IMS Integration Suite Enhancements

IMS Integration Suite

- IMS TM Resource Adapter
  - IMS Connector for Java
- IMS SOAP Gateway
- IMS MFS Web Support
- IMS DB Resource Adapter
  - IMS JDBC Connector
- IMS DLIModel utility
- IMS XML DB

2

The IMS™ Integration Suite is a collection of IMS middleware functions and tools that support your IMS on demand systems and your distributed IMS application environment. Some of these middleware functions and tools are generally available, whereas others might be part of an open beta program or technology preview program. All of these functions and tools require IMS Version 9 (or later), and are not supported for previous releases of IMS.

The IMS Integration Suite includes the following IMS middleware functions and tools:

IMS TM Resource Adapter - formally know as IMS Connector for Java
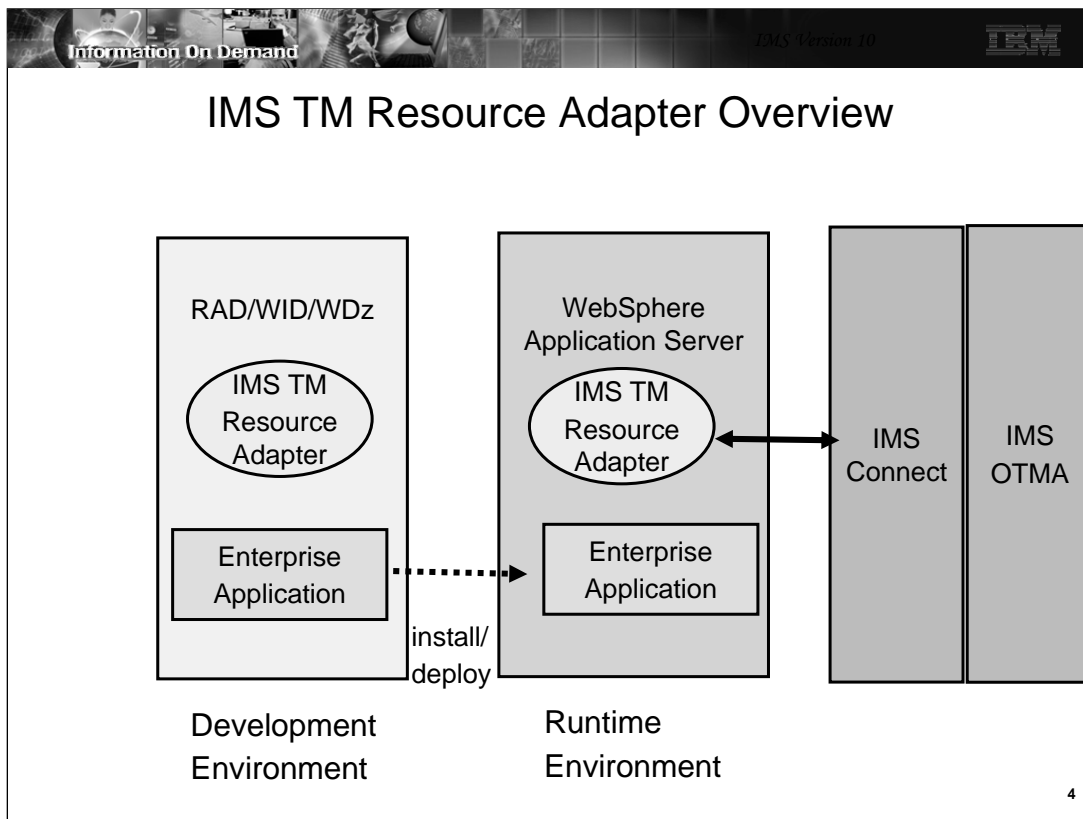
IMS SOAP Gateway

IMS MFS Web support

IMS DB Resource Adapter – formally known as IMS JDBC Connector

IMS DLIModel utility

IMS XML DB

IMS TM Resource Adapter

3

IMS TM Resource Adapter Overview

```
┌──────────────────┐  ┌──────────────────┐  ┌────────┐  ┌────────┐
│ RAD/WID/WDz      │  │ WebSphere        │  │        │  │        │
│                  │  │ Application Server│  │        │  │        │
│   ╭──────────╮   │  │   ╭──────────╮   │  │        │  │        │
│   │ IMS TM   │   │  │   │ IMS TM   │◄──┼──┼─► IMS   │  │  IMS   │
│   │ Resource │   │  │   │ Resource │   │  │ Connect│  │  OTMA  │
│   │ Adapter  │   │  │   │ Adapter  │   │  │        │  │        │
│   ╰──────────╯   │  │   ╰──────────╯   │  │        │  │        │
│  ┌────────────┐  │  │  ┌────────────┐  │  │        │  │        │
│  │ Enterprise │┈┈┼┈┈┼┈►│ Enterprise │  │  │        │  │        │
│  │ Application│  │  │  │ Application│  │  │        │  │        │
│  └────────────┘  │  │  └────────────┘  │  │        │  │        │
└──────────────────┘  └──────────────────┘  └────────┘  └────────┘
       install/
       deploy
   Development          Runtime
   Environment          Environment
```

4

IMS TM Resource Adapter is a J2EE Connector Architecture (J2C) resource adapter that supports IBM Java integrated development environments (IDEs) to create Java applications that can access IMS transactions. The IDEs import C, COBOL and PL/I source to generate the Java data bindings for the IMS transaction input/output message segments.

This diagram shows the dual functions of IMS TM Resource Adapter. To provide IDE support and to provide runtime environment support to access IMS applications using IMS Connect as the interface to IMS via OTMA.

The following IDEs are supported by IMS TM Resource Adapter:

Rational Application Developer (RAD)

Rational Software Architect

WebSphere Integration Developer (WID)

WebSphere Developer for zSeries (WDz)

WebSphere Studio Application Developer Integration Edition (WSAD-IE)

The IMS TM Resource Adapter also provides a plug-in for a J2EE application server by using the standardized system contracts. IMS TM Resource Adapter provides runtime support for WebSphere Application Server on distributed and z/OS platforms

A J2EE Application Server provides the runtime environment for application code.

WebSphere Process Server (WPS) – extends an application server by providing a

business process engine that executes business processes.

WID is the development tool used to create standards based business processes that are deployed to WPS.
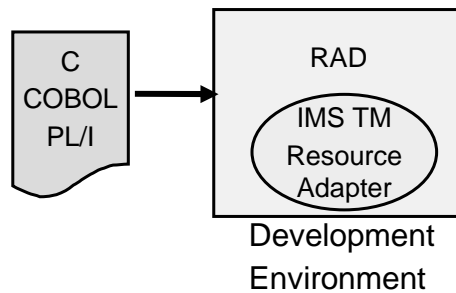
IMS TM Resource Adapter

- Pre-IMS 10 enhancements
  - PL/I source message structures
  - Commit Mode 1, Sync Level CONFIRM
  - Retrieve output using alternate clientID on shareable persistent sockets
  - Enhanced SSL cipher configurability

5

This is the list of enhancements IMS TM Resource Adapter provided after IMS V9 and before IMS V10.
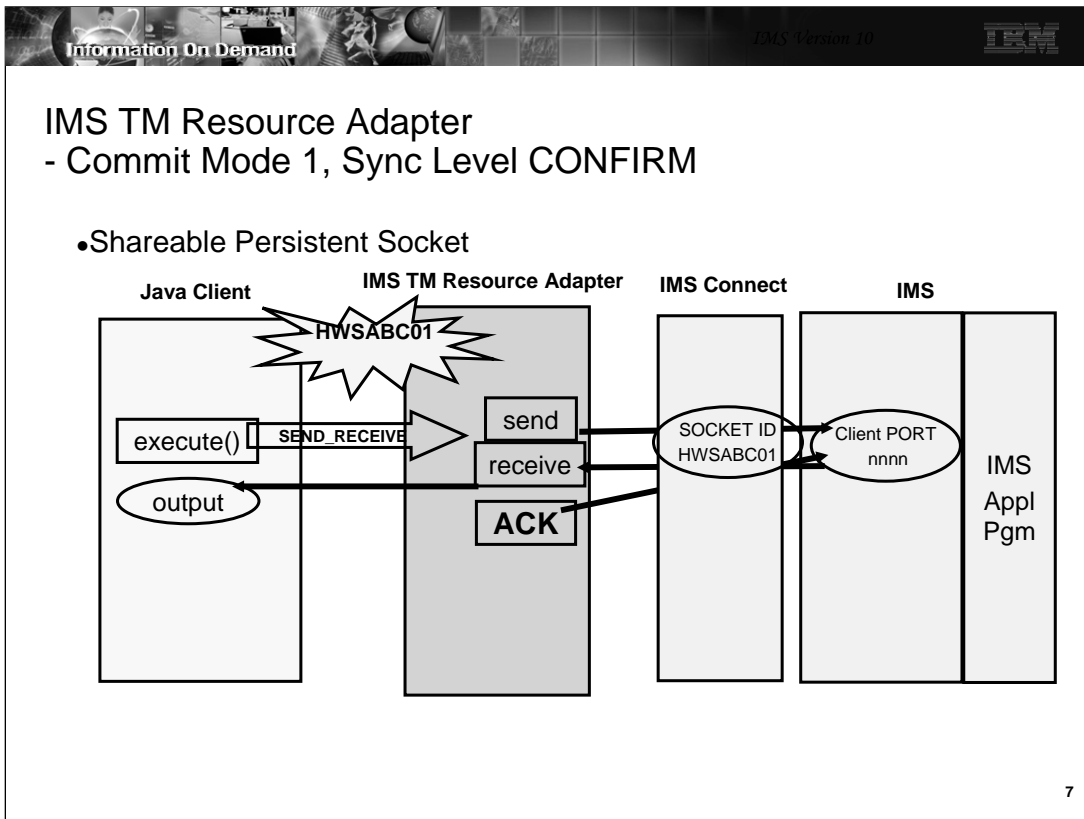
PL/I source message structures

- Capability to parse PL/I copybooks
  - Used to create Java data bindings for the input and output messages
- Implementation
  - Shipped in Rational Application Developer
- Benefits
  - Simplifies building Java data bindings for IMS PL/I application programs

```
C            RAD
COBOL    ┌─────────────┐
PL/I  →  │  IMS TM     │
         │  Resource   │
         │  Adapter    │
         └─────────────┘
         Development
         Environment
```

6

RAD imports C , COBOL or PL/I definitions of the IMS transaction input and output messages and creates Java data bindings for the input and output messages.

Prior to this support the customer would have to create a COBOL copybook version of the IMS PL/I input/output message structure.

IMS TM Resource Adapter
- Commit Mode 1, Sync Level CONFIRM

- Shareable Persistent Socket

Java Client — **IMS TM Resource Adapter** — **IMS Connect** — **IMS**

HWSABC01

execute() — SEND_RECEIVE — send / receive — ACK — SOCKET ID HWSABC01 — Client PORT nnnn — IMS Appl Pgm

output

7

Commit Mode 1 is only supported using a shareable persistent socket.

IMS TM Resource Adapter uses persistent socket connections to communicate with IMS Connect. A persistent socket connection prevents the overhead of opening and closing the socket for each use by an application. IMS TM Resource Adapter provides two types of persistent socket connections.

1.Dedicated persistent – a dedicated persistent socket connection is restricted to Commit Mode 0 (Commit then Send protocol) and the client application provides the clientid to be used as the socket identifier.

2.Shareable persistent – a shareable persistent socket connection supports Commit Mode 0 and Commit Mode 1 (Send then Commit protocol). However, The client application does not provide a clientid for a socket identifier. IMS TM Resource Adapter generates a clientid.

The support for CONFIRM provides the ability for IMS TM Resource Adapter to respond ACK to the output message which activates the IMS sync-point process. In this case the client application is not running under a Global Transaction.

This provides the ability for IMS TM Resource Adapter to notify IMS that it has received the IMS application output message without requiring a Global Transaction.

In this example the Java client is using a shareable persistent socket Commit Mode 1 Sync Level CONFIRM. IMS TM Resource Adapter generates HWSABC01 for a socket ID and sends the input message to the IMS application program via IMS Connect and OTMA. To provide a message queue anchor for the Commit Mode 1 output message OTMA creates a transaction pipe (TPIPE) and will use the TCP/IP generated Client Port number provided by IMS Connect for the TPIPE name. When IMS TM Resource Adapter receives the output message from IMS Connect it responds with an ACK. IMS Connect will send the ACK to OTMA to activate the IMS sync point process.

