SHARE in Orlando 2015

SHARE Educate · Network · Influence

Application Programming in the IMS World

Kenny Blackman
kblackm@us.ibm.com
Suzie Wendler
wendler@us.ibm.com
zGrowth IMS Washington Systems Center

Thursday March 5,2015

#16398 - Cedar

#SHAREorg

SHARE is an independent volunteer-run information technology association that provides education, professional networking and industry influence.

60 YEARS OF SHARE Influencing IT Since 1955

Copyright (c) 2015 by SHARE Inc. Except where otherwise noted, this work is licensed under http://creativecommons.org/licenses/by-nc-sa/3.0/

IMS 13 APAR PM78158 MPP,JMP,IFP regions PARDLI capability

- Current
  - For BMPs, PARDLI=1 means all DL/I processing is to be performed in the IMS control region to prevent control region system 113 abends resulting from system X22 abends in the BMP region
- Change
  - APAR PM78158 provides the ability to specify the PARDLI parameter for JMP, MPP, and IFP regions.
    - Note using PARDLI=1 for MPP, JMP, or IFP regions can seriously degrade performance. Use of PARDLI=1 for MPP, JMP, or IFP regions is intended only for application debugging purposes if needed.

IMS 13 APAR PM86872 IMS Timing Services and connecting to External Subsystems.

- Current
  - Application programs running in IMS dependent regions using STIMER= may not be terminated with ABENDU240 while in a long running call to an External Subsystem.
  - ABENDU240 was delayed until after the External Subsystem returned to IMS.
- Change
  - ABENDU240 will now be enforced in IMS dependent regions that are running in an External Subsystem (ESS) when time expires using IMS Timing Services.

ICAL Enhancements

Support for Truncated Messages

- **New “RECEIVE” subfunction code**
  - **With an expanded response area**
    - **Retrieves the response message after an ICAL “SENDRECV” is issued with an inadequate response area specification and gets partial data (AIB RC X’100’, AIB RS X’00C’)**
  - **IMS 13 keeps a copy of the entire response message in the control region private storage**
    - **Until a subsequent ICAL “SENDRECV”, syncpoint, or application termination**
- **Addresses**
  - **Partial response message due to inadequate application specification**
- **Benefit**
  - **Provides the ability to complete the retrieval of a reply message**
    - **Without having to re-issue a complete ICAL “SENDRECV” and associated network transmission costs**

ICAL subfunction RECEIVE

- **Format:**
  - >>-ICAL--*aib*--*response area*--><

- **AIB**
  - ***AIBSFUNC*** **value “RECEIVE”**
  - ***AIBOAUSE*** **is used as an input and output parameter based on AIBSFUNC**
    - **For the “RECEIVE” call**
      - **Contains the length of the response area**
  - ***AIBOALEN*** **= request area length**
    - **Used as an *output parameter for “RECEIVE”***
      - **When complete response is returned in *response area*, this field is 0**
      - **If partial data is returned (AIB RC X'100', RS X'00C'), this field contains the actual length of the response message**

ICAL sub-function RECEIVE …

- **Usage example:**
  - **ICAL --*aib—request area, response area***
    - ⇨ **AIBSFUNC (SENDRECV)**
    - ⇨ **AIBOAUSE – Response area length**

  - ➤ **CALL is issued → AIBRETRN=x'100', AIBREASN='00C'**
    - × **Specified length of the output response area is too small**
    - × **AIBOAUSE= length of the data that was returned in the response area**
    - × **AIBOALEN = the actual length of the entire response message**

  - ➤ **Using the value in the previous AIBOALEN and leveraging the new support which keeps the message in IMS CTL region private, retrieve the entire response:**

  - **ICAL --*aib— response area***
    - ⇨ **Where *response area* has been expanded to contain the entire message**
    - ⇨ **AIBSFUNC (RECEIVE)**
    - ⇨ **AIBOAUSE – new response area length**

  - ➤ **CALL is issued successfully**
    - ✓ **AIBOAUSE – length of the response in the response area**
    - ✓ **AIBOALEN – set to 0 because the call successfully returned the entire response**

ICAL sub-function RECEIVE …

- ICAL “RECEIVE” is only valid if previous ICAL “SENDRECV” failed

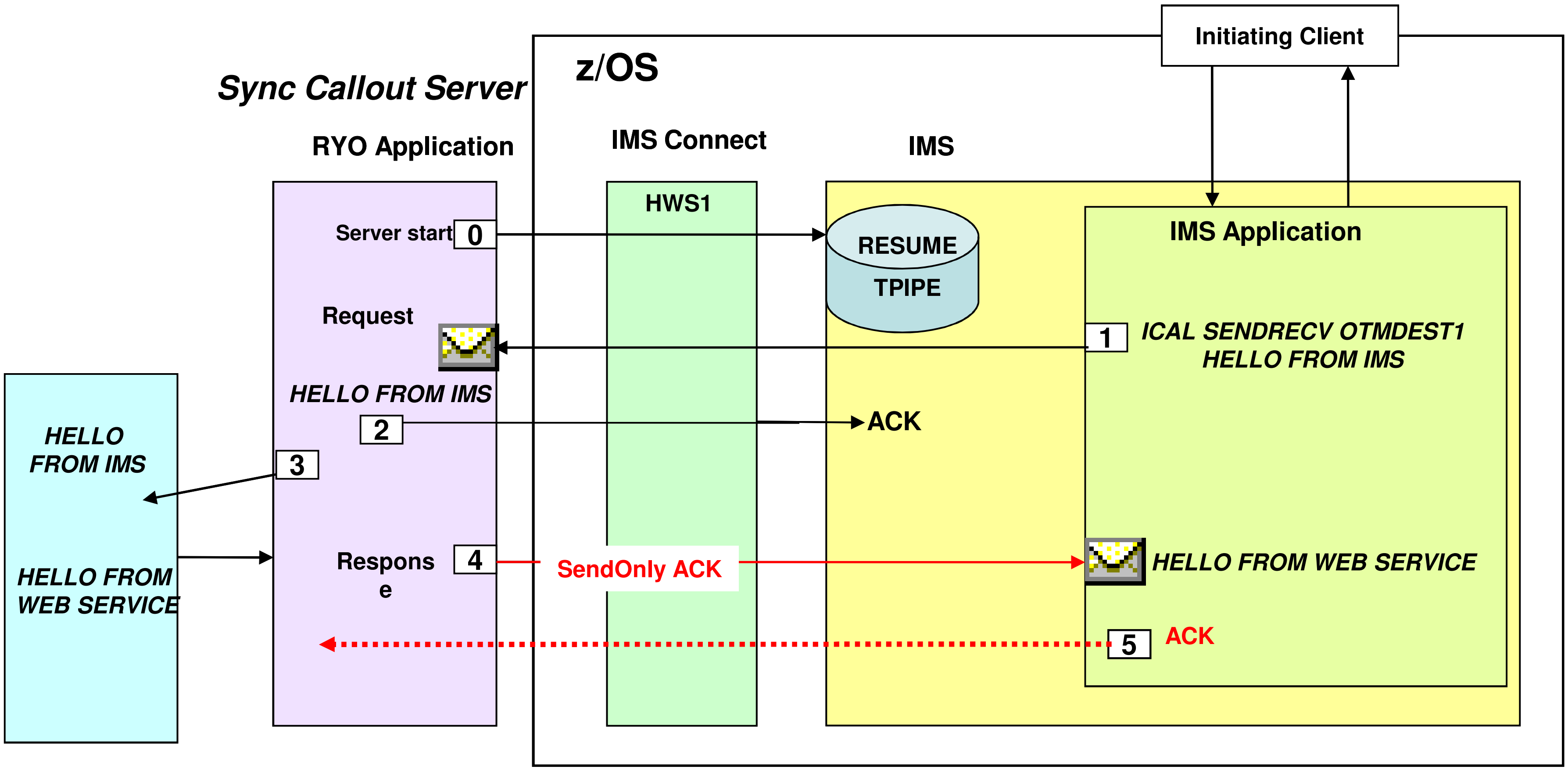
- Response data is available for retrieval until:
  - A new ICAL call with sub-function code SENDRECV is issued
  - When the IMS application reaches a syncpoint
    - Checkpoint for an BMP application
  - Abnormal termination

AIBUTKN

- **New AIB field - AIBUTKN**
  - Provides optional specification of a 1-8 byte map name
  - Included in the OTMA state data prefix to be sent to the callout destination
  - IMS 12: PM73135/UK82636

- **Benefit**
  - Ability to send a name to a remote ICAL destination that can be used for message formatting or service identification purposes

IMS 12 Synchronous Callout SendOnly Ack SPE ...



