# Scaling Your Applications with the IMS Catalog – C06/C15

Richard Tran r
IMS Open Database Development Lead
2015-03-17

IMS Technical Symposium 2015

# Please Note

## Agenda

**The IMS catalog**

Using the IMS catalog within the cloud

IMS Explorer Demo

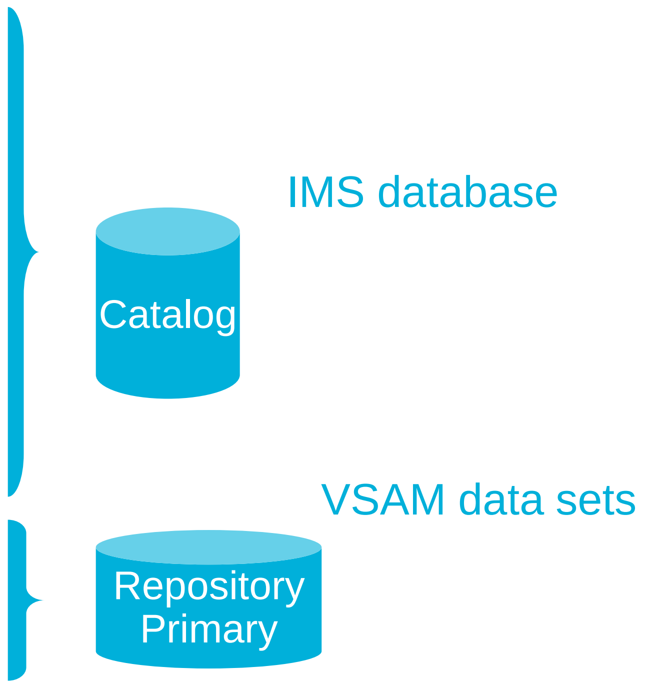
## What is the IMS catalog?

## IMS Catalog features

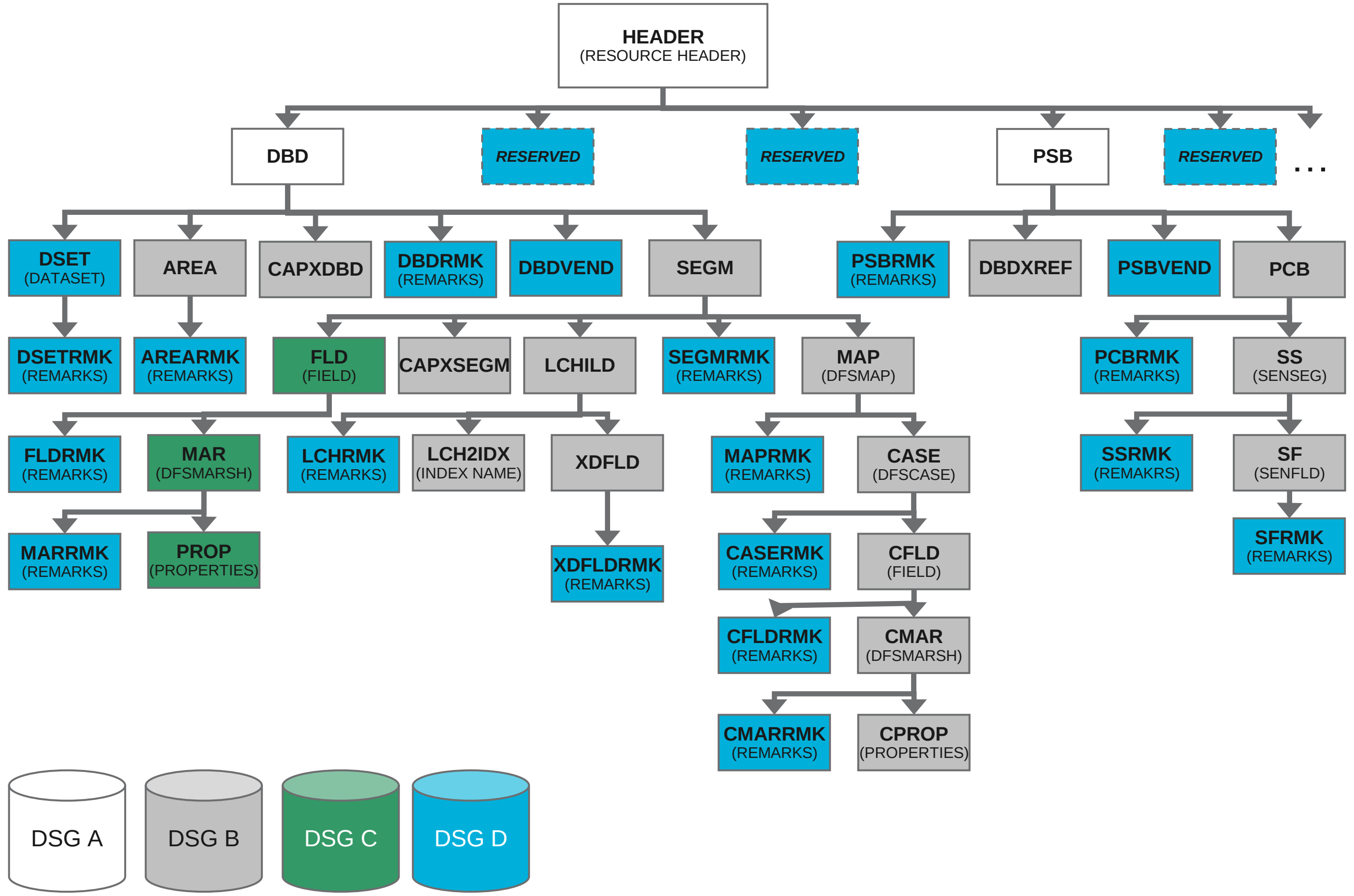
- Provides a trusted source of both database and application metadata
- Enables better integration for both mobile and cloud workloads
- Allows for better workload scalability

# Types of technical metadata and storage medium

- DB
  - PSB/DBD resources
    - Database structure definitions
    - Physical database definitions
    - Segment definitions
    - Field definitions
  - Application
    - Data types
    - Application defined fields
    - Encodings
    - Redefines
    - User defined types
    - Structures
- TM
  - MODBLKS resources
    - Program definitions
    - Transaction definitions

