

Integration

Information Management software



IMS service stream enhancements

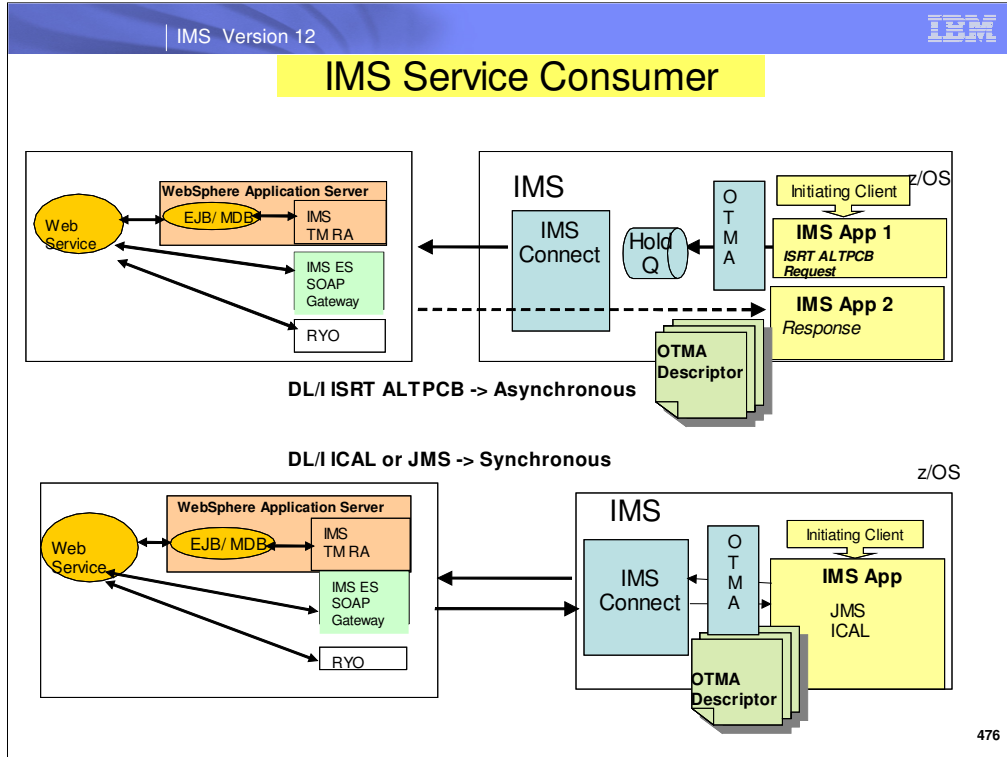
- **OTMA/Connect callout enhancements**
 - 10/11 Callout PM09696/PM09695/PM14812, PM12172/PM12142
 - 10/11 Callout Error Handling PM20541/PM20292
 - 10/11 Top Down support for Service Provider
- **Threadsafe support for IMS DBCTL**
- **Add CCTL TRANCODE/TASK ID to TYPE08 Log Record**
- **IMS 12 HALDB Catalog Metadata**
- **IMS Java**
 - 10 and 11 Persistent JVM in MPP, BMP, and IFP
 - 10 and 11 Open DB Type 2 Universal Driver for WAS for z/OS
 - 11 Universal JDBC Driver updates for JDBC Connection
 - 11 SQL support for HALDB, DEDB and HDAM databases
 - Type 4 Universal Drivers for IMS XML DB

IMS has been further enhancing IMS 10 and IMS 11 through the service process.

IMS service stream enhancements...

- **IMS Enterprise Suite**
 - Connect API for C and Java
 - SOAP Gateway
 - DLIModel Utility
 - Explorer for Development
- **IMS SOA Integration Suite**
 - IMS MFS Web Enablement DBCS support
 - IMS TM Resource Adapter MFS services
 - IMS 10/11 TM Resource Adapter Transaction Expiration support
 - Web 2.0 Solution for WebSphere sMash (web samples/doc)

Callout Enhancements



IMS 10 introduced callout support for IMS application program outbound access. This slide is a review of the asynchronous and synchronous callout function

Synchronous Callout Enhancements

- New ICAL AIB AIBERRXT for why ICAL failed
 - Benefit -extended reason codes provide additional diagnostic support
- New DFS4688E for ICAL internal processing error
 - ICAL times out with an AIBERRXT extended reason code
 - Benefit – simplify problem determination and reduce region wait time
- OTMA is enhanced to validate tpipe names more efficiently
 - Benefit - reduce the CPU cycles used for input transactions

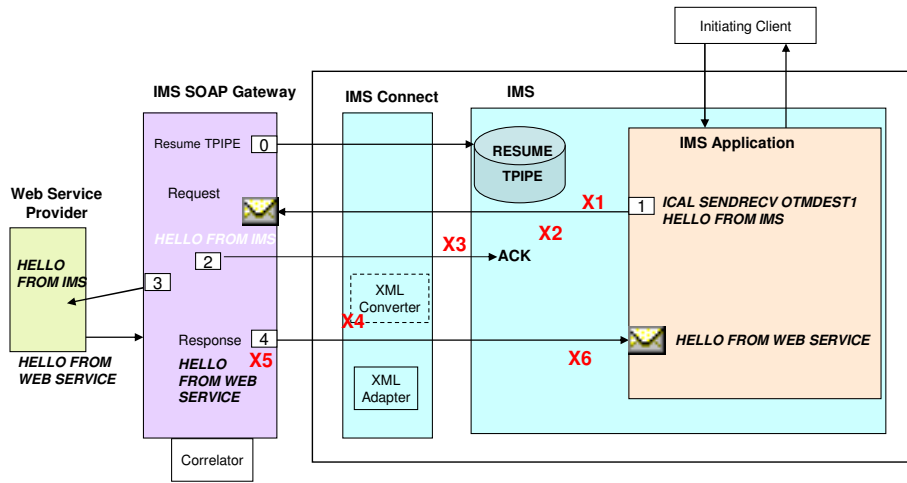
IMS 10 APAR PM20292
PTF UK61703
IMS 11 APAR PM20293
PTF UK62053

IMS introduced the DL/I ICAL for synchronous outbound access from an application program in IMS 10. The Callout Enhancements SPEs provides additional functions for IMS Connect customers using the ICAL for callout.

They include:

- IMS 10 PM20292 UK61703 / IMS 11 PM20293 UK62053
 - New ICAL AIB return codes /reason will be introduced for IMS Connect error response
 - New AIBERRXT extended reason codes will be introduced for ICAL
 - New DFS4688E to report an ICAL processing error. When this error occurs ICAL can be timed out with an extended reason code and a X'67D0' log record is written.
- OTMA has also been enhanced to not check the tpipe name if the message is being sent to a known tpipe. This reduces the CPU cycles for processing the OTMA transactions.

Synchronous Callout Enhancements...



X1: ICAL cannot be sent out
 X2: ICAL time out
 X3: late ACK received after time out
 X4: XML converter in error

X5: external server already commit before time out
 X6: external server already commit, but IMS fails to process the response

Synchronous Callout Enhancements...

- Enhancement to /DIS TMEMBER TPIPE SYNC command
- ENQCT DEQCT QCT INPCT OPT MODE SYNCOT NO-COT RTQ STATUS SMEM
 - RTQ new field
 - Number of queued Resume TPIPE requests
 - Replace NO-RSP with NO-COT
 - show the accumulated ICAL counts
- Benefit
 - Enhances information details for synchronous callout processing

APAR
IMS 10 PK80756/PM09695
IMS 11 PK91373/PM12142

479

The /DIS TMEMBER TPIPE SYNC command results in:
ENQCT DEQCT QCT INPCT OPT MODE SYNCOT NO-COT RTQ STATUS SMEM

New fields include:

- RTQ - The number of queued Resume TPIPE requests to be processed. This field is displayed only for the command issued with the SYNC keyword.
- Replace NO-RSP with NO-COT - The current number of ICAL messages received for this TPIPE. If the number is greater than or equal to 65535, it will be reset to 1.

10/11 Synchronous Callout Enhancements...

- Resume TPIPE for ICAL
 - Delayed ACK/NAK support
 - Benefit - Prevent Client hang when ICAL timeout occurs before ACK
- TPIPE cleanup function extended to ICAL messages
 - TPIPE will be deleted after 3 checkpoints
 - Benefit – Improves CPU processing and Memory management

APAR

IMS 10 PM09696/PM09695/PM14872

IMS 11 PM12172/PM12142

480

IMS 10/11 Callout enhancements also include: Delayed ACK/NAK support - IMS 10 UK57873 PM09696/PM09695 , IMS 11 PM12172/PM12142

After the client issues the Resume Tpipe request and retrieves the message the client sends an ACK to acknowledge receiving the message.

IMS OTMA enhancements include:

- Delayed ACK support which is used to prevent a Client hang when their ACK for Resume TPIPE message is received after OTMA has already timed out the TPIPE. IMS Connect will indicate to OTMA a "delayed ACK" and OTMA will reject a late or invalid ACK/NAK message with a new NAK message with sense code x'2B' so the client receives a response. This new NAK message will only be issued when TMAMDACK flag is set in the late or invalid ACK/NAK messages.
- The Tpipe cleanup function has been extended to support a tpipe with ICAL messages. The Tpipe used for Callout message will be deleted after 3 checkpoint if the following conditions are true:
 - No active CM0 and CM1 message on the tpipe
 - No hold queue messages on the tpipe
 - No trace is activated on the tpipe
 - Tpipe is not stopped
 - No active resume tpipe with AUTO or SINGLE WAIT option

10/11 Synchronous Callout Enhancements...

