


Figure 19-6 The ODBA from WebSphere Application Server for distributed platforms

WebSphere Information Integrator Classic Federation for z/OS

WebSphere Information Integrator Classic Federation for z/OS provides SQL access to mainframe databases (including IMS databases) and files with transactional speed and enterprise scale without mainframe programming. Using WebSphere Information Integrator Classic Federation for z/OS applications can:

- _ Map mainframe assets into relational views.
- _ Build integrated views across mainframe databases and files co-resident on a single z/OS instance.
- _ Issue SQL SELECT, INSERT, UPDATE, and DELETE statements from ODBC, JDBC, or a call level interface (CLI) to access mainframes, databases, and files.

Figure 19-7 shows an overview of WebSphere Information Integrator Classic Federation.

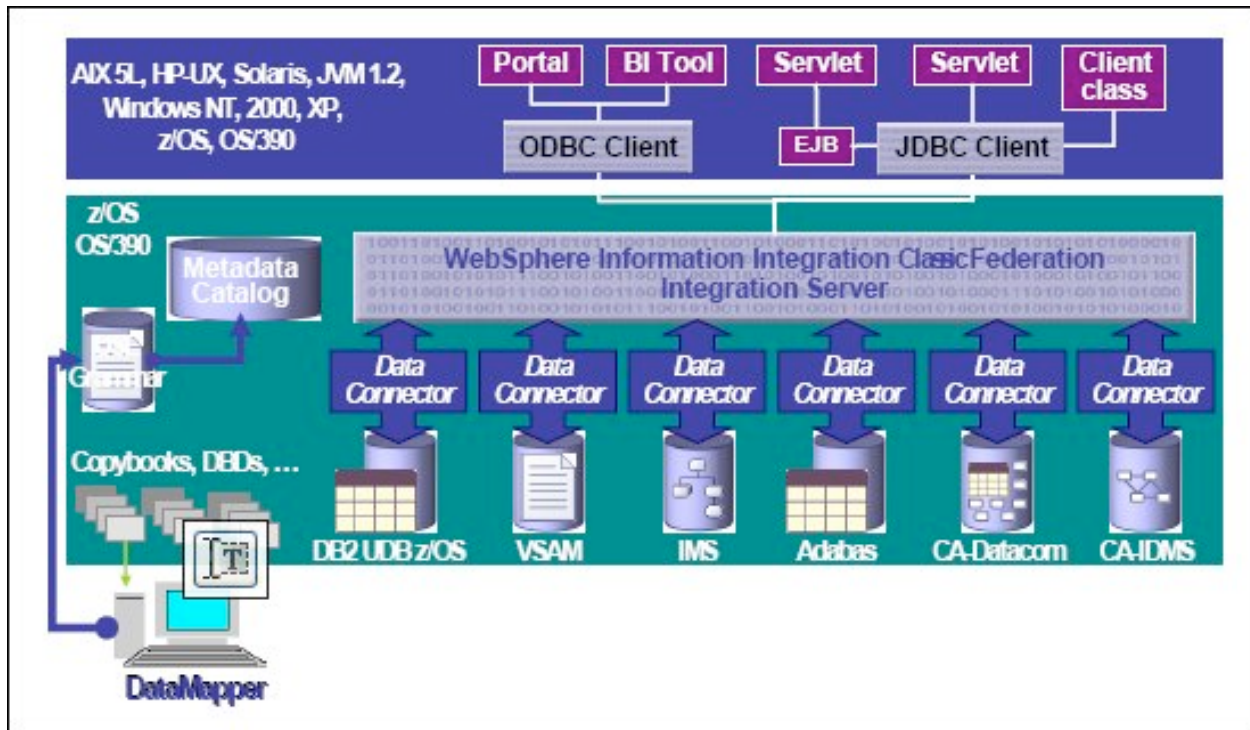


Figure 19-7 Overview of WebSphere Information Integrator Classic Federation for z/OS

IMS MFS Web Services and MFS Web Enablement

IMS MFS Web Services lets you define a service from a Message Format Service (MFS) source and then deploy it to WebSphere Application Server and make it available as an EJB service and a Java service.

You build an MFS-based service by importing MFS source files into the WebSphere Studio Application Developer Integration Edition and generating Web Service Definition Language (WSDL) files. The WSDL files are based on application interface information that is stored in the MFS source. The WSDL files contain details about the service including:

- _ Interface elements that expose the operations and messages provided by the service
- _ Provider-specific binding elements that describe how the service interface is implemented
- _ Service and port elements that enable client applications to locate the service

MFS Web Services are built by generating Web Service Definition Language (WSDL) files. The MFS WSDL files contain only device information. For the input part of the WSDL message, only device fields that map to input message fields are used; and on the output side, only device fields that map to output message fields are used.

