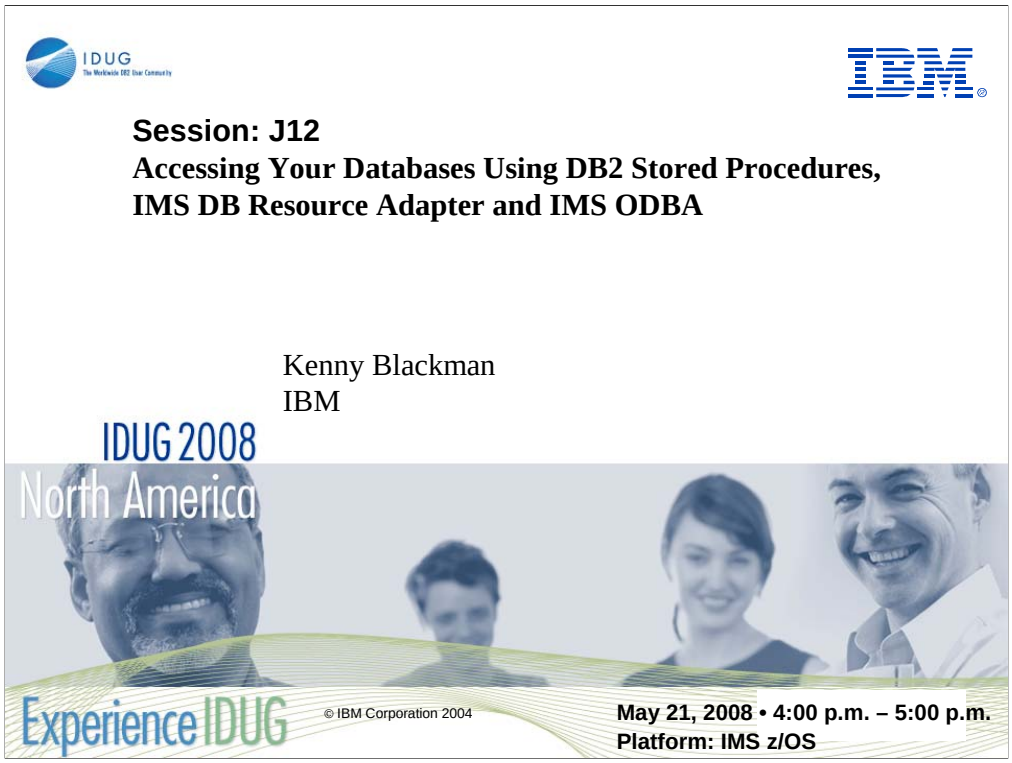


You have DB2 for z/OS and IMS Databases and you have client applications that need to access your data using Java Stored Procedures. In this presentation we describe how to configure and use the IMS DB Resource Adapter in a Stored Procedure to access your IMS databases. The presentation will also include setup examples.

Agenda

- How Stored Procedures can be used to access IMS
- How to configure DB2 and IMS to support Java Stored Procedures
- How RRS is used to provide syncpoint coordination
- What is the IMS DB Resource Adapter
- How the IMS DLIModel utility can be used to create a Java Database view of an IMS database

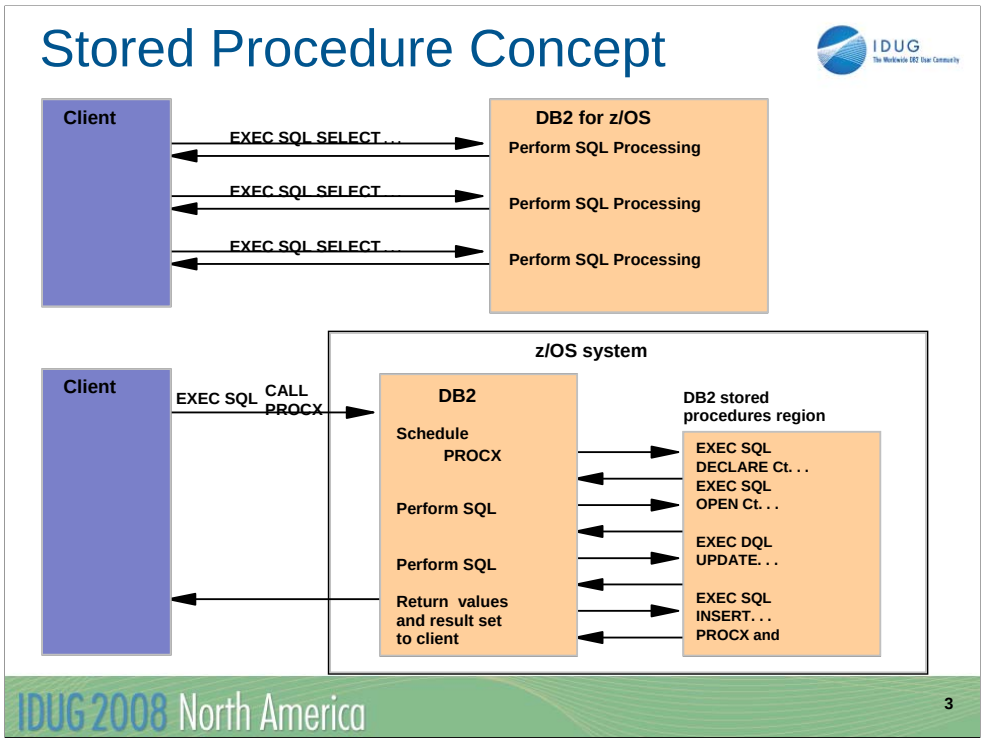
How Java Stored Procedures can be used to access IMS