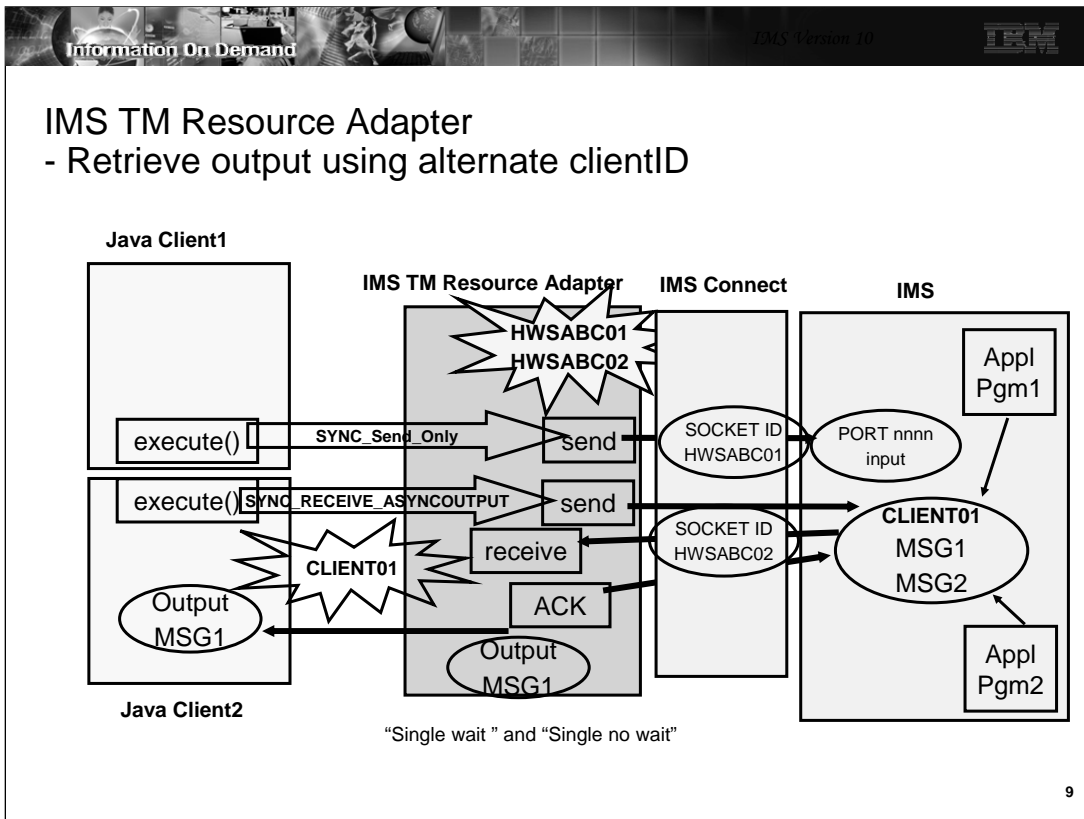Note the Java Client application cannot respond with an ACK.

Commit Mode 1, Sync Level CONFIRM

- Implementation
  - Shipped in Rational Application Developer
- Considerations
  - IMS TM Resource Adapter sends ACK
  - CM1 (Send-then-Commit) Time-out Control
- Benefits
  - IMS TM Resource Adapter can notify IMS that it has received the IMS application output message without requiring a Global Transaction
  - Reduce the need of a compensating transaction
  - IMS TM Resource Adapter compatible with APPC sync level support

8

Commit Mode One Synclevel None has the potential to discard an output message without a ROLLBack of the database in the event IMS Connect is not able to deliver the output message to IMS TM Resource Adapter. As a result, a compensation transaction may have to be executed to reverse the database changes. Adding the Commit Mode One Synclevel Confirm could reduce the necessity of using a compensation transaction because an ACK will be performed by IMS TM Resource Adapter.

APPC provides CM1 sync levels NONE,CONFIRM and SYNCPT. Customers who have APPC client applications accessing IMS using CM1 sync level CONFRIM can now provide IMS TM Resource Adapter clients with the same sync levels.

IMS TM Resource Adapter
- Retrieve output using alternate clientID

"Single wait " and "Single no wait"

For Shareable Persistent Sessions the client application can not specify a clientID since it is generated by IMS TM Resource Adapter. This can prevent the retrieving of failed IOPCB output or ALTPCB output messages. For the current IMS TM Resource Adapter support the message originating client application can only request asynchronous output while the generated id for the socket remains connected.

In this example Java Client1 submits a Send_Only input message to IMS. IMS TM Resource Adapter generates HWSABC01 Socket ID and sends the input message to IMS Connect. The IMS application program does not respond to the I/O PCB but does insert a message to ALTPCB using CLIENT01 for a TPIPE name. Java Client2 submits a SYNC_RECEIVE_ASYNCOUTPUT to IMS TM Resource Adapter to obtain the output queued to CLIENT01 TPIPE. IMS TM Resource Adapter generates HWSABC02 socket ID and uses the new "Resume tpipe with alternate clientid" protocol to retrieve the output message.

Notice with this protocol a different Java Client application can be used to retrieve asynchronous output generated from processing the originating java client application request.

IMS TM Resource Adapter supports the "Single wait " and "Single no wait" options.

For the Single message with wait option only one message is returned for a RESUME TPIPE request. If no message is present OTMA waits for a message to arrive and then sends that single message to IMS Connect. The timer set on RESUME TPIPE can expire before a message is returned to IMS Connect. If that occurs, IMS Connect will NAK the message when sent to IMS Connect by OTMA.

For Single message nowait only one message is returned following the RESUME TPIPE. If no message is present OTMA does not wait for a message and the IMS Connect timer causes a timeout to occur based on the timeout value specified.

Retrieve output using alternate clientID

- Implementation
  - Requires IMS Connect APAR PK24907
  - Only "Single wait " and "Single no wait" options supported
- Error considerations
  - IMS Connect cannot send output
    - NACK do not de-queue output message
- Benefits
  - Can use shareable persistent socket to retrieve asynchronous output
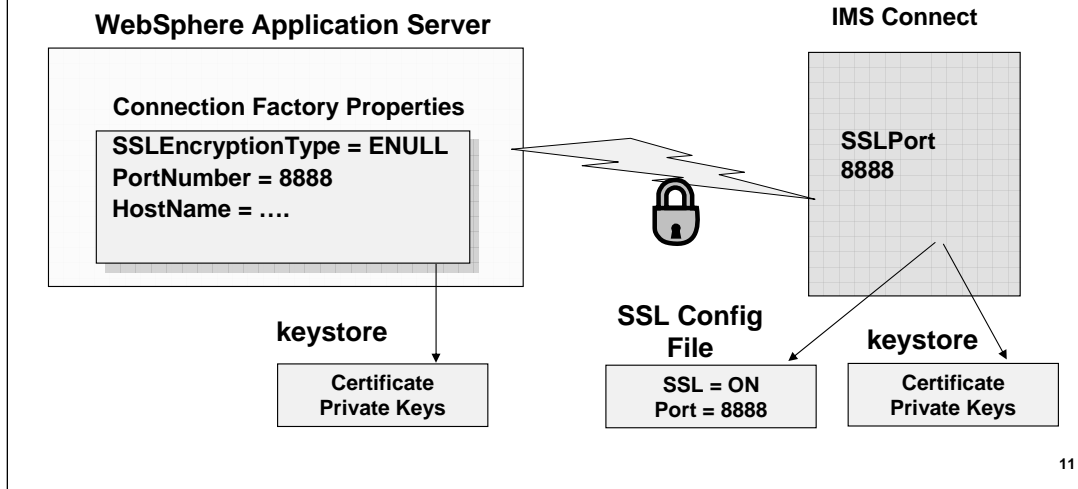 - Simplify IMS Connector for Java Client programming

10

IMS Connect APAR PK24907 supports a new protocol "Resume tpipe with alternate clientid" that provides the IMS TM Resource Adapter client application the ability to specify an alternate clientid in the SYNC_RECEIVE_ASYNCOUTPUT interaction.

If IMS Connect cannot send the output message to IMS TM Resource Adapter then IMS Connect will respond NACK to OTMA. OTMA will not de-queue the output message.

IMS TM Resource Adapter

- Enhanced SSL cipher configurability
  - ENULL value for the SSLEncryptionType property.

WebSphere Application Server

IMS Connect

Connection Factory Properties

SSLEncryptionType = ENULL
PortNumber = 8888
HostName = ….

SSLPort
8888

keystore

Certificate
Private Keys

SSL Config File

SSL = ON
Port = 8888

keystore

Certificate
Private Keys

11

The Secure Socket Layer (SSL) protocol ensures that the transfer of sensitive information over the Internet is secure.

Currently, IMS TM Resource Adapter allows customers to specify only "strong" or "weak" as the SSLEncryptionType for all connections created by an IMS connection factory. Enhanced SSL cipher configurability allows customers to specify a new value, "ENULL" for SSLEncryptionType. When "ENULL" is specified, IMS TM Resource Adapter will attempt to use a cipher spec whose name contains the string "NULL"

Null encryption will allow for authentication to take place during the SSL handshaking process as is currently the case. Once the handshaking process for a socket has completed including authentication as required, all messages will flow in the clear over that socket.

An example is the passing of non-confidential data between IMS TM Resource Adapter and IMS Connect, both of which are located behind a firewall.

IMS TM Resource Adapter

- Enhancements for IMS 10
  - Asynchronous Callout to EJB/MDB
  - Change RACF password using IMS TM Resource Adapter and IMS Connect
  - IMS TM Resource Adapter EWLM's policy-based performance management
  - IMS SOA Composite Business Application support
  - CM0 multiple PURG calls

12

This is the list of enhancements IMS TM Resource Adapter provides for IMS V10.

Asynchronous Callout to EJB
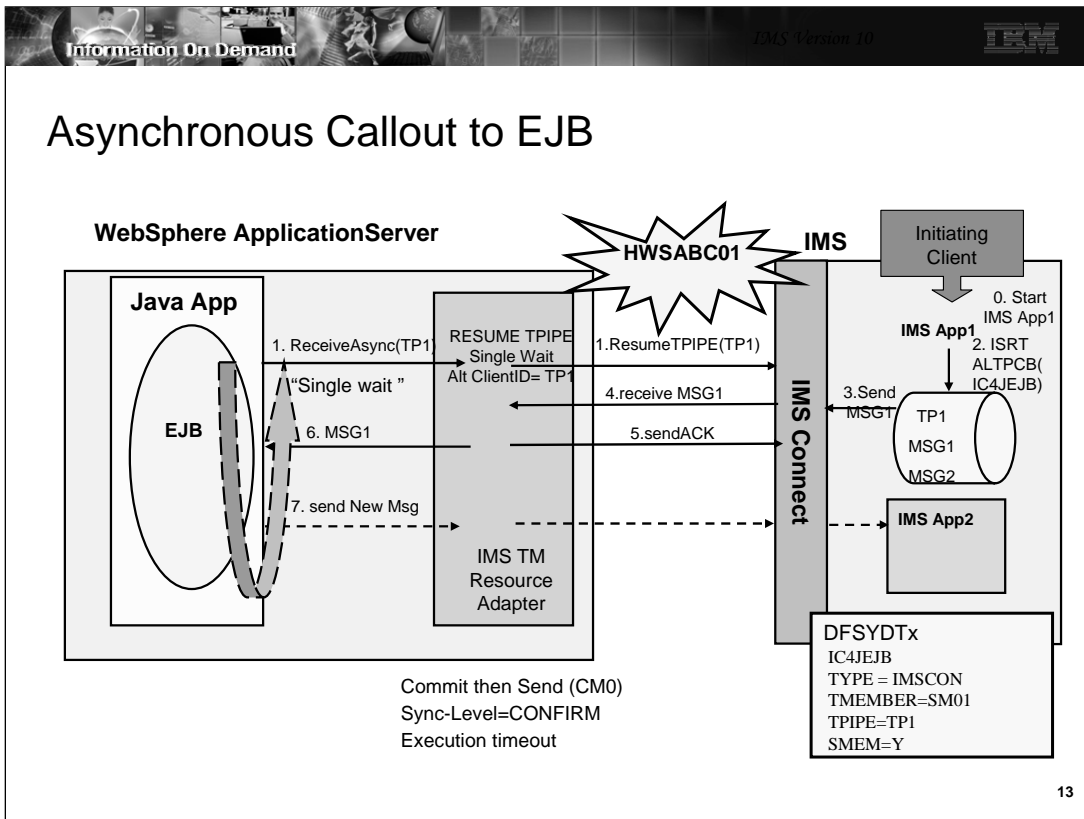
IMS TM Resource Adapter has been enhanced to support access to Enterprise Java Beans (EJB) from an IMS application program. In this scenario an EJB uses the SYNC_RECEIVE_ASYNC_OUTPUT_SINGLE_WAIT (1) alt-clientID=TP1 to request a message queued to TP1. IMS TM Resource Adapter uses the IMS Connect "Resume tpipe with alternate clientid" protocol to wait for a message to retrieve (1). IMS Connect receives the request and notifies OTMA to send a message. An already scheduled (0) IMS Application program does an insert to an ALTPCB for a destination defined in an OTMA destination routing descriptor (2). The OTMA destination routing descriptor is used to define the OTMA TPIPE for the en-queue of the message (TP1). OTMA sends the message to IMS Connect (3) and IMS Connect will return the message to IMS TM Resource Adapter (4). IMS TM Resource Adapter receives the message and notifies IMS Connect with an ACK(5). IMS Connect will forward the ACK to OTMA and the message will be de-queued.