- IMS ES SOAP Gateway response data XML conversion error
 - IMS Connect Adapter for ICAL
 - HWSA0345E ADAPTER HWSXMLA0 ENCOUNTERED AN ERROR
 - The client socket connection is disconnected
 - OTMA is not informed and IMS application waits for ICAL timer to pop
 - Previous to APAR support
 - HWSA0345E ADAPTER HWSXMLA0 ENCOUNTERED AN ERROR
 - The client socket connection is disconnected
 - OTMA is not informed and IMS application waits for ICAL timer to pop
 - APAR support provides
 - HWSA0345E ADAPTER HWSXMLA0 ENCOUNTERED AN ERROR
 - The client socket connection is disconnected
 - OTMA is informed and IMS application receives AIB return/reason code
 - Benefit – simplify problem determination and reduce region wait time

APAR

IMS 10 PM20541 UK61453 and PM20292

IMS 11 PM20543 UK62199 and PM20293

481

The Callout Enhancements SPEs provide additional functions for IMS Connect customers using the ICAL for callout.

To enable the functions provided by IMS 10 PM20541 UK61453 /IMS 11 PM20543 UK62199 you will need to apply IMS 10 PM20292/IMS 11 PM20293.

Previous to this APAR, IMS Connect would discard Sync Callout responses from SOAP Gateway that encountered XML conversion errors without informing OTMA. This results in the IMS Application Program ICAL having to wait until time out to be notified. With the APAR IMS Connect will send a NAK to OTMA for sync callout response XML adapter errors so the ICAL doesn't have to wait for the timeout.

New ICAL AIB return code/reason code are provided to the IMS application program:

- rc X'108' rsn X'588' with any XML data that was converted is returned to IMS application program response area
- rc X'108' rsn X'58C' with no data returned to IMS application program response area

10/11 Callout Enhancements Resume TPIPE for Asynchronous and Synchronous

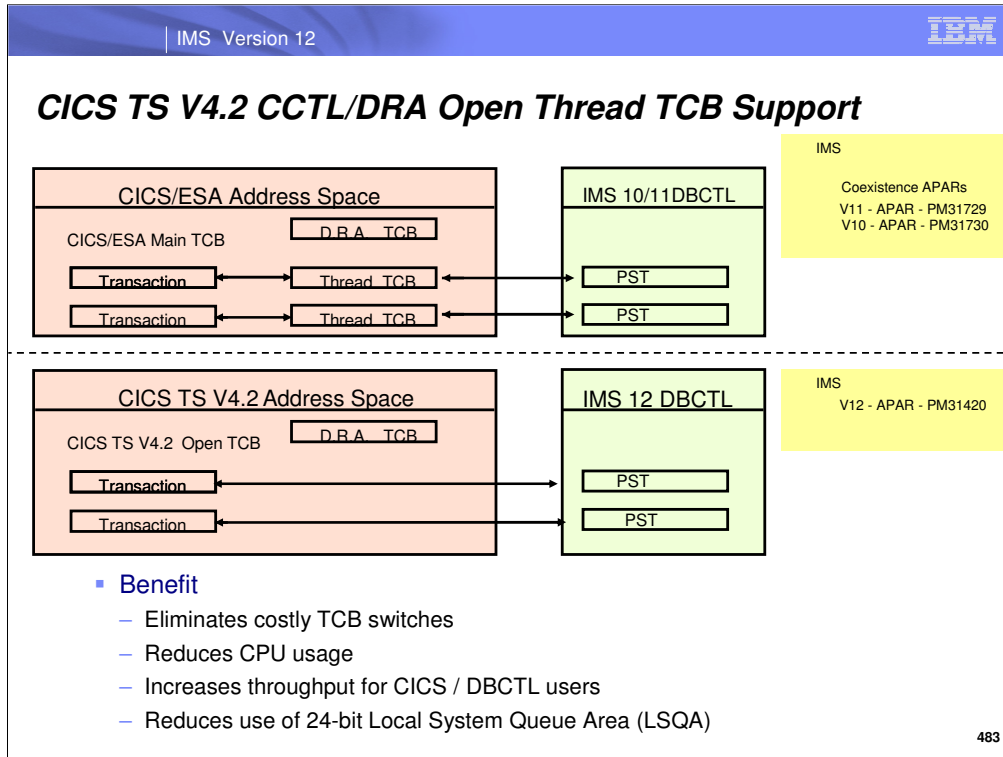
- OTMA Resume TPIPE enhancements
 - Client can queue multiple Resume TPIPE requests
 - Client can cancel a Resume TPIPE request for the Hold Queue
 - Resume TPIPE for NOAUTO or SINGLE-NO-WAIT options
 - IMS Connect informed when there is no message on the Hold Queue
 - IMS Connect will immediately send a timeout RSM to the client.
- Benefits
 - Multi-Resume TPIPE processing per Client
 - More efficient Hold queue processing and client notification

IMS 10 APAR PK80756
PTF UK50877
IMS 11 APAR PK91373
PTF UK55186

482

Review - A *transaction pipe* (tpipe) is a logical connection between a client and OTMA. For each tpipe, OTMA maintains a connection between the queue and the client. For callout processing OTMA creates a Hold Queue based upon a user defined tpipe name. On a Resume TPIPE request the client specifies the user defined Hold Queue tpipe name to obtain the IMS application program callout message for Asynchronous or Synchronous callout processing.

The callout enhancements enable a client to provide an 8 byte token on the Resume Tpipe to specify multiple resume tpipe requests and process them in order. A client can request the cancellation of a specific resume tpipe request for the hold queue based on the 8 byte token. When a resume tpipe token is not used, the multiple resume tpipe requests cannot be queued properly and the most current resume tpipe request will be the one processed by OTMA. The other resume tpipe requests will be ignored by OTMA. OTMA sends a protocol command to inform the OTMA client (IMS Connect) that the TPIPE's hold queue does not contain a message or response for the current Resume TPIPE request. IMS Connect notifies the Resume TPIPE client that no message is available.



Without this enhancement, the IMS CCTL DRA attaches dedicated thread TCBs in the CCTL address space. PSB schedule requests are assigned one of these thread TCBs. All subsequent thread related DL/I requests result in the application's task being suspended, processing switches onto the DRA thread TCB to complete the DL/I request. Upon returning, the application's task is resumed, switching processing off the DRA thread TCB which is then suspended awaiting the next DL/I request. This sequence of events is repeated for each DL/I call, syncpoint requests and thread termination at which time the DRA thread TCB becomes available for a new PSB schedule request.

IMS V12 CCTL DRA Open Thread TCB (OTT) enhancement provides the option for CCTL exploiters to direct the DRA not to attach dedicated DRA thread TCBs.

This avoids the overhead of TCB switching and lead to improved parallel processing. CICS TS 4.2 extends the threadsafe support to the DBCTL / DRA interface taking advantage of the DRA Open Thread TCB support.

IMS Type 08 LOG RECORD

IMS 11 APAR PM24076
PTF UK61382

- **DFSLOG08**
 - LINTSY2 Field
 - For CCTL - set value to Task ID and Trancode
 - For ODBA - set value to AER Thread Jobname
 - LINTSUB2 new flag field indicate LINTSY2 value
 - LINTCCTL CCTL Task id/Trancode
 - LINTAER AER Jobname
 - LINTCTSK new field added to end of 08 log record
 - Contains CCTL task id
- **Benefit**
 - Provides additional information for analysis of CCTL/ODBA threads

484

In order to offer better analysis of DRA/AER threads, additional information is necessary in the IMS TYPE08 log record in the form of the CCTL transaction code or job id, and the CCTL task id. For an AER thread, this additional information is the jobname of the AER thread. DFSDASPO has been modified to set the CCTL trancode/id or the AER jobname in field LINTSY2 in the TYPE08 log record. Code has been added to set new bits LINTCCTL or LINTAER as appropriate in new field LINTSUB2 in the 08 log record. Code has also been added to set new field LINTCTSK with the CCTL task id in the 08 record. Prior to this maintenance, field LINTSY2 was explicitly set to blanks for DBCTL and ODBA threads. DFSLOG08 has been modified to add new flag field LINTSUB2 with new flag bits LINTCCTL and LINTAER indicating that field LINTSY2 contains the CCTL trancode/id or the AER jobname. Additionally, new field LINTCTSK has been added to the end of the TYPE08 log record to contain the CCTL task id.

IMS Java

10/11 Persistent JVM in MPP, BMP, and IFP

- Provides the ability to use Java in IMS Dependent Regions
 - MPP,BMP,IFP
 - Add ENVIRON= and JVMOPMAS= parms to launch JVM
 - ENVIRON= path to z/OS Java JVM, IMS Java classes
 - JVMOPMAS= path to IMS Java subroutine
 - DB2 z/OS PK93123 Java SQL calls via IMS ESAF
 - RRS is not required
- Benefits
 - leverage Java code from existing COBOL applications
 - Resolves create/destroy JVM performance problem

IMS 10 PTF UK58284
 APAR PK82214
 IMS 11 PTF UK58285
 APAR PM00360

486

Customers who have need to leverage Java code from their existing COBOL applications have used Enterprise COBOL to create the interface between the programming languages. Enterprise COBOL, however, creates/destroys the Java Virtual Machine (JVM) every time the Java code is called from and returns to the COBOL code. This causes extremely poor performance in IMS MPP,IFP and BMP regions. This SPE resolves this performance issue by allowing IMS to manage the JVM which means that the JVM is only created during dependent region initialization and is only destroyed when the MPP/BMP/IFP region terminates.

To provide this capability the JCL parms (ENVIRON= and JVMOPMAS=) are now supported for MPPs, BMPs, IFPs. DB2 provides an SPE for JDBC call support via IMS ESAF.

The software requirements include:

For z/OS 1.9 or 1.10 LE (Language Environment) UK52873 APAR PK99010

For z/OS 1.11 or above LE (Language Environment) UK53437 APAR PM00482

For DB2:

DB2 z/OS V8 UK52961 APAR PK93123

DB2 z/OS V9 UK52962 APAR PK93123

10/11 Type 2 Universal Drivers

- Provides type-2 connectivity when accessing IMS in the same LPAR
 - IMS Java enabled Dependent Regions
 - IMS 10 PM13967, PM02734 UK57317 IMS 11 PK86498 UK57312
 - New Java Dependent Region Resource Adapter (imsutm.jar)
 - Provides Java access to IMS TM services
 - WAS z/OS
 - Uses ODBA
 - IMS 10 PM13216,PM14770 IMS 11 PK99686 UK63383 PK14766 UK63384
 - CICS
 - Uses CCTL
 - IMS 10 PM34894, PM38665, IMS 11 PM38666 , IMS 12 PM38667
- Benefits
 - Provides JDBC 3.0 and Variable length segments support
 - ODBA support for local transaction mode. This allows a local commit or backout to be made.