Objective 2: How to configure DB2 and IMS to support Java Stored Procedures

Objective 3: How RRS is used to provide syncpoint coordination

Objective 4: Understand how client application can access the Stored Procedure and process the Result Set

Objective 5: How the IMS DLIModel utility can be used to create a Java Database view of an IMS database

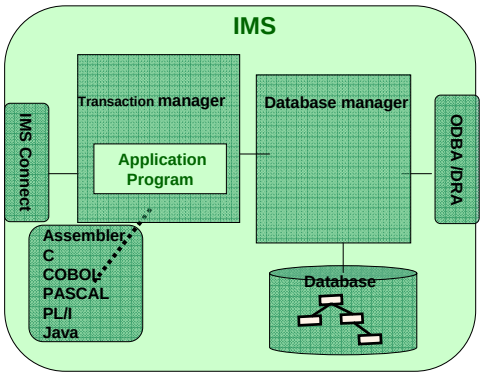


You can see from the following illustration that the Network traffic was significantly reduced.

Rather than having a Send/Receive for each SQL statement there is 1 Send and 1 Receive

What is IMS?

- Transaction Management System
- Multilingual Application Program server
- Database Management System

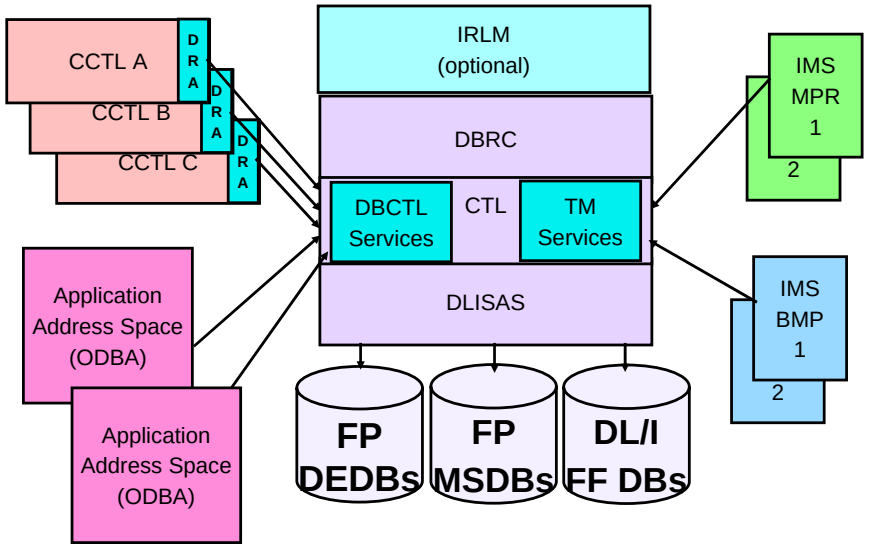


Services accessed without knowledge of underlying implementation

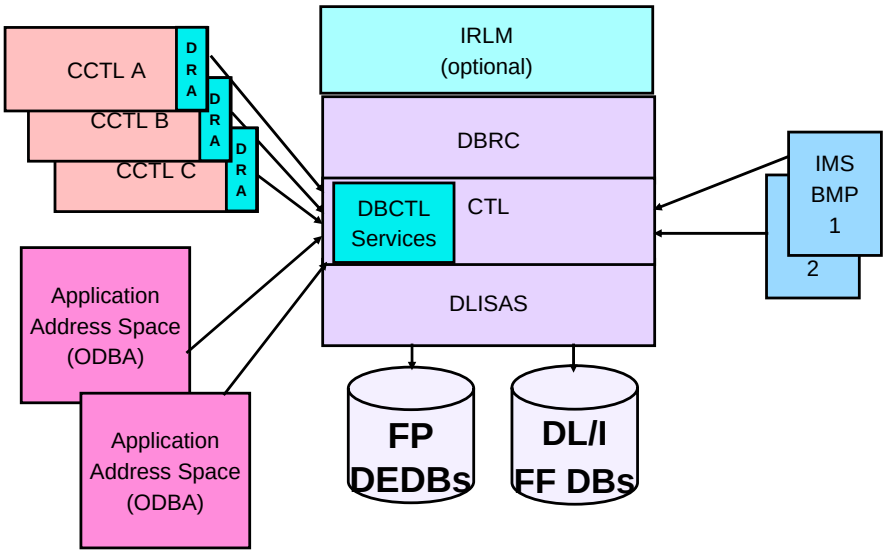
Services written in C# running on .Net platforms and services written in Java running on Java EE platforms can both be consumed by a common composite application