IMS TM Resource Adapter receives the message and returns it to the EJB (6).

An optional step is the EJB creates a response to the message request. IMS TM Resource Adapter sends the response to IMS (7). This results in the scheduling of an IMS Application program to process the response.

Asynchronous Callout to MDB
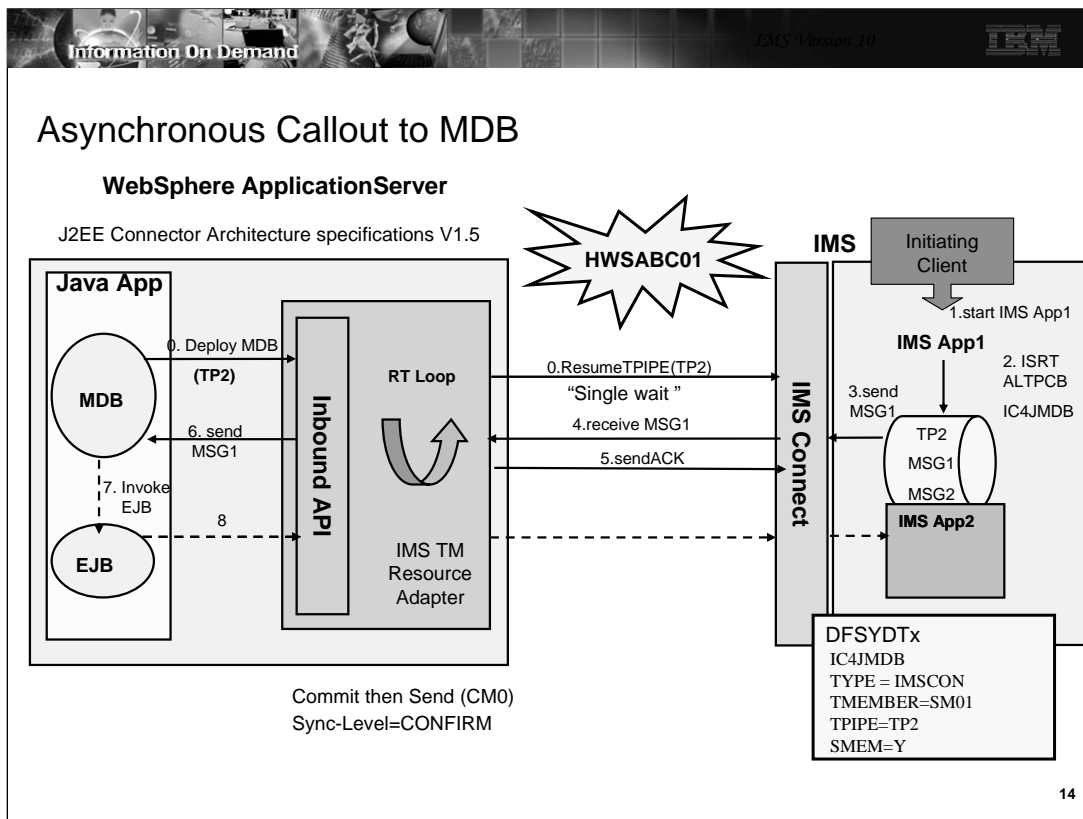
IMS TM Resource Adapter also supports access to Message Driven Beans (MDB) from an IMS application program. The J2EE Connector Architecture specifications V1.5 describes how an enterprise application can access a J2EE application environment. This is known as Inbound processing. IMS TM Resource Adapter is providing Inbound processing for IMS application programs. From an IMS application perspective this is Asynchronous Callout processing.

In this scenario IMS TM Resource Adapter uses the IMS Connect "Resume tpipe with alternate clientid" protocol to wait for a message to retrieve (0).

An already scheduled(1) IMS Application program does an insert to an ALTPCB for a destination defined in an OTMA Descriptor (2). The OTMA Descriptor is used to define the OTMA TPIPE for en-queue of the message.

OTMA sends the message to IMS Connect (3) and IMS Connect will return the message to IMS TM Resource Adapter (4). IMS TM Resource Adapter receives the message and notifies IMS Connect with an ACK(5). IMS Connect will forward the ACK to OTMA and the message will be de-queued.

An MDB running inside WebSphere Application server will use the standard JCA 1.5 interfaces to listen to callout request from IMS applications. IMS TM Resource Adapter receives the message send it to the MDB (6).

The MDB will invoke the EJB (7) to process the message.

An optional step is the EJB creates a response to the message request. IMS TM Resource Adapter sends the response to IMS (8). This results in the scheduling of an IMS Application program to process the response.

Asynchronous Callout

- Implementation
  - ◆ IMS V10 Connect Resume TPIPE with Alternate Clientid
  - ◆ IMS V10 OTMA Resume TPIPE Security
  - ◆ IMS V10 OTMA Destination Routing Descriptors
- Migration
  - ◆ No impact to existing IMS Application
    - ▪ ISRT ALTPCB
      - • the OTAM Destination Routing Descriptor provides destination information
- Benefits
  - ◆ Provides ability for IMS application to access a Web Service
    - ▪ IMS application does not have to change

15

IMS TM Resource Adapter

- RACF Password Change
  - Provides ability to change RACF password using IMS TM Resource Adapter
- Implementation
  - IMS Connect V10 Client Password Change Request
- Benefits
  - Customers can now change their RACF password after expiration

LLLL | IRM | OTMA | LLZZ **HWSPWCH** *old-password/new-password1/new-password2* | EOM

16

IMS TM Resource Adapter has been enhanced to change a RACF password during interactions with IMS Connect.

When a J2EE application requires access to an IMS transaction that is password protected the RACF password may have expired. Using this facility the J2EE application can prompt the client for an new password and provide the new password to facilitate completion of the business process.

IMS TM Resource Adapter

- Client sample

```
package sample.ims.hwspwcdh;
        import com.ibm.connector2.ims.ico.*;

    public class HWSPWCHIMSConnect
    {
        public static void main(String[] args)
        {
            try
            {
                INMSG input = new INMSG();
                input.setInll((short) 31);
                input.setInzz((short) 0);
                input.setIncmd("HWSPWCH CURPSWD/NEWPSWD/NEWPSWD");
                HWSPWCHIMSProxy proxy = new HWSPWCHCmdIMSProxy();
                proxy.setINMSG(input);
                // Run the IMS service.
                proxy.execute();

        }
```

17

IMS TM Resource Adapter

- Enterprise WorkLoad Mangement (EWLM)
  - Built into z/OS V1 R8 or OA12935 for z/OS V1 R7
- Implementation
  - WebSphere Application Server V6.1 EWLM enable instrumentation
  - IMS V10 OTMA
  - IMS TM Resource Adapter obtains EWLM correlator from WAS
    - CM0 Send_Receive and all CM1 flows will be instrumented by EWLM
- Benefits
  - Ability to track workflow from

```
LLLL | IRM | OTMA User Data (EWLM 64 byte correlator) | LLZZ TRANCODE data / EOM
```

18

The Enterprise Workload Management (EWLM) support requires the EWLM CICS/IMS Support that is built into z/OS Version 1 Release 8 or provided in APAR OA12935 for z/OS Version 1 Release 7.

IMS TM Resource Adapter obtains the EWLM correlator from WebSphere Application Server, then passes the correlator to IMS by inserting the correlator into the OTMA prefix for the transaction. When the transaction returns to IMS TM Resource Adapter after processing is complete, IMS TM Resource Adapter reports the status of the transaction to WAS.

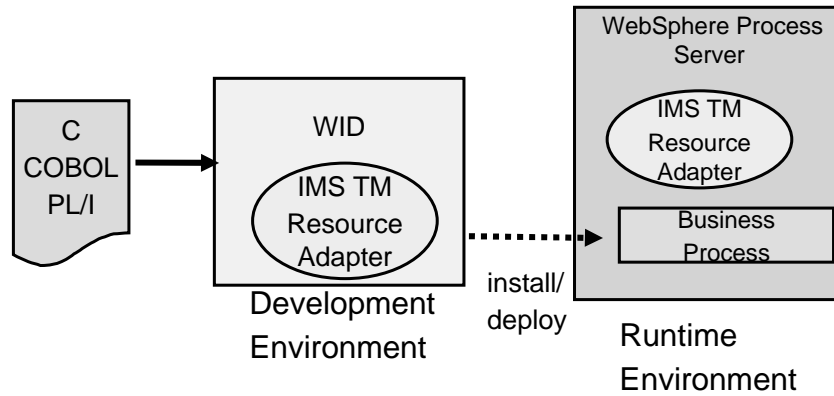If the incoming transaction does not contain an EWLM correlator, the existing function is unchanged.

CM0 Send Only flows will not be instrumented by EWLM. IMS TM Resource Adapter does not receive a correlator for CM0 send only transactions. Therefore, the transaction will be sent to IMS with the correlator field zeroed out, and IMS will call the existing non-EWLM version of the WLM services

If the correlator is provided then IMS TM Resource Adapter will pass it to IMS. If the correlator is not provided then the OTMA prefix field will be set to zero.

If an EWLM enabled transaction is routed from a V10 IMS to a backlevel IMS, the transaction passes through the backlevel system with the EWLM data unchanged and the existing WLM function is unchanged on the backlevel system. No APAR is required for prior versions of IMS.

IMS TM Resource Adapter

- IMS SOA Composite Business Application support
  - Provides multi-connection IMS Conversational Transaction support
    - Only available to IMS TM Resource Adapter clients
    - Supports WebSphere Process Server Business Process Flows

```
┌──────────┐        ┌──────────────────┐      ┌──────────────────────┐
│ C        │        │      WID         │      │ WebSphere Process    │
│ COBOL    │───────▶│  ┌────────────┐  │      │       Server         │
│ PL/I     │        │  │  IMS TM    │  │      │  ┌────────────────┐  │
└──────────┘        │  │ Resource   │  │······▶│  │    IMS TM      │  │
                    │  │  Adapter   │  │      │  │   Resource     │  │
                    │  └────────────┘  │      │  │   Adapter      │  │
                    └──────────────────┘      │  └────────────────┘  │
                       Development      install/│  ┌────────────────┐  │
                       Environment      deploy │  │   Business     │  │
                                               │  │   Process      │  │
                                               │  └────────────────┘  │
                                               └──────────────────────┘
                                                  Runtime
                                                  Environment
```