487

A new SPE provides Open Database Universal drivers (DB resource adapter ,JDBC and DLI) for type-2 connectivity when accessing IMS databases in the same logical partition (LPAR) from IMS java enabled dependent regions MPP,IFP,JMP,BMP and JBP regions. A new Java Dependent Region resource adapter `imsutm.rar` provides Java access to traditional IMS services such as message queue read/write control, program switching, unit of work control when running in a JMP or JBP. System DLI calls are also provided by the JDR resource adapter.

SPEs were also created to provide type-2 connectivity when accessing IMS databases in the same logical partition (LPAR) from WAS z/OS and CICS runtimes.

For the IMS TM runtimes, commit/rollback calls from the JDBC Connection or the DLI PSB interface implementations are not allowed. Two-phase commit (XA) transaction processing is not supported.

The IMS Universal drivers are built on industry standards and open specifications, and provide more flexibility and improved support for connectivity, data access methods, and transaction processing options

IMS Java Dependent Region Resource Adapter (imsutm.jar)

▪ **Java Classes**

- ApplicationFactory
 - Factory for creating Application objects
- Application
 - Starting point of a TM application.
 - Used to create Transaction, MessageQueue and IOMessage.
- Transaction
 - Represents an IMS TM unit of work. Used to commit, rollback, checkpoint & restart
- IOMessage
 - Class to represent input, output and SPA messages of IMS TM
- MessageQueue
 - Provides services to send and receive messages from an IMS message queue or SPA
- MessageDestinationSpec
 - Used to store information like the LTERM, MFS MOD and alternate PCB names
- SaveArea
 - Used during CHKP/XRST calls to store/retrieve application data
- DLICall Interface
 - Used for IMS System Service calls

488

The IMS Java dependent region resource adapter is a set of Java classes and interfaces that support IMS TM services and IMS message queue processing within IMS Java enabled MPP, BMP, IFP, JMP and JBP dependent regions. The IMS Java dependent region resource adapter provides Java application programs with the following DL/I functionality:

- Access to IMS TM services
- Access to IMS message queues to read and write messages
- Ability to perform program switches
- Ability to perform commit and rollback processing
- Access to GSAM databases
- Database recovery (CHKP/XRST)

11 Universal JDBC Driver updates for JDBC Connection

IMS 11 APAR PM12893
PTF UK59628

- Support DBCS
- Support new optional connection properties for JDBC compliant users
 - dpsbOnCommit
 - Prevents stale connections in a non-managed connection pool
 - maxRows
 - To limit the amount of result set data that IMS returns
 - fetchSize
 - To optimize the amount of data that IMS returns across a network
- SQL aggregate function
 - COUNT(*)
 - To provide number of segments returned in a result set
- Benefits
 - Tools such as Cognos 8 BI and Data Source Explorer can access IMS DB

489

The Universal Drivers have been updated to support Double Byte Character Set (DBCS) (PM12893/UK59628). This SPE provides new optional connection properties for JDBC compliant tools/applications that can be used when creating a JDBC connection to an IMS database.

dpsbOnCommit - by default this property is set to false. When dpsbOnCommit is set to true and a commit is called with no other active statements then the allocated PSB is automatically deallocated (DPSB) before the JDBC consumer places the connection back into the connection pool to prevent connections going stale.

maxRows - generic queries such as `SELECT * FROM DBPCB.SEGMENT` are often executed. This causes IMS to return large amounts of data. To limit the amount of data that IMS returns to a client the maxRows property can be set

fetchSize - by default a JDBC's fetchSize is set to 1. Having a fetch size of 1 causes a network request for every row/segment returned. Setting the fetchSize to a larger value will allow for less network chatter and quicker responses time.

The SQL `COUNT(*)` aggregate function is now supported and is used to count the number of rows/segments in a database table. The result is the number of rows/segments in the set. Note: This causes a scan of the entire IMS database

11 Universal JDBC Driver updates examples

- `dpsbOnCommit`

- Connection URL interface-

```
jdbc://ims://<HOST>:<PORT>/<DATABASE_NAME>:dpsbOnCommit=true;
```

- `maxRows`

```
SELECT * FROM DBPCB.SEGMENT
```

- Programming interface –

```
Statement.setMaxRows(int fetchSize)
```

- Connection URL interface –

```
jdbc://ims://<HOST>:<PORT>/<DATABASE_NAME>:maxRows=50;
```

- `fetchSize`

- Programming interface

```
Statement.setFetchSize(int fetchSize)
```

- Connection URL interface

```
jdbc://ims://<HOST>:<PORT>/<DATABASE_NAME>:fetchSize=100;
```

- SQL aggregate function

- COUNT (*)

```
SELECT COUNT(*) FROM DBPCB.SEGMENTA
```

This slide shows a coding example for the JDBC driver updates

10/11/12 Universal Drivers New SQL Syntax support

- **Support for alternate NOT EQUAL (<>) operator**
 - Previously only supported '!='
 - SELECT * FROM TABLE WHERE A != 1
 - Now can also issue the equivalent call:
 - SELECT * FROM TABLE WHERE A <> 1
- **Support for INNER JOIN ON**
 - Application relates two segments together in the same hierarchical path
 - SELECT * FROM HOSPITAL INNER JOIN WARD ON HOSPITAL.HOSPCODE = WARD.HOSPITAL_HOSPCODE
- **Support for FETCH FIRST n ROW(S)**
 - Only retrieves the first n rows specified for a call.
 - SELECT * FROM TBL FETCH FIRST 5 ROWS ONLY
- **Benefits**
 - Adds additional standard SQL support

APAR

10 PM14770

11 PM17522

12 PM37191

A join is a relational operation in a SELECT statement that retrieves data from two or more tables, based on matching column values. The data in the tables is linked into a single result. The IMS Universal Drivers and the Explorer support the inner join operator. In an inner join, each row of the first table is combined with each row of the second table, keeping only the rows where the join condition is true. For IMS the inner join can only be used between two segments in the same hierarchical path.

IMS 11 SQL support for HALDB, DEDB and HDAM

- **New A or G command codes used to search on a range of keys**
 - A - Clear positioning and start the call at the beginning of the database.
 - G - Prevent randomization exit processing and search the database sequentially
- **IMS Universal Drivers**
 - DLI Driver
 - Programmer can use command codes A and G
 - JDBC Driver
 - Driver uses command codes for ranged queries on a root key
- **Benefits**
 - enables the return of records that are not in sequence

IMS 11 APAR	
Full Function	PM08746
Universal Driver	PM11977
DLIModel Utility	PM11121

Three APARs, Full Function (PM08746), Universal Drivers (PM11977) and DLIMODEL utility (PM11121), will need to be applied to use this new function. The A or G command codes can be used to search on a range of keys to return all values for DEDB, HDAM, or PHDAM databases or for PHDAM and PHIDAM databases with a Partition Selection Exit routine. This enables the return of records that are not in sequence.

A - Clears positioning and start the call at the beginning of the database.

G - Prevents randomization or the calling of the HALDB Partition Selection exit routine and search the database sequentially.

IMS 11 SQL support for HALDB, DEDB and HDAM examples

- **IMS Universal DLI Driver**

- Programmer can use command codes A and G
 - `ssaListInstance.addCommandCode("ROOTSEG", SSAList.CC_A);`

- **IMS Universal JDBC Driver**

- SQL parsed by IMS Universal Driver will use A and G command codes
 - `SELECT DBPCB.ROOTSEG.NAME,DBPCB.ROOTSEG.ROOTKEY FROM DBPCB.ROOTSEG WHERE ROOTKEY>='R1210010000A' AND ROOTKEY<'R1210040000A'`
- SQL parsed by IMS Universal Driver will NOT use A and G command codes
 - `SELECT DBPCB.ROOTSEG.NAME,DBPCB.ROOTSEG.ROOTKEY FROM DBPCB.ROOTSEG WHERE ROOTKEY='R1210010000A' OR ROOTKEY='R1210040000A'`
 - split the statement into two different DL/I search queries

This slide is a coding example to demonstrate when IMS Universal drivers will use command codes after SQL parse

11 Universal Driver for XML-DB

- Support the Universal driver interface introduced in 11
 - type-2 and type-4
- New JDBC syntax for XML support
 - To define data
 - DLIDatabaseView with XML datatype
 - new DLTypeInfo("ASegmentFieldAXML", "ASegmentFieldA.xsd", DLTypeInfo.XML)
 - To retrieve data
 - `ResultSet rs = st.executeQuery("SELECT ASegmentFieldAXML FROM DBPCB.SEGMENTA");`
 - To store data
 - `PreparedStatement ps = conn.prepareStatement("INSERT INTO DBPCB.SEGMENTA" + " (ASegmentFieldAXML) VALUES (?)");`
- Benefits
 - Simplifies API by replacing User Defined Functions

IMS 11 APAR PM17522
PM14766

494

IMS 11 introduced the IMS Universal Drivers. However, at the time of GA the Universal Drivers did not support IMS XML-DB. Java applications that use the type-2 IMS Universal drivers must reside on the same logical partition (LPAR) as the IMS subsystem. Java applications that use the type-4 TCP/IP connectivity IMS Universal drivers can reside on the same logical partition (LPAR), on a different LPAR from the IMS subsystem or on a distributed platform.

PM17522 provides the service maintenance for IMS XML-DB support and also introduces new syntax for supporting XML. To use IMS Universal JDBC driver support for XML, you need to define the XML datatype column fields for storing and retrieving XML data.