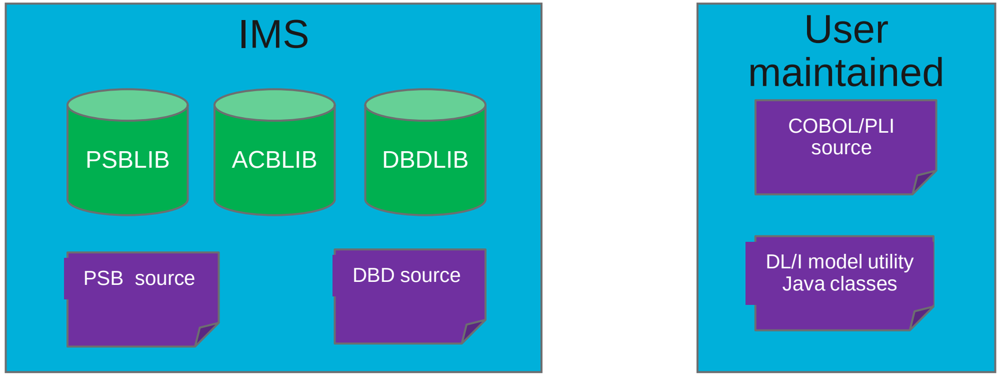


# Physical catalog structure



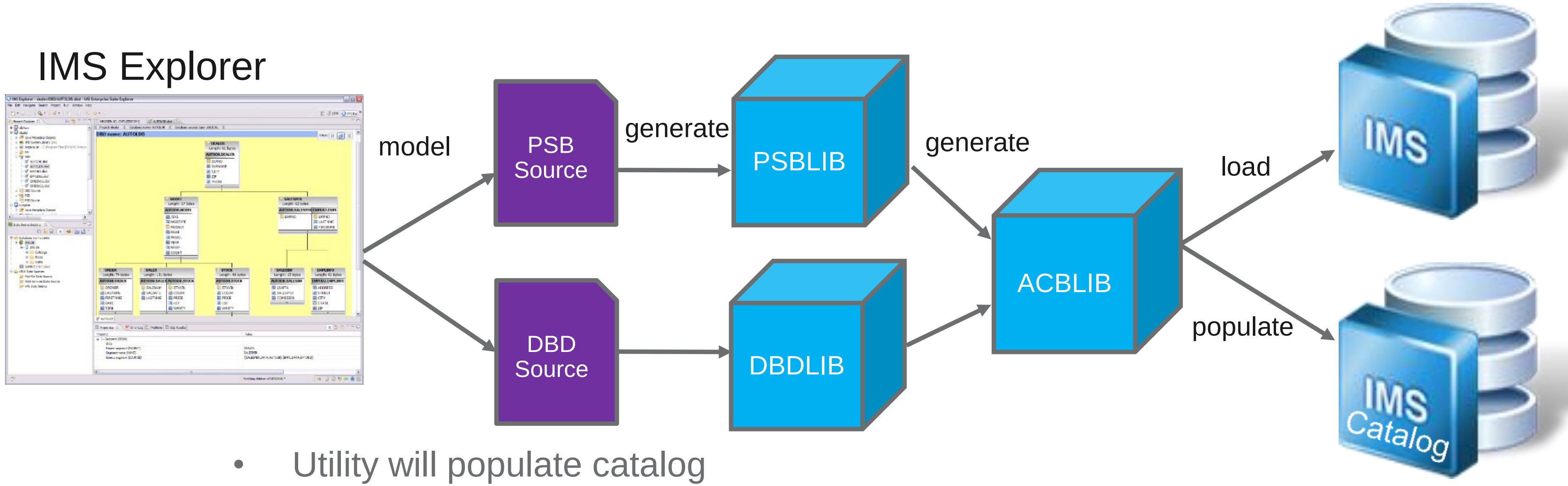
# Before the IMS Catalog

- Databases partially defined in DBD
  - Only searchable fields needed by applications
  - Remaining segment data is not defined
- Remaining database definition in applications
  - COBOL copybook maps segment data
  - Applications can have different segment mappings



# IMS Catalog life cycle

IMS Explorer



- Utility will populate catalog
- ACBGEN will populate ACBLIB and catalog
  - Populate ACBLIB with *standard* ACB info and extended info
  - Populate catalog with *extended* info
- Key points
  - Only way to update catalog is via the ACBGEN process
  - Extended info stored in ACBLIB members for recoverability
  - Extended info is acquired via the IMS Explorer

# Application metadata pre-catalog

## DBD

```
SEGM  NAME=HOSPITAL,
      PARENT=0,
      BYTES=900,
      RULES=(,HERE),
      SSPTR=0,
      ENCODING=Cp1047
FIELD NAME=(HOSPCODE,SEQ,U),
      BYTES=12,
      START=3,
      TYPE=C,
FIELD NAME=(HOSPLL),
      BYTES=2,
      START=1,
      TYPE=X,
FIELD NAME=(HOSPNAME),
      BYTES=17,
      START=15,
      TYPE=C,
```

## Copybook

```
01 HOSPITAL.
  05 HOSPLL             PICTURE S9(3) COMP.
  05 HOSPITAL_CODE      PICTURE X(12).
  05 HOSPITAL_NAME      PICTURE X(17).
```

## Java Metadata file

```
private DLITypeInfo[] PCB01HOSPITALArray() {
 DLITypeInfo[] PCB01HOSPITALArray=  {
   new DLITypeInfo("HOSPLL", DLITypeInfo.SMALLINT, 1, 2,
     "HOSPLL", DBType.DEDB, false),
   new DLITypeInfo("HOSPITAL_CODE", DLITypeInfo.CHAR, 3, 12,
     "HOSPCODE", DLITypeInfo.UNIQUE_KEY,
      DBType.DEDB, true),
   new DLITypeInfo("HOSPITAL_NAME", DLITypeInfo.CHAR, 15, 17,
      "HOSPNAME", DBType.DEDB, false)
  };
  return PCB01HOSPITALArray;
}
```

# Application metadata with new macro definition

DBD

```
SEGM  NAME=HOSPITAL,
      PARENT=0,
      BYTES=900,
      RULES=(,HERE),
      SSPTR=0,
      ENCODING=Cp1047
FIELD NAME=(HOSPCODE,SEQ,U),
      BYTES=12,
      START=3,
      TYPE=C,
FIELD NAME=(HOSPLL),
      BYTES=2,
      START=1,
      TYPE=X,
FIELD NAME=(HOSPNAME),
      BYTES=17,
      START=15,
      TYPE=C,
```

Copybook

```
01 HOSPITAL.
   05 HOSPLL              PICTURE S9(3) COMP.
   05 HOSPITAL_CODE       PICTURE X(12).
   05 HOSPITAL_NAME       PICTURE X(17).
```

DBD++

```
SEGM  NAME=HOSPITAL,
   EXTERNALNAME=HOSPITAL,
   PARENT=0,
   BYTES=900,
   RULES=(,HERE),
   SSPTR=0,
   ENCODING=Cp1047
FIELD NAME=(HOSPCODE,SEQ,U),
   EXTERNALNAME=HOSPITAL_CODE,
   BYTES=12,
   START=3,
   TYPE=C,
   DATATYPE=CHAR
DFSMARSH ENCODING=Cp1047,
   INTERNALTYPECONVERTER=CHAR
FIELD NAME=(HOSPLL),
   EXTERNALNAME=HOSPLL,
   BYTES=2,
   START=1,
   TYPE=X,
   DATATYPE=SHORT
DFSMARSH ,
   INTERNALTYPECONVERTER=SHORT
FIELD NAME=(HOSPNAME),
   EXTERNALNAME=HOSPITAL_NAME,
   BYTES=17,
   START=15,
   TYPE=C,
   DATATYPE=CHAR
DFSMARSH ENCODING=Cp1047,
   INTERNALTYPECONVERTER=CHAR
```

# Application metadata with catalog

## DBD++

```
SEGM  NAME=HOSPITAL,
      EXTERNALNAME=HOSPITAL,
      PARENT=0,
      BYTES=900,
      RULES=(,HERE),
      SSPTR=0,
      ENCODING=Cp1047
FIELD NAME=(HOSPCODE,SEQ,U),
      EXTERNALNAME=HOSPCODE,
      BYTES=12,
      START=3,
      TYPE=C,
      DATATYPE=CHAR
DFSMARSH ENCODING=Cp1047,
      INTERNALTYPECONVERTER=CHAR
FIELD NAME=(HOSPLL),
      EXTERNALNAME=HOSPLL,
      BYTES=2,
      START=1,
      TYPE=X,
      DATATYPE=SHORT
DFSMARSH ,
      INTERNALTYPECONVERTER=SHORT
FIELD NAME=(HOSPNAME),
      EXTERNALNAME=HOSPNAME,
      BYTES=17,
      START=15,
      TYPE=C,
      DATATYPE=CHAR
DFSMARSH ENCODING=Cp1047,
      INTERNALTYPECONVERTER=CHAR
```

## Catalog XML – GUR DL/I

```
<segment imsName="HOSPITAL" name="HOSPITAL" encoding="Cp1047">
  <dedb>
    <bytes maxBytes="900" />
  </dedb>
  <field imsDatatype="C" imsName="HOSPCODE" name="HOSPITAL_CODE" seqType="U">
    <startPos>3</startPos>
    <bytes>12</bytes>
    <marshaller encoding="Cp1047">
      <typeConverter>CHAR</typeConverter>
    </marshaller>
    <applicationDatatype datatype="CHAR" />
  </field>
  <field imsDatatype="C" imsName="HOSPITAL_NAME" name="HOSPNAME">
    <startPos>15</startPos>
    <bytes>17</bytes>
    <marshaller encoding="Cp1047">
      <typeConverter>CHAR</typeConverter>
    </marshaller>
    <applicationDatatype datatype="CHAR" />
  </field>
  <field imsDatatype="X" imsName="HOSPLL" name="HOSPLL">
    <startPos>1</startPos>
    <bytes>2</bytes>
    <marshaller>
      <typeConverter>SHORT</typeConverter>
    </marshaller>
    <applicationDatatype datatype="SHORT" />
  </field>
</segment>
```

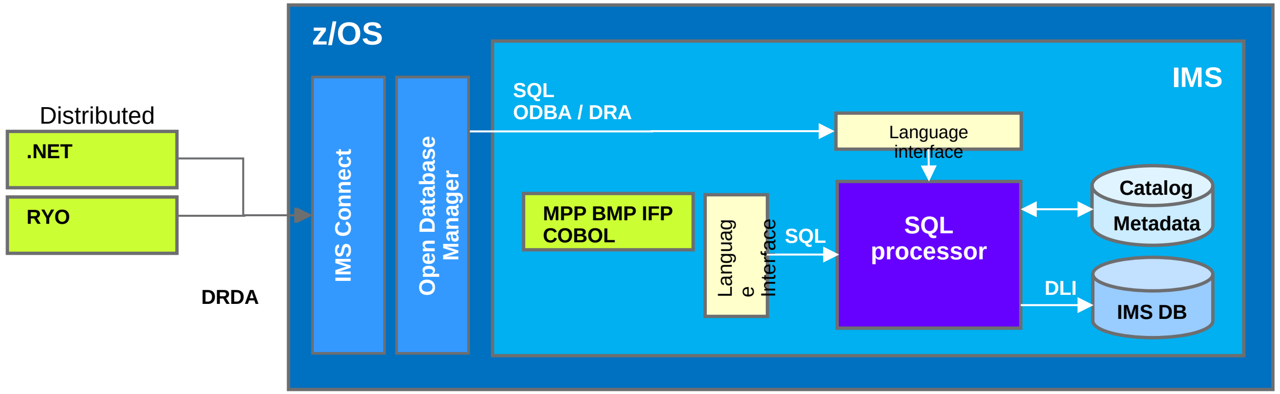
## IMS features leveraging the IMS catalog