19

IMS SOA Composite Business Application support

- Migration
  - Replace Connection ID with Conversation Token
  - Only supports CM1 Synch level NONE or CONFIRM
  - IMS TM Resource Adapter
    - Used to pass IMS conversation token
- Considerations
  - IMS TM Resource Adapter Client must manage IMS conversation token
  - Supports only one IMS
- Benefits
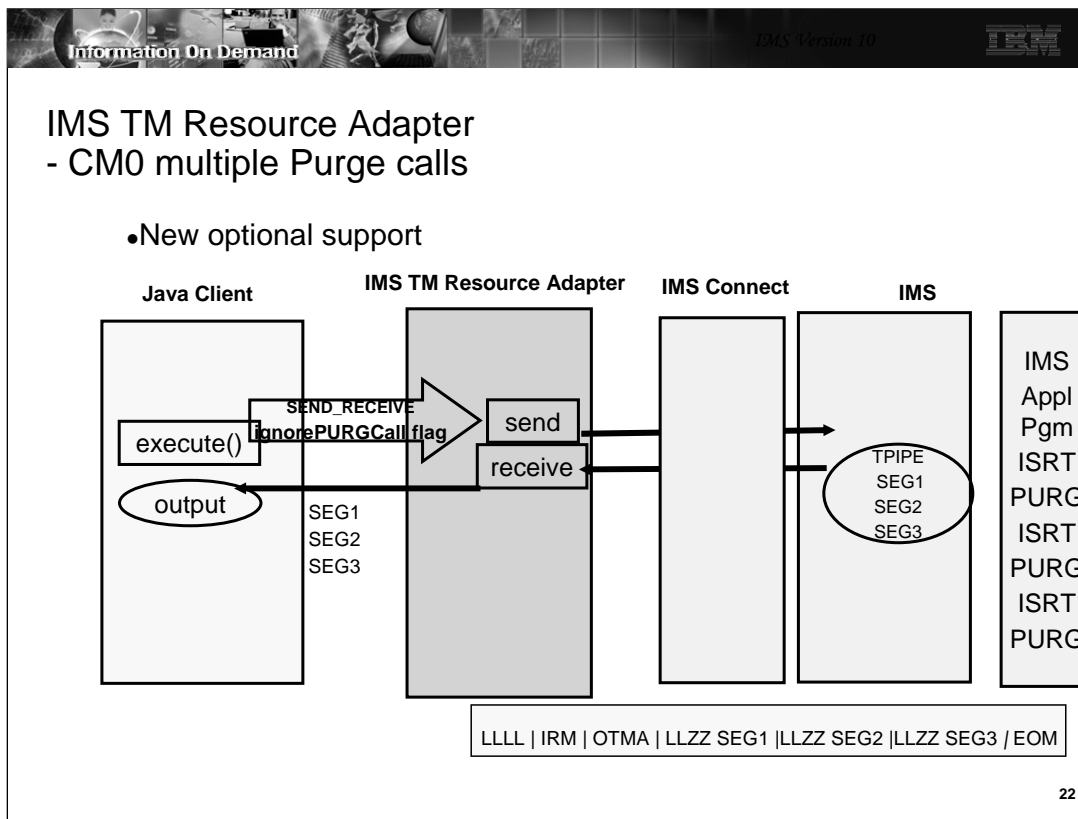 - Provides multiple connections IMS conversational transaction support

20

Because an IMS Conversational transaction is a commit mode 1 type of transaction, IMS Connect only supports commit mode 1 conversational input messages with synch level of none or confirm from IMS TM Resource Adapter client applications.
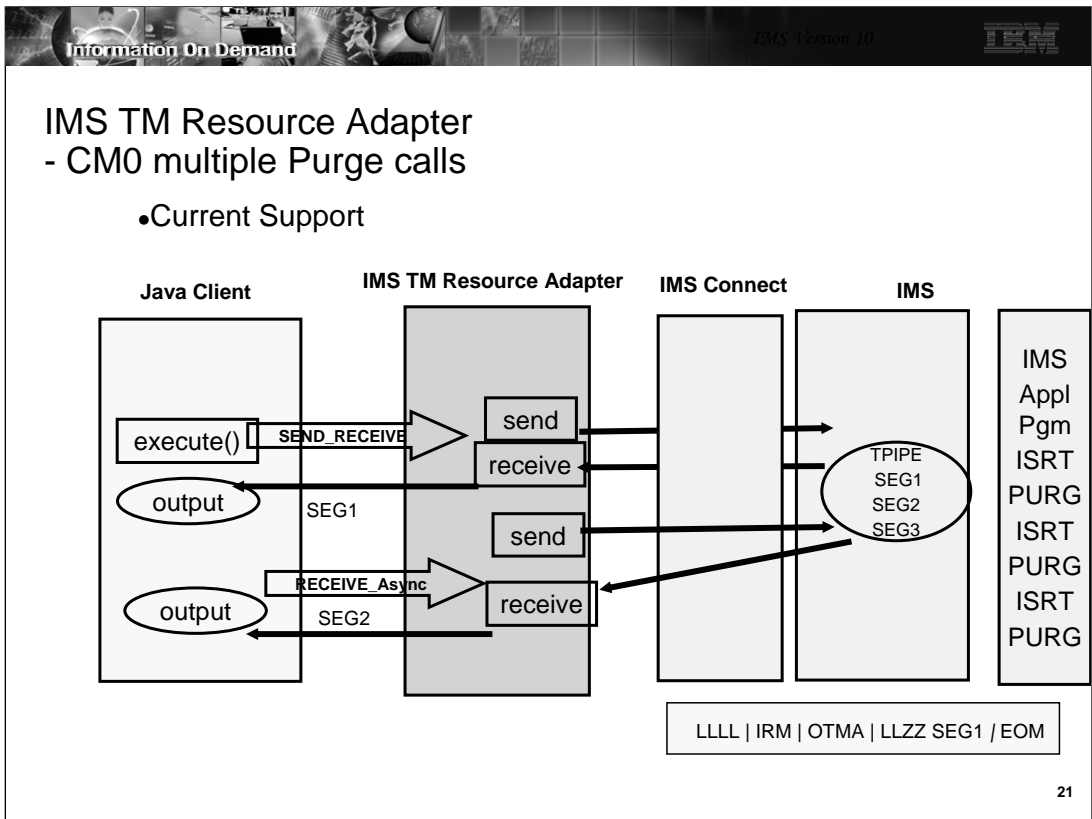
IMS TM Resource Adapter
- CM0 multiple Purge calls

When there are multiple ISRT and PURG calls in the IMS application, OTMA sends one response message with multiple output segments if the client application uses Commit Mode 1 interaction. If the client application uses Commit Mode 0, OTMA will send multiple output response messages, one for each PURG call. OTMA introduces an enhancement so that the customer using CM0 with multiple PURG calls in the TP PCB can ignore the PURG calls to generate one output message with multiple segments. This provides consistent output message processing for CM0 and CM1.

A new property called ignorePURGCall will be added to the IMSInteractionSpec.

IMS TM Resource Adapter
- CM0 multiple Purge calls

● New optional support

Java Client   **IMS TM Resource Adapter**   **IMS Connect**   **IMS**

execute()

SEND_RECEIVE
IgnorePURGCall flag

send

receive

output

SEG1
SEG2
SEG3

TPIPE
SEG1
SEG2
SEG3

IMS
Appl
Pgm
ISRT
PURG
ISRT
PURG
ISRT
PURG

LLLL | IRM | OTMA | LLZZ SEG1 |LLZZ SEG2 |LLZZ SEG3 / EOM

22

Commit Mode 1 is only supported using a shareable persistent socket.

IMS Connector for Java uses persistent socket connections to communicate with IMS Connect. A persistent socket connection prevents the overhead of opening and closing the socket for each use by an application. IMS Connector for Java provides two types of persistent socket connections.

1.Dedicated persistent – a dedicated persistent socket connection is restricted to Commit Mode 0 (Commit then Send protocol) and the client application provides the clientid to be used as the socket identifier.

2.Shareable persistent – a shareable persistent socket connection supports Commit Mode 0 and Commit Mode 1 (Send then Commit protocol). However, The client application does not provide a clientid for a socket identifier. IC4J generates a clientid.

The support for CONFIRM provides the ability for IC4J to respond ACK to the output message which activates the IMS sync-point process. In this case the client application is not running under a Global Transaction.

This provides the ability for IMS Connector for Java to notify IMS that it has received the IMS application output message without requiring a Global Transaction.

In this example the Java client is using a shareable persistent socket Commit Mode 1 Sync Level CONFIRM. IC4J generates HWSABC01 for a socket ID and sends the input message to the IMS application program via IMS Connect and OTMA. To provide a message queue anchor for the Commit Mode 1 output message OTMA creates a transaction pipe (TPIPE) and will use the TCP/IP generated Client Port number provided by IMS Connect for the TPIPE name. When IC4J receives the output message from IMS Connect is responds with an ACK. IMS Connect will send the ACK to OTMA to activate the IMS sync point process.

Note the Java Client application cannot respond with an ACK.

IMS TM Resource Adapter

- CM0 multiple PURG calls
  - Provides CM1 and CM0 consistent output message processing
  - IMS V10 OTMA optional output processing enhancement
- Implementation
  - IMS TM Resource Adapter Client
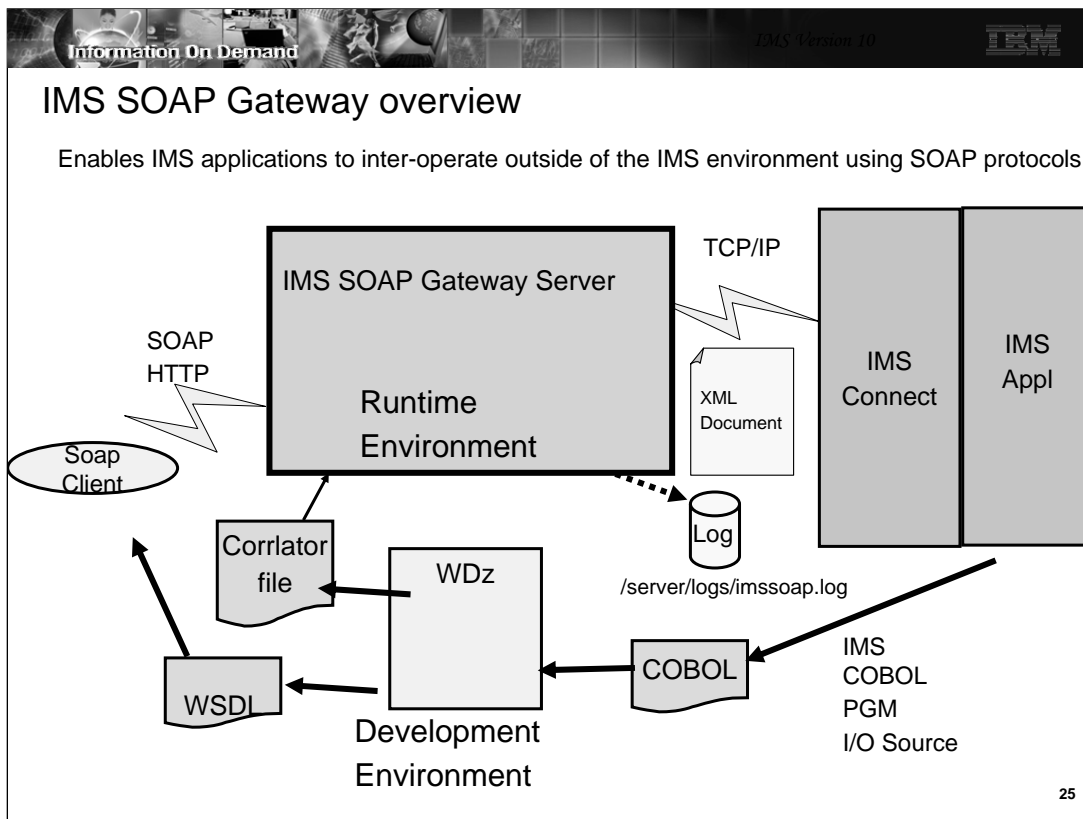   - IMSInteractionSpec new ignorePURGCall property

23

When there are multiple ISRT and PURG calls in the IMS application, OTMA sends one response message with multiple output segments if the client application uses Commit Mode 1 interaction. If the client application uses Commit Mode 0, OTMA will send multiple output response messages, one for each PURG call. OTMA introduces an enhancement so that the customer using CM0 with multiple PURG calls in the TP PCB can ignore the PURG calls to generate one output message with multiple segments. This provides consistent output message processing for CM0 and CM1.

A new property called ignorePUGRCall will be added to the IMSInteractionSpec.

IMS SOAP Gateway

24

IMS SOAP Gateway is a separate product that is available free of charge from the IMS Web site

IMS SOAP Gateway overview

Enables IMS applications to inter-operate outside of the IMS environment using SOAP protocols

25

The IMS SOAP Gateway is an XML-based connectivity Web service solution that is an HTTP endpoint for a SOAP client and supports access to IMS applications in a Service-Oriented Architecture (SOA) environment. This diagram shows a SOAP client accessing IMS via the IMS SOAP Gateway Server. SOAP (simple object access protocol) is a protocol for exchanging the XML-based messages over a network using HyperText Transfer Protocol (HTTP).

To deploy an IMS application as a Web service, you need to create a WSDL (Web service description language) file.

A WSDL file is an XML document that describes a Web service. WSDL files are used by others (for example, the client that invokes the service) to discover the service and to understand how to invoke the service. It specifies the location of the service and the operations that the service exposes.