DLIDatabaseView with XML datatype

The following example shows how to define XML column datatype fields in a Java metadata class for decomposed mode. In this example, an XML datatype column field named "ASegmentFieldAXML" is defined that is associated with the "ASegmentFieldAXML.xsd" XML schema..

```
// The following describes Segment: SEGMENTA ("SEGMENTA") in PCB: DBPCB ("DBPCB") static
DLTypeInfo[] DBPCBSEGMENTAArray= {
new DLTypeInfo("ASegmentFieldAXML", "ASegmentFieldA.xsd", DLTypeInfo.XML)
```

The new JDBC syntax is then used to retrieve and update IMS databases

Retrieve data

```
ResultSet rs = st.executeQuery("SELECT ASegmentFieldAXML FROM
DBPCB.SEGMENTA");
```

Store data

```
PreparedStatement ps = conn.prepareStatement("INSERT INTO DBPCB.SEGMENTA"
+ " (ASegmentFieldAXML) VALUES (?)");
```

Restrictions:

The IMS Universal JDBC driver does not support side segments.

The XQuery API function introduced in IMS V10 is only supported with the IMS classic JDBC driver

Universal Drivers Variable Length Segments support

- **IMS Universal Drivers are now sensitive to the LL field of a Variable Length Segment**
 - The IMS Universal DB resource adapter and IMS Universal JDBC driver
 - Internally manage the LL field on behalf of the application
 - The IMS Universal DL/I driver
 - Applications are responsible for managing the LL field
- **Benefits**
 - Extends capability support for DL/I database processing

APAR

11 - PM14766

12 - PM25951

495

The universal drivers are now sensitive to the data portion of a variable length segment contains a 2 byte LL field.

Example:

DBD specifies maximum and minimum number of bytes

```
DBD  NAME=HOSPDBD, ...  
SEGM NAME=WARD, PARENT=HOSPITAL, BYTES=(maxbytes, minbytes)
```

The IMS Universal DB resource adapter and IMS Universal JDBC driver internally manage the LL field on behalf of the application. Applications that use the IMS Universal DL/I driver are responsible for managing the LL field.

Universal Driver Interfaces

APAR

10 PM14770

11 PM17522

12 PM37191

- **Public Converter Interfaces**
 - Universal Drivers internal type converters made public
 - Used to convert the data type to a binary representation
 - Benefits
 - Open IMS Java data type functionality to application developers
- **AssemblerCall interface in Universal Drivers**
 - Allows users to call an Assembler module from within Java
 - Benefits
 - Extend interoperability with Assembler

The Public Converter Interface is a data-conversion routine for IMS to use when converting field data from the data type that IMS uses to physically store data to a data type expected by an application program.

The IMS Universal Drivers provide default converter routines for the specified field data type. Customers can create their own data type converter routines to replace the default routines!

IMS 12 HALDB Catalog Metadata

Catalog Metadata definition in DBD and PSB source

- **IMS 12 HALDB Catalog**
 - Contains information about IMS program resources, database resources, and relevant application metadata
 - Database Metadata
 - describes a physical database
 - Application Program Metadata
 - describes application program view of the data in the database
 - Arrays and Structures
 - Field data types and data type conversion
 - Field redefines
 - Alternative Field maps for a segment
 - ACBGEN
 - Used to auto-populate the catalog
- **Benefits**
 - consolidates the application program data requirements into trusted location

498

The IMS Catalog is new in IMS 12. It contains information about IMS program resources, database resources, and relevant application metadata that IMS controls. It is a High Availability Large Database (HALDB) that can be accessed by DL/I calls or SQL statements to look at data in the catalog in the same way that you retrieve data from any other HALDB. It is a trusted source of metadata which is derived from the database descriptions (DBDs) and program specification blocks (PSBs). Additionally, the ability to code the elements described by an application program metadata in a DBD provides the ability to consolidate the data requirements of your application programs into the IMS Catalog. You can populate the IMS Catalog when you generate the ACB members or you can populate the IMS catalog in a separate job after the ACB generation process is complete.

IMS ensures that the catalog contains accurate information.

Immediate Benefits

- All Universal drivers leverage the IMS catalog
 - Trusted online (IMS) source for metadata used by the drivers
 - Direct access to IMS metadata in the catalog
 - No longer require the separate Java metadata class
 - Virtual deployment support
 - No longer file-system dependent for metadata
 - Virtual deployment support
 - Metadata is trusted and up-to-date

PSB PCB statements changes

- **PCB**
 - **EXTERNALNAME=**
 - An optional alias for the PCB label or the PCBNAME= parameter
 - Java application programs use the external name to refer to the PCB
 - **REMARKS=**
 - Optional user comments. A 1- to 256-character field. Added to PSBGEN statements

An optional alias for the PCB label or the PCBNAME= parameter. Java application programs use the external name to refer to the PCB.

DBD statement changes

- **DBD**
 - ENCODING=
 - Specifies the default encoding of all character data in the database defined by this DBD.
 - This value can be overridden in individual segments or fields.
- **SEGM**
 - EXTERNALNAME=
 - An optional alias for the NAME= parameter used by Java application programs to refer to the segment.
- **FIELD**
 - CASENAME=
 - The name of the map case when alternative mappings are defined for the fields in a segment
 - DATATYPE=
 - Specifies the external data type of the field.
 - EXTERNALNAME=
 - An optional alias for the NAME= parameter used by Java application programs to refer to the field.
- **DFSMARSH**
 - Specifies the DATATYPE converter routine to transform binary data to external data format
- **DFSMAP**
 - Defines alternative field mappings in a segment.
- **DFSCASE**
 - Defines a map case for a segment type that uses DFSMAP alternative field mapping.
- **REMARKS=**
 - Optional user comments. A 1- to 256-character field. Added to DBDGEN statements

501

ENCODING - Specifies the default encoding of all character data in the database defined by this DBD. The default is ENCODING=Cp1047, which specifies EBCDIC encoding.

This value can be overridden in individual segments or fields.

DATATYPE – (Note before the catalog this information was provided by the application program copybooks). An IMS provided or custom written data type converter routine transforms the binary data type stored by IMS to the data type specified on the DATATYPE parameter.

ARRAY BINARY BIT BYTE CHAR DATE DECIMAL DOUBLE FLOAT INT
LONG OTHER SHORT STRUCT TIME TIMESTAMP XML

DFSMAP - You can define multiple alternative field maps for a single sequence of bytes in a segment definition. When an application program accesses a sequence of bytes in a segment instance that uses field mapping, the application program must evaluate the control field to determine which field map is in effect. Each field map in a segment definition is defined to IMS by coding a DFSCASE statement. The fields that make up a field map are defined by FIELD statements that specify the name of the map case on the CASENAME parameter. A set of map cases that map the same sequence of bytes in a segment share the same control field. In the segment definition, a DFSMAP statement links the set of related map cases to the control field. The map case definitions must specify the name of the DFSMAP statement on the MAPNAME parameter in the DFSCASE statement.

Application programming with the IMS 12 Catalog

- **IMS catalog resident PSBs for application programs**
 - DFSCP000 - High-level assembler and COBOL applications
 - DFSCP002 PL/I - applications
 - DFSCP003 - PASCAL applications
- **The following PCBs are included to support different catalog processing models:**
 - DFSCAT00 - The primary PCB to access all data in the catalog database..
 - DFSCATSX - Use this PCB to access the catalog database via the catalog secondary index.
 - DFSCATX0 - Use this PCB to process the catalog secondary index database.
- **All catalog processing is performed with PROCOPT=G.**
- **GUR call**
 - application programs can use the Get Unique Record (GUR) DL/I call to retrieve catalog database record

502

The IMS catalog database is accessible to standard IMS DB application programs when it is enabled for your IMS system. The IMS catalog database stores application and database metadata in a format that is accessible to standard IMS DB application programs. This information includes database definitions, program specifications, and user comments. Any application program can read this information, but the catalog database is write-protected and can be updated only by authorized system utilities such as the IMS catalog populate utility (DFS3PU00). By default, the IMS catalog is named DFSCD000. The DFSC prefix is replaced with an alias prefix if one is defined to IMS.

The IMS catalog secondary index contains a single segment type, DBDPSB. It is logically linked to the DBDXREF segment type in the IMS catalog database, which is included in all catalog records for IMS PSBs. You can use the catalog secondary index to determine which IMS programs reference a specific user database without processing the entire IMS catalog.

By default, the IMS catalog is named DFSCX000. The DFSC prefix is replaced with an alias prefix if one is defined to IMS.

IMS does not require user PSBs to contain a PCB for the IMS catalog database or secondary index. The catalog PSBs DFSCP000, DFSCP002, and DFSCP003 are dynamically attached to any user PSB that makes a DL/I call to the catalog database or issues an INT DB QUERY call. Each PSB is intended for use by a different type of application program. Restriction: The IMS catalog PSBs are not dynamically attached to generated PSBs or GSAM-only PSBs.

You might need to increase the size of your storage pools, up to the maximum size of the catalog PSB in each pool multiplied by the number of user PSBs that concurrently access the catalog.

IMS automatically increases the space to attach the catalog PSB with user PSB:

96 bytes of additional space are allocated for each user PSB in the PSB CSA storage pool.

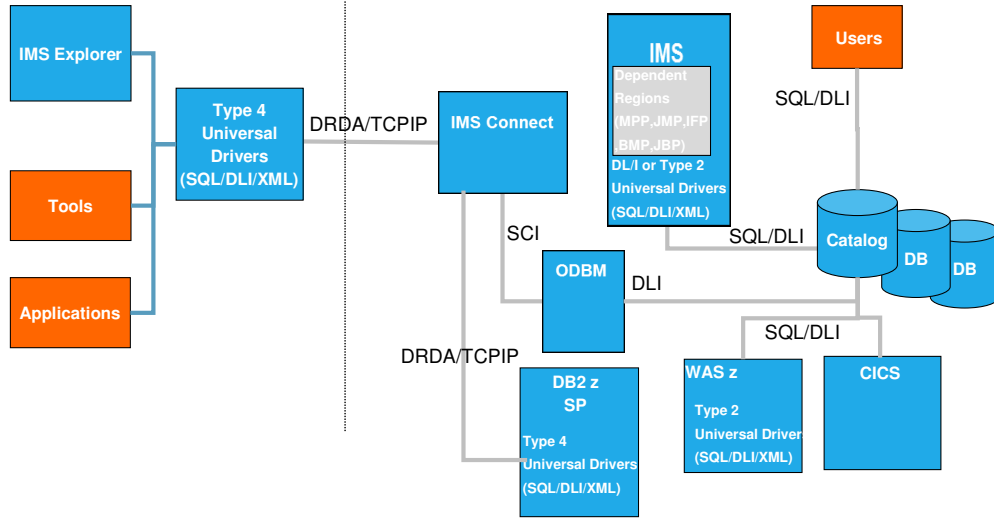
The catalog PSB occupies 12kb in the DLIPSB pool and 500 bytes CSAPSB pool for each user PSB that is using the catalog PSBs.