## New IMS V13 features based on the IMS catalog

- Native SQL support
  - .NET Data Provider
- Database Versioning

# COBOL and .NET access through SQL

- SQL support for COBOL directly access IMS Catalog for database metadata
  - No need to generate metadata for use in applications
  - No need to reference copybooks for metadata
- Consolidated SQL processor for both host (COBOL) and distributed applications (.NET/RYO)



# 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 Overview (cont’d)

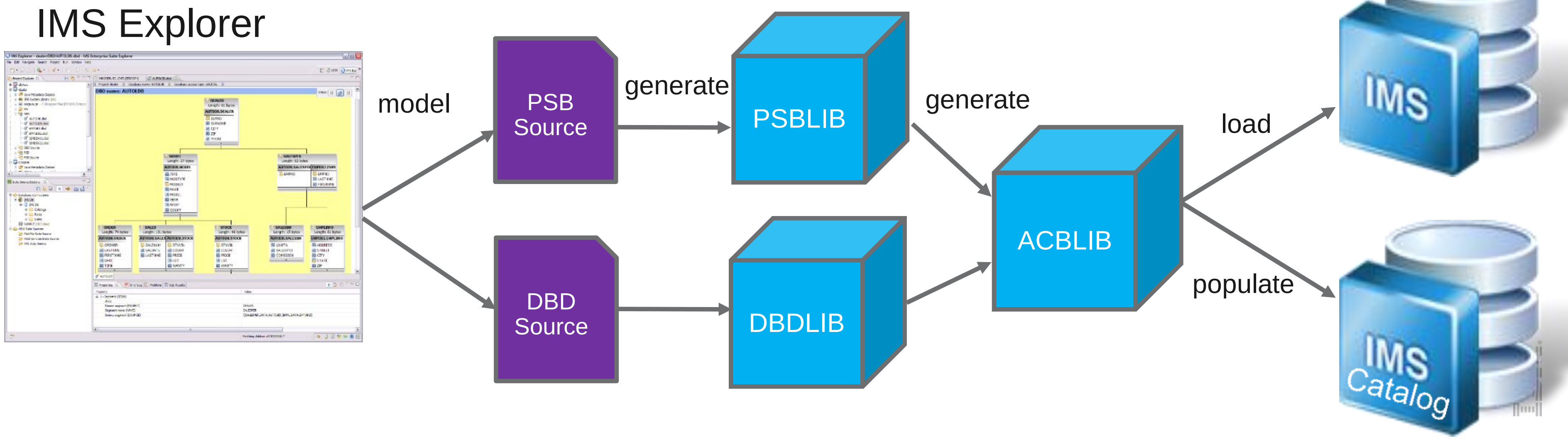
- Database Versioning requires enablement of the **IMS catalog**
  - DBD definitions for versioned databases must be in the **IMS catalog**
- Database Versioning must be enabled
- Versioning is at the DBD level
  - Users define the version of a database definition on the DBD
  - Version numbers must be maintained in incremented values
- Application programs select the desired database version by
  - Specifying the version number on the PCB of the PSB
  - Specifying the version number on a DL/I INIT call

## Planned IMS V14 feature based on the IMS catalog

- Dynamic Database support
  - Using the Data Definition Language (DDL)
- IMS Managed ACBLIB
- Read more about V14:
  - http://www-01.ibm.com/software/data/ims/v14/

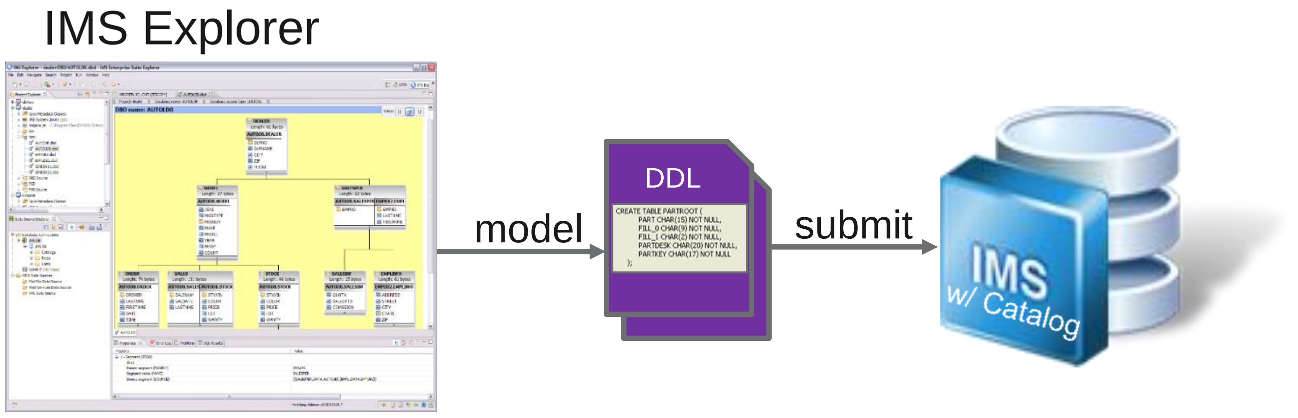
# Current IMS catalog environment

- IMS loads from the ACBLIB
- The IMS catalog is populated from the ACBLIB
- Typically requires both a DBA and a SYSPROG to model and build database resources



# Future IMS catalog environment

- Databases can be dynamically defined to the IMS catalog through the DDL standard similar to other relational databases
- IMS will now load directly off of the IMS catalog
  - No longer requiring ACBLIB or any of the gen process
- Database changes can be initiated from a DBA,  sysprog, or an application developer depending on permissions



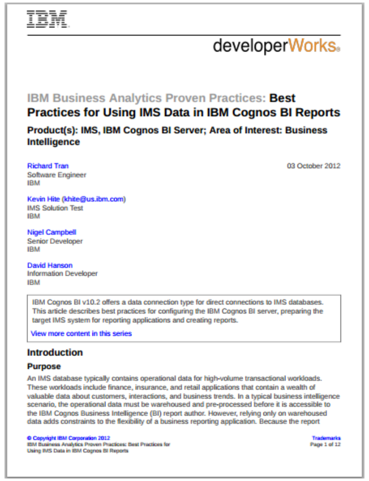
```
CREATE TABLE PARTROOT (
        PART CHAR(15) NOT NULL,
        FILL_0 CHAR(9) NOT NULL,
        FILL_1 CHAR(2) NOT NULL,
        PARTDESK CHAR(20) NOT NULL,
        PARTKEY CHAR(17) NOT NULL
   );
```

## IMS catalog integration capabilities

# Portfolio integration

| Use case | Solution |
|---|---|
| ▪ BI, dashboarding, reporting of IMS data | ▪ QMF<br>▪ Cognos 10.2 BI |
| ▪ Merge HDFS data with trusted OLTP<br>▪ IT analytics (log data) | ▪ IBM InfoSphere BigInsights |
| ▪ Bring analytics to the data | ▪ IBM DB2 Analytics Accelerator |
| ▪ Visualize entire big data landscape | ▪ IBM Watson Explorer |

# Cognos 10.2 BI with IMS Data



IBM

developerWorks

IBM Business Analytics Proven Practices: **Best Practices for Using IMS Data in IBM Cognos BI Reports**

**Product(s): IMS, IBM Cognos BI Server; Area of Interest: Business Intelligence**

Richard Tran — 03 October 2012
Software Engineer
IBM

Kevin Hite (khite@us.ibm.com)
IMS Solution Test
IBM

Nigel Campbell
Senior Developer
IBM

David Hanson
Information Developer
IBM

IBM Cognos BI v10.2 offers a data connection type for direct connections to IMS databases. This article describes best practices for configuring the IBM Cognos BI server, preparing the target IMS system for reporting applications and creating reports.