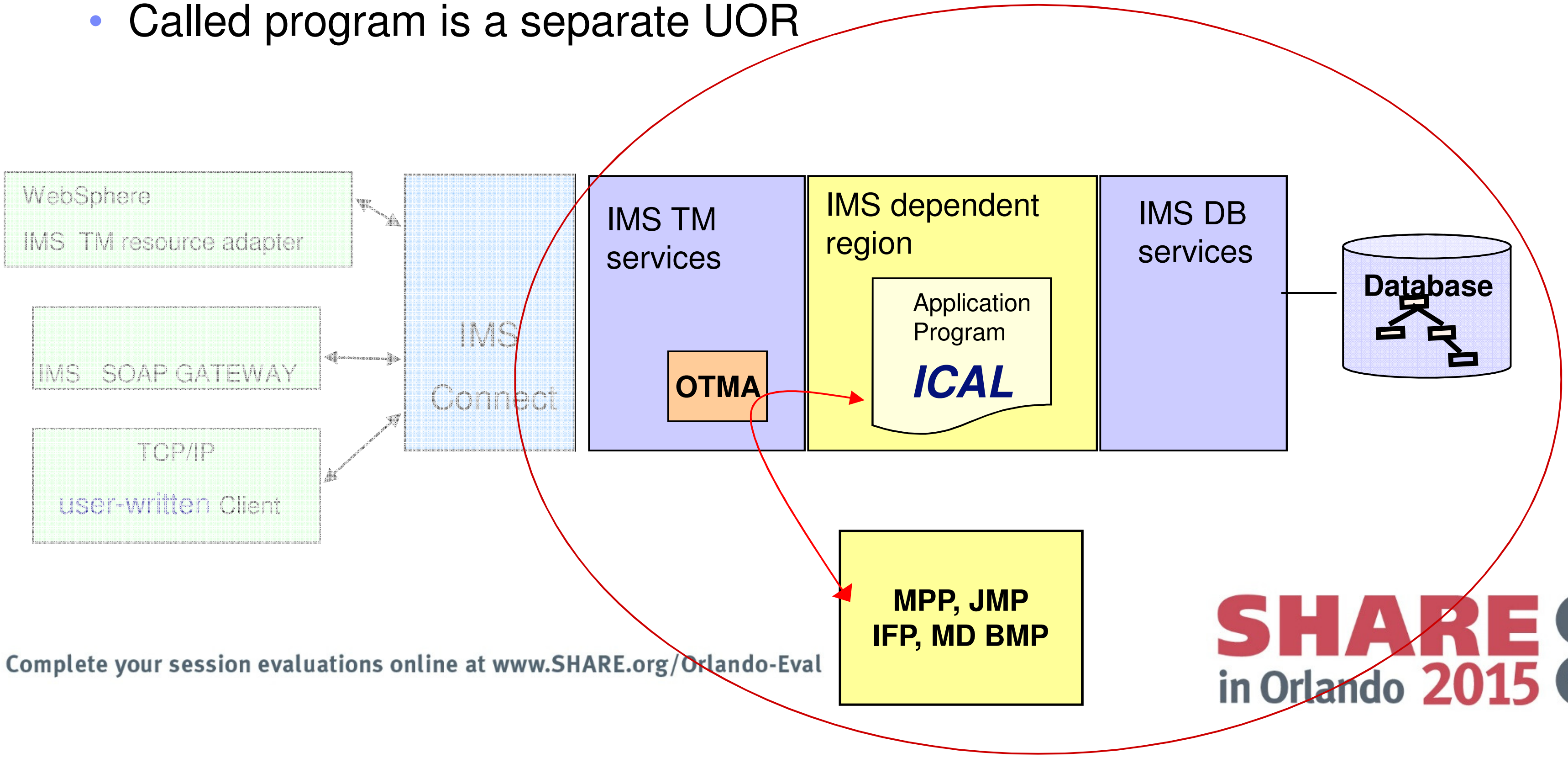
- IMS Connect 12: APAR PM39569 (PTF UK74666)
- IMS OTMA 12: APAR PM39562 (PTF UK74653)

IMS 12 Synchronous Callout SendOnly Ack SPE

- **IMS 13 APAR for Callout Send-Only ACK SPE**
  - **The ACKs were sent back with the complete response message text**
    - **This could be very large**
  - **IMS 13 APAR PM90943 allows the OTMA Client to request that the request message text not be send back with the ACK**
  - **IMS Connect 13 APAR PI10653 adds flag IRM_F1_SOARSP to allow the IMS Connect Client to request that the request message text not be returned with the ACK**

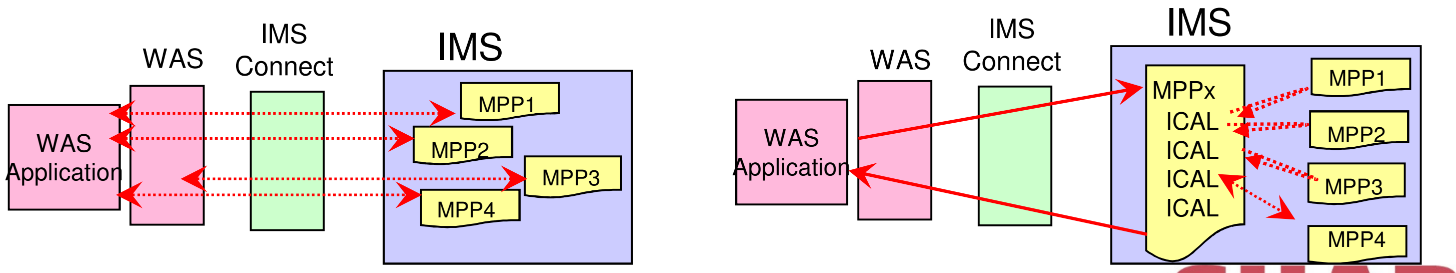
Synchronous Program Switch

- **New** capability that enhances the DL/I ICAL support
  - Allows an IMS application program to *synchronously* call and wait for a reply from *another IMS application program*
    - Within the calling program's UOR
    - Called program is a separate UOR



Synchronous Program Switch…

- Benefits
  - Modernization of the IMS application infrastructure
    - Provides an internal service flow of IMS transactions to complete a business process
      - In the same IMS or a different IMS
  - Implementation of a Process Server or Broker inside IMS
    - Reduces unnecessary network traffic when accessing multiple applications in the same IMS or IMSplex