GUR – This call functions like a GU followed by a series of GNP calls and is provided to reduce the number of processing steps required to retrieve a complete catalog record for a DBD or PSB. Data returned in the form of a single XML instance document for a particular catalog record. You can also issue other DL/I read calls to process the catalog database in the same way as any other IMS database.

GUR Call

- **New “GUR” DL/I call**
 - **Get Unique Record**
 - Restricted to use with IMS Catalog database
- **Functions like a GU followed by a series of GNP calls**
- **Returns the entire database record in one call**
 - Saves overhead of issuing GU & GNP to retrieve all the metadata for a catalog member
 - Using an AIB token, the call can be continued if the I/O area is too small for entire catalog database record
- **Data returned will be in XML format**
 - Matches the XML schema
- **Support added for DFSDDLTO and IMS REXX**

Runtime Catalog Access



IMS Enterprise Suite

IMS Enterprise Suite V2.1

- IMS Enterprise Suite is a product within the IMS SOA Integration Suite.
 - Connect APIs
 - IMS Enterprise Suite Explorer for Development
 - IMS Enterprise Suite DLIModel utility plug-in
 - IMS Enterprise Suite SOAP Gateway
 - Java Message Service (JMS) API
- Support SMP/E and the IBM Installation Manager.
- Supports IMS 10, 11 and 12.

The IMS Enterprise Suite V2R1 is a separate, no charge product that provides IMS application development and integration. It is a collection of IMS middleware functions and tools that simplify your use of IMS applications and data while supporting your IMS on demand systems and your distributed IMS application environment. It also delivers innovative new capabilities for connectivity to your IMS environment, expands application development, extends standards and tools for a Service Oriented Architecture (SOA), eases installation, and provides simplified interfaces. It includes the Java Message Service (JMS) API used for the IMS Java application Callout function, the Connect API for Java and for C - providing a simple API to interact with IMS Connect, the IMS SOAP Gateway for connectivity to/from IMS applications and data along with its open source , WS-Security and Business Events, the Explorer for Development for application development, and the DLI Model Utility plug in for local Java metadata development. Also provided is the Installation Manager and SMP/E support to ease installation and management on the distributed and z/OS platforms.

IMS Enterprise Suite V2.1 Connect API for Java New features

- **Support for IMS synchronous callout**
- **Support for IMS type-2 commands**
 - IMS and IMS 12 Connect type-2 command input
 - Return the response data as either a raw XML document or as a parsed response object.
- **CM0 ACK NOWAIT transaction support**
 - CM0 ACK NOWAIT transaction support with IMS 12 Connect
- **Configurable CM0 send-and-receive no response message**
 - configured to request a DFS2082 message
- **Support for IMS 12 RACF return codes**
 - Connect API for Java provides the `getRacfReturnCode` and `getRacfReturnCodeString` methods to retrieve the RACF error information.

Synchronous callout - IMS 10 introduced IMS application program synchronous callout support. The Connect API for Java can now be used to retrieve and respond to the callout request.

Benefit – simplifies RYO application development

Type 2 Commands - The Connect API for Java now supports sending IMS and IMS 12 Connect type-2 commands. The Connect API for Java can return the response data as either a raw XML document or as a parsed response object.

Benefit – enables RYO distributed type-2 command processing

CM0 ACK NoWait for user-written clients - The CM0 ACK NoWait enhancement includes two new features for user-supplied IMS Connect clients:

- For commit-then-send (CM0) send-and-receive transactions, the option to eliminate the final receive on the client side by suppressing the timeout notification sent by IMS Connect after the client acknowledges receipt of the response.

- The ability to determine the protocol level supported by a given instance of IMS Connect.

Benefit - reduce processing costs and increase the number of CM0 send-and-receive transactions that can be input to IMS over a time.

In IMS 12, the NoWait option on CM0 ACK messages requires IMS Connect protocol level 2. User-supplied IMS Connect clients can determine which protocol level a given instance of IMS Connect supports by checking any CSM returned by the IMS Connect instance. The ability to determine the protocol level of a given instance of IMS Connect can be useful in environments where multiple versions of IMS Connect are used.

Configurable CM0 - This enhancement provides the ability to request a DFS2082 message instead of a timeout error when a CM0 send-and-receive message does not receive a response from the IMS application

Benefit - reduce processing costs and increase the number of CM0 send-and-receive transactions that can be input to IMS over a time.

RACF return codes - IMS Version 12, the IMS Connect request status message (RSM) contains a 2-byte RACF return code when a RACF security failure occurs during message processing.

Benefit -The RACF return code indicates what caused the security failure and is useful for diagnosing security configuration problems

IMS Enterprise Suite Connect API for Java

Apar PM05893 UK54997

Enhancements

- Transaction Expiration
 - IMS Connect timeout activates OTMA transaction expiration
 - Benefit- better utilization of CPU, and storage after a timeout occurs in IMS Connect.
- SSL keystore and truststore loaded
 - FileInputStream or URL object
 - Benefit - provides greater flexibility in managing keystores and truststores.
- Support return of MFS Modname
 - Benefit- Client application can use the modname to determine a response format
- Rename samples.jar to ImsESConnectApiForJavaV1R1Samples.jar
 - Updated to eliminate the need to manually create a logs directory
 - Benefit – Improves usability of the sample JAR file,

508

Enhancements made in APAR PM05893 are provided as total replacement .jar UK54997 and are included in IMS Enterprise Suite V2.1

Enhancements:

1. The Connect API for Java now supports the transaction expiration function available in IMS Version 10 and later. The new boolean property, `otmaTransactionExpiration`, can be used to instruct IMS Connect to notify OTMA to override the default transaction expiration time with a value that matches the current IMS Connect timeout value.

Benefit- This enhancement leads to better utilization of CPU, storage, network, and other resources after a timeout occurs in IMS Connect.

2. The Connect API for Java now supports the use of the `InputStream` object or file URLs to load the SSL keystore and truststore.

Benefit -This enhancement provides greater flexibility in managing keystores and truststores.

3. The `mfsModname`, is accessible via the `getMfsModname()` getter method in the `TmInteraction` interface and is populated when the response message includes a `REQMOD` segment containing a modname.

Benefit- The application can use the modname to determine a response format.

4. The file `samples.jar` is renamed to `ImsESConnectApiForJavaV1R1Samples.jar` so that it is easily recognizable as part of the IMS Enterprise Suite Connect API for Java component.

Benefit - The usability of the sample JAR file, `samples.jar`, is improved by eliminating the need to manually create a logs directory.

IMS Version 12 IBM

WebSphere sMash and IMS Web 2.0 IMS Connect API for Java

```

    graph LR
      subgraph Windows
        sMash[sMash http runtime]
        API[IMS Connect Java API]
        Files[Connection and TmInteraction properties files]
        sMash --- API
        Files --- API
      end
      Browser[Browser client] --> sMash
      API --> zOS
      subgraph zOS
        IMSConnect[IMS Connect]
        subgraph OTMA [OTMA]
          IMSApp[IMS App]
        end
      end
  
```

- **WebSphere sMash**
 - lightweight runtime for creating and running RESTful services
 - Groovy, PHP, and Java through the IMS Connect for Java API
 - sMash application is responsible for
 - Preparing input data for IMS application
 - Interpreting output data from IMS application
 - Configuring connection and interaction configuration property files read in by API during execution
- **Benefit :** Ability to quickly build and execute Web 2.0-based applications that access IMS

509

This slide shows an example of an application that uses IMS Connect API for Java to access IMS transactions.

The Connection and TmInteraction properties files provided with the Connect API for Java installation can be used to specify the connection parameters and the parameters that determine the type of interaction to perform with IMS Connect.

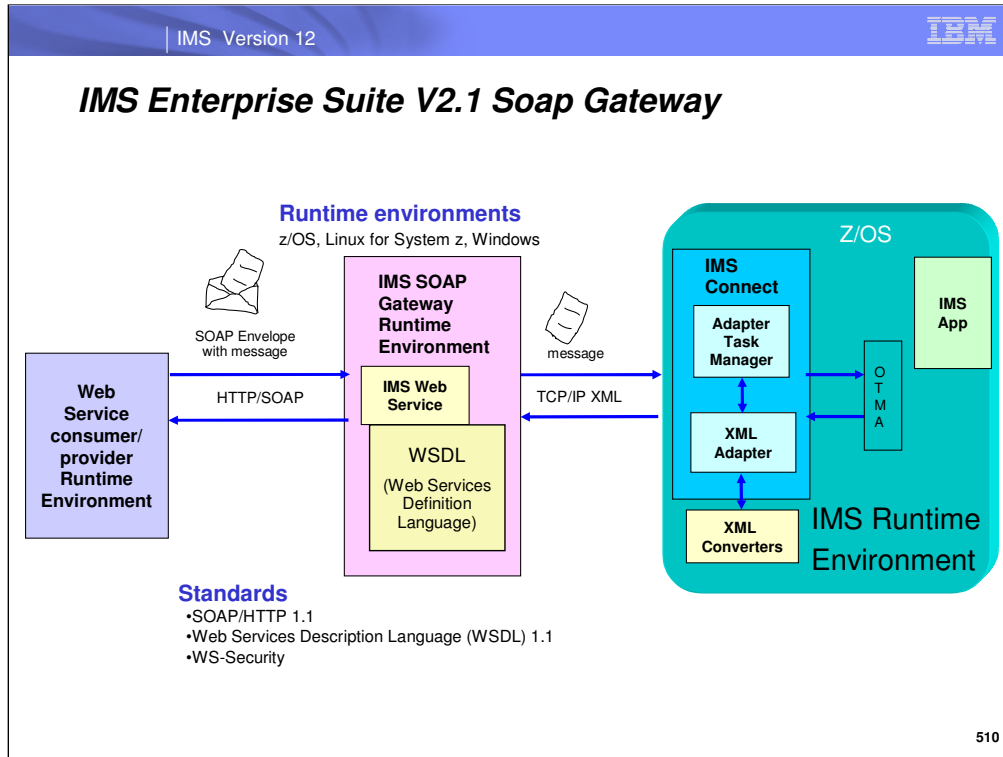
The IMS Web 2.0 Solution for WebSphere sMash allows for a more lightweight and cost-effective solution in generating IMS RESTful services from IMS Transactions for use in Web Oriented Architecture such as Web Mashups.