View more content in this series

**Introduction**

**Purpose**

An IMS database typically contains operational data for high-volume transactional workloads. These workloads include finance, insurance, and retail applications that contain a wealth of valuable data about customers, interactions, and business trends. In a typical business intelligence scenario, the operational data must be warehoused and pre-processed before it is accessible to the IBM Cognos Business Intelligence (BI) report author. However, relying only on warehoused data adds constraints to the flexibility of a business reporting application. Because the report

© Copyright IBM Corporation 2012 — Trademarks
IBM Business Analytics Proven Practices: Best Practices for Using IMS Data in IBM Cognos BI Reports — Page 1 of 12

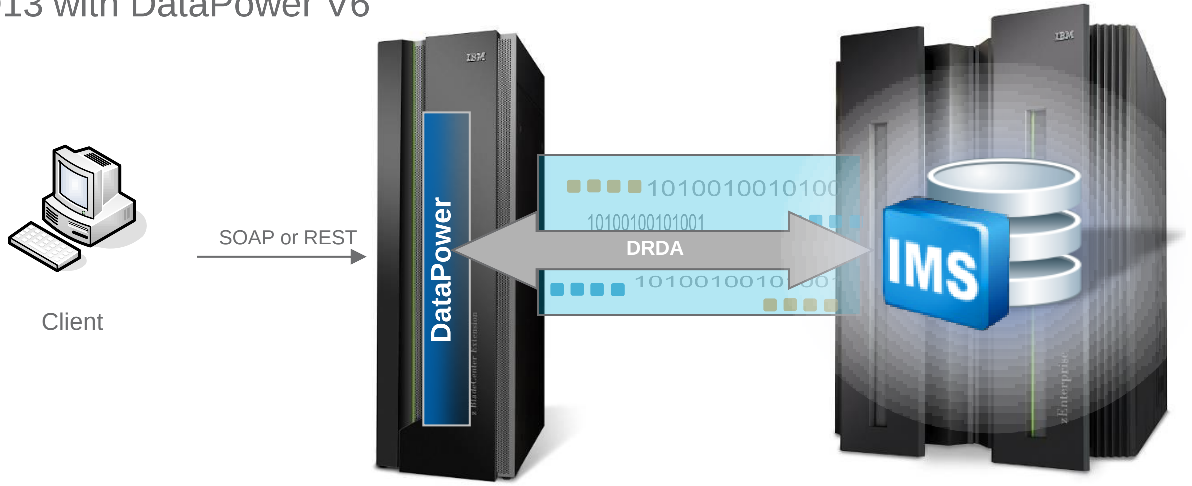
## Get Started Today!

- developerWorks article available here

- Certified against IMS 12 using IMS Open Database technology
  - Universal JDBC driver
- Real-time analytics

# IMS Integration
## “Information as a Service”

- DataPower  provides a standard WS façade to IMS
  - SOAP or REST call is mapped to a JDBC (DRDA) invocation
- Exposes database content (information) *as a service*
- Leverages extensive Web Services security and management capabilities of DataPower to more securely expose critical data to the enterprise
- Available 6/2013 with DataPower V6



# Watson Explorer V10 delivers cognitive exploration

## Watson Explorer

Search, visualize, and explore information from internal and external content through 360-degree information applications

## Watson Content Analytics

**Analyze**, visualize, and discover insight in unstructured data through NLP and text analytics

Now part of Watson Explorer Advanced Edition

## Watson Explorer V10 Now available!

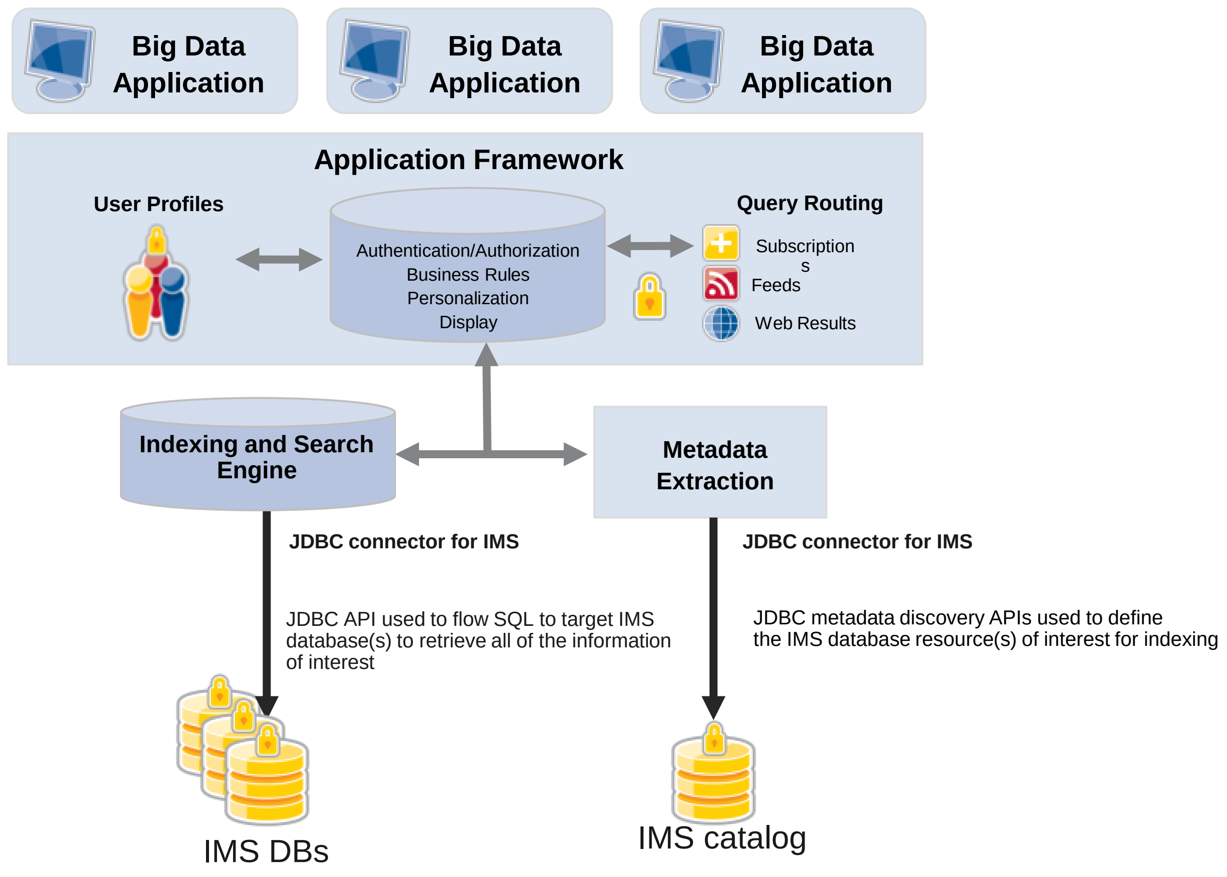
## Watson Developer Cloud

Interpret information to enhance, scale, and accelerate human expertise through cognitive capabilities

## Cognitive Exploration

- ✓ Search and exploration across many different sources
- ✓ Content analytics
- ✓ Cognitive insights
- ✓ Delivered in a 360-degree information application

# Seamless IMS integration in Watson Explorer

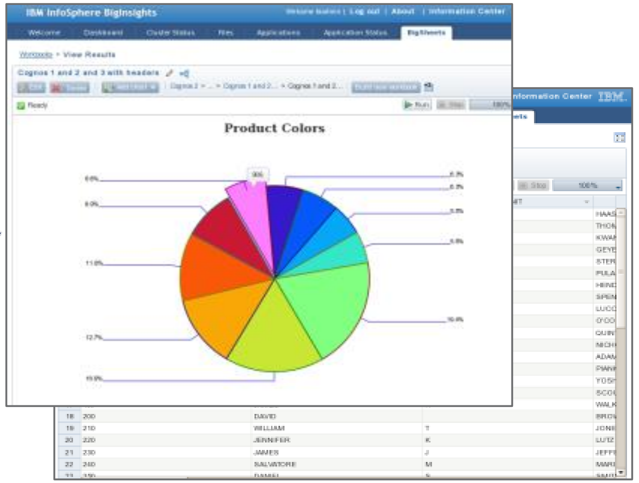


# Enhancing IMS analytics on System z with Big Data

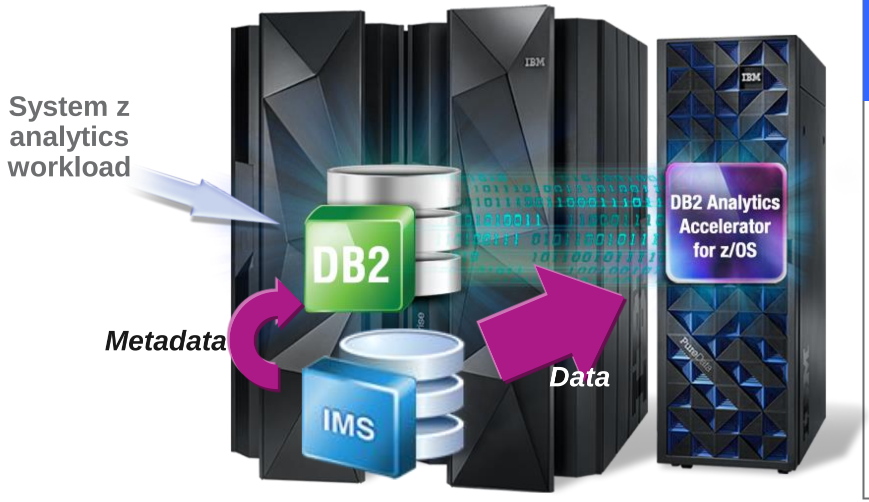
- **Much of the world's operational data resides on z/OS**
- **Unstructured data sources are growing fast**
- **There is a need to <u>merge</u> this data with trusted OLTP data from System z data sources**
- **IMS provides the connectors and the DB capability to allow BigInsights v2.1.2.0 (3/13/2014) to easily and efficiently access the IMS data source**