WebSphere Developer for zSeries (WDz) can be used to generate a WSDL file.

IMS SOAP Gateway Deployment Utility

- The Deployment Utility has the following Options.

```
===============================================================================
Enable your IMS application as a Web Service :
  Task 1: Enable your IMS application as a Web Service from start to finish

 Administrative tasks :
  Task 2: Start IMS SOAP Gateway
  Task 3: Stop IMS SOAP Gateway
  Task 4: Update IMS SOAP Gateway properties
  Task 5: Create, Update or View correlator properties for Web Service
  Task 6: Create, Update, Delete or View connection bundle
  Task 7: Deploy the WSDL file
  Task 8: Generate Java client code
  Task 9: Undeploy Web Service
  Task 10: Exit deployment utility
===============================================================================
> Enter your selection here:
```

26

The IMS SOAP Gateway also provides a deployment utility that generates the connection bundle, the correlator file, and the Server properties file. The connection bundle specifies the connection and security properties between IMS SOAP Gateway, IMS Connect, and IMS. The correlator file specifies transaction, and runtime properties and the information that the IMS SOAP Gateway needs to match incoming requests to the appropriate back-end IMS application. It also identifies the Connection Bundle. The Server Properties file is used to configure the IMS SOAP Gateway Server runtime environment

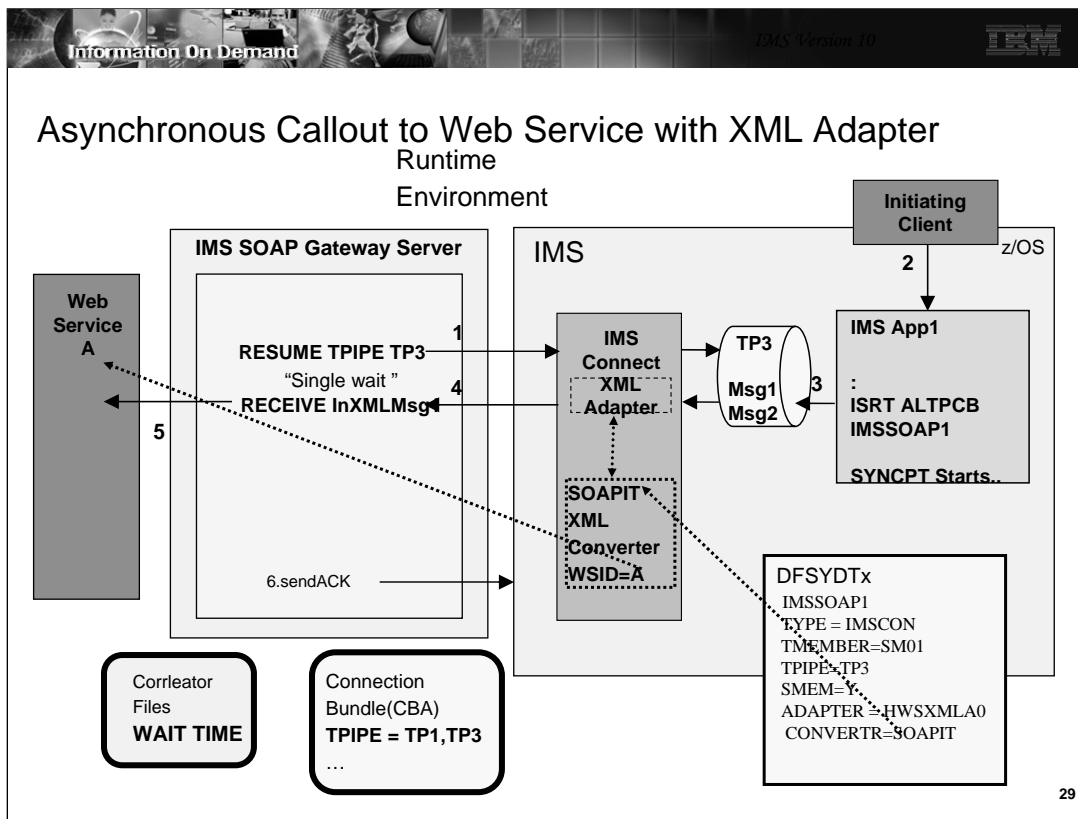The Deployment Utility runs in the Windows DOS Command prompt.

IMS SOAP Gateway V10 Enhancements

- Support on z/OS
- Asynchronous Callout to Web Service
  - Requires IMS V10
    - IMS Connect Resume TPIPE with Alternate Clientid
    - IMS OTMA Resume TPIPE Security
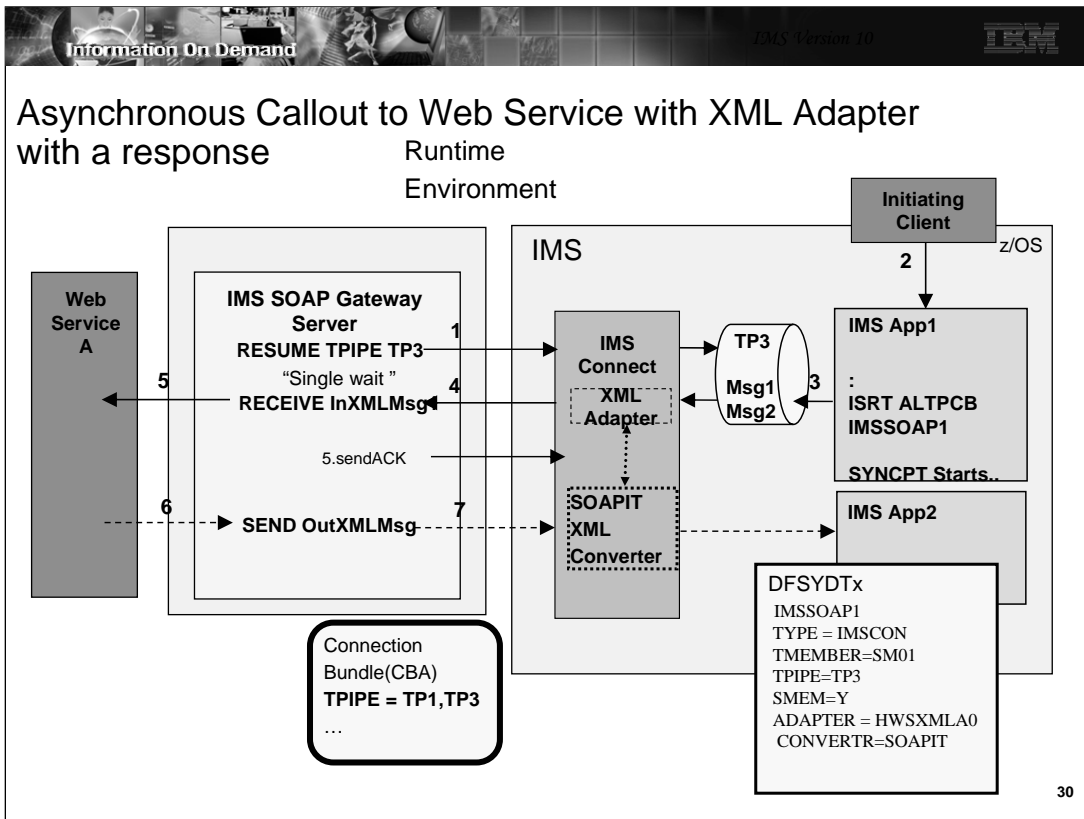   - IMS OTMA Destination Routing Descriptors

27

Support on zOS with IMS 10

Windows, AIX,zLinux

Web Service Clients, e.g. Microsoft .Net, SAP, Java, etc..

SOAP

IMS SOAP Gateway

XML

z/os

IMS

Connect

XML Adapter

OTMA

IMS App

XML

DB

Web Service Clients, e.g. Microsoft .Net, SAP, Java, etc..

SOAP

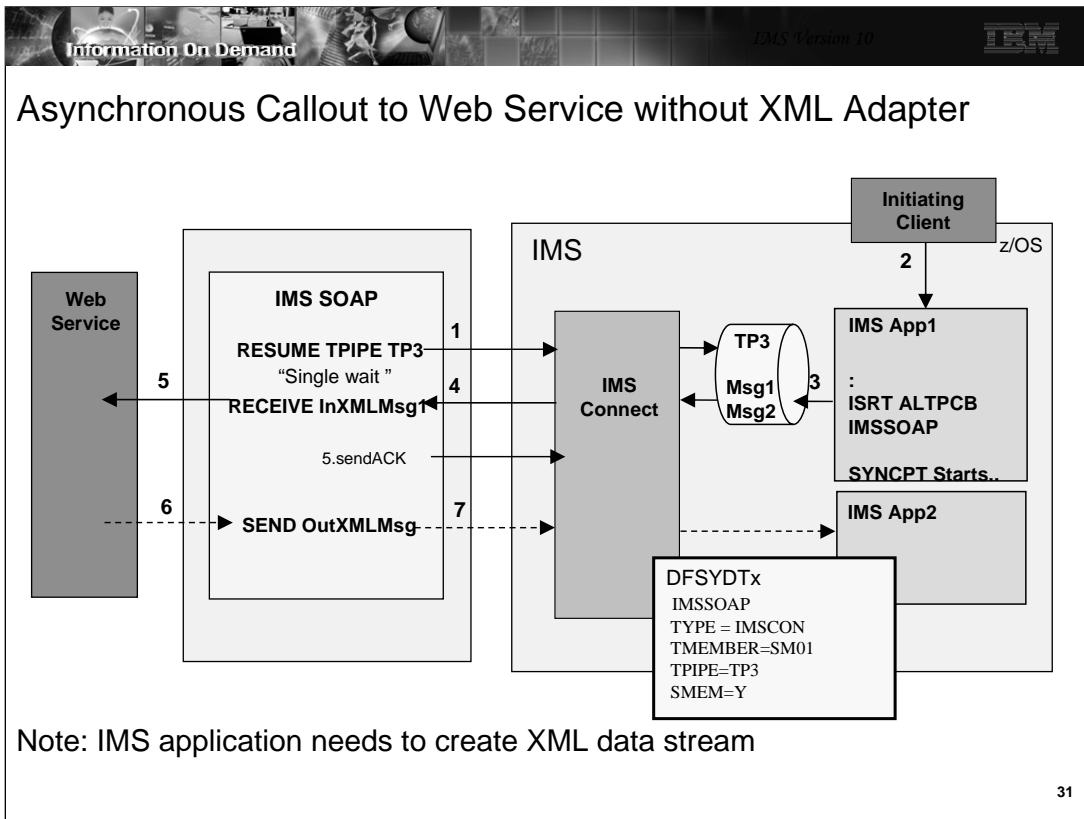IMS SOAP Gateway

28

Asynchronous Callout to Web Service with XML Adapter

In this scenario an IMS Application program does an insert to an ALTPCB whose destination is an OTMA Descriptor. The OTMA Descriptor is used to define the OTMA TPIPE for en-queue of the message.

Steps 1, 2 and 3 schedule the IMS application which results in the insert to ALTPCB defined by the OTMA Descriptor that represents IMS SOAP Gateway.

IMS SOAP Gateway will use the Resume TPIPE Alt-ClientID protocol to retrieve the message(1). IMS Connect will invoke the COBOL adapter to perform XML transformation of the message and send the message to the IMS SOAP Gateway(4). IMS SOAP Gateway receives the messages and sends it to the Web Service(5).

An optional step is the Web Service creates a response to the message request. IMS SOAP Gateway sends the response to IMS (6). This results in the scheduling of an IMS Application program to process the response.

Asynchronous Callout to Web Service with XML Adapter with a response

In this scenario an IMS Application program does an insert to an ALTPCB whose destination is an OTMA Descriptor. The OTMA Descriptor is used to define the OTMA TPIPE for en-queue of the message.

Steps 1, 2 and 3 schedule the IMS application which results in the insert to ALTPCB defined by the OTMA Descriptor that represents IMS SOAP Gateway.

IMS SOAP Gateway will use the Resume TPIPE Alt-ClientID protocol to retrieve the message(1). IMS Connect will invoke the COBOL adapter to perform XML transformation of the message and send the message to the IMS SOAP Gateway(4). IMS SOAP Gateway receives the messages and sends it to the Web Service(5).