According to IBM's own estimates, SOA has a market opportunity of $65 billion this year. Gartner says that by 2008, SOA will provide the basis for 80 percent of new IT transformation projects.

IMS TM/DB Subsystem Structure



IMS DBCTL Subsystem Structure



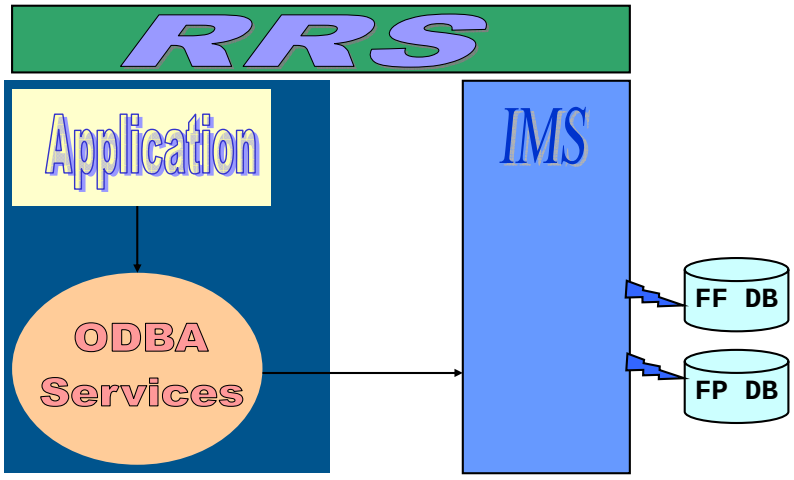
What is ODBA

- ODBA = Open DataBase Access
- DL/I API using the IMS AIB (Application Interface Block)
- Allows more direct access to IMS Databases
- Uses Resource Recovery Services (RRS)
  - RRS is the sync point coordinator
  - Allows for distributed syncpoint coordination

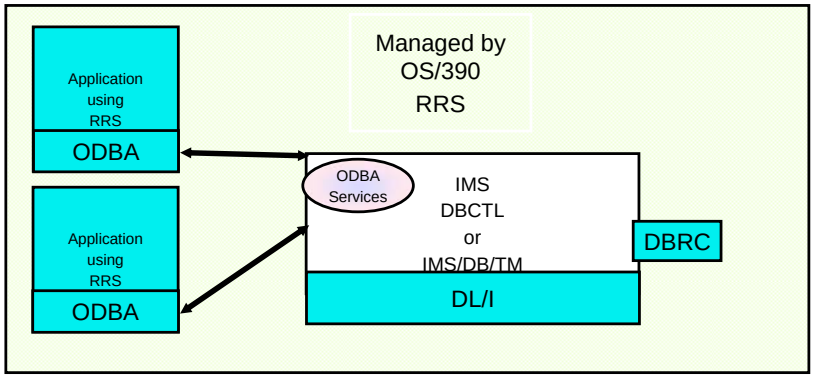
ODBA connection to IMS

- Connects to any IMS online subsystem with DB
  - DB/TM or DBCTL
- Connect to one or more IMS subsystems
  - ODBA must be on the same LPAR as IMS(s)
- One or more ODBA address spaces can connect to a single IMS
- ODBA must reside on the same LPAR as IMS

Application address space

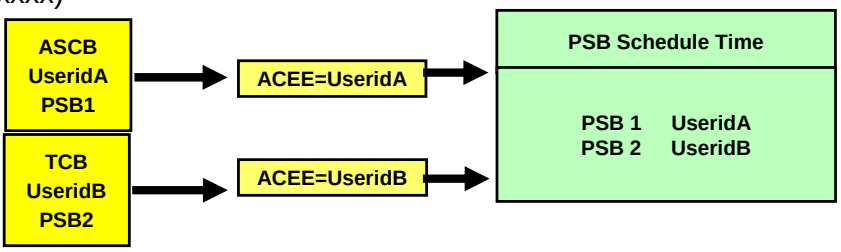


Multiple Applications to one IMS system



ACEE PSB Security

- ODBASE=Y/N
  - ACEE - Authenticated USERID
  - The IMS application group resource class (AIMS or Axxxxxxx)