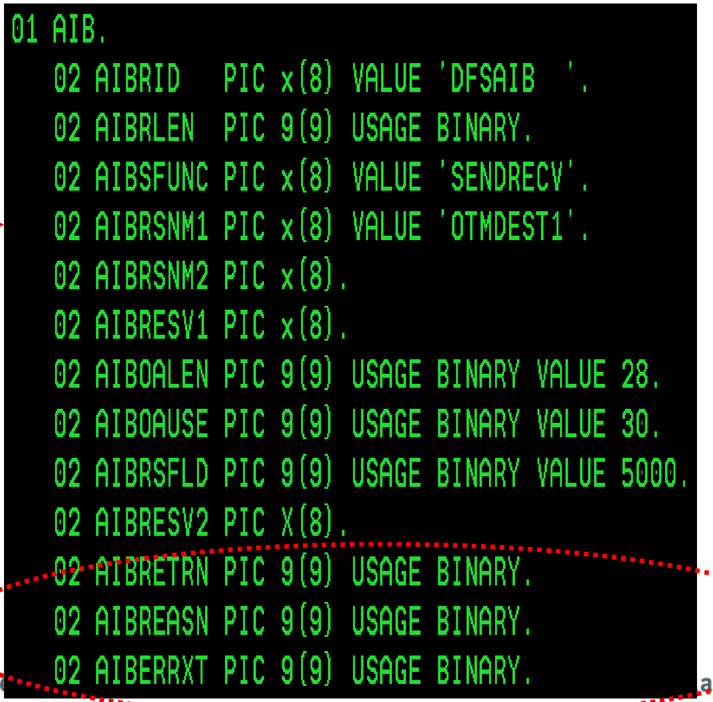


The DL/I ICAL call

Same Format

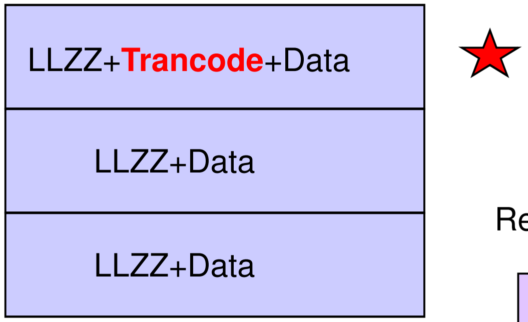
>>-ICAL--aib--request_area--response_area--------------------><

Call Name	DB/DC	DBCTL	DCCTL	DB Batch	TM Batch
ICAL	X		X		



```
01 AIB.
   02 AIBRID   PIC x(8) VALUE 'DFSAIB  '.
   02 AIBRLEN  PIC 9(9) USAGE BINARY.
   02 AIBSFUNC PIC x(8) VALUE 'SENDRECV'.
   02 AIBRSNM1 PIC x(8) VALUE 'OTMDEST1'.
   02 AIBRSNM2 PIC x(8).
   02 AIBRESV1 PIC x(8).
   02 AIBOALEN PIC 9(9) USAGE BINARY VALUE 28.
   02 AIBOAUSE PIC 9(9) USAGE BINARY VALUE 30.
   02 AIBRSFLD PIC 9(9) USAGE BINARY VALUE 5000.
   02 AIBRESV2 PIC X(8).
   02 AIBRETRN PIC 9(9) USAGE BINARY.
   02 AIBREASN PIC 9(9) USAGE BINARY.
   02 AIBERRXT PIC 9(9) USAGE BINARY.
```

Request Data (example of multi-segment):

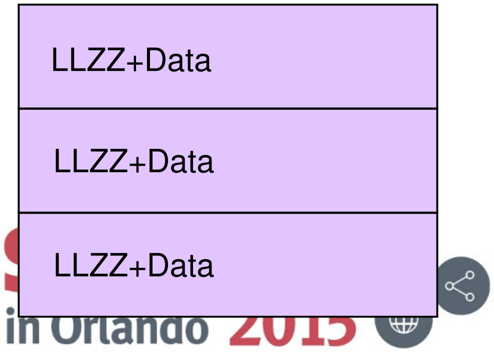


LLZZ+**Trancode**+Data

LLZZ+Data

LLZZ+Data

Response Data in multi-segment:



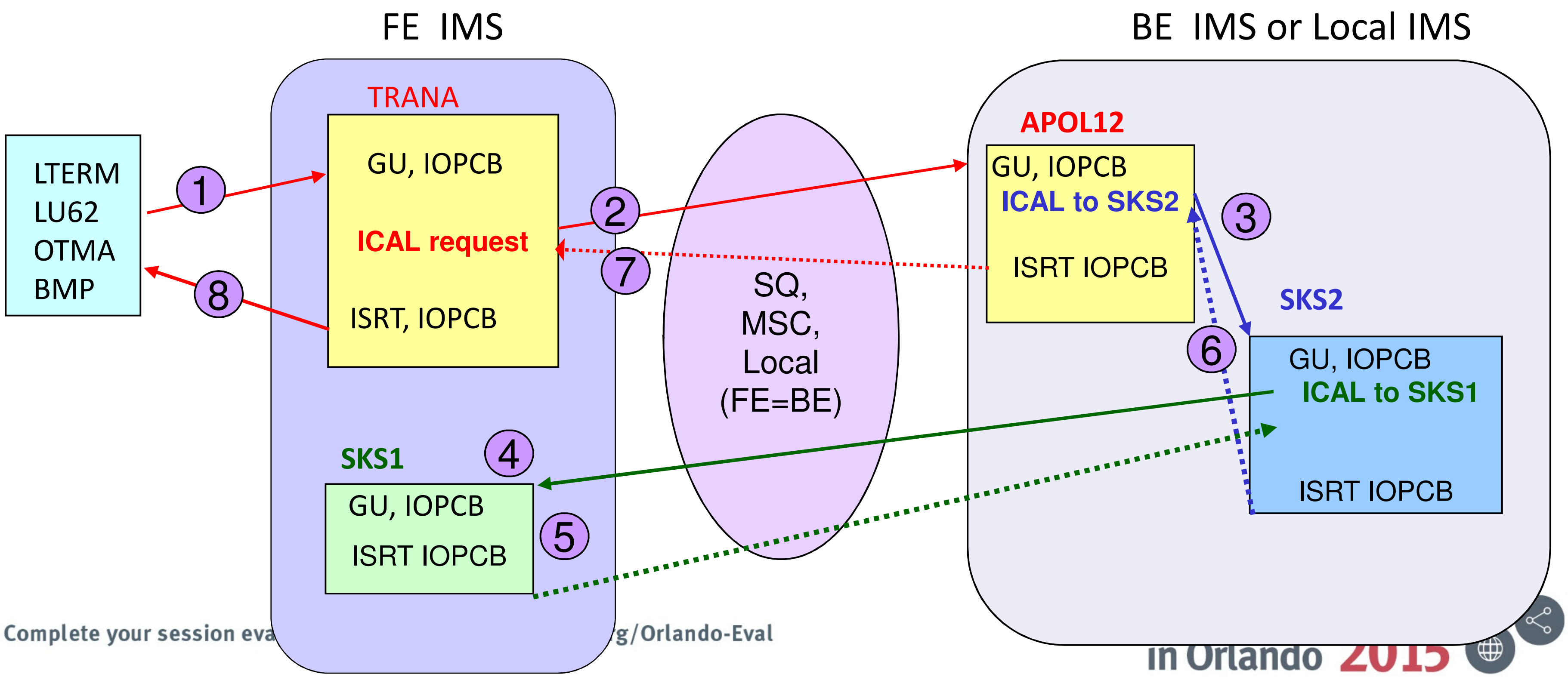
LLZZ+Data

LLZZ+Data

LLZZ+Data

Application Examples…

- The IMS application environment supports recursive requests
  - ICAL to ICAL
    - Across a single or multiple IMS systems



OTMA Transaction Expiration and Shared Queues SPE

- **SPE: APAR/PTFs**
  - **IMS 10: PM05985 (UK75413/UK75414)**
  - **IMS 11: PM05984 (UK74312/UK74313)**
  - **IMS 12: PM46829 (UK75415/UK75416)**