WebSphere sMash is a runtime that allows for developing RESTful services Java (a popular object oriented computer language), PHP (a popular scripting language), and Groovy (a scripting language based on Java) and includes its own http stack for running these services. It is considered lightweight in the sense that it does not need an actual application server to deploy its services.

The IMS Web 2.0 Solution for WebSphere sMash shows how the IMS Connect API for Java (a free offering from IMS) can be used in all three languages (Groovy, Java and PHP) to connect to IMS and launch a transaction. Once the output of an IMS Transaction is returned to WebSphere sMash, the programmer can manipulate that information however they want prior to returning it to the end user. This allows for greater flexibility than the IMS Web 2.0 Solution for IBM Mashup Center which is all GUI/wizard based. This Solution is provided as a document only that contains instructions and sample code. The reason for this is that both IMS Connect API and WebSphere sMash are available as separate deliverables.

Developers can quickly build and execute Web 2.0-based applications that access IMS transactions. The Solution is available on the IMS Website.

<http://www-01.ibm.com/software/data/ims/mashup.html>



IMS Enterprise Suite V2.1 SOAP Gateway is a web services solution that enables IMS applications to interoperate outside of the IMS environment through the SOAP protocol to provide and request services that are independent of platform, environment, application language, or programming model. For V2.1 SOAP Gateway can only be used on z/OS®, Linux on System z®, and Windows systems. When IMS applications are enabled as web service providers, 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 the IMS applications.

With the IMS Connect XML adapter your IMS applications do not need to change.

When IMS applications are enabled as web service consumers, they can callout to any external web services and receive the response in the same or a different transaction. SOAP Gateway is compliant with the industry standards for web services, including SOAP/HTTP 1.1 and Web Services Description Language (WSDL) 1.1. SOAP Gateway also supports web services security (WS-Security), where user identity or security token is authenticated on a per-message basis. Authentication information can also be passed on a per-web service basis, where the user information is defined in the connection bundle for the web service.

IMS Enterprise Suite V1.1 SOAP Gateway Fix Pack 2 - Review

- **IMS 10/11 Fix Pack 2 enhancements:**
 - Store web service information in a runtime cache
 - Benefit – performance
 - A new command-line utility, SOAP Gateway management utility (iogmgmt)
 - Benefit - facilitates automation of administrative tasks
 - Multi-operation messages for IMS as a service provider
 - Benefit - consolidate service request into one WSDL
 - ERROR and FATAL messages can be directed to Write To Operator (WTO)
 - Benefit - facilitate server startup message monitoring

511

These enhancements are also included in IMS ES V2.1 SOAP Gateway. For both IMS Version 10 and IMS Version 11 customers, Fix Pack 2 offers the following enhancements:

•SOAP Gateway provides enhanced performance by storing web service information in a runtime cache. Instead of reading the information from a file system, SOAP Gateway can process request messages more efficiently, and manage changes to web service information with greater flexibility.

•A new command-line utility, SOAP Gateway management utility, facilitates automation of administrative tasks and provides management flexibility. The SOAP Gateway management utility replaces the SOAP Gateway deployment utility.

•Multi-operation messages are now supported for the provider scenarios. Rational Developer for System z Version 8.0.1 and later can generate correlators and converters for messages that contain multiple operations.

•All ERROR and FATAL messages can be directed to Write To Operator (WTO) to facilitate message monitoring.

IMS Enterprise Suite V1.1 SOAP Gateway – Fix Pack 2...

- IMS 11 Fix Pack 2:
 - Security
 - Client authentication via Java keystore (JKS) on all platforms.
 - Client and Server authentication by SAF keyrings
 - AT-TLS z/OS Communication Server z/OS
 - Benefit
 - Open platform certificate management
 - Leverage z/OS platform system services
 -
 - Top-down PL/I application generation from a WSDL file
 - Rational Developer for System z Version 8.0.1 and later.
 - IMS Connect XML converter drivers support via APAR PM16945
 - Benefit
 - Generate new IMS PL/I service provider applications

For IMS Version 11 customers, Fix Pack 2 offers the following enhancements for the web service provider scenario (an IMS application is enabled as a web service):

- Client authentication is supported by using the Java keystore (JKS) on all platforms.
- JKS (Java Keystore) is file or RACF database used for certificates and key management
- Server authentication and client authentication is supported by using System Authorization Facility (SAF) keyrings through the Application Transparent Transport Layer Security (AT-TLS) feature in IBM® z/OS Communications Server on the z/OS platform.

You can generate a PL/I application template from a WSDL file by using Rational Developer for System z Version 8.0.1 and later. After adding your business logic to the generated PL/I application, you can enable the application as a web service on SOAP Gateway. This approach is known as the *top-down* development scenario in Rational Developer for System z. The generated PL/I top-down XML converter drivers require IMS Version 11 APAR PM16945.

IMS Enterprise Suite V1.1 SOAP Gateway Fix Pack 2 ...

- **Web service security**
 - UsernameToken Profile 1.0 user name tokens
 - UserID and password sent to IMS Connect
 - SAML 1.1 unsigned sender-vouches tokens
 - UserID sent to IMS Connect
 - IMS Connect must be RACF=N
 - Java Authentication and Authorization Service (JAAS) user exit
 - Benefit
 - users are authenticated on a per-message basis
 - custom authentication user exit to perform additional security checks

513

Web services security support provides dynamic authentication of users on a per-message basis. When WS-Security is enabled for a web service, SOAP Gateway extracts the user information from the WS-Security header and propagates the information to IMS. When WS-Security is not enabled, user ID and password information is provided by the connection bundle on a per-web service basis.

UsernameToken Profile Version 1.0 - For a user name security token, a SOAP Gateway client must provide the user ID and password to SOAP Gateway in the SOAP message security header. SOAP Gateway extracts this user ID and password from the SOAP message and passes them with the payload data to IMS Connect for authentication.

•if IMS Connect security is enabled (RACF=Y), IMS Connect authenticates the user ID and password. IMS Connect then passes the user ID and data to OTMA. IMS Connect does not pass any password information to OTMA.

•If IMS Connect security is disabled (RACF=N), IMS Connect simply passes on the user ID and data to OTMA without authentication.

Security Assertion Markup Language (SAML) Version 1.1 unsigned sender-vouches tokens - SAML is an XML-based OASIS standard for exchanging user identity and security attribute information. You can send a Security Assertion Markup Language (SAML) 1.1 sender-vouches token from a client application to SOAP Gateway to dynamically authenticate the user per message. SOAP Gateway passes the user ID and the payload data. When SAML tokens are used, IMS Connect security must be turned off because SAML tokens do not contain password information. If IMS Connect security is turned on, the authentication would fail.

JAAS - You can plug in your own authentication module to intercept an inbound message to perform additional checks by using a Java Authentication and Authorization Service (JAAS) module.

IMS Enterprise Suite V1.1 SOAP Gateway Fix Pack 3 Enhancements – IMS 11 required

- **Idle Connection cleanup**
 - Checks for and deletes idle connections
 - Benefit
 - improve memory and CPU usage
- **Connection error recovery**
 - socket reconnect support
 - recycle IMS Connect without the need to restart SOAP Gateway
 - Benefit
 - maximize reuse of existing connections and reduce connectivity errors
- **IMSSOAPCalloutIVPSERVICE web service**
 - setup verification for the web service consumer scenario

Fix Pack 3 enhancements require IMS 11 and are included in IMS Enterprise Suite V2.1 SOAP Gateway.

•**Idle Connection Cleanup** - The SOAP Gateway server creates one connection pool for each unique IMS Connect host name and port number with which it communicates. Each connection pool can contain multiple socket connections. An idle connection cleanup feature is added to enhance connection pool management. SOAP Gateway can periodically clean up idle connections in the connection pool to improve overall memory and CPU usage. You can specify the cleanup interval and the minimum number of connections to keep. By default, idle connection cleanup is disabled. You can enable it with the `iogmgmt -prop` command.

•**Connection error recovery** - SOAP Gateway now tries to reestablish a stale connection in a connection pool when the previous connection to the same IMS Connect instance encountered an error. This socket reconnect support increases the fault-tolerance of SOAP Gateway when a temporary network connection issue occurs or when IMS Connect is recycled. You can recycle IMS Connect as part of your system maintenance without the need to restart SOAP Gateway. The restriction in Fix Pack 2 to restart SOAP Gateway when IMS Connect is restarted is lifted. This function provides maximum reuse of existing connections and reduces number of errors a client would see because of IMS Connect availability issues. SOAP Gateway does not resubmit the request to avoid duplicate request errors. Upon receiving an error, the client can choose to resubmit the request.

•**Web Service IVP** - A web service is provided for environment setup verification for the web service consumer scenario. This `IMSSOAPCalloutIVPSERVICE` web service is automatically deployed when you start the SOAP Gateway server. The IMS application that issues the callout requests and the related OTMA destination descriptors and XML converters are provided in IMS Version 11 APAR PM31226.