# Enhancing IMS analytics on System z with Big Data

- **Observation points lead to new business opportunities**
- **Observation points gleaned from both archived data and live data**
- **Score business events, track claims evolution, and more**
- *Make the data available to people who can do something meaningful with it*



# IBM zEnterprise and DB2 Analytics Accelerator


System z analytics workload
DB2
IMS
Metadata
Data
DB2 Analytics Accelerator for z/OS
IBM
PureData

***The hybrid computing platform on zEnterprise***

- *Supports transaction processing and analytics workloads concurrently, efficiently and cost-effectively*
- *Delivers industry leading performance for mixed workloads*

**DB2 Analytics Accelerator and DB2 for z/OS**

A self-managing, hybrid workload-optimized database management system that runs each query workload in the most efficient way, so that each query is executed in its optimal environment for greatest performance and cost efficiency

# IDAA use cases with IMS data

## Make better decisions faster

Large volume reporting of combined IMS and DB2 assets

## Better understand your customers

Leverage full breadth of transactional data for analytics

## Trust your data

Ensure consistency of data relationships between IMS and DB2

IBM Software Group
Technical White Paper

August 2014

**Accelerate business insights with IMS transactional data**

**Introduction**

IBM® System z® servers are home to 80 percent of the world's data,[1] a large part of which is Information Management System (IMS) transactional data. IMS applications and data continue to form the core of the world's critical online transaction processing (OLTP) workloads. IMS data on System z servers is widely viewed as the single version of the truth, from which data stores and warehouses are populated.

The System z platform offers high availability, scalability, security and performance for OLTP workloads. The data that resides on the System z platform and is targeted for analytics on the System z platform has those same qualities of service. Bringing analytics directly to the data, rather than moving data to the analytics, makes sense. It preserves these qualities of service to ensure timely, accurate, and secure insight, and reduces costs of moving data off the platform.

The insights that analytics provide must be delivered efficiently across an organization, with high quality and proper governance. Moving data off of a System z platform and across platforms is not only costly; it literally ages data out of its productive use and impedes the delivery of business insights. Where transactional data originates plays an increasingly important role in analytics. Some percentage of that 80 percent of data is not being analyzed at all due to the performance and cost challenges of moving data.