- **Enhancements**
  - **Options when transaction expiration occurs at application GU time**
    - **Option to suppress or display symptom dumps and DFS554A messages**
    - **Option to return input message instead of DFS3688I**
  - **Improved routing capability of Shared Queues back-end ALTPCB output**
  - **Improved usability of /DIS TMEMBER TPIPE command**

OTMA Descriptors

- OTMA destination descriptor enhancements
  - TYPE={MQSERIES}
    - Provides asynchronous callout and messaging support (ISRT ALTPCB)
  - EXIT={YES | NO}
    - Specifies whether or not the OTMA exits are to be called
- Corresponding enhancement to IMS Type-2 OTMADESC commands
  - [CREATE | UPDATE | DELETE |QUERY]   OTMADESC
    - Recovered across warm and emergency restarts
- New/changed member control cards in DFSYDTx requires an IMS COLD start to take effect (not new to IMS 13)
- Benefits
  - Simplifies asynchronous messaging to WMQ
  - Removes the need to code the OTMA exits, DFSYPRX0 and DFSYDRU0
  - Provides dynamic change capability with the Type-2 commands

IMS 12 SPE Enhancement

SSA Qualify By Position and Length

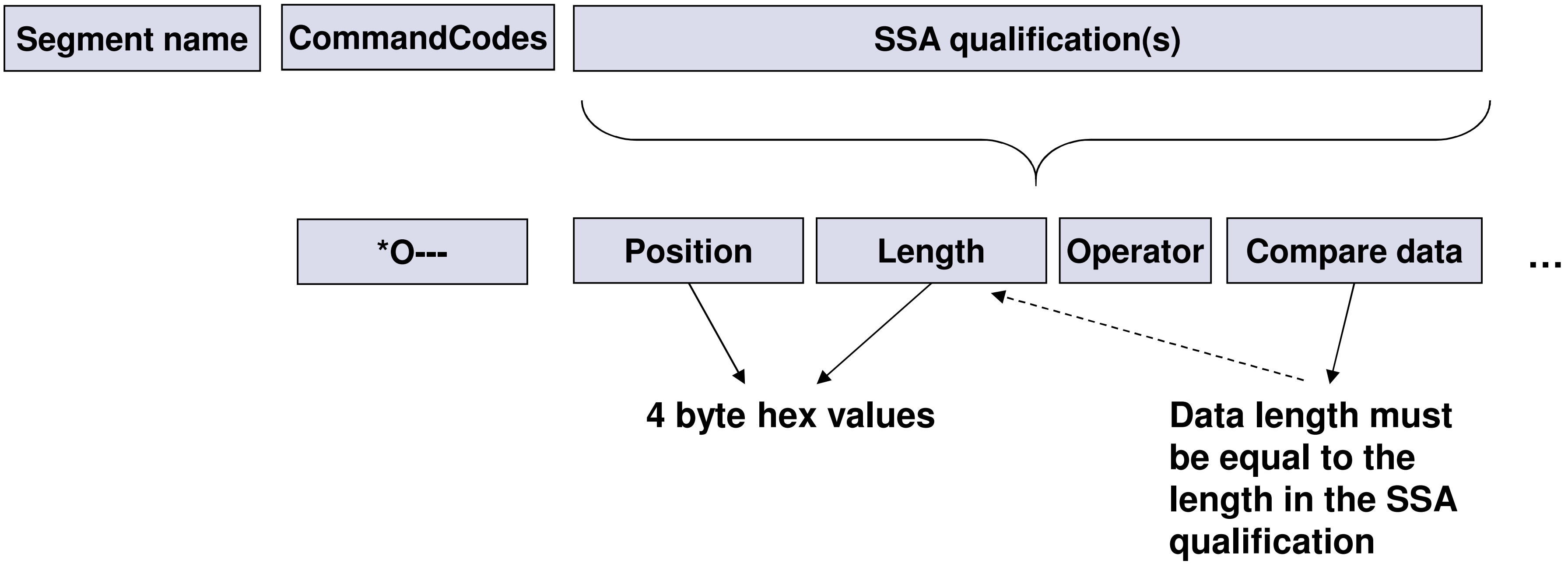
SSA Enhancement - Qualify by Position

- IMS 12 APAR PM65139 / PTF UK81837 & UK81838
  - New SSA command code “O”
  - Enhanced database SSA processing with ability to search for data in a segment by specifying a field position and length instead of a field name
  - Contains core IMS database code

- IMS 12 APAR PM69378 / PTF UK81917
  - Enhanced IMS Universal Drivers to allow SQL predicates containing ‘columns’ not defined in the DBD by internally converting ‘columns’ to position and length for SSA qualification
  - Contains IMS universal driver code

SSA Enhancement - Qualify by Position

- New SSA using “O” command code with position/length



SSA Enhancement - Qualify by Position

- New SSA with “O” command code, position and length

DBD

Field	Offset	Len
Labname	1	5
Street	10	20
State	30	2

Database

```
0          1          2          3
12345678901234567901235678901
SVL   DEV 555 BAILEY AVE     CA
ARC   RSC 650 HARRY RD       CA
```

COBOL Copybook

Field	Offset	Len
Labname	1	5
Type	6 ⟷	3
Street	10	20
State	30	2

```
GU IBMLABS *O(0000000100000005EQSVL   )
GU IBMLABS *O(0000000100000005EQARC   )
GU IBMLABS *O(0000001E00000002EQCA)
GU IBMLABS *O(0000000600000003EQDEV)
```

Position Length

‘bb‘ Status Code: all segments returned successfully

SSA Enhancement - Performance Consideration

- Performance will be similar to a search on a non-key field
- IMS will scan the database looking for field match(es)
- Qualification of the root key will help reduce the impact
- If business need requires searching on a non-key field
  - Consider defining the non-key field as a searchable field in the DBD

Fast Path Secondary Index Enhancement

- IMS 13 enhances the DEDB secondary index that was added in IMS 12
  - Add ability to use Boolean Operators to Segment Search Arguments (SSA)
    - AND = * or &
    - OR = + or |
  - Support specific Command Codes with Secondary Index search field

- Benefits
  - New and simplified programming opportunities with DEDBs
    - Allows ability to refine DL/I calls to Fast Path DEDBs
    - Commands supported when secondary index is accessed as a DEDB