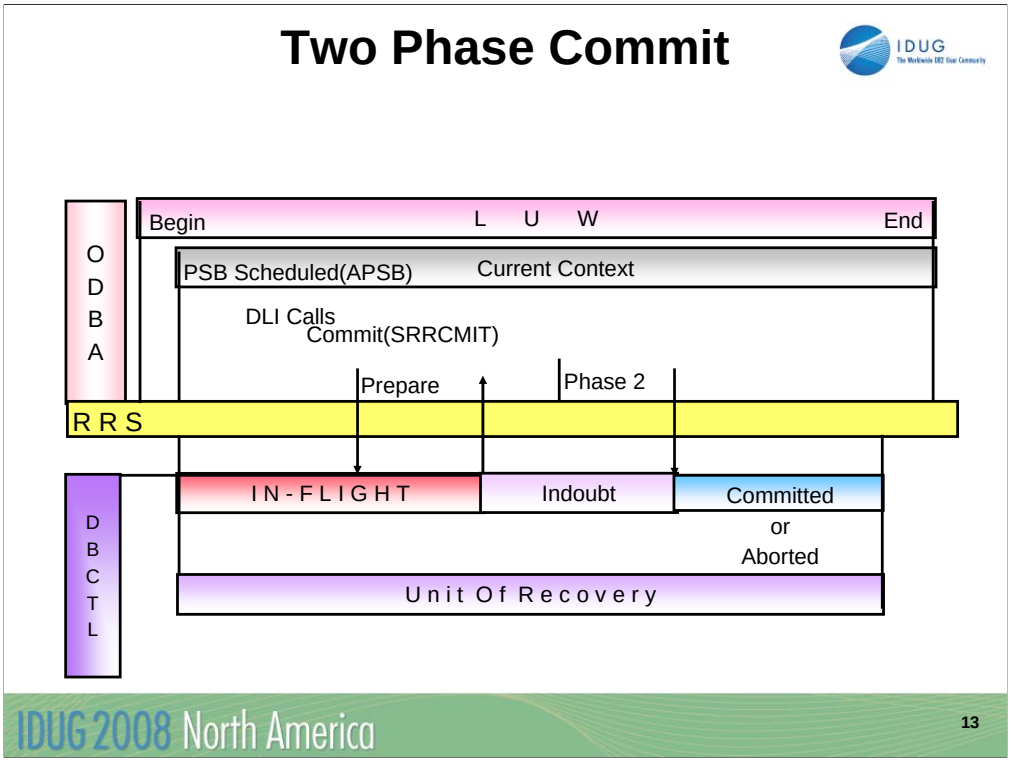


USERID from JOBCARD = ASCB
USERID from THREAD = TCB

DFS2855A APSB SECURITY CHECK FAILED...
and
ABENDU0438

Resource Recovery Services(RRS)

- A sync-point manager to coordinate the two-phase commit process
- Commit and Backout callable services for application programs
- A mechanism to associate resources with an application instance
- Manages Unit of Recovery(UR)



The *two-phase commit* protocol is a set of actions used to make sure that an application

program either makes *all* changes to the resources represented by a single unit of recovery

(UR), or makes *no* changes at all. This protocol verifies that either all changes or no changes

are applied even if one of the elements (such as the application, the system, or the resource

manager) fails. The protocol allows for restart and recovery processing to take place after

system or subsystem failure.

The two-phase commit protocol is initiated when the application is ready to commit or back

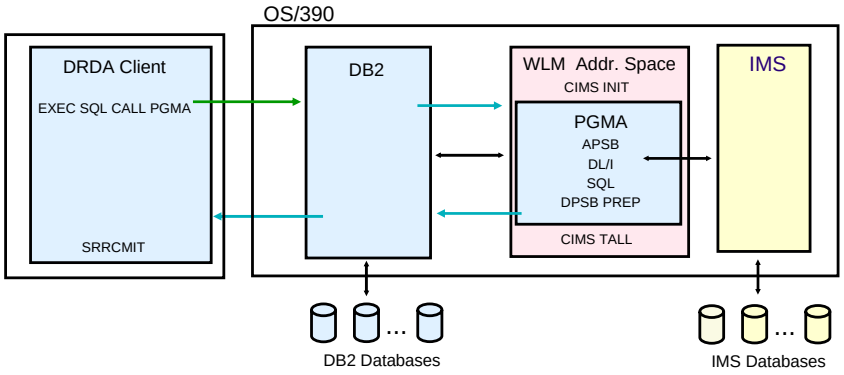
out its changes. The application program that initiates the commit or backout does not need to

know who the syncpoint manager is or how two-phase commit works. This is hidden from the

application; a program simply calls a commit or backout service and receives a return code

indicating whether this has been successfully completed.