A Web services client, which can be a Java or EJB application using a Web Services

Invocation Framework (WSIF), looks up the WSDL file generated from the MFS importer. Based on the WSDL, the Web services client starts the service. The service request then invokes the MFS Adapter.

The MFS Java or EJB clients use the MFS Adapter to translate a service message into a byte stream to input to IMS Connector for Java and also to translate a response in byte stream to a service message, as shown in Figure 16-1.

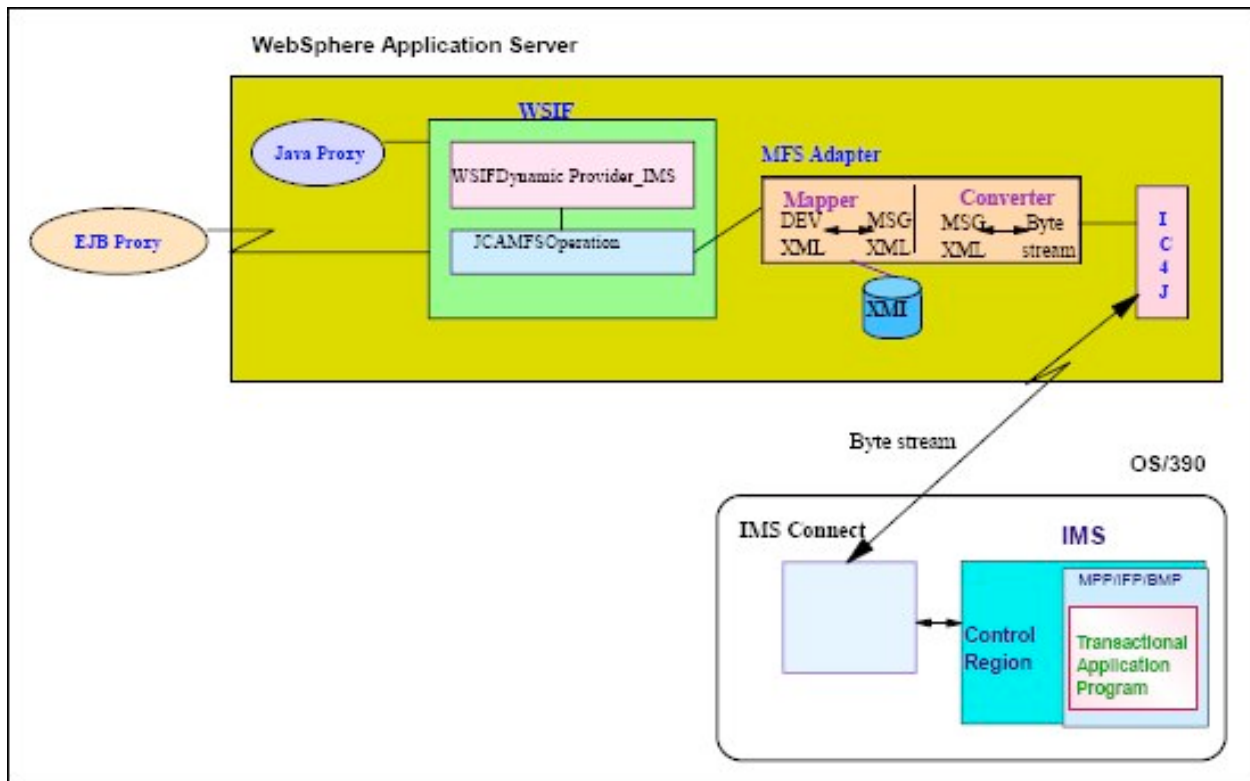


Figure 16-1 MFS Web Services in WebSphere Application Server

IMS MFS Web Enablement

IMS MFS Web Enablement modernizes existing MFS-based IMS transactions in a business-to-consumer (B2C) environment. The support Web-enables existing or new IMS MFS-based applications in IBM WebSphere Application Server and interactively renders them for display in standard browsers such as Microsoft Internet Explorer and Mozilla Firefox.

The MFS Web Enablement tooling utility support consists of the MFS XML Utility and the MFS Importer. MFS Web Enablement runtime support consists of application instance servlets, sample cascading style sheets, the MFS Servlet, and the MFS Adapter.