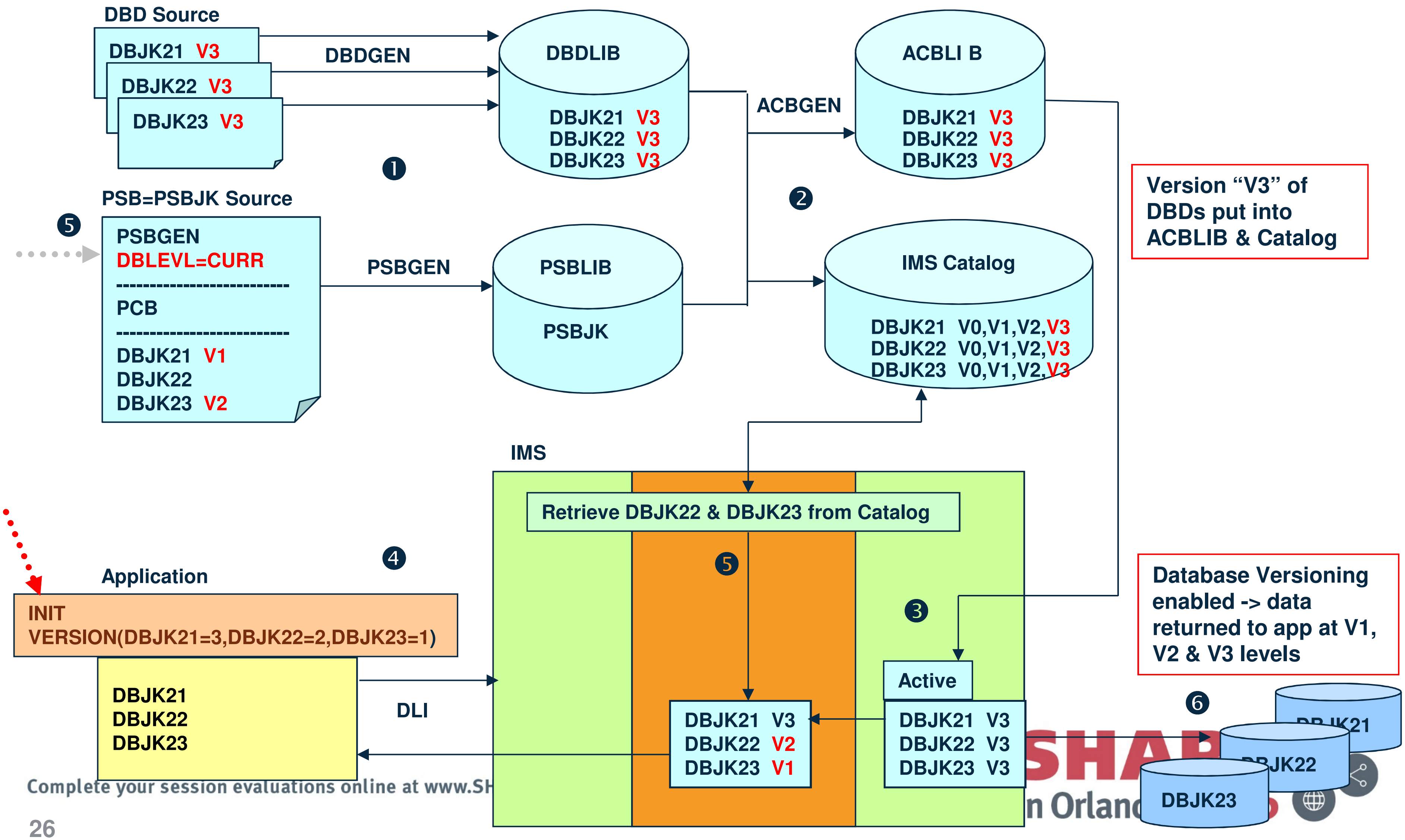
Database Versioning

Database Versioning Overview

- Database Versioning provides the ability to assign user-defined version identifiers to different versions of a database structure
  - Enables structural changes to a database while providing multiple views of the physical IMS data to application programs

- Applications referencing a new physical database structure can be brought online without affecting applications that use previous database structures
  - Applications which do not require sensitivity to the new physical structure, do not need to be modified and can continue to access the database

Database Versioning



DBD Source

DBJK21 V3
DBJK22 V3
DBJK23 V3

DBDGEN

DBDLIB
DBJK21 V3
DBJK22 V3
DBJK23 V3

❶

ACBGEN

ACBLI B
DBJK21 V3
DBJK22 V3
DBJK23 V3

❷

PSB=PSBJK Source

❺

PSBGEN
DBLEVL=CURR

PCB

DBJK21 V1
DBJK22
DBJK23 V2

PSBGEN

PSBLIB
PSBJK

IMS Catalog
DBJK21 V0,V1,V2,V3
DBJK22 V0,V1,V2,V3
DBJK23 V0,V1,V2,V3

Version “V3” of DBDs put into ACBLIB & Catalog

IMS

Retrieve DBJK22 & DBJK23 from Catalog

❹

Application

INIT VERSION(DBJK21=3,DBJK22=2,DBJK23=1)

DBJK21
DBJK22
DBJK23

DLI

❺

DBJK21 V3
DBJK22 V2
DBJK23 V1

❸

Active

DBJK21 V3
DBJK22 V3
DBJK23 V3

Database Versioning enabled -> data returned to app at V1, V2 & V3 levels

❻

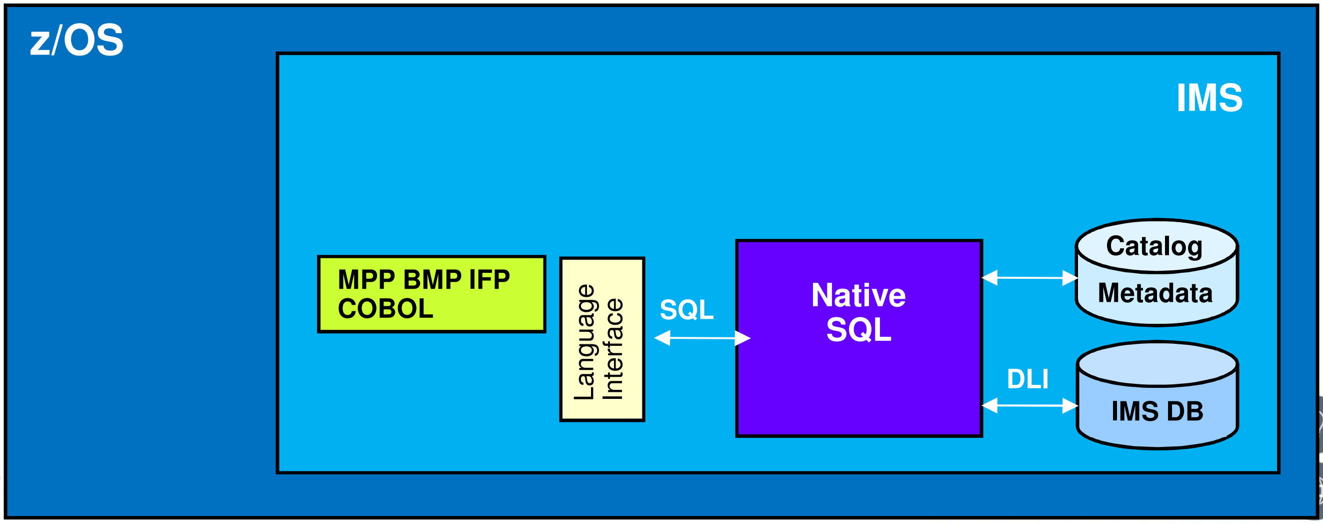
DBJK21
DBJK22
DBJK23

Complete your session evaluations online at www.SH

IMS Native SQL Support for COBOL

IMS 13 SQL Support

- Native SQL COBOL
- Provides standard SQL keywords to easily access IMS data
  - ✓ SELECT, INSERT, UPDATE, DELETE
  - ✓ Uses Dynamic SQL programming model
  - ✓ Converts SQL statements to DL/I calls
  - ✓ Supports a subset of SQL keywords that are currently supported by IMS Universal JDBC driver
- Uses database metadata in IMS Catalog
  - ✓ No need to generate metadata for use in applications