ODBA DB2 Stored Procedure Example



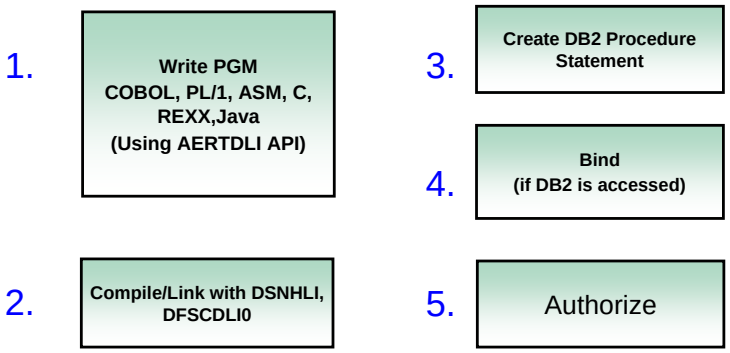
- DB2 stored procedure example
  - Requires WLM managed stored procedures address spaces
  - DRDA Client issues SQL for stored procedure
  - DB2 invokes stored procedure
  - Stored procedure does SQL and DL/I calls
  - Client program does commit when stored procedure returns
    - or DB2 can issue SRRCMIT

This example shows a DRDA client issuing an SQL call which invokes a DB2 stored procedure. The call could be issued by any DB2 program, including one running in the MVS system where DB2 and IMS are executing.

The DB2 Stored Procedure must run in a Workload Manager (WLM) established stored procedures address space. This is required for support of recoverable resources (RRS/MVS). DB2 Version 5 or a later release is required for the use of these WLM established address spaces.

DB2 Stored Procedure to ODBA Setup

- ODBA requires z/OS RRS



To use ODBA requires Z/OS Resource Recovery Services.

The steps to create an ODBA stored procedure are described on this foil.

1. Write the program using an IMS supported language, such as COBOL, PL/1, ASM, C, REXX and Java, using the AERTDLI call API to pass the input and out parameters to the program.

2. Compile and link the program using the DSNHLI proc.

3. Create the DB2 Procedure statement identifing this stored procedure, an example is included on the next foil.

4. Bind the application (this step is only needed if DB2 is also being called by this program.

5. Authorize the use of this Stored Procedure using the DB2 "GRANT BIND..." command.

DB2 Stored Procedure to ODBA Setup...

```
CREATE PROCEDURE      DSN2.JAVASP1(
 INOUT COMMAND CHAR(8)     CCSID EBCDIC,
 INOUT CUST_ID    INT,
 INOUT EMAIL CHAR(40)    CCSID EBCDIC,
 INOUT COMPANY_NAME   CHAR(20)    CCSID EBCDIC,
 INOUT LAST_NAME      CHAR(20)    CCSID EBCDIC,
 INOUT FIRST_NAME     CHAR(20)    CCSID EBCDIC,
 INOUT ADDRESS        CHAR(30)    CCSID EBCDIC,
 INOUT CITY           CHAR(20)    CCSID EBCDIC,
 INOUT STATE          CHAR(7)     CCSID EBCDIC,
 INOUT COUNTRY        CHAR(20)    CCSID EBCDIC,
 INOUT FLAG           CHAR(1)     CCSID EBCDIC,
 OUT   CALL_STATUS    CHAR(40)    CCSID EBCDIC,
 OUT   NEW_CUSTID     INT,
 OUT   OUT_INT2       INT,
 OUT   AIBRETRN       INT,
 OUT   AIBREASN       INT)
 FENCED
 RESULT SETS 0
 EXTERNAL NAME JAVAPGSP
 LANGUAGE COBOL
 PARAMETER STYLE GENERAL
 NOT DETERMINISTIC
 NO SQL
 NO DBINFO
 NO COLLID
 WLM ENVIRONMENT DB2JODBA
 ASUTIME LIMIT 50
 STAY RESIDENT NO
 PROGRAM TYPE MAIN
 SECURITY DB2
 RUN OPTIONS 'TRAP(OFF),RPTOPTS(OFF),TERMTHDAC(QUIET),NONOVR'
 COMMIT ON RETURN YES;
 GRANT EXECUTE ON PROCEDURE DSN2.JAVASP1 TO PUBLIC;
```


This foil shows an example of the Create Procedure statement used in developing our DB2 Cobol Stored Procedure to ODBA.

DB2 Stored Procedure to ODBA Setup...