IMS Enterprise Suite V2.1 SOAP Gateway New features

- **Web service security**
 - SAML 1.1 signed sender-vouches tokens
 - SAML 2.0 unsigned sender-vouches tokens
 - Benefit
 - protects the message integrity by using token validation
- **Java separation**
 - SOAP Gateway server on a separately installed IBM SDK
 - Multiple SOAP Gateway servers can share a single instance of the IBM SDK
 - Benefit
 - Smaller installation footprint
- **Enhanced connection error recovery**
 - SOAP Gateway actively checks the status of socket connections
 - terminates stale connections and creates new socket connections.
 - This enhancement requires
 - APAR PM29137 for IMS Version 12
 - APAR PM27324 for IMS Version 11
 - HWSSMPL1 exit on IMS Connect
- **Benefit**
 - Reduces connection recovery response time

IMS™ Enterprise Suite Version 2.1 SOAP Gateway adds the support for Security Assertion Markup Language (SAML) 1.1 sender-vouches signed tokens and SAML 2.0 unsigned tokens for the web service provider scenario. SAML is an XML-based standard developed by Security Services Technical Committee (SSTC) of Organization for the Advancement of Structured Information Standards (OASIS) for exchanging user identity and security attributes information between communicating parties at the SOAP message level. The SAML standard provides the means by which authentication and authorization assertions can be exchanged across web service transactions, and how a security identity can be obtained and transferred from one business entity to another.

A signed SAML token further protects the message integrity by enabling the recipient of the token to validate authenticity of the token and assert SAML token identity and attributes based on the trust relationship with the token issuer. SAML 2.0 is a newer standard that introduces features such as session management, attribute profiles, encryption, metadata specifications, and pseudonyms.

You can also run SOAP Gateway on a previously installed IBM SDK for operational efficiency, enhanced serviceability, and smaller installation footprint. Multiple SOAP Gateway server instances can share a single instance of the IBM SDK.

When connection errors occur between SOAP Gateway and the target IMS Connect, SOAP Gateway actively checks the status of socket connections in the pool, terminates stale connections, and creates new ones. In this release, SOAP Gateway takes advantage of an IMS Connect enhancement by sending a ping request to the IMS Connect HWSSMPL1 user exit. The reply is returned directly to SOAP Gateway without being routed through IMS so the response time is shorter.

This enhancement requires APAR PM29137 for IMS Version 12, or APAR PM27324 for IMS Version 11, and the configuration of the HWSSMPL1 exit on IMS Connect.

IMS Version 12

IMS Enterprise Suite V2.1 Explorer for Development

Generate SQL to access IMS data

Edit PSB sensitive segments and attributes

See database relationships change DBD field attributes

Status	Operation	Date	HOSPLL	HOSPCODE	HOSPNAME
✓	Succeed: select * from pcb...	8/2	1	R 1210010000A	ALEXANDRIA
✓	Succeed: select * from pcb...	8/2	2	R 1210020000A	SANTA TERESA
✓	Succeed: select * from pcb...	8/2	3	R 1210030000A	SANTA CLARA
✓	Succeed: select * from pcb...	8/2	4	R 1210040000A	NEW ENGLAND

516

The IMS™ Enterprise Suite Explorer for Development (IMS Explorer) is an Eclipse-based graphical tool that simplifies IMS application development tasks such as updating IMS database and program definitions, and using standard SQL to manipulate IMS data. Its graphically-driven editors allow the user to display the segment hierarchy for any IMS database, including logical relationships and secondary indexes. It also provides user assistance in the form of rich GUI controls and contextual help to reduce IMS development effort.

The Explorer's graphical editors can be useful for the importing, visualization, and editing of IMS database and program definitions. You can also use the IMS Explorer to easily access and manipulate data stored in IMS by using standard SQL.

IMS Enterprise Suite V2.1 Explorer for Development

- Visualization and editing of IMS Database and Program Definitions
 - Provide graphical editors to:
 - Display IMS database hierarchical structures
 - Display/create/edit PSBs
 - Change/add fields on a DBDs
 - Import Cobol CopyBooks and PL/I Structures to a database segment*
 - Generate of DBD and PSB source
- Ability to easily access IMS data using SQL statements
 - Leveraging IMS Universal JDBC driver
- Ability to access the IMS Catalog
- Connectivity to the z/OS system
 - Browse a Data Set and submit JCLs
 - Import and export DBD and PSB source files from a Data Set to the IMS Explorer, and vice-versa
- Supports cross-product integration:
 - IBM® Rational® Developer for System z®
 - IBM Optim™ Development Studio
 - IBM Problem Determination Tools Plug-ins for Eclipse
- Benefit
 - Simplify IMS application development tasks

*Requires RDz 8

The IMS ES Explorer can be used to simplify of IMS application development tasks.

IMS Enterprise Suite V2.1 Explorer for Development

- **IMS 12 HALDB Catalog Metadata**
 - Universal Driver Type 4 connection
 - Can add COBOL or PL/I metadata to the catalog
 - Direct update of the catalog metadata is not available
 - Requires intermediate ACBGEN to populate the catalog with the extended information
- **PSBs and DBDs editing**
 - Send updated DBDGEN and PSBGEN source files to the host
 - run DBDGENs, PSBGENs, ACBGENs, and catalog population

Supports the IMS 12 HALDB Catalog Metadata processing

IMS Version 12

IMS Enterprise Suite V2.1 Explorer for Development

View physical IMS database structure

Logical relationship between databases

Additional properties of a segment or field

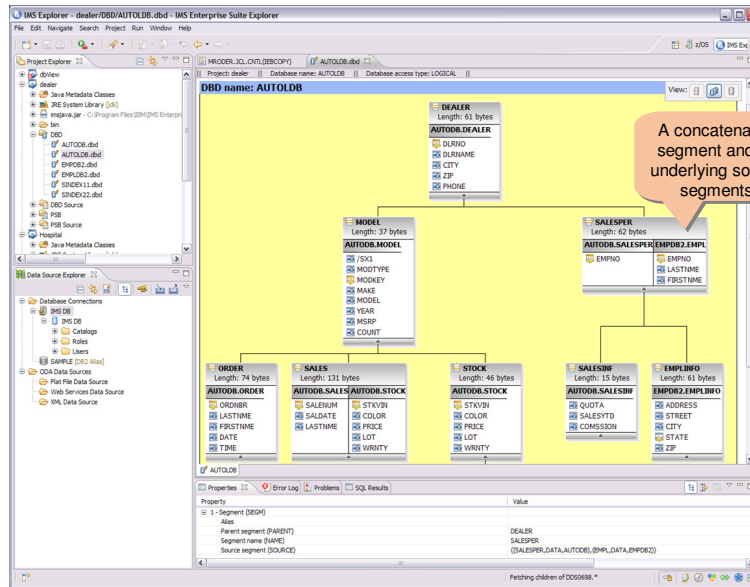
Property	Value
Segment statement	
Length (BYTES)	61
Parent segment (PARENT)	0
Segment name (NAME)	DEALER
Source segment (SOURCE)	
List of fields	

519

The following slides provide screen views of the various function provided by the Explorer.

This slide show the IMS database structures relationships.

IMS Enterprise Suite V2.1 Explorer for Development View logical IMS database structure



This slide is a continuation of the IMS database structures view.

IMS Version 12

IMS Enterprise Suite V2.1 Explorer for Development Build

PCB definition

PCB name: PCB01 | All Segments (Edit Sensitivity) |

HOSPITAL
Total length: 900

- HOSPLL
- HOSPCODE
- HOSPNAME

PAYMENTS
Total length: 900

- PATMLL
- PATNUM
- AMOUNT

WARD
Total length: 900

- WARDLL
- WARDNO
- WARDNAME
- PATCOUNT
- NURCOUNT
- DOCCOUNT

PATIENT
Total length: 900

- PATLL
- PATNUM
- PATNAME

```

PCB NUMBER 5      DB      DEDBJN21
-----
PCB      TYPE=DB, DBDNAME=DEDBJN21, POS=M, PROCOPT=A, KEYLEI
        PCBNAME=PCB01
SENSEG  NAME=HOSPITAL, PARENT=0
SENSEG  NAME=PAYMENTS, PARENT=HOSPITAL
SENSEG  NAME=WARD, PARENT=HOSPITAL
SENSEG  NAME=PATIENT, PARENT=WARD
SENSEG  NAME=ILLNESS, PARENT=PATIENT
SENSEG  NAME=TREATMNT, PARENT=ILLNESS
SENSEG  NAME=DOCTOR, PARENT=TREATMNT
SENSEG  NAME=BILLING, PARENT=PATIENT
-----
*      PCB NUMBER 6      DB      IVPDB1
        
```

Point and click to select SENSEGs

Generated PSB source

521 521

This slide shows how updates to PSB source result in the generation of new PSB source that can be sent to PSBGEN utility

IMS Version 12 IBM

IMS Enterprise Suite V2.1 Explorer for Development Query IMS database with Universal JDBC driver

Create SQL scripts with Select, Update, Delete, and Insert statements

Build a SQL statement using the SQL Builder with content assistance

View the SQL results

PSB = schema

DB PCB = database

Database segments = tables

522 522

This slide shows how the Data Source Explorer function can use the IMS Universal Drivers to access IMS databases. (requires IMS 11 or later)

IMS Version 12

IMS Enterprise Suite V2.1 Explorer for Development

Browsing Data Sets and Submitting JCL's

The screenshot displays the IMS Enterprise Suite V2.1 Explorer for Development interface. The main window is titled "z/OS - MRODER.JCL.CNTL(IEBCOPY) - IMS Enterprise Suite Explorer". The interface is divided into several panes:

- Data Sets Pane (Left):** Shows a tree view of data sets under the qualifier "MRODER". A context menu is open over the "IEBCOPY" data set, with options: "New Data Set Member...", "Open", "Submit Job", and "Delete". A callout bubble labeled "Browse data sets" points to this pane.
- JCL Editor Pane (Right):** Displays the JCL code for the "IEBCOPY" job. A callout bubble labeled "Edit a data set member" points to the JCL code.
- Jobs Pane (Bottom Left):** Shows a list of jobs. A callout bubble labeled "View JES output" points to the "MRODER*" job entry.
- Console Pane (Bottom Right):** Shows the output of the job, including the message "z/OS DSN=MRODER.JCL.CNTL(IEBCOPY) - saved".