## Get Started Today!

- Technical Whitepaper and "how-to" guide available here

## Agenda

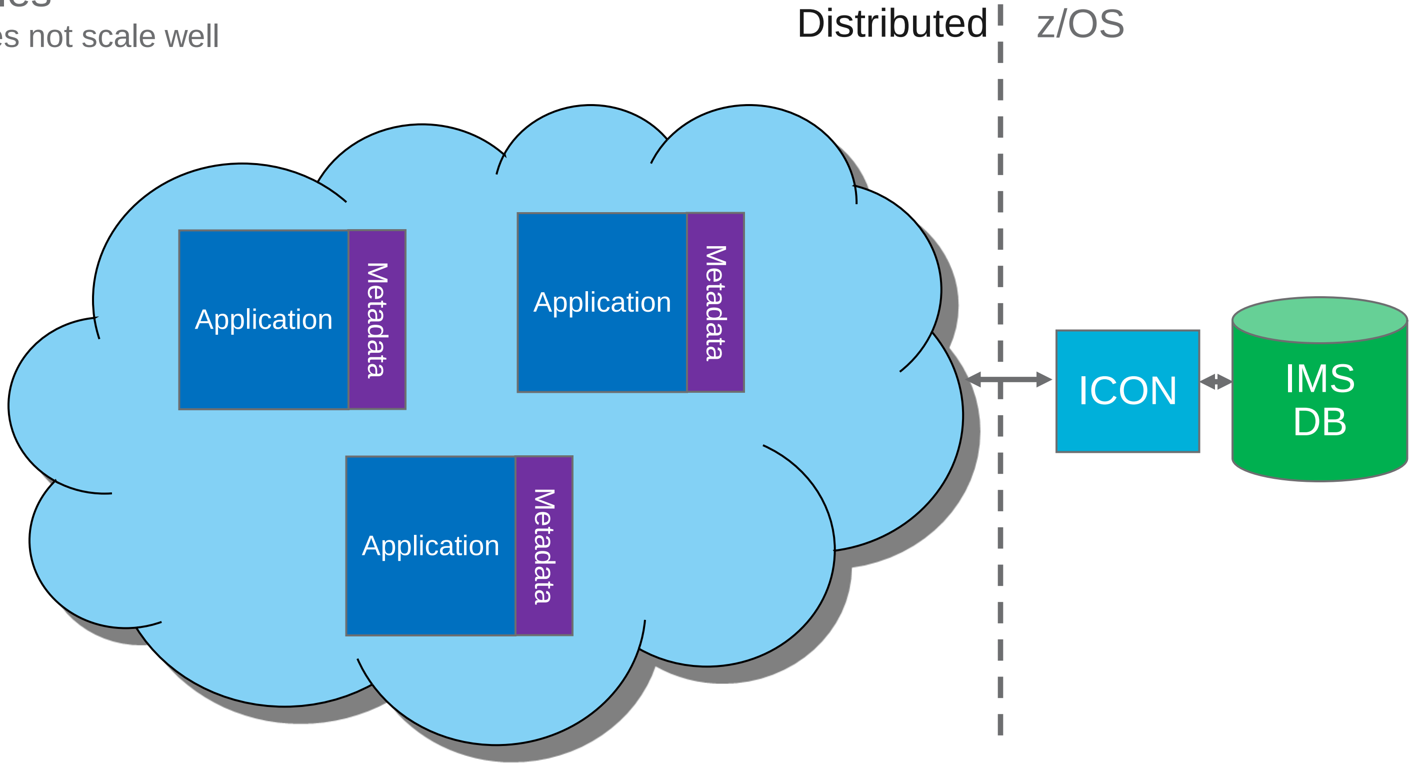
The IMS catalog

**Using the IMS catalog within the cloud**

IMS Explorer Demo

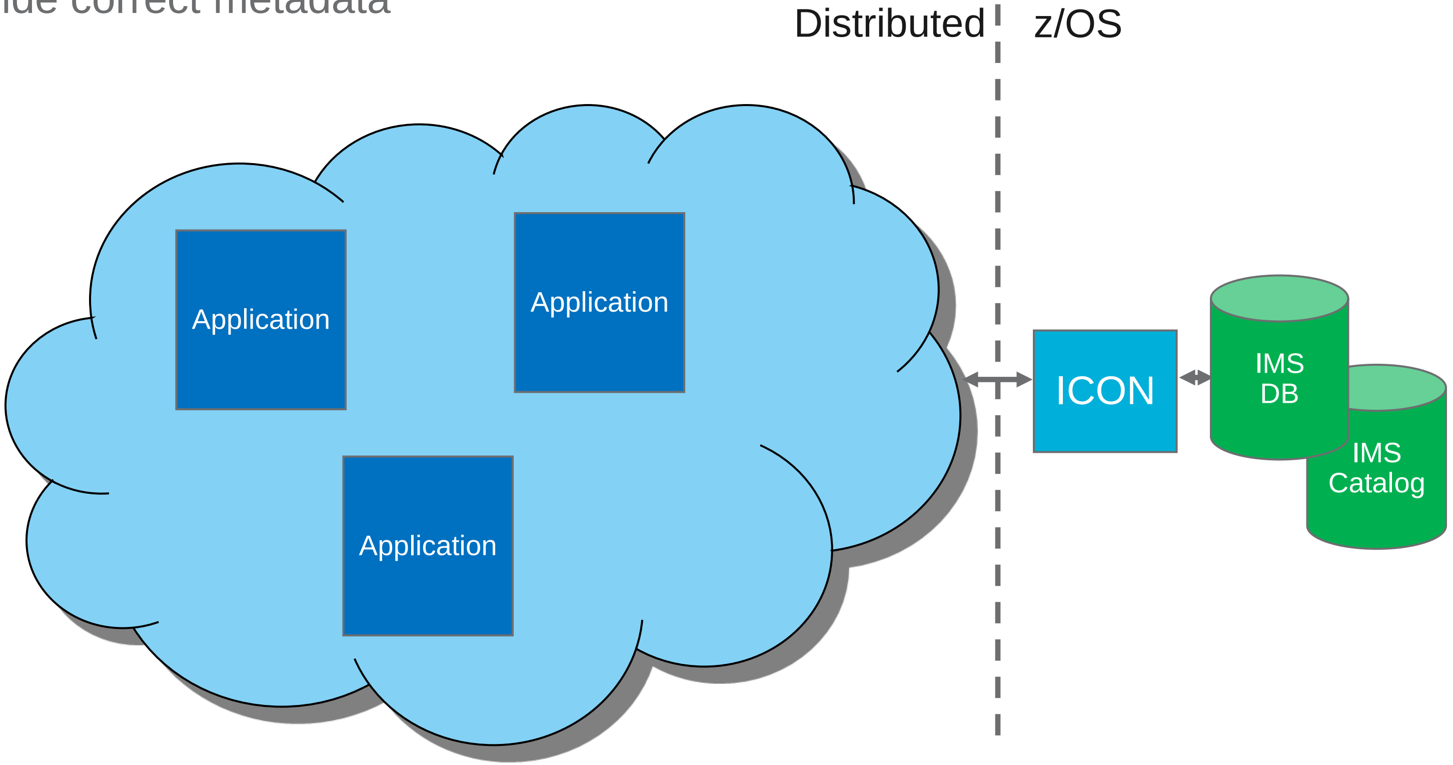
# IMS cloud deployment pre-catalog

- Before the IMS catalog, metadata was contained within localized Java files
  - Does not scale well



# IMS cloud deployment with catalog

- Applications can replicate as often as needed and rely on catalog to provide correct metadata

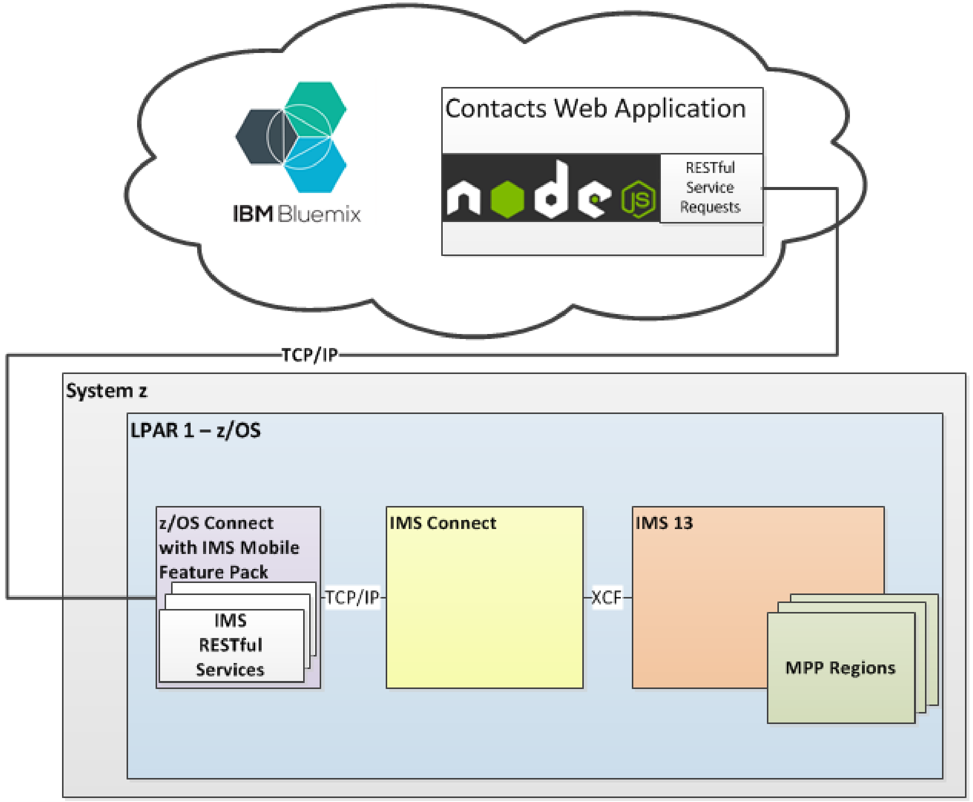


# Intended Support for Database REST Services

- User Story
  - The solution architects have decided to
    - expose an IMS DB query as several RESTful services using z/OS Connect and the IMS Mobile Feature Pack
    - use IMS Explorer for Development (Eclipse tooling) which is required to deploy and test the IMS RESTful services in z/OS Connect
    - use a JavaScript-based web server that will leverage the new Contacts-based RESTful services provided
    - use the Node.js runtime for the JavaScript server
    - host the web application on IBM’s cloud platform, IBM Bluemix

# Intended Support Architecture

- Architectural Diagram



# Agenda

The IMS catalog

Using the IMS catalog within the cloud

**IMS Explorer Demo**

# Additional Resources

- The IMS catalog
  - [http://www.redbooks.ibm.com/abstracts/redp4812.html?Open](http://www.redbooks.ibm.com/abstracts/redp4812.html?Open)
- IMS Native SQL Application Programming Guide
  - [http://www-01.ibm.com/support/knowledgecenter/SSEPH2_13.1.0/com.ibm.ims13.doc.apg/ims_appprog_sql.htm](http://www-01.ibm.com/support/knowledgecenter/SSEPH2_13.1.0/com.ibm.ims13.doc.apg/ims_appprog_sql.htm)
- IMS database versioning
  - [http://www-01.ibm.com/support/knowledgecenter/SSEPH2_13.1.0/com.ibm.ims13.doc.rpg/ims_over13_db_dbver.htm](http://www-01.ibm.com/support/knowledgecenter/SSEPH2_13.1.0/com.ibm.ims13.doc.rpg/ims_over13_db_dbver.htm)
- IMS Explorer for Development
  - [http://www-01.ibm.com/support/knowledgecenter/SS9NWR_3.1.0/com.ibm.ims.explorer31.doc/wb_container_imsexplorer.htm?lang=en](http://www-01.ibm.com/support/knowledgecenter/SS9NWR_3.1.0/com.ibm.ims.explorer31.doc/wb_container_imsexplorer.htm?lang=en)
- IMS and Cognos
  - [http://www.ibm.com/developerworks/library/ba-pp-infrastructure-cognos_specific-page630/](http://www.ibm.com/developerworks/library/ba-pp-infrastructure-cognos_specific-page630/)
- IMS and Big Data
  - Attend Session #6128 – 10/30 10am in Banyan C

## Thank You!

## Demo Backup

# IMS Enterprise Suite Explorer for Development

- Easier visualization and editing of IMS Database and Program Definitions
  - Provide graphical editors to:
    - Display IMS database hierarchical structures
    - Display/create/edit PSBs
    - Add/Edit fields in DBDs
  - Import Cobol CopyBooks and PL/I Structures into database segments
  - Generates DBD and PSB source for Catalog or non-Catalog enabled systems.

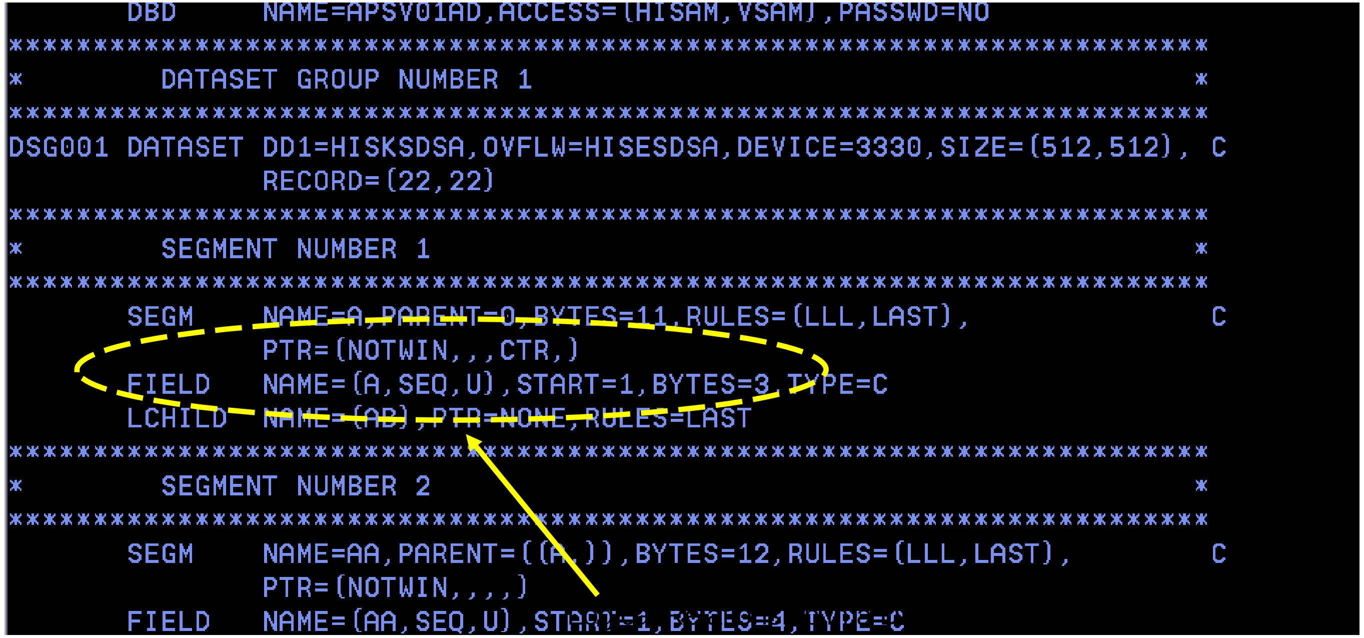
- Ability to easily access live IMS data using SQL statements. A graphical SQL statement builder is provide to help beginners.
  - Leveraging IMS Universal JDBC driver