- DB2 SP requires a WLM DB2 Stored Procedure Address Space

```
//*******************************************************
**
//DB2JODBA PROC RGN=0K,APPLENV=DB2JODBA,DB2SSN=DB2J,NUMTCB=8
//IEFPROC EXEC PGM=DSNX9WLM,REGION=&RGN,TIME=NOLIMIT,
//        PARM='&DB2SSN,&NUMTCB,&APPLENV'
//STEPLIB  DD  DISP=SHR,DSN=CEE.SCEERUN
//         DD  DISP=SHR,DSN=DB2V610J.TEST.SDSNLOAD
//         DD  DISP=SHR,DSN=DSN610.SDSNLOAD
//         DD  DISP=SHR,DSN=IMS610P.PGMLIB
//         DD  DISP=SHR,DSN=IMS610P.RESLIB
//DFSRESLB DD  DISP=SHR,DSN=IMS610P.RESLIB
```

DB2SSN

- DB2 SUBSYSTEM NAME

APPLENV

- WLM ENVIRONMENT statement (previous page)

DFSRESLB

- Indicates DB2 is to init ODBA support

To run a DB2 Stored Procedure calling ODBA requires a WLM DB2 Stored Procedure Address space to be set up.

This foil shows sample JCL that was used for our sample to IMS Customer data. Since our sample does not access DB2, there are no DB2 libraries included in the STEPLIB of this proc.

The DB2SSN name describes the DB2 Subsystem being accessed.

The APPLENV name matches the WLM ENVIRONMENT statement included in the "Create Procedure" listed on the previous page.

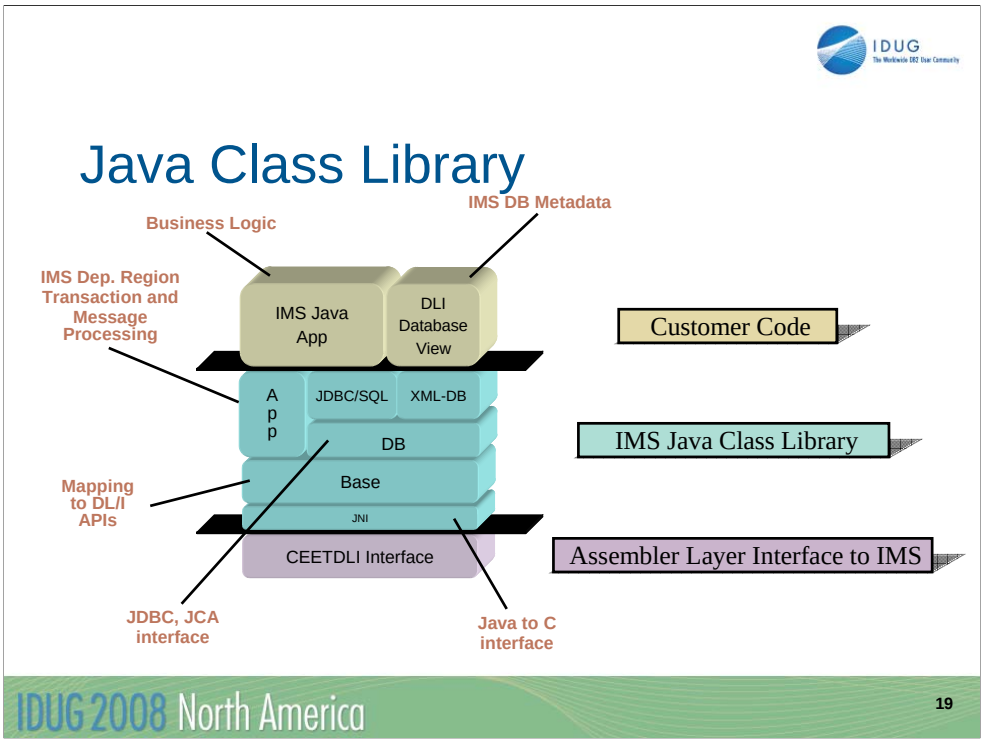
DB2 COBOL Sample

- COBOL code

```
PROCEDURE DIVISION
        USING  IO-COMMAND,IO-LAST-NAME,IO-FIRST-NAME,
          IO-ADDRESS,IO-CITY,IO-STATE,IO-COUNTRY,
          OUT-MESSAGE,OUT-AIBRETRN,OUT-AIBREASN.
        MOVE APSBNME to AIBRSNM1.        set PSB NAME in AIB
        MOVE TDBCTLID to AIBRSNM2.       set value for DFSPRP table
        CALL 'AERTDLI' USING APSB, AIB.  allocate the PSB
        MOVE DPCBNME to AIBRSNM1         set DB PCB NAME in AIB

        CALL 'AERTDLI' USING GET-HOLD-UNIQUE, AIB, IOAREA, SSA

        SET ADDRESS OF DBPCB TO AIBRESA1  set address of DBPCB from AIB
        IF DBSTATUS = 'GE'
          ...
        MOVE APSBNME to AIBRSNM1.        set PSB NAME in AIB
        MOVE SFPREP to AIBSFUNC.         set PREP subfunction in AIB
        CALL 'AERTDLI' USING DPSB, AIB.  deallocate the PSB
```



Bottom – c i/face (depending on the environment 3 interfaces) – have to drop down to c (thin jni layer)