An optional step is the Web Service creates a response to the message request. IMS SOAP Gateway sends the response to IMS (6). This results in the scheduling of an IMS Application program to process the response.
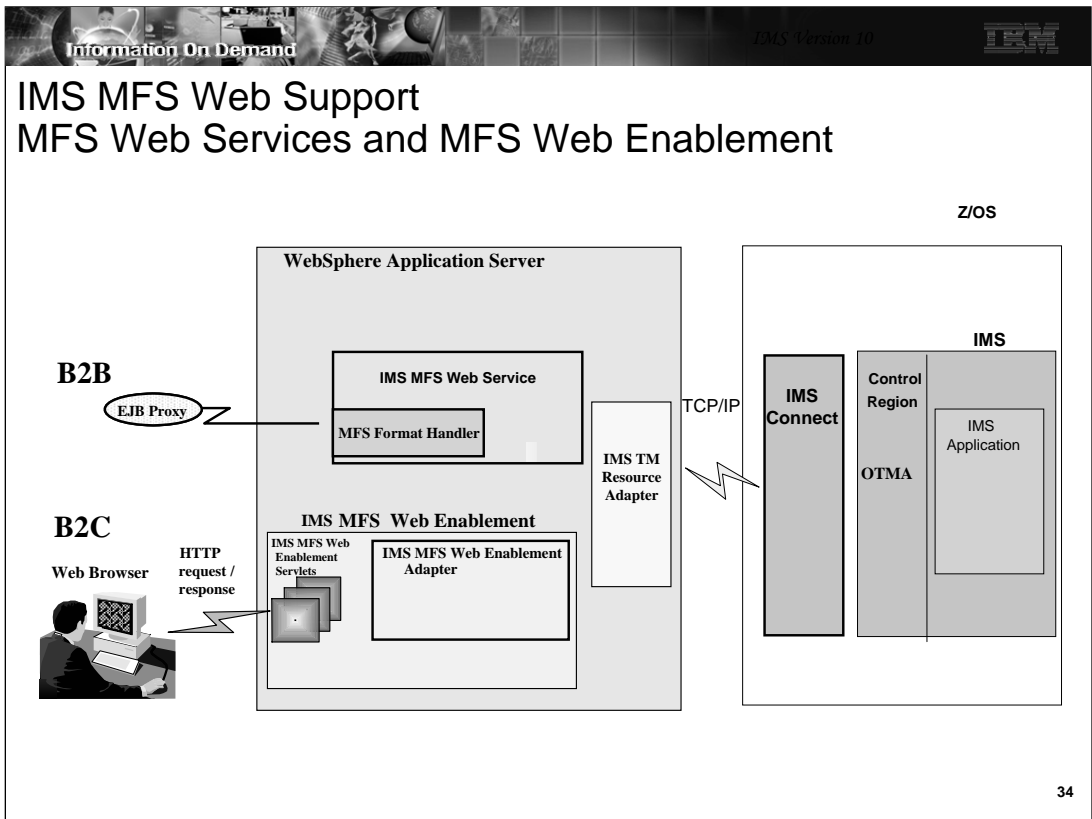
Asynchronous Callout to Web Service without XML Adapter

Note: IMS application needs to create XML data stream

In this scenario an IMS Application program does an insert to an ALTPCB whose destination is an OTMA Descriptor. The OTMA Descriptor is used to define the OTMA TPIPE for en-queue of the message.

Steps 1, 2 and 3 schedule the IMS application which results in the insert to ALTPCB defined by the OTMA Descriptor that represents IMS SOAP Gateway.

IMS SOAP Gateway will use the Resume TPIPE Alt-ClientID protocol to retrieve the message(1). IMS Connect will invoke the COBOL adapter to perform XML transformation of the message and send the message to the IMS SOAP Gateway(4). IMS SOAP Gateway receives the messages and sends it to the Web Service(5).

An optional step is the Web Service creates a response to the message request. IMS SOAP Gateway sends the response to IMS (6). This results in the scheduling of an IMS Application program to process the response.

Asynchronous Callout

- Migration
  - ISRT ALTPCB
    - OTMA Destination Routing Descriptor
- Benefits
  - Provides ability for IMS application to access a Web Service

32

The OTMA Callout Descriptors need to be created.

IMS MFS Web Support

33

IMS MFS Web Support
MFS Web Services and MFS Web Enablement

Z/OS

WebSphere Application Server

B2B

EJB Proxy

IMS MFS Web Service

MFS Format Handler

IMS TM Resource Adapter

TCP/IP

IMS Connect

IMS

Control Region

OTMA

IMS Application

B2C

Web Browser

HTTP request / response

IMS MFS Web Enablement

IMS MFS Web Enablement Servlets

IMS MFS Web Enablement Adapter

34

IMS MFS Web support provides two solutions to re-use existing MFS-based IMS business logic.

1. IMS MFS Web services support allows you to submit MFS-based IMS transactions as Web services.

2. IMS MFS Web enablement support replaces the MFS online processor and allows you to seamlessly navigate MFS-based Web pages that all have a similar look and feel.

MFS Web Enablement
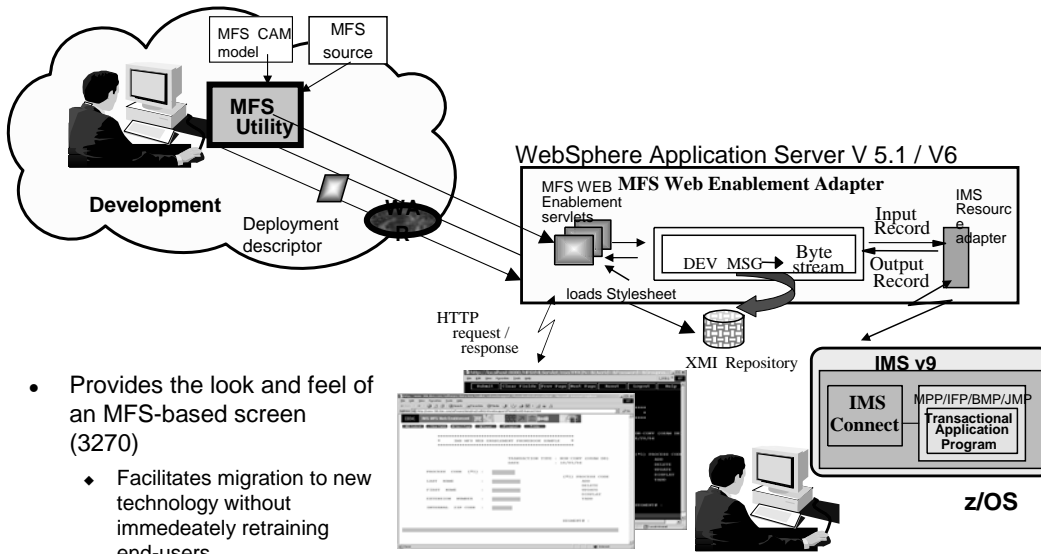
- Provides the look and feel of an MFS-based screen (3270)
  - ◆ Facilitates migration to new technology without immedeately retraining end-users

35

MFS Web Services – SOA support

- A capability that takes advantage of MFS source to generate web services

36

IMS DB Resource Adapter

37
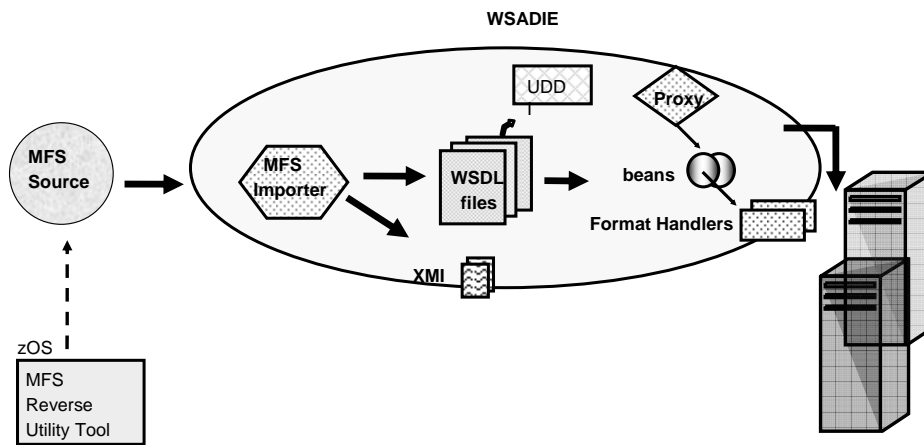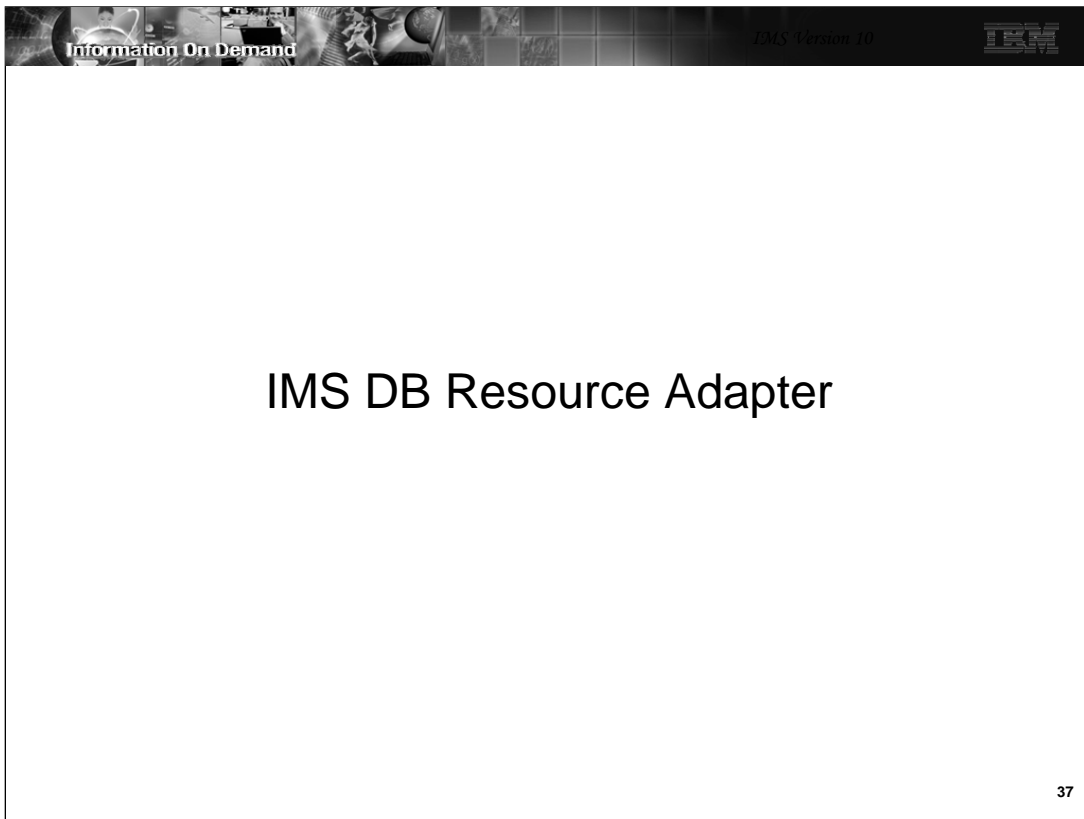
The IMS DB Resource Adapter is a function of IMS that enables a programmer with minimal IMS knowledge to write Java application programs that access IMS databases

IMS DB Resource Adapter
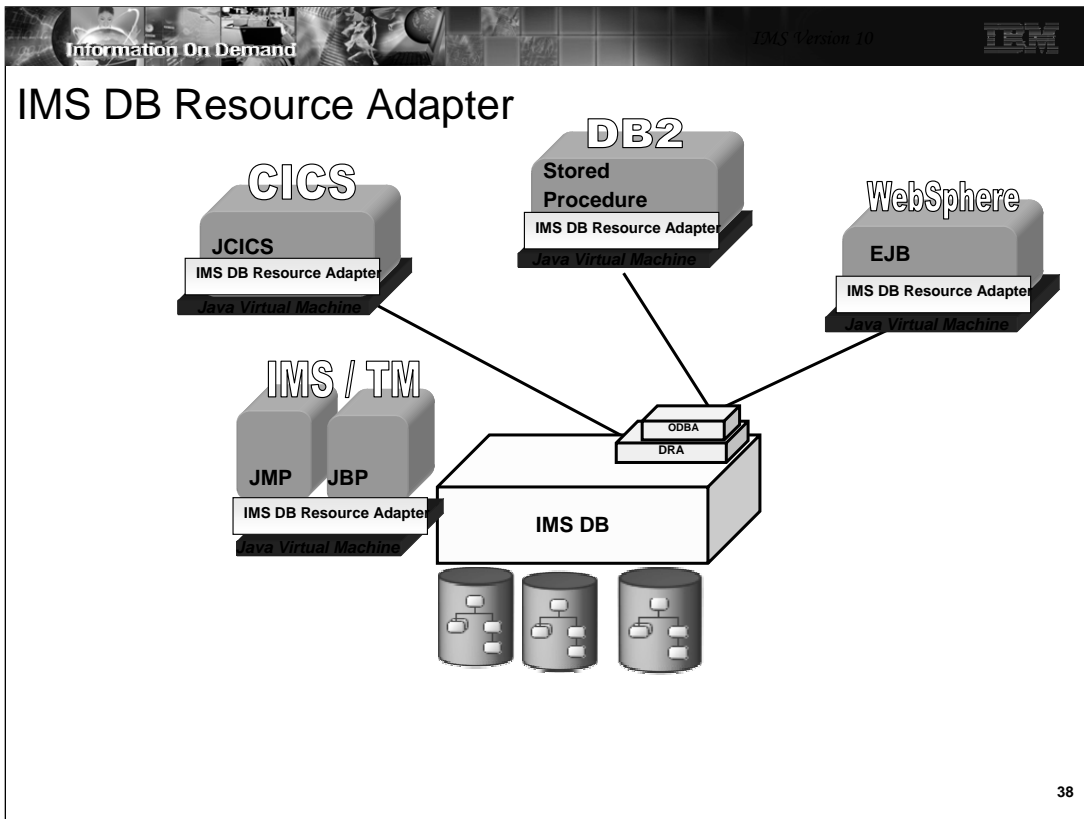
The IMS DB Resource Adapter enables JDBC access to IMS DB from IMS TM JMP/JBP environments, CICS Java application, DB2 Java Stored procedure, and Enterprise Java Beans running on WebSphere distributed and z/OS environments.