At development time, the IMS MFS XML Utility invokes the MFS Importer to parse MFS source files of existing MFS-based IMS transactions and generates Java classes and

metadata files for runtime processing. The Java classes are packaged into a Web application archive (WAR) file and deployed on WebSphere Application Server.

During run time, the HTTP request from a browser is sent to the Web application, which works in conjunction with the IMS MFS Web Enablement runtime component. If this request is the initial request, a new HTTP session is created and the initial Web page is returned, which simulates the 3270 type terminal blank screen. Subsequent requests containing input data, which can be transaction or command, are transformed into an input byte stream to be sent across to the IMS host application. The output byte stream coming back from the application is transformed into XML data objects that are stored in the HTTP session. The XML data object contains one or more physical pages to be displayed as Web pages. Web pages are returned in the HTTP response, one page per 3270-PA1-equivalent paging request.

As depicted in Figure 17-1, the IMS Installation Verification Program Phonebook application is invoked to display a phonebook entry. The input request, coming in from the Web browser, is transformed into an input byte stream and sent to the IMS host application. The output byte stream is transformed, and the Web page containing output data is generated and returned.

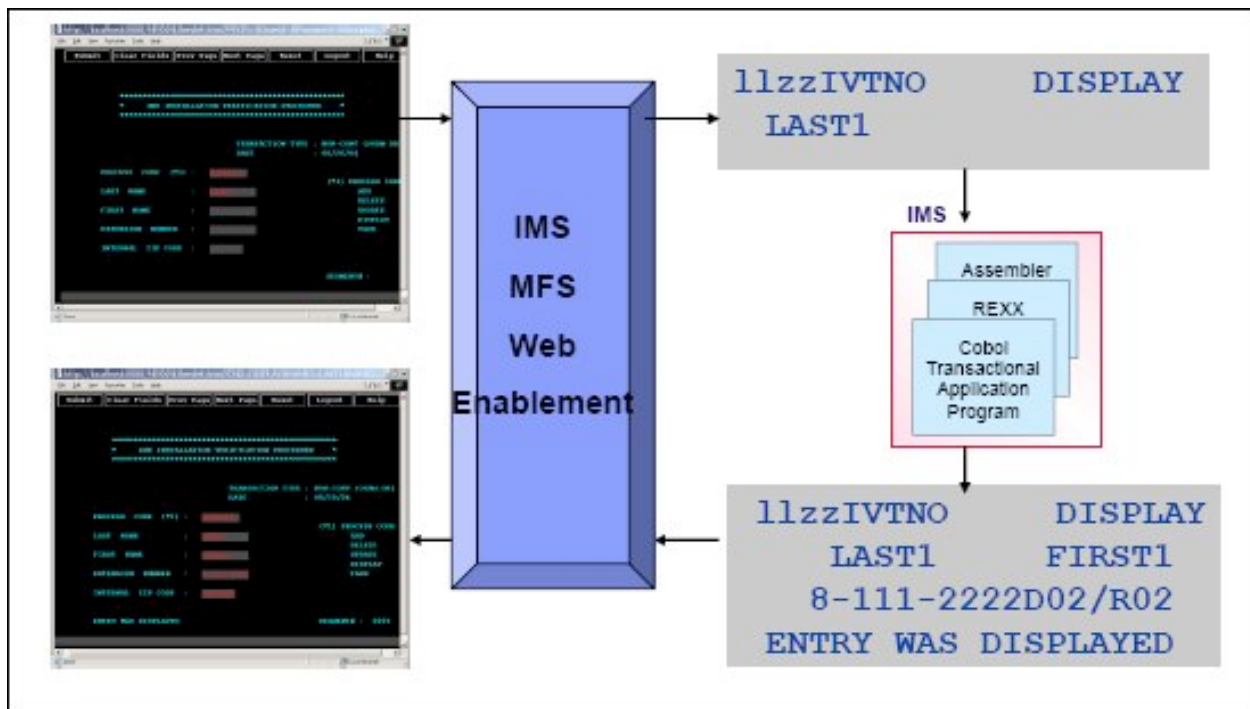


Figure 17-1 IMS IVP Phonebook application as an MFS Web-enabled Web service

IMS MFS Web Enablement runtime support

MFS Web Enablement runtime support consists of the instance servlets, MFS Servlet, and MFS Adapter. The support runs on WebSphere Application Server and requires IMS Connect and IMS Connector for Java.