Base – 1-1 mapping of the way ims works under covers / in java build SSAs & make db calls using dli

Db – really what is turning this in to our jdbc driver/ making sql calls

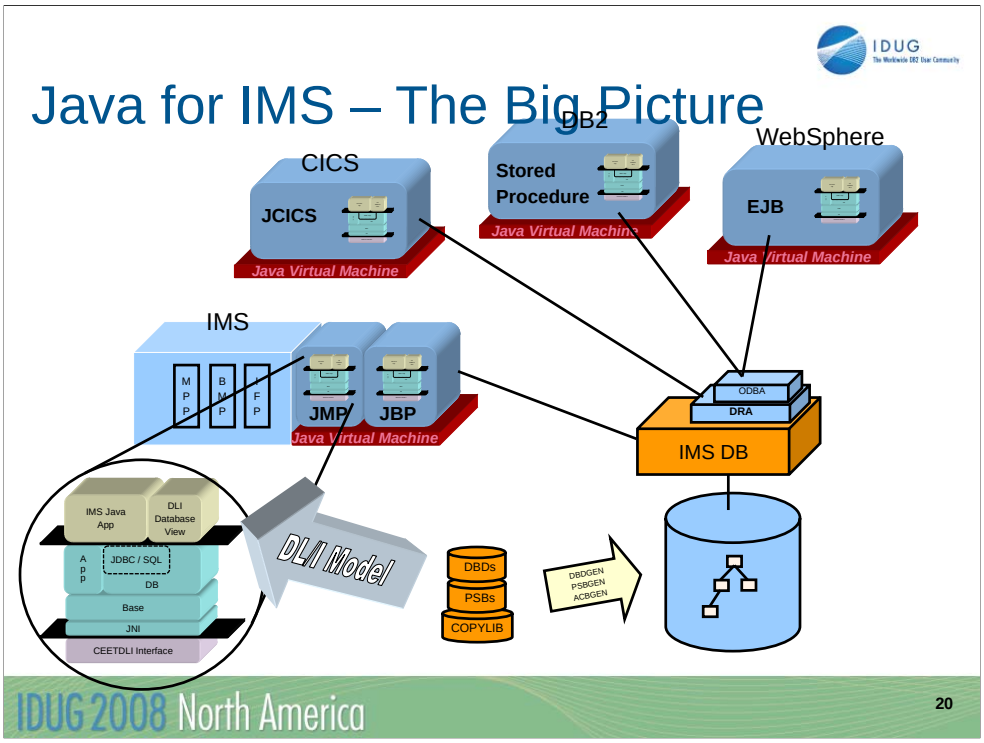
App - running in an ims dependent region and offers reading/writing messages to ims message queue

Customer code – not worry about the dli / only jdbc

Dbview- ims does not have online metadata; good – now we added new stuff (XML

bad –

Tooling – generates database view called dli model utility



Showing all he environs of imsjava

Jmp analogous to mpp

Jbp to bmp

Cics

Db2 sp

Was ejbs

All run on jvm/ persistent reusable jvm/ issues performance – bring up own jvm/interpreter and run app/ 2-3 mill instructions/ leave master jvm up and reset the worker jvm/ each java app has a fresh view of the jvm

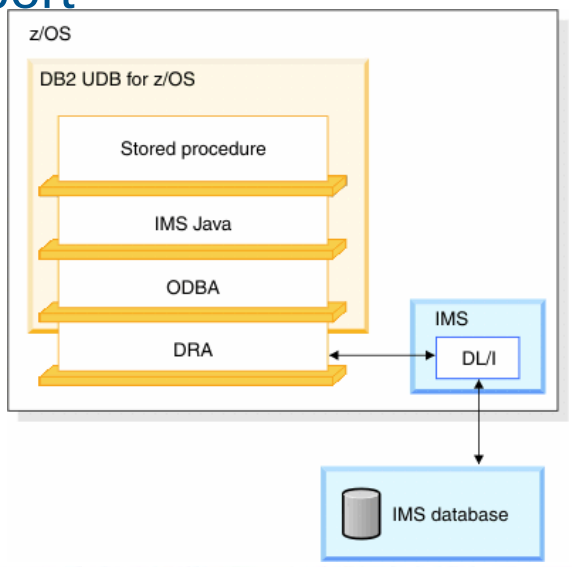
The diff in all 4 environs is how often the reset is done