523

This slide shows how you can browse z/OS datasets and view job output results.

IMS SOA Integration Suite Enhancements

The IBM® IMS™ SOA Integration Suite is a collection of IMS middleware functions and tools that support your IMS on demand systems and your distributed IMS application environment.

IMS SOA Integration Suite

www.ibm.com/ims

- **IMS Solutions for Java development**
 - Write Java applications to access IMS databases and process IMS transactions
 - XML Database support
- **IMS MFS Web solutions**
 - support the reuse of existing MFS-based IMS business logic on the web
- **IMS TM Resource Adapter**
 - create Java Platform, Enterprise Edition applications to access IMS transactions
 - callout requests to external Java EE applications from IMS applications
- **IMS Web 2.0 solutions for TM and DB**
 - integrate IMS TM and DB assets into Web 2.0 applications

This slide identifies the functions included in the IMS SOA Integration Suite.

IMS SOA Integration Suite Enhancements

- IMS 10 MFS Web Enablement DBCS support

APAR IMS 10 PM23985

- MFS Parser
 - Generate XMI files with both DBCS and ASCII data
- MFS Stylesheet
 - New stylesheets supporting DBCS

- Benefit

- Provides DBCS keyword support in MFS source



526

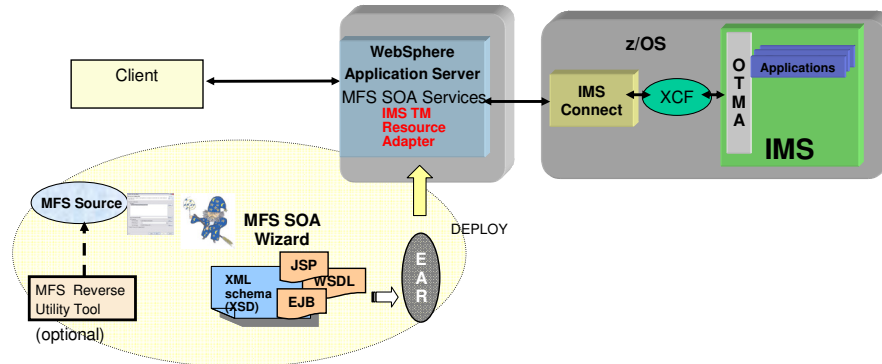
PM23985

The IMS SOA Integration Suite MFS Web Enablement provides the tooling utility and runtime support to enable existing or new IMS MFS-based applications for the Web, and interactively render them for display in standard browsers such as Microsoft® Internet Explorer and Mozilla Firefox. In addition, the MFS XML Utility provides the MFS Importer with existing source files that the MFS Importer can use to generate the XML Metadata Interchange (XMI) files metadata. The runtime support, running on WebSphere® Application Server, provides a servlet to dynamically render MFS-based Web pages on browsers by using cascading stylesheets (CSSs), and an adapter that loads the metadata generated by the MFS XML Utility and communicates with host IMS (conversational) applications through the J2EE-compliant IMS TM Resource Adapter.

IMS MFS SOA Support

- Message Formatting Service SOA transforms existing MFS-based IMS application into MFS J2C Services (Web Services, EJB, or a JSP)
 - IBM Integration Designer
 - IBM Business Process Manager
- Benefit
 - Provides MFS transaction support for Business Process Choreography (B2B)

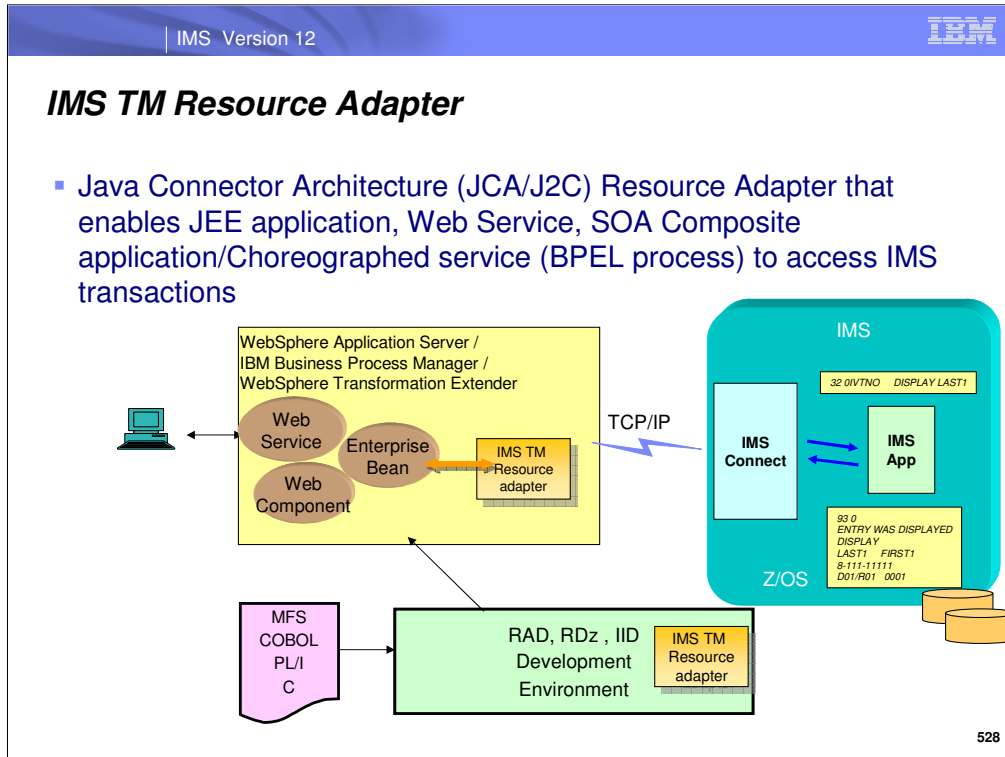
APAR IMS 10 PM13664 / IMS 11 PM23831



527

Message Format Service service-oriented architecture (MFS SOA) support enables the reuse of existing MFS-based IMS application program business logic as web services, web pages, or Enterprise JavaBeans (EJB) components, or integrate your MFS applications in business processes. MFS SOA support uses the IMS TM resource adapter on both z/OS or distributed platforms to communicate with IMS Connect and IMS.

These APARS provides the ability to use the IBM Integration Designer to generate MFS-based services and integrate them into Business Process Execution Language (BPEL) processes that can be deployed to IBM Business Process Manager.



The 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 MFS , COBOL , PL/I and C 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
- IBM Integration Developer (IID)
- Rational Developer for System z (RDz)

The IMS TM Resource Adapter also provides a plug-in for a J2EE application server (runtime environment for application code) by using the standardized system contracts. It provides runtime support for WebSphere Application Server on distributed and z/OS platforms. The IBM Business Process Manager extends an application server by providing a business process engine that executes business processes. The IBM Integration Developer is the development tool used to create standards based business processes that are deployed to the Business Process Manager.

IMS TM Resource Adapter Version 12 New features

- **Support for WebSphere Application Server Version 8 and its resource workload routing function**
 - new resource workload routing function that offers failover and subsequent failback
- **Support for multiple data stores per IMS activation specification for callout messages**
 - This enhancement enables a single message-driven bean (MDB) to pull callout messages from more than one IMS data store.
- **Data store connection failure recovery for callout messages**
 - When the backend IMS is not available, in addition to the attempt to reconnect to IMS Connect, the resource adapter will also attempt to reconnect to the IMS data store.
- **Support for IMS V12 OTMA DFS2082 messages for commit-then-send CM0 transactions**
 - if the IMS application does not reply to the IOPCB or complete a message switch to another transaction, OTMA issues a DFS2082 message to the client to indicate the transaction terminated with no reply.
 - This enhancement enables you to convert send-then-commit (CM1) transactions into CM0 transactions without having to modify your applications.
- **Support for IMS Version 12 RACF return codes**
 - In IMS V12, if a RACF security failure occurs, IMS Connect includes a 2-byte return code from the RACF RACROUTE REQUEST=VERIFY command.

529

WAS V8 introduces a new function called Resource Workload Routing, which includes data source and connection factory failover and subsequent failback from a predefined alternate or backup resource. This function enables an application to easily recover from resource outages, such as database failures, without having to embed alternate resource and configuration information.

Support for multiple data stores enables users to specify more than one IMS datastore in the IMSActivationSpec such that a single MDB can be used to pull callout messages from more than one IMS datastore.

The following changes are made to enhance the reconnect capability for retrieving callout messages:

- a) The resource adapter will attempt to reconnect to an IMS if the IMS datastore is not available.
- b) The log and traces are enhanced to remove unnecessary stack traces and provide informational messages to indicate a successful reconnection.
- c) A new -1 value is provided for the retryLimit property in the IMSActivationSpec to allow user to try to reconnect forever.

The resource adapter exploits this IMS 12 function by introducing a new InteractionSpec property, CM0Response, to allow customers to request the DFS2082 message for a CM0 input transaction.

The IMS TM resource adapter code is modified to display the RACF return code, reason code, and the error description to assist users in debugging the RACF error.

IMS TM Resource Adapter Transaction Expiration Support

Apar IMS 10 PM01727
IMS 11 PM01822

- **Invoke message level transaction expiration**
 - New IMSInteractionSpec boolean property added along with getter and setter
 - transExpiration
 - getTransExpiration()
 - setTransExpiration()
 - Execution times out OTMA discard or dequeue the transaction
- **Benefit**
 - reduce the CPU cycles for processing the unwanted transactions

530

PM01727/PM01822 (PM01727/PM01822)

Transaction Expiration support is extended to IMS TM Resource Adapter (TM RA). This support enables clients to use the transaction expiration function which allows IMS to expire the input transactions without processing them, to reduce the CPU cycles for processing the unwanted transactions.