IMS 13 SQL support for COBOL Solution Highlights

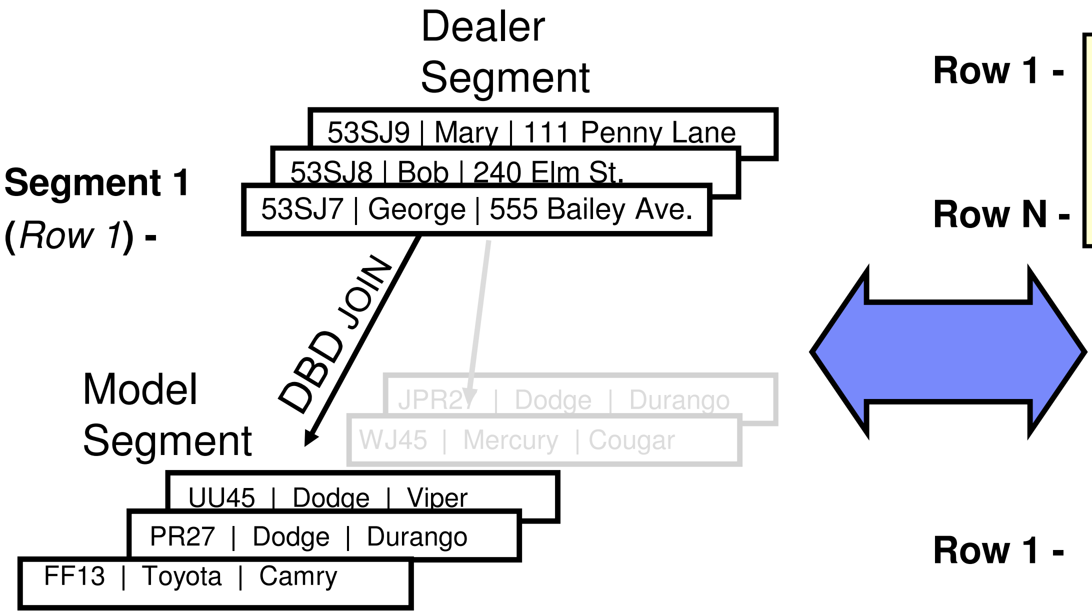
- SQL support for COBOL
  - Use Dynamic SQL as a query language for COBOL programs to access IMS database
  - EXEC SQLIMS is the interface to execute IMS SQL calls

- Native SQL in IMS
  - Process SQL calls natively by the IMS subsystem
  - Still perform DL/I database call processing to IMS DB
  - Provide a consolidated way for SQL processing
  - Uses database metadata in IMS Catalog

- Support IMS TM/DB (MPP, IFP, BMP) and DBCTL BMP

Hierarchical to Relational Terminology Mapping

Hierarchical Design



Relational Design

Dealer Table

		DealerID	DealerName	DealerAddress
Row 1 -	**0**	53SJ7	George	555 Bailey Ave.
	1	53SJ8	Bob	240 Elm St.
	2	53SJ9	Mary	111 Penny Ln.
Row N -	...	...	...	...

Relational JOIN

Model Table

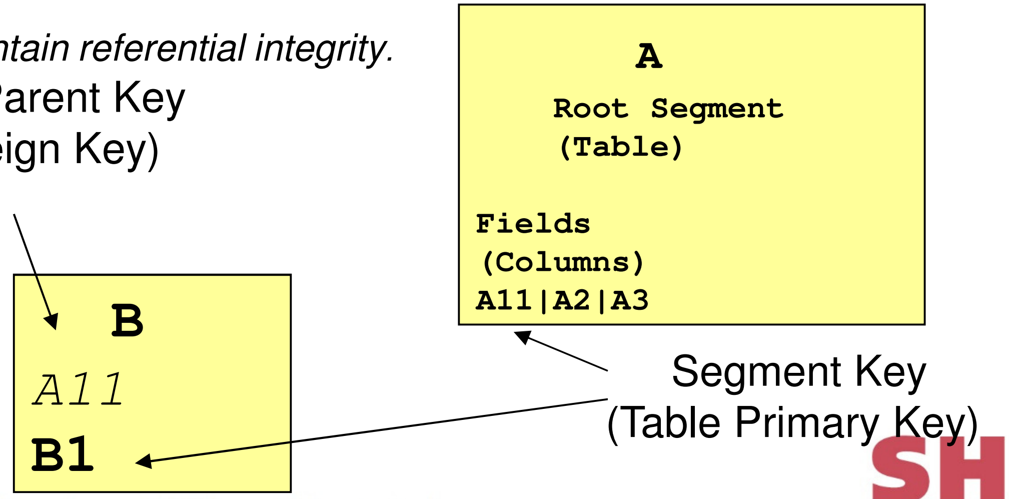
	ID	Make	Model	Dealer	
Row 1 -	UU45	Dodge	Viper	53SJ7	**0**
	PR27	Dodge	Durango	53SJ7	**0**
	FF13	Toyota	Camry	53SJ7	**0**
	JR27	Dodge	Durango	53SJ8	1
	WJ45	Mercury	Cougar	53SJ8	1
Row N -	...	...	...	...	...

Note: **Segment Names ~ Table Names**
Segment Instances ~ Table Rows
Segment Field Names ~ Column Names
Segment unique key ~ Table primary key
IMS foreign key field ~ Table foreign key
PCB ~ Schema

SHARE in Orlando 2015

Solution highlights - IMS foreign keys Referential constraint

- IMS cannot insert a dependent segment unless the parent segment exists
  - IMS has built-in foreign keys in each segment which are comprised of keys of each parent segment
    - Exist in the key feedback area not physically stored in the IMS database
      - For INSERT operations the Foreign Keys s are used to establish the correct position in the hierarchy
        - Values aren’t actually inserted as they already exist in the database



IMS Foreign Key - *maintain referential integrity.*
(Segment Parent Key
Table Foreign Key)

A
Root Segment
(Table)
Fields
(Columns)
A11|A2|A3

B
A11
B1

Segment Key
(Table Primary Key)

IMS Catalog Metadata and SQL

```
PCB   TYPE=DB,DBDNAME=AUTODBD,PROCOPT=G,KEYLEN=4,PCBNAME=AUTOGPCB
       EXTERNALNAME=DealerDBRead
PCB   TYPE=DB,DBDNAME=AUTODBD,PROCOPT=A,KEYLEN=4,PCBNAME=AUTOAPCB
      EXTERNALNAME=DealerDBUpdate
       PSBGEN PSBNAME=AUTOPSB
```

PSB

```
SELECT * FROM AUTOGPCB.DEALER
SELECT * FROM DealerDBUdate.DealerTable
UPDATE DealerDBUdate.DealerTable SET
DELETE FROM DealerDBUdate.DealerTable
INSERT INTO DealerDBUdate.DealerTable
```

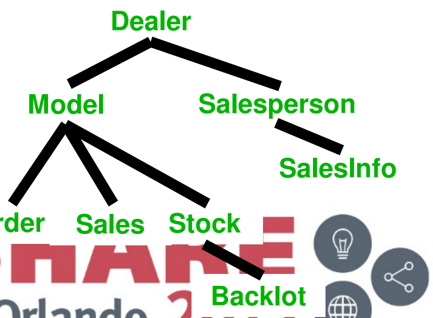
```
01 Dealer_Segment
    02 Dealer_ID  PIC 9(6) COMP.
    02 Dealer_Name PIC X(20).
    02 Dealer_Address PIC X(30).
```

```
DBD NAME=AUTODBD, ACCESS=DEDB,
SEGM NAME=DEALER,PARENT=0,EXTERLNAME=DealerTable
SEGM NAME=MODEL,PARENT=DEALER
SEGM NAME=ORDER,PARENT=MODEL
SEGM NAME=SALES,PARENT=MODEL
SEGM NAME=STOCK,PARENT=MODEL
SEGM NAME=BACKLOT,PARENT=STOCK
. . .
```

COBOL COPYBOOK

```
PSB    AUTOPSB
PCB    AUTOGPCB    EXTERNALNAME DealerDBRead
PCB    AUTOAPCB    EXTERNALNAME DealerDBUpdate
DBD    AUTODBD
SEGM DEALER         EXTERNALNAME  DealerTable
```

DBD



IMS Catalog Metadata

Solution Details – Key application elements

- Delimit SQL statement using EXEC SQLIMS ... END-EXEC
- Dynamic SQL programming model
  - Must call PREPARE to process SQL statement
- Host variables
  - Use for both send and receive data processed by IMS
- SQL communication area (SQLIMSCA)
  - Structure used by IMS to provide status feedback
  - SQLIMSCODE (error code), SQLIMSSTATE (state), SQLIMSERRM (error message)
- SQL description area (SQLIMSDA)
  - DESCRIBE statement IMS provides information to an application program about a prepared statement
  - FETCH statement application program describes a host variable or buffer that is to be used to contain an output value from a row of the result.