- MFS instance servlet
Instance servlets are generated by the MFS XML Utility tool. An instance servlet records user-specified IMS Connector for Java connection properties, MFS style sheet file path, MFS XMI repository file path, and WebSphere Application Server platform information. The specifications are used by the MFS Servlet.
- MFS Servlet
The MFS Servlet is the super class of all MFS instance servlets. The MFS Servlet handles HTTP requests and responses to and from Web browsers. The MFS Servlet is responsible for the connection state management, interaction with the MFS Adapter, and the rendering of MFS XMI objects using the style sheet to dynamically generate Web pages.
- MFS Adapter
The MFS Adapter loads metadata XMI files into XML data objects for data transformation to and from byte stream that IMS applications understand. It works in conjunction with the MFS Servlet and IMS Connector for Java to supply input and to handle output for MFS-based IMS transactions.

Figure 17-2 depicts data flowing through MFS Web Enablement runtime components, IMS Connector for Java, and IMS Connect to IMS application programs and back.

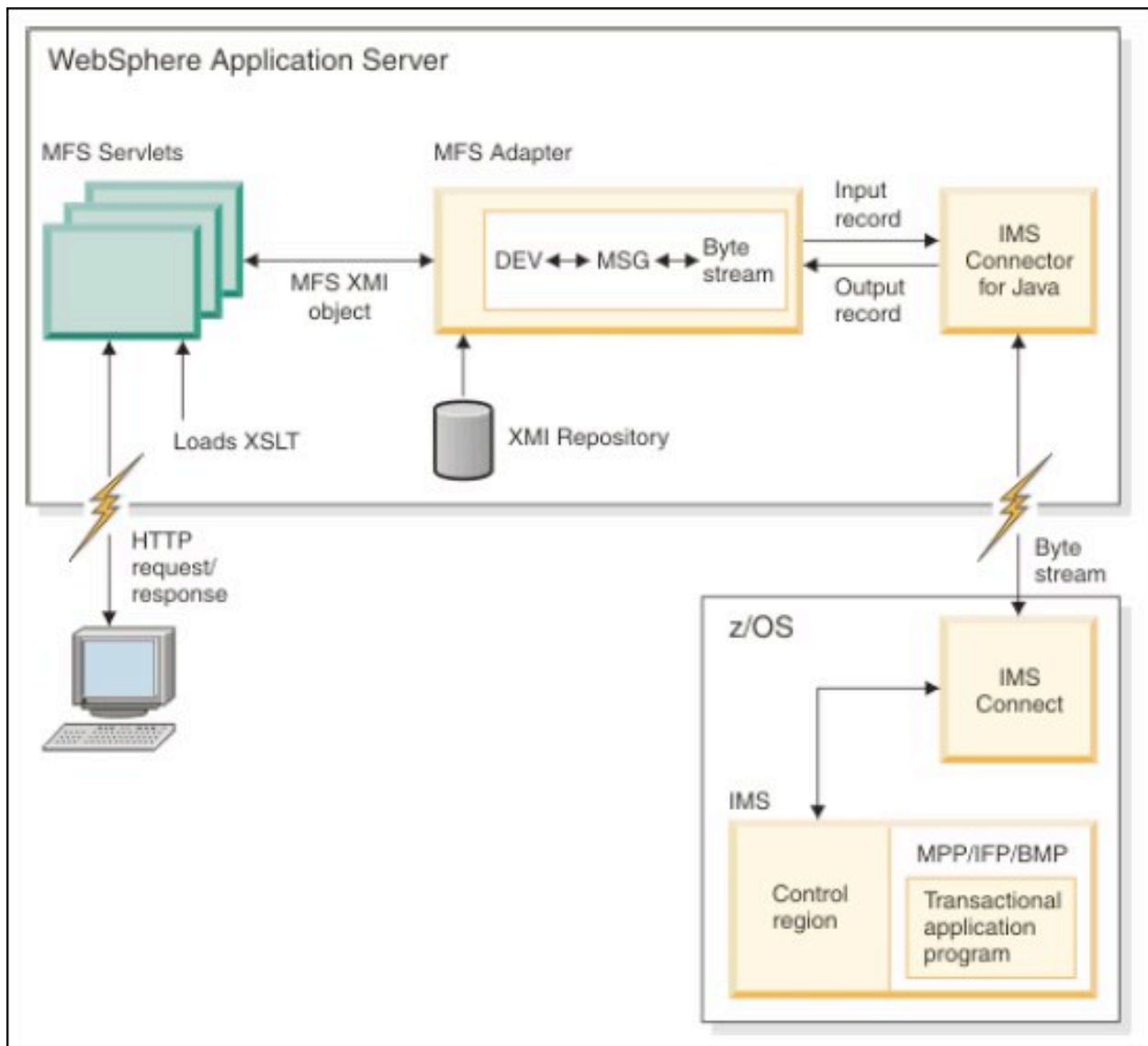


Figure 17-2 Data flowing through MFS Web Enablement runtime components

IMS SOAP Gateway

IBM IMS SOAP Gateway is a Web services solution that enables IMS applications to interoperate outside of the IMS environment through Simple Object Access Protocol (SOAP) to provide and request services independently of platform, environment, application language, or programming model.