IMS DB Resource Adapter

- DB2 Java Stored Procedure return DB2 Result Sets
- GSAM Metadata
- IBM SDK V5 for z/OS support

39

The IMS DB Resource Adapter has been enhanced to support DB2 Java Stored Procedures returning DB2 result sets instead of returning IMS result sets. GSAM access has been enhanced to use IMS Java Metadata generated by DLIModel Utility. Changes were also made to support IBM SDK V5 for z/OS.

DB2 Java Stored Procedure return DB2 Result Sets

- Implementation
  - IMS DB Resource Adapter
    - DB2 global temporary table (GTT)
  - DB2 Stored Procedures Client
    - Stored procedure clients are READ_ONLY
- Migration
  - Update Stored Procedure to specify type of DB2 Global Temp Table
  - Modify the output parameters to return a result set instead of individual fields
  - Update the client application to process the DB2 Result Set
- Benefits
  - Simplifies processing of IMS Results Sets data
  - Provides data independent processing
  - Consistent Result Set Processing with DB2

40

The IMS DB Resource Adapter has been enhanced to create a DB2 global temporary table (GTT) in the DB2 Stored Procedure Address Space and moves the IMS data returned from IMS ResultSet into the DB2 GTT. The DB2 Result Set returned to the client application will iterate the DB2 GTT for result set processing.

The stored procedure clients are READ_ONLY. This means that the client cannot update or delete IMS data using the DB2 result set returned to it by the stored procedure because it is not possible send a change in a DB2 temp table back to an IMS database

Existing DB2 Stored Procedures using the IMS V10 DB Resource Adapter can continue to run without modifications

To update existing stored procedures to return DB2 result sets instead of individual fields you need to update the stored procedure to specify the use of a created or declared type of DB2 Global Temp Table. After the Stored Procedure application queries the IMS database the IMS DB Resource Adapter API is used to convert the IMS ResultSet to a DB2 Result Set using the GTT.

The output parameters defined for the Stored Procedure need to be changed to return a result set instead of individual fields.

The client application needs to be modified to accept a DB2 result set instead of IMS individual fields.

The use of DB2 Results Sets provides a consistent and easier programming interface for processing output from a DB2 Stored Procedure that uses IMS JDBC Connector to access IMS data.

IMS DB Resource Adapter

- GSAM Metadata support
  - Ability to utilize the IMS Java data conversion routines
  - Ability to utilize IMS Java DLIDatabaseView metadata

- Migration
  - Existing Java Batch Processing applications not affected
  - To convert an existing application
    - Use IMS V10 DLIModel utility to create GSAM DLIDatabaseView
    - Use new GSAM fields generated by DLIModel utility
    - Use new GSAM API

- Benefits
  - Automatic data conversion is done by the Java class libraries
  - Supports all data types to be stored into and read from a GSAM database record

41

IMS DB Resource Adapter support for GSAM was enabled by Small Programming Enhancement PQ93785/UQ93241. This provided a Java Batch Processing application program with a Java API to open read write and close GSAM databases. However , the application was required to know the complete details of the GSAM database

Given a DLIDatabaseView name, the IMS Java class library will use the metadata information to capture the correct sequence of bytes in the record, read that sequence of bytes, and convert that sequence of bytes into the appropriate data type as defined by the metadata.

Java Batch Processing applications that use the IMS V9 GSAM database support will not be affected

The DLIModel utility for IMS V10 provides new GSAM record & field options. This will be described in the DLIModel utility section.

IMS V10 GSAM enables all of the supported data types to be stored into and read from a GSAM database record and provides automatic data conversion on behalf of the application. GSAM database support is now consistent with the IMS Java class libraries support for all other IMS database types.

IMS DB Resource Adapter

- IBM SDK V5 for z/OS support
  - ◆ Contains a re-engineered Java 2 virtual machine
    - ▪ Provides cache class sharing
      - • Replaces persistent reusable function
    - ▪ z/OS APAR OA11519
      - • Cache class sharing
    - ▪ IBM 31-bit and 64-bit SDKs for z/OS, Java 2 Technology Edition, Version 5 SDK and Runtime Environment User Guide
      - • http://www-03.ibm.com/servers/eserver/zseries/software/java/pdf/sdkguide.zos.pdf
  - ◆ IMS Java Dependent Region Programming Model
    - ▪ IMS Java and OO COBOL Interoperability
      - • GU and CHKP processing

42

z/OS delivers a complete Java 2 Software Developer Kit (SDK).

Previous versions of the SDK provided a reusable function to support transactional runtime environments like IMS. This capability allowed the Java Virtual Machine (JVM) to be initialized during IMS Java dependent region startup and to be "re-set" after the IMS application program completed processing. This avoided the overhead of loading the JVM for each IMS appplication program schedule.

The new SDK provides a Class Sharing capability to replace the persistent reusable function.

z/OS APAR OA11519 is recommended for cache class sharing.

The User Guide can be downloaded from the specified URL.

IMS message GU/CHKP processing was altered for Java Dependent Regions (JDRs) when JDRs were initially implemented. For JDRs, an IMS message GU meant to get the message only, do not perform sync point processing. CHKP processing meant perform sync point processing only, do not get the next message. In the standard IMS model, message GU and CHKP meant perform sync point and retrieve the next input message.

For IMS V10, IMS JDRs GU/CHKP processing will be consistent with the IMS standard model.

IBM SDK V5 for z/OS support

- Migration for IMS Java Dependent Regions setup
  - ◆ Xresettable is not supported
    - Must remove **-**Xresettable , -Xjvmset , -Xscmax
  - ◆ XPLINK= and JVMOPWKR= are obsolete and will be ignored if specified
  - ◆ The following JVM options are not supported
    - "-Dibm.jvm.shareable.application.class.path="
    - "-Dibm.jvm.trusted.middleware.class.path="
  - ◆ Pathname for the application class files and IMS Java ".jar" files:
    - -Djava.class.path**=**

      or
    - -Xoptionsfile**=**
  - ◆ ENVIRON=
    - LIBPATH= change to SDK V5
    - DEBUG= can be removed
      - -Xdebug **e**nvironment variable can be specified

43

The serial reusability feature of the IBM SDK for z/OS, version 1.4.2 (31-bit) and earlier is not supported. If you specify –Xresettable -Xjvmset or -Xscmax the JVM will issue an error message and will not start.

The JVM shared library libjvm.so is installed in directory jre/bin/j9vm.

The IMS Java dependent region ENVIRON= member needs to have the LIBPATH changed to LIBPATH=JavaHome/bin/j9vm:JavaHome/bin:ImsjavaPath

ENVIRON= can now be used to specify any Java environment variables. This can be used to replace the DEBUG= parameter. The Java environment variables are presented as Java system properties at runtime and are therefore accessible by a Java application running in the JDR.

-Xdebug starts the JVM with the debugger enabled. By default, the debugger is disabled

IBM SDK V5 for z/OS support

- Migration for IMS Java Dependent Regions programming model
  - Current applications do not need to change unless:
    - Applications that use U118 Abend for ROLLBACK
    - Applications that use explicit CHKP call
      - Now receives next input message after CHKP call

44

An IMS Java application that does not issue explicit ROLLBACK but uses U118 abend for rollback processing must be changed. An IMS Java application that uses the IMS Java hierarchical database interface for CHKP call processing must be able to process the next message off the message queue.

IBM SDK V5 for z/OS support

- Migration for IMS Java Application Programs
  - com.ibm.ims.application.IMSApplication is deprecated
    - Recommend begin changing existing IMS Java applications
  - com.ibm.ims.base.DLISecondaryIndexInfo class has been removed
    - DLIModel generated meta data
      - No impact unless application explicitly used the class
  - com.ibm.ims.db.SecondaryIndexInfo class has been renamed
    - com.ibm.ims.base.SecondaryIndexInfo class
      - DLIModel generated meta data
      - No impact unless application explicitly used the class

45

The IMS DB Resource Adapter class com.ibm.ims.application.IMSApplication is deprecated in IMS Version 10.

A class marked as deprecated is considered obsolete but still works in IMS Version 10.

You do not have to change your applications, but it is highly recommended.

The com.ibm.ims.base.DLISecondaryIndexInfo class has been removed from the library. This will only impact you if you did not use DLIModel to generate the meta data classes (the database view) or if you use the DLISecondaryIndexInfo class explicitly in your code.

The com.ibm.ims.db.SecondaryIndexInfo class has been renamed to com.ibm.ims.base.SecondaryIndexInfo.

You will be impacted only if you use the class directly in your code. The metadata that is generated by the DLIModel utility is not affected.

IBM SDK V5 for z/OS support
- IMS Java Application sample API

Current

remove

public class CustomerApplication extends IMSApplication {

 public static void main(String args[]) {

    CustomerApplication myapp = new CustomerApplication();

    myapp.begin();

remove

Modified

public class CustomerApplication {

 public static void main(String args[]) {

    CustomerApplication myapp = new CustomerApplication();

46

The applications that subclass IMSApplication can be modified as follows:

Remove the "extends IMSApplication" from the class declaration line.

For example, "public class CustomerApplication extends IMSApplication"

becomes "public class CustomerApplication".

The main method of the application no longer needs to call the IMSApplication.begin() method.

Instead, the main method can directly call the public void doBegin() method or simply move the logic from the doBegin() method to the main method and delete the doBegin() method.