Handling errors

- SQL communication area (SQLIMSCA)
  - Structure used by IMS to provide status feedback
  - The SQL INCLUDE statement is used in the COBOL application to provide the declaration of the SQLIMSCA

```
EXEC SQLIMS INCLUDE SQLIMSCA
```

- The main elements in the SQLIMSCA are:
  - SQLIMSCODE – A return code represents a successful or failed SQL operation
    - Example -8004
  - SQLIMSSTATE – Common codes for error conditions which conform to the SQL standard
    - Example 58030
  - SQLIMSERRM – Error message text
    - Example IMS returned the PCB status code

SQL descriptor area (SQLIMSDA)

- SQLIMSDA stores information about prepared SQL statements or host variables.
  - SQLIMSDA header
  - SQLIMSVAR entry
    - each column or host variable is described

```
EXEC SQLIMS INCLUDE SQLIMSDA
```

- Can be read by IMS or the application program
  - Read by application program after a DESCRIBE statement
  - Read by IMS for the host variables set by the application program

IMS Native SQL Support for COBOL solution

- Compile IMS program using COBOL compiler with the SQL(IMS) option
  - Create an executable program to be run in IMS.
  - IMS co-processor knows when a particular SQL statement begins and ends by the following delimits for SQL statements:

```
– EXEC SQLIMS
    SQL-STATEMENT
– END-EXEC.
```

  - Translate SQL statement to a COBOL CALL statement

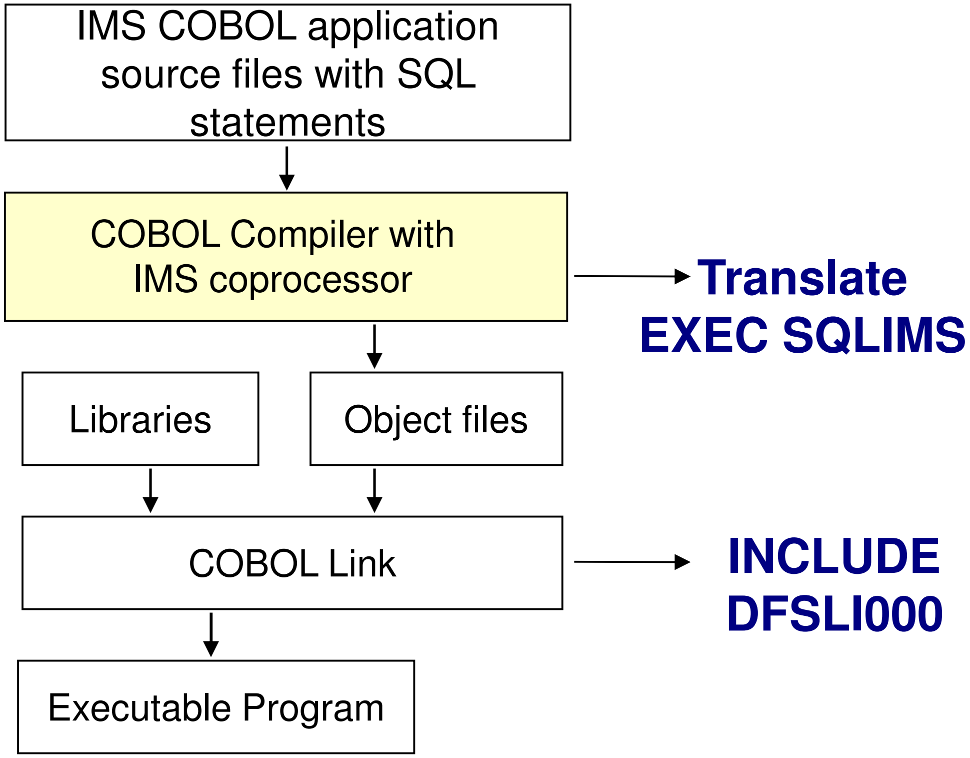
```
*EXEC SQLIMS FETCH . . .
CALL SQLTDLI USING SQL-PARMLIST
```

- SQLTDLI
  - non-language-specific interface added to DFSLI000

IMS coprocessor

- Compile IMS SQL COBOL application with IMS coprocessor
- Pre-process EXEC SQLIMS statements in COBOL source
- Integrated with Enterprise COBOL V5.1
- Specify ‘SQLIMS’ compiler option to compile COBOL program with IMS SQL calls