- Connectivity to the z/OS system
  - Browse a Data Set, submit JCL jobs, and view job output.
  - Import DBD and PSB source files from a Data Set to IMS Explorer, and export generated source back to the host.

# IMS Enterprise Suite Explorer for Development cont’d

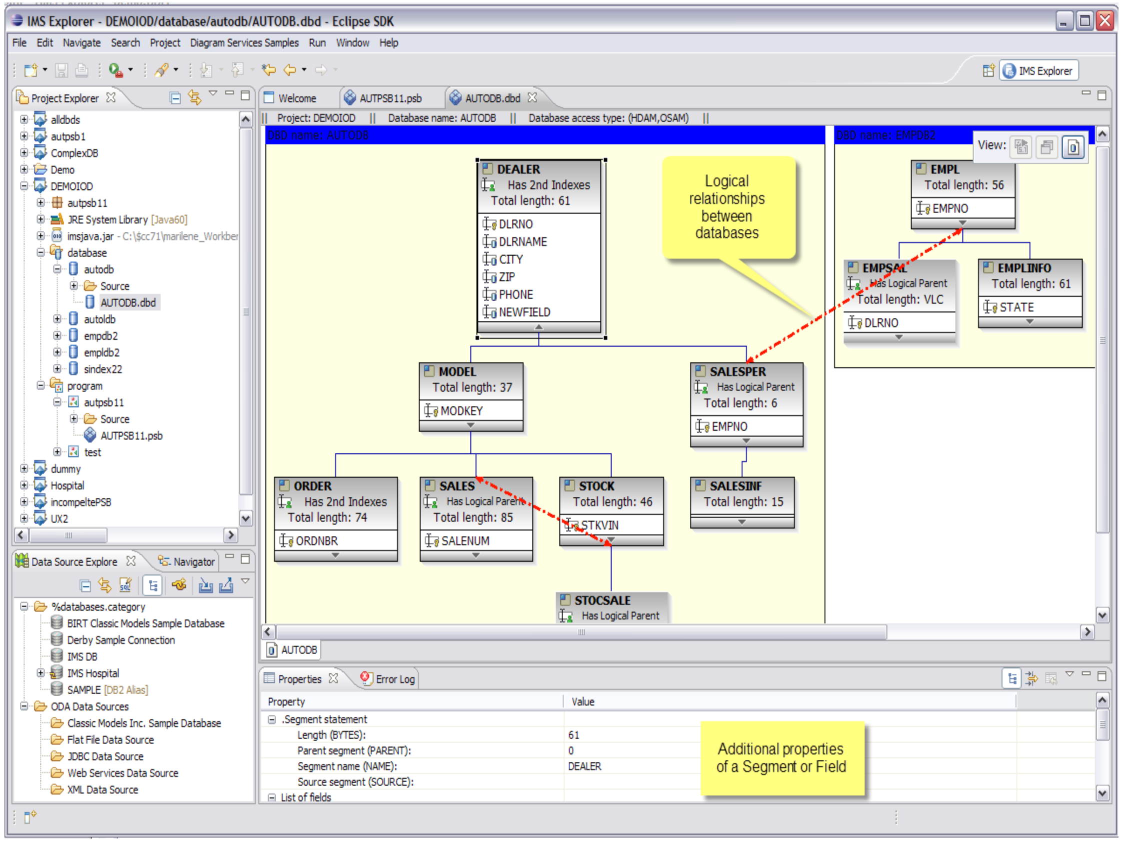
- pureQuery code generation
  - Easily create a services layer separating the application development roles between the database and query tuning specialists and the business logic developers.
- Database web service generation
  - Develop and test an SQL query and then generate a deployment package to expose the query as a web service.
- Catalog Navigation view
  - Navigate through the PSBs/DBDs on a given IMS catalog. Search based on partial resource names or built in queries and then import resources for editing or launch graphical modeling editors in read only mode.
- Transaction unit test support
  - Create a bucket of test cases with a UI enabling you to easily tweak input messages to test different code paths in your transactions and inspect transaction output messages.

# Displaying an IMS Database Structure via Green Screen



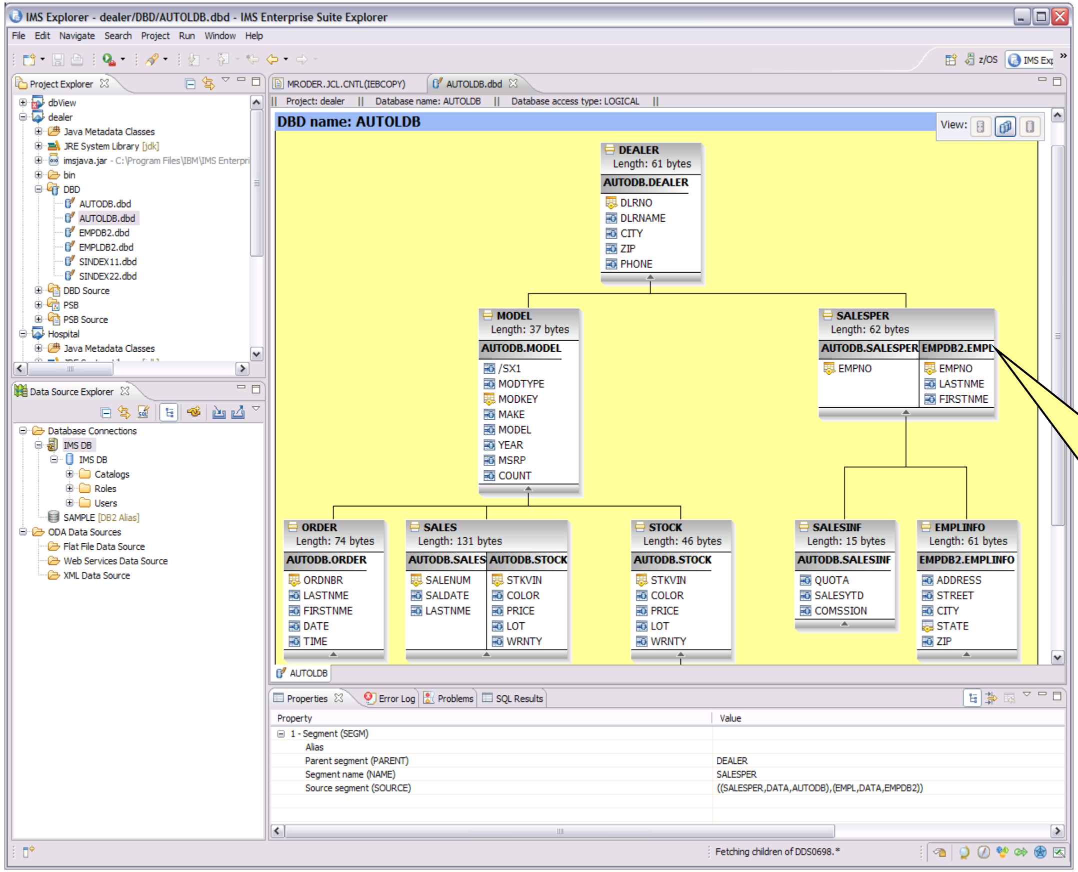
```
       DBD     NAME=APSV01AD,ACCESS=(HISAM,VSAM),PASSWD=NO
***********************************************************************
*         DATASET GROUP NUMBER 1                                      *
***********************************************************************
DSG001 DATASET DD1=HISKSDSA,OVFLW=HISESDSA,DEVICE=3330,SIZE=(512,512),  C
               RECORD=(22,22)
***********************************************************************
*         SEGMENT NUMBER 1                                            *
***********************************************************************
       SEGM    NAME=A,PARENT=0,BYTES=11,RULES=(LLL,LAST),               C
               PTR=(NOTWIN,,,CTR,)
       FIELD   NAME=(A,SEQ,U),START=1,BYTES=3,TYPE=C
       LCHILD  NAME=(AB),PTR=NONE,RULES=LAST
***********************************************************************
*         SEGMENT NUMBER 2                                            *
***********************************************************************
       SEGM    NAME=AA,PARENT=((A,)),BYTES=12,RULES=(LLL,LAST),         C
               PTR=(NOTWIN,,,,)
       FIELD   NAME=(AA,SEQ,U),START=1,BYTES=4,TYPE=C
```