Cics – resets for every transaction

Was – never resets the jvm/ app has to know if status has changed

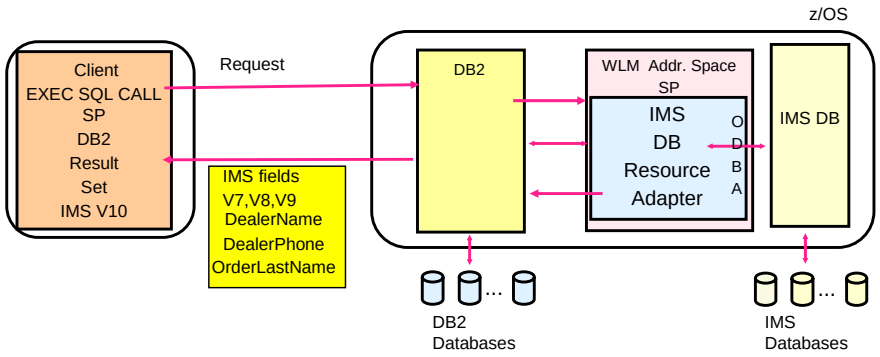
Db2 – every 256 trans

DB2 Stored Procedure Support



IMS DB Resource Adapter

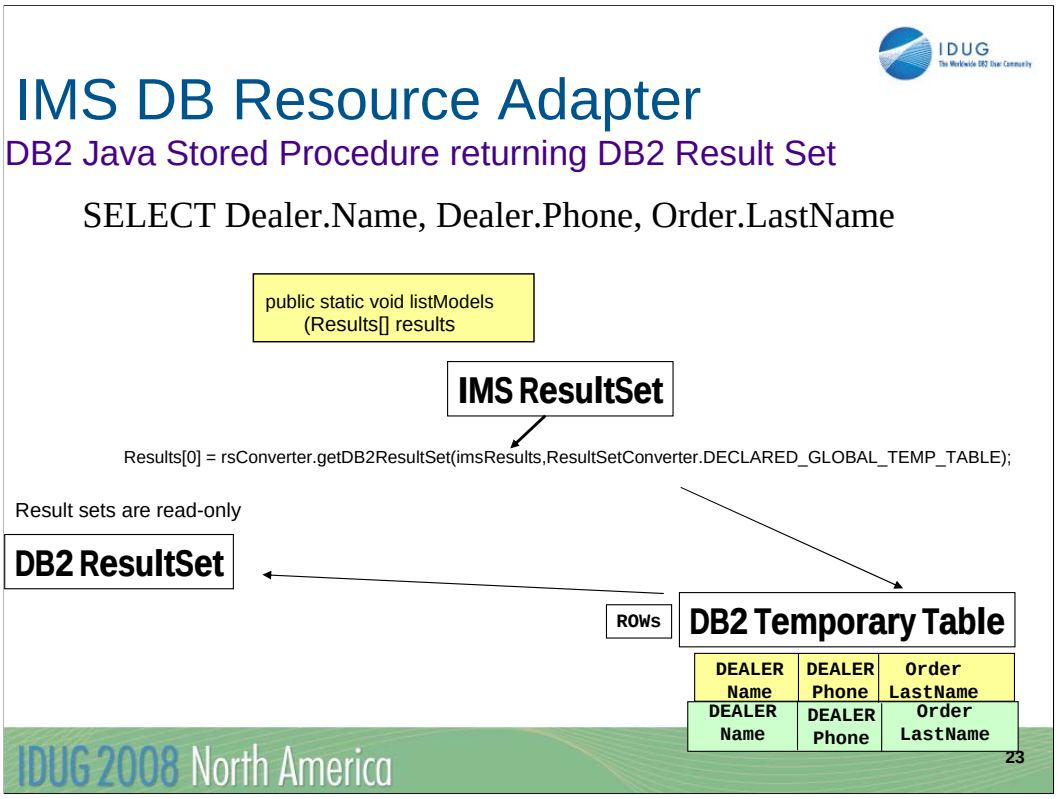
DB2 Java Stored Procedure Support



Stored procedures are application programs executed by DB2 in response to an SQL CALL statement

When the DB2 Stored Procedure needs to access IMS DB the IMS Open Database Access (ODBA) function is activated.

Notice in IMS V7,V8 and V9 individual IMS fields are returned to the client. In IMS V10 a DB2 Result Set is returned to the client.



By embedding the IMS data within a DB2 result set the type of data and the amount of data does not need to be known when the stored procedure is developed.

The DB2 global temporary table (GTT) is either created or declared to be used in the process of converting an IMS result set to a DB2 result set.

The IMS DB Resource Adapter ResultSetConverter class is used to convert an IMS result set instance into a DB2 result set instance. This method takes in an instance of com.ibm.ims.db.DLIResultSet object which is IMS DB Resource Adapter's implementation of the java.sql.ResultSet interface and converts it into an instance of the DB2 implementation of the java.sql.ResultSet interface

Note the ResultSet[] results replaces

(String[] dealername1, String[] dealerphone1, String[] orderlastname1,

String[] dealername2, String[] dealerphone2, String[] orderlastname2)

J12

Accessing Your Databases Using DB2 Stored Procedures, IMS DB Resource Adapter and IMS ODBA

Kenny Blackman

IBM

kblackm@us.ibm.com