The IMS SOAP Gateway can assist an organization in the following areas:

- _ Enterprise modernization
- _ Application development
- _ Business-to-business (B2B) integration
- _ Service-oriented architecture (SOA) implementation

For the latest information about IMS SOAP Gateway, visit:

<http://www.ibm.com/software/data/ims/soap/>

IMS SOAP Gateway allows you to enable your IMS application to become a Web service. Different types of client applications, such as Microsoft .NET, Java, and third-party applications, can submit SOAP requests into IMS to drive the business logic of your IMS applications.

IMS SOAP Gateway is compliant with the industry standards for Web services, including SOAP/HTTP 1.1 and WSDL 1.1. This allows your IMS assets to interoperate openly with various types of applications.

With IMS SOAP Gateway, you can be flexible about how you want your data to be handled. For example, you can have the data send the XML data from the client to your IMS environment that can be handled by your new or enhanced IMS application. The data can be stored directly in XML using the IMS XML DB function, or you can transform the XML data into data bytes.

IMS SOAP Gateway consists of two main components:

- _ IMS SOAP Gateway deployment utility
The end-to-end deployment utility enables you to set up properties and create runtime code that IMS SOAP Gateway uses to enable IMS applications as Web services.
- _ IMS SOAP Gateway server
The IMS SOAP Gateway server processes SOAP messages. It receives the SOAP message from the client application, converts it to an IMS input message, and sends it to IMS through IMS Connect. It then receives the output message from IMS and converts it to a SOAP message to send back to the client.

The diagram in Figure 18-1 shows how IMS SOAP Gateway helps you process the SOAP message from your client at run time.

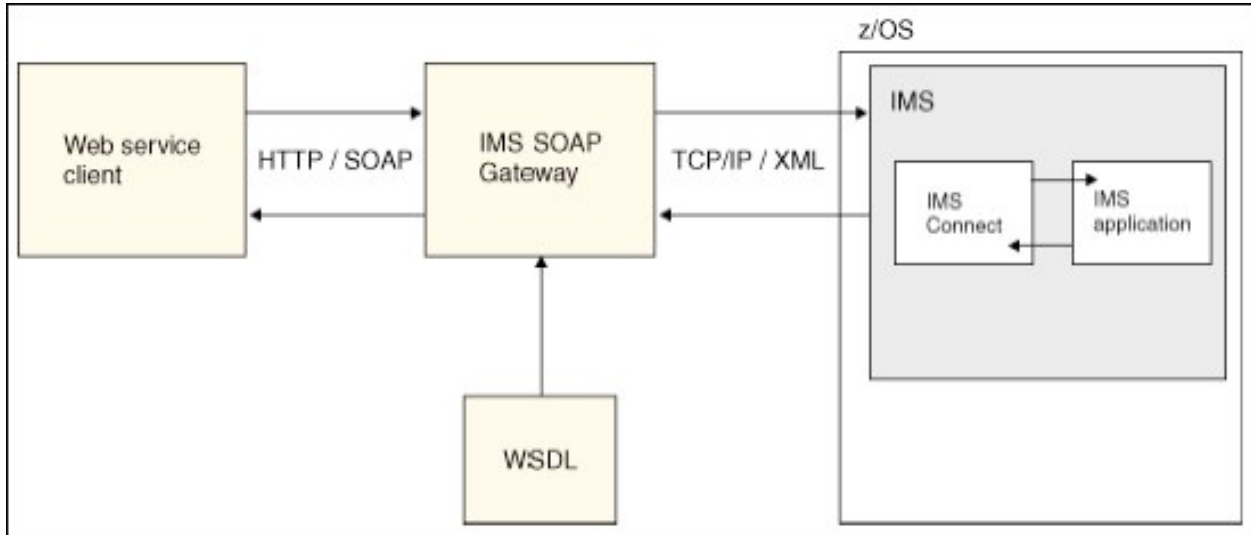


Figure 18-1 IMS SOAP Gateway