IMS COBOL SQL application compiled and linked



IMS COBOL application source files with SQL statements

COBOL Compiler with IMS coprocessor

Translate EXEC SQLIMS

Libraries

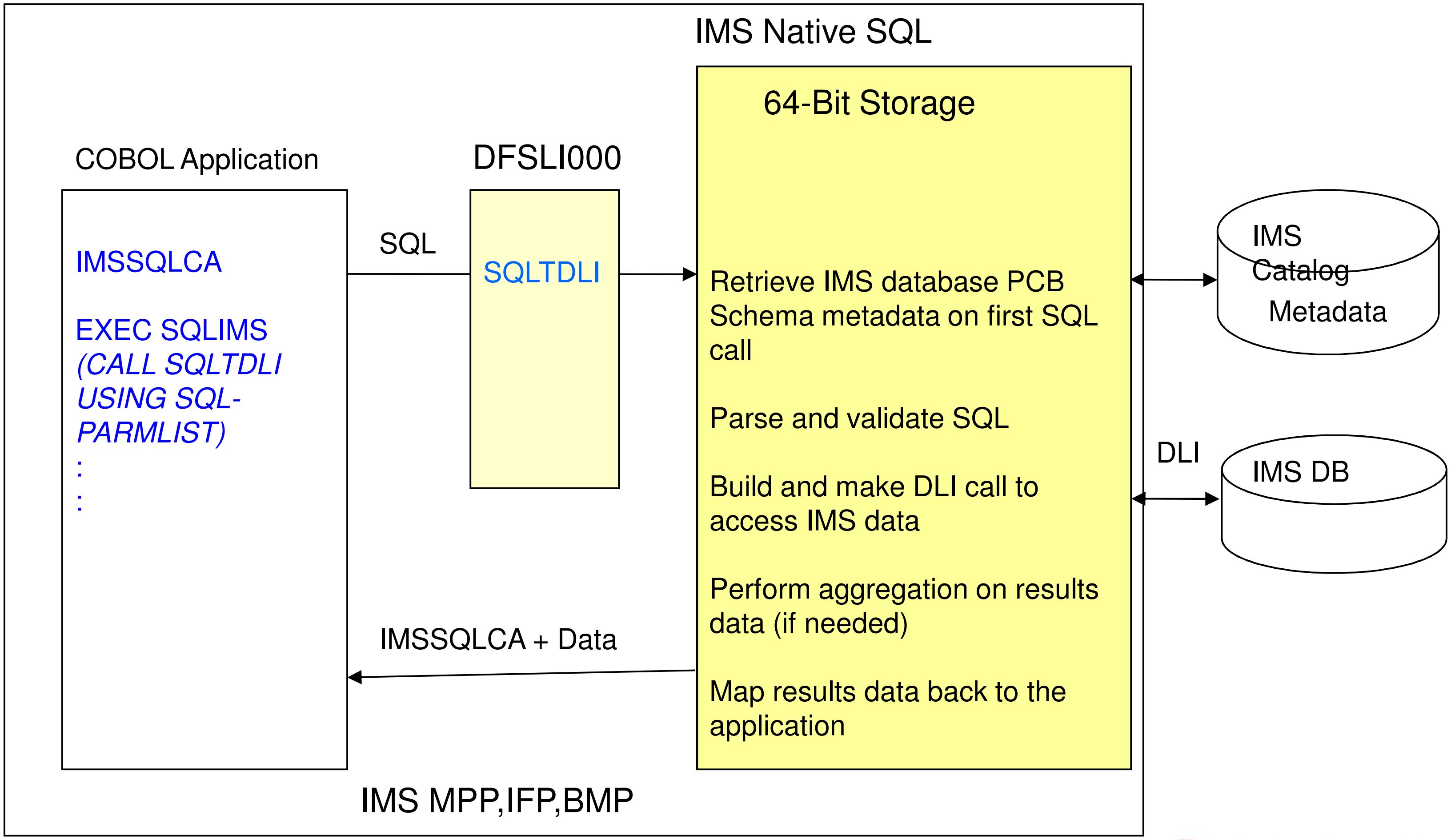
Object files

COBOL Link

INCLUDE DFSLI000

Executable Program

IMS SQL Call Request Handler



SQL considerations and restrictions for COBOL

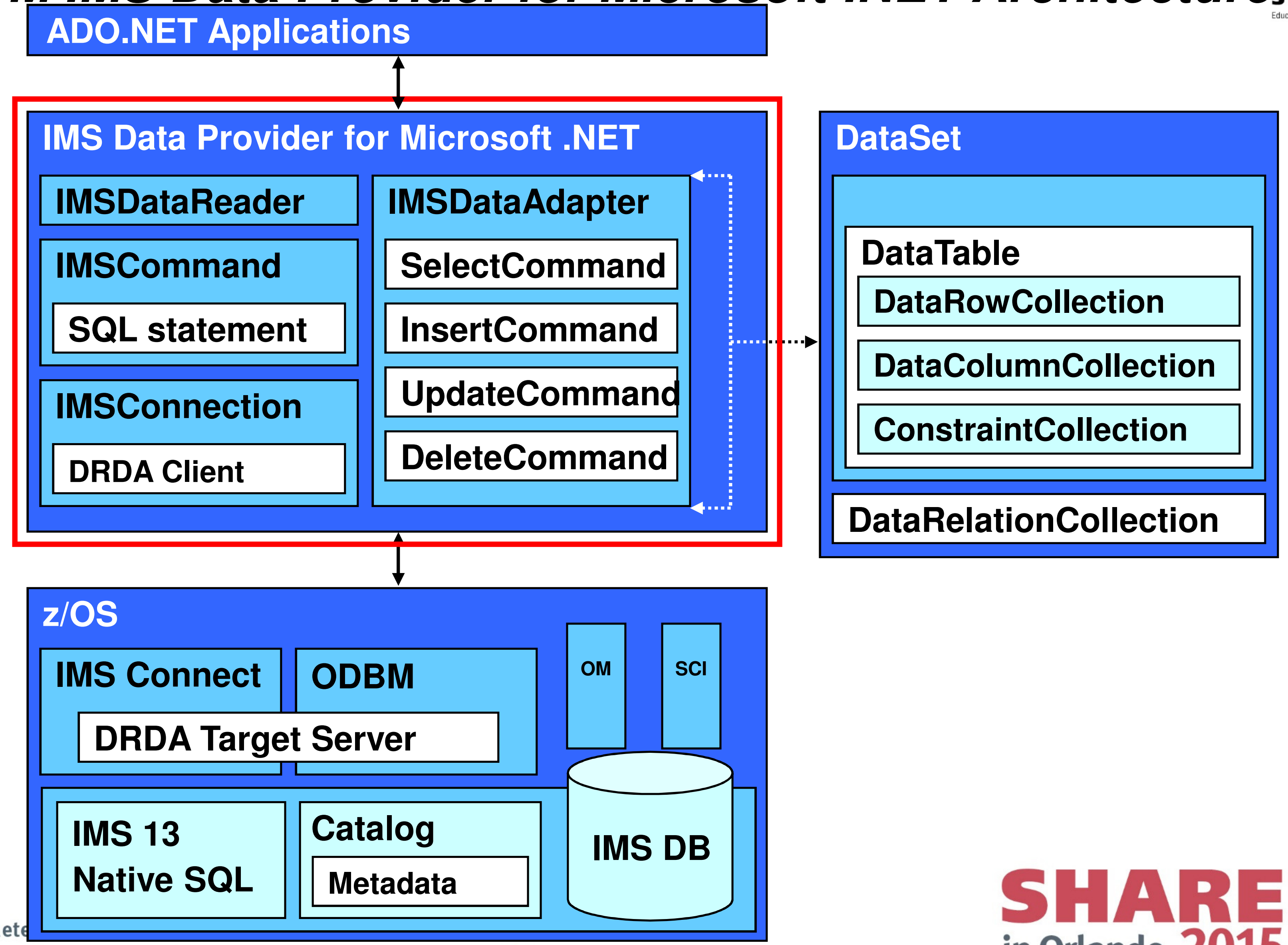
- A subset of SQL keywords is supported.
  - Aggregate functions and XML are not supported by COBOL SQL in SELECT statements.
  - SQL COMMIT and ROLLBACK keywords are not supported.
    - use IMS DB system services call to commit or roll back your database changes
- Batch and DB Batch are not supported.
- IBM® CICS® Transaction Server for z/OS® and DB2® for z/OS stored procedures to IMS are not supported..
- The IMS catalog must be enabled to use SQL support for COBOL..
- Specify at least 12M for your IMS dependent region size for running a COBOL SQL application.
- Only one cursor and SQL statement can be active at a time in the application.
- For IMS database services, GSAM, IMS TM, and message processing services, continue to use DL/I API.
- Dynamic SQL statement is supported. Static SQL is not supported
- Only EBCDIC CCSID 37 and 1140 codepages for the COBOL CODEPAGE option are supported.
- Note The IMS Universal Database resource adapter and IMS Universal JDBC driver internally manage the LL field on behalf of the application
  - For SQL support for COBOL, COBOL applications are responsible for managing the LL field

Performance

- Recommendations
  - Fully qualify all tables (segments) and columns (fields) in SQL statements
    - Specify the schema (PCB) name
  - Always use PREPARE call for SQL statement that is going to be executed multiple times
  - Consider using FETCH or cursors to select a set of rows and then process the set either one row at a time or one rowset at a time

IMS Enterprise Suite V3.1
IBM IMS Data Provider for Microsoft .NET

IBM IMS Data Provider for Microsoft .NET Architecture



IMS Universal Driver Enhancements

ESAF support in Java Dependent Regions (JDR)

- With IMS 13, the ESAF interface can be used in JMP/JBP regions to access any ESAF defined to the IMS control region
    - WebSphere MQ, DB2, WOLA (WebSphere Optimized Local Adapter)
- Support for the SSM= parameter on the JMP/JBP dependent region startup JCL
- Only one ESS connection method allowed per JMP/JBP
    - Default ESS connection method is DB2 RRSAF
        - No impact to existing users
    - Need to specify ESAF as the connection method by specifying SSM= in the JMP/JBP dependent region JCL
- Provides support for all types of ESAF interfaces
- WebSphere MQ and WOLA can now be accessed via JMP/JBP regions

IMS 12 APAR PI30300 ***: All users of the IMS Universal Drivers and*** CSL ODBM Input user exit

- service allows users of the CSL ODBM Input user exit the capability to alter the PSB and/or alias names before an APSB call
  - The IMS Universal Drivers code has been modified to support a new DRDA codepoint, sent via the ACCRDBRM response, that allows ODBM to change the PSB and/or alias name via the ODBM Input user exit.
  - The IMS Universal Drivers will receive any altered PSB and/or alias names via the ACCRDBRM response from ODBM.