IBM SDK V5 for z/OS support

- Benefits
  - IMS participates in industry standards
  - Can be used with existing IMS Java applications
 - Reduces virtual memory consumption
  - Reduces JVM startup time

47

IMS DLIModel utility

48

DLIModel Utility

GUI
(Free
Eclipse
Plug-in)

Eclipse
3.0.1
3.0.2
WDz

BPXBATCH
DLIModel
Utility

49
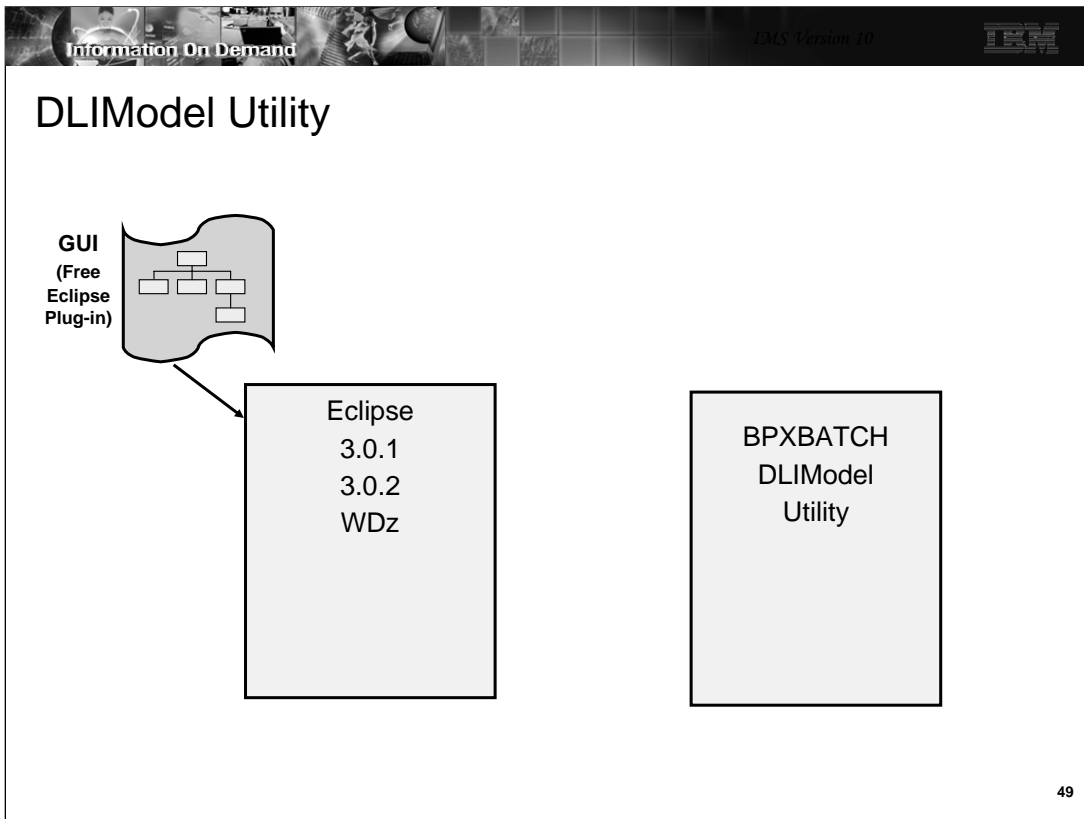
Two versions of the IMS DLIModel utility are available:

1. An IMS-shipped version that runs from System Services or from the z/OS® BPXBATCH utility

2. A technology preview IMS Web free download version that runs as a plug-in to Eclipse

The GUI can be installed in an Eclipse 3.0.1 or 3.0.2 level tool. It can also be installed in WebSphere Developer for z IDE.
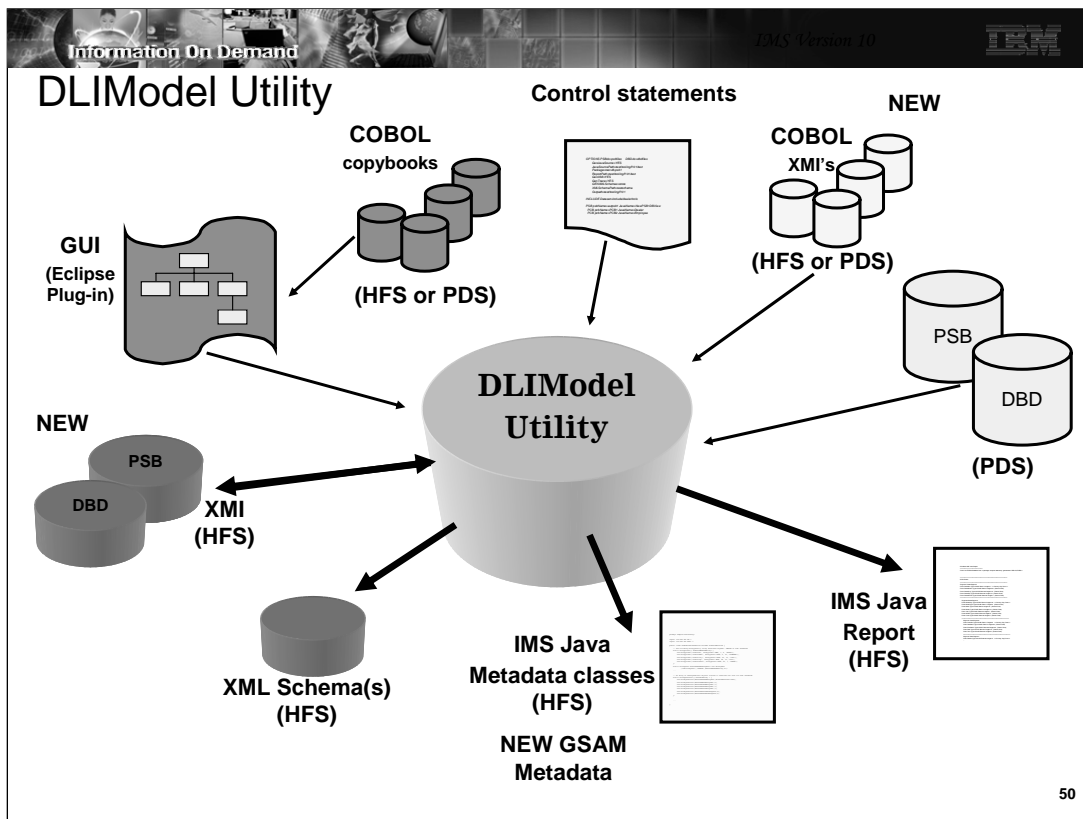
DLIModel Utility

The IMS DLIModel utility has been enhanced to generate XMI from PSB and DBD source. The generated XMI can also be used as input to the DLIModel utility.

GSAM now uses the GSAMDLIDatabaseView IMS Java class for metadata information about the GSAM database. The DLIModel Utility now supports GSAM databases.

Input:

Shows the inputs and outputs of the DLIModel utility. The actions of the utility are directed by control statements that you supply. PSB and DBD source members are read from their PDS or PDSE data sets and parsed by the utility to build an in-memory object model of the database structure and the PSB's view of that structure.

Note IMS COBOL copybooks can only be processed by the GUI and the BPXBATCH utility can only process COBOL XMI representations of the COBOL copybooks.

Output:

The utility generates various outputs that were requested through control statements. You can specify to have an IMS Java metadata class be generated for the PSB processed, together with a corresponding easy-to-read DLIModel Java Report for the Java programmer to use.

You can specify an XMI description of the entire in-memory model (one description covers the PSB and all DBDs processed in the run).

You can also request a detailed trace file of the utility execution if one is necessary for problem resolution.

IMS Java metadata classes

The DLIModel utility produces the necessary metadata classes needed to develop IMS Java applications. However, the Java developer needs only to reference the DLIModel Java Report for information about the classes.

DLI Model Java Report

The DLIModel Java Report summarizes the structure of the IMS databases in a way that allows you to create IMS Java applications and to code SQL queries against the databases. With the DLIModel Java Report, you do not have to interpret the syntax of the IMS Java classes or refer to the DBD or PSB source.

XMI Description of databases

An XMI file, written in UTF-8 encoding, is produced by the utility if you specify genXMI=YES in the OPTIONS control statement. It describes all of the PCBs and their referenced DBDs processed in the entire run of the utility. The XMI that is produced by the utility is based on a metamodel of IMS database defined in UML. This model is a package with a number of inheritance relationships to the OMG Common Warehouse Metamodel (CWM). However, only the IMS package itself is included and used in the DLIModel utility.

XML Schema

The generated XML schema, written in UTF-8 encoding, is an XML document that describes an IMS database based on a PCB. An XML schema is required to retrieve or store XML in IMS. IMS uses an XML schema to validate an XML document that is either being stored into IMS or being retrieved from IMS. The XML schema, not the application program, determines the structural layout of the parsed XML document in the database during storage and the of the generated XML document during retrieval.

DLIModel Utility GUI Download

This is the IMS web site to download the DLIModel Utility Plug-in

DLIModel Utility GUI

- Migration
  - Eclipse version 3.0.1 or 3.0.2
  - Download DLIModel Utility plug-in
- Benefits
  - Direct import of COBOL Copybook
  - IMS Perspective for Integrated Development Environment

52

IMS XML DB

53

IMS XML DB allows applications to view a traditional IMS database as an XML database and to use an IMS database to store XML documents.

XML Database - Review

- Introduces a way to view/map native IMS hierarchical data to XML documents
  - ◆ Aligns IMS Database (DBD) with XML Schema
- Allows the retrieval and storage of IMS Records as XML documents with no change to existing IMS databases
  - ◆ Enables query of IMS data using XQuery

54
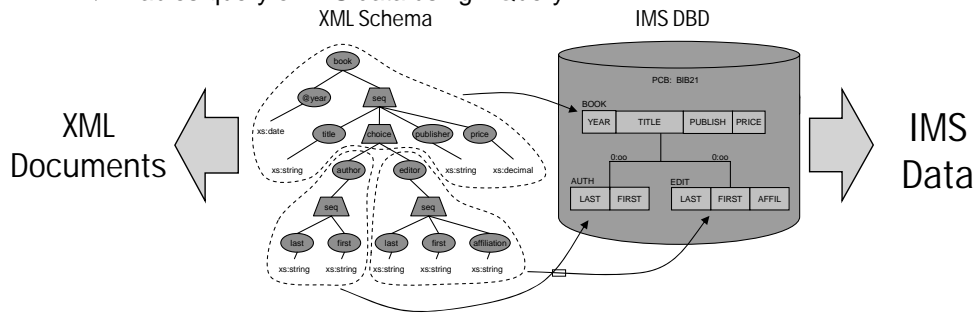
IMS DB Resource Adapter XML APIs

IMS V9 XML API

XML CLOB

```
<B>
  <f1>        </f1>
  <f2>        </f2>
  <f3>        </f3>
<B>
```

XML document

```
SELECT retrieveXML( B)
   FROM PCB C
   WHERE C.FIELD….
```

```
<xsd:annotation>
   <xsd:appinfo>
      <ims:DLI mode="store"
PSB="MYPOPSB"
PCB="MYPOPCB"
<xsd:element name="A">
<xsd:element name="field1"
```

DLIModel Generated XML schema

IMS V10 XQUERY API

```
<B>
  <f1>        </f1>
  <f2>        </f2>
  <f3>        </f3>
<B>
```

```
SELECT retrieveXML( B,
   'for $x in /B
   where  $x/@f1 = z
   return  <namelist>
             {$x/B/f4}
          </namelist>')
   FROM PCB C
   WHERE C.FIELD…
```

Af1(x)

B1   B2   D1   D2

C1   C2f6(Y)   **IMS DB**

```
<xsd:
   <xsd:appinfo>
      <ims:DLI mode="store"
PSB="MYPOPSB"
PCB="MYPOPCB"
<xsd:element name="A">
<xsd:element name="field1"
```

DLIModel Generated XML schema

XQUERY - an evolving
standard API designed to query and format XML data

IMS XQUERY processor
- Extends IMS XML DB
- Uses XML schemas generated by DLIModel

55

IMS DB Resource Adapter XQuery API

- XQuery is a language for querying XML data
  - Note XQuery is currently a read only language
- IMS XQuery extends IMS V9 XML database support
  - View entire IMS database as XML Document and return specific IMS data
- Supports the XQuery 1.0 and XPath 2.0 Data Model
  - XQuery is based on the structure of XML to provide query capabilities
- Extends IMS DB Resource Adapter API

  SELECT retrieveXML(, **<xquery>**)

  FROM <pcb.segment>

  WHERE <predicate>
- Package in it's own jar file:
  - imsxquery.jar

56

XQuery is a functional programming language that was designed by the World Wide Web Consortium (W3C) to meet specific requirements for querying XML data.

XQuery is based on the structure of XML and leverages this structure to provide query

capabilities for the same range of data that XML stores.

The IMS DB Resource Adapter XQuery support extends the *retrieveXML* User Defined Function (UDF) by adding a second parameter. The second parameter allows the passing of an XQuery 1.0 expression.

The expression is evaluated relative to the retrieveXML context and returned to the result set as a CLOB value.

This implementation views the entire IMS DB as an XML document and enables the return of specific IMS data based on the XQuery.

For IMS XQuery support the XQuery 1.0 and XPath 2.0 Data Model serves two purposes.

First, it defines the information contained in the input to be used by the IMS XQuery processor.

Second, it defines all permissible values of expressions in the XQuery, and XPath languages that can be evaluated by the IMS XQuery processor.

The IMS DB Resource Adapter is packaged in imsjava.jar.

The IMS XQuery function resides in a separate package (imsxquery.jar).

IMS DB Resource Adapter XQuery

- Benefits
  - IMS participates in industry standards
  - Can use Application Development tools that support XQuery
  - Can be used with existing IMS data and IMS XML Schemas

57