## Displaying a physical IMS Database Structure



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

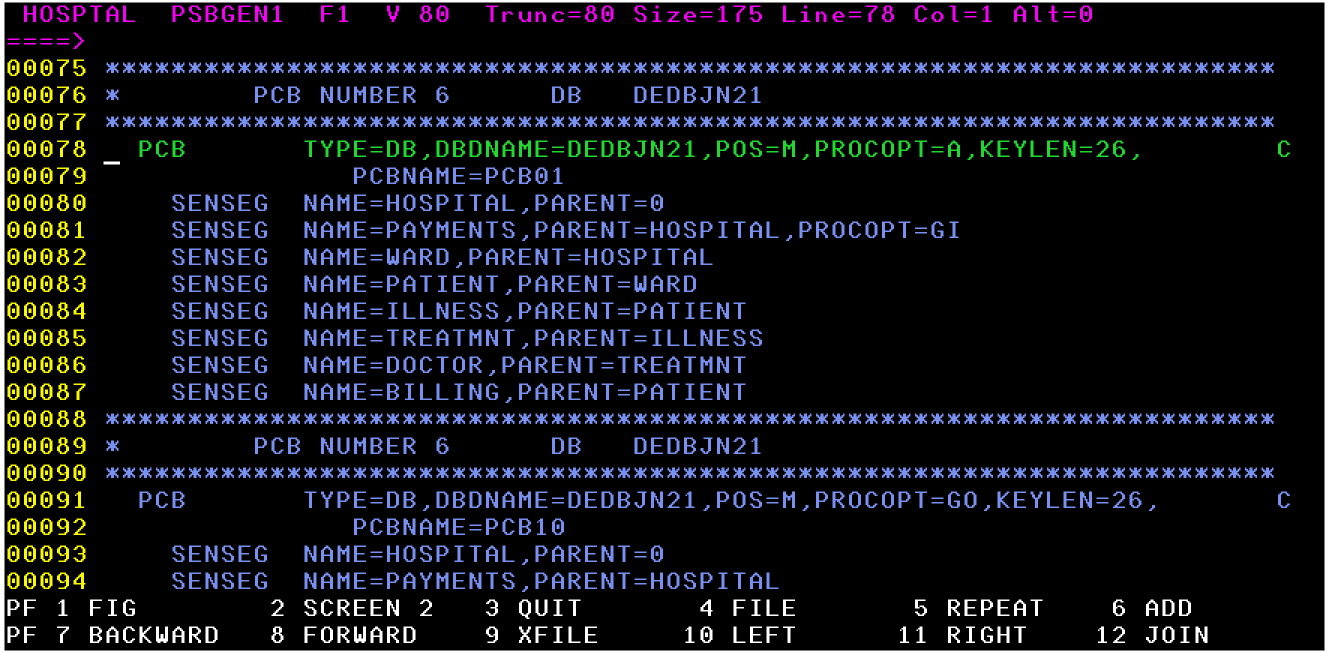
# Displaying a logical IMS Database



| Property | Value |
|---|---|
| 1 - Segment (SEGM) | |
| Alias | |
| Parent segment (PARENT) | DEALER |
| Segment name (NAME) | SALESPER |
| Source segment (SOURCE) | ((SALESPER,DATA,AUTODB),(EMPL,DATA,EMPDB2)) |

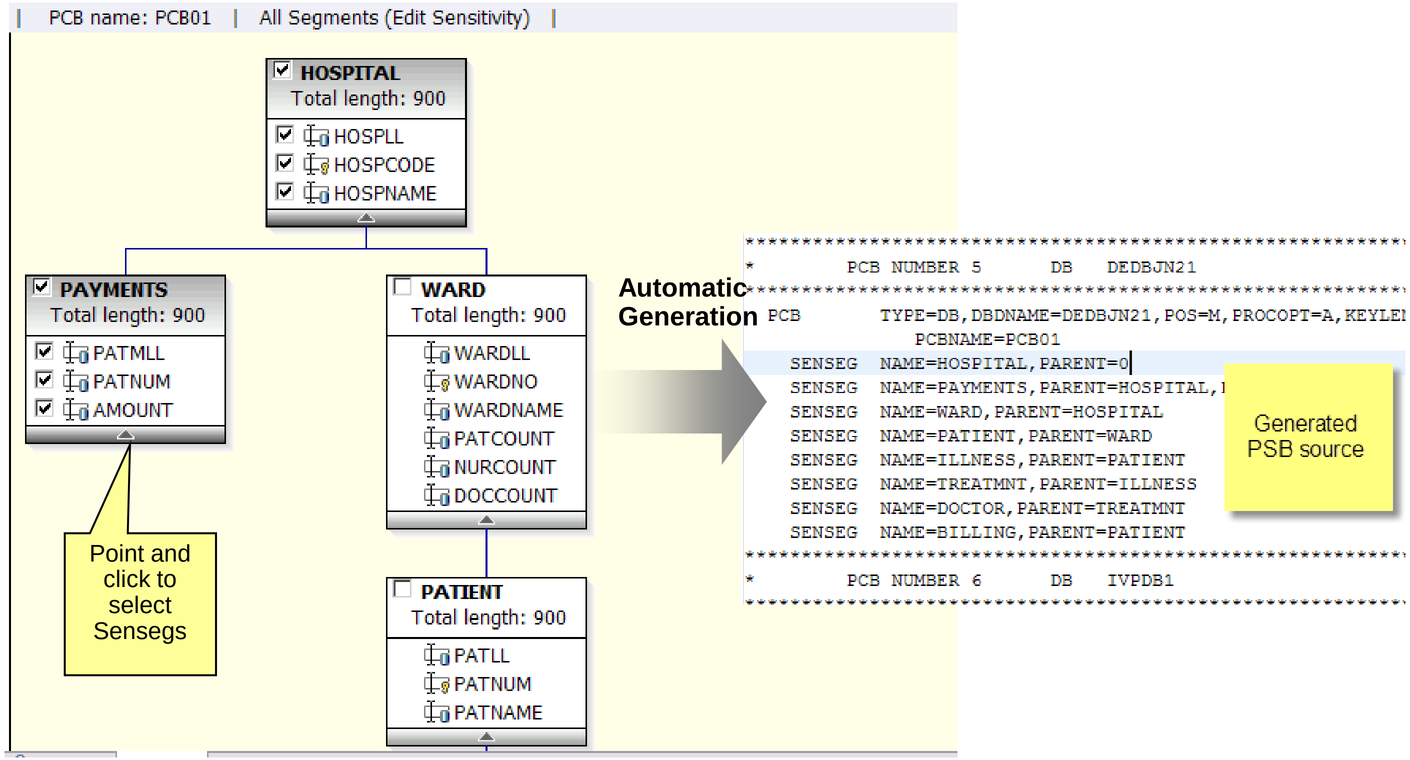
A concatenated segment and its underlying source segments

# PSB and PCB Definitions via Green Screen



```
 HOSPTAL  PSBGEN1  F1  V 80  Trunc=80 Size=175 Line=78 Col=1 Alt=0
====>
***********************************************************************
*        PCB NUMBER 6       DB    DEDBJN21
***********************************************************************
  PCB       TYPE=DB,DBDNAME=DEDBJN21,POS=M,PROCOPT=A,KEYLEN=26,        C
               PCBNAME=PCB01
    SENSEG  NAME=HOSPITAL,PARENT=0
    SENSEG  NAME=PAYMENTS,PARENT=HOSPITAL,PROCOPT=GI
    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    DEDBJN21
***********************************************************************
  PCB       TYPE=DB,DBDNAME=DEDBJN21,POS=M,PROCOPT=GO,KEYLEN=26,       C
               PCBNAME=PCB10
    SENSEG  NAME=HOSPITAL,PARENT=0
    SENSEG  NAME=PAYMENTS,PARENT=HOSPITAL
PF 1 FIG         2 SCREEN 2    3 QUIT        4 FILE        5 REPEAT     6 ADD
PF 7 BACKWARD    8 FORWARD     9 XFILE      10 LEFT       11 RIGHT     12 JOIN
```

# Building a PCB definition with IMS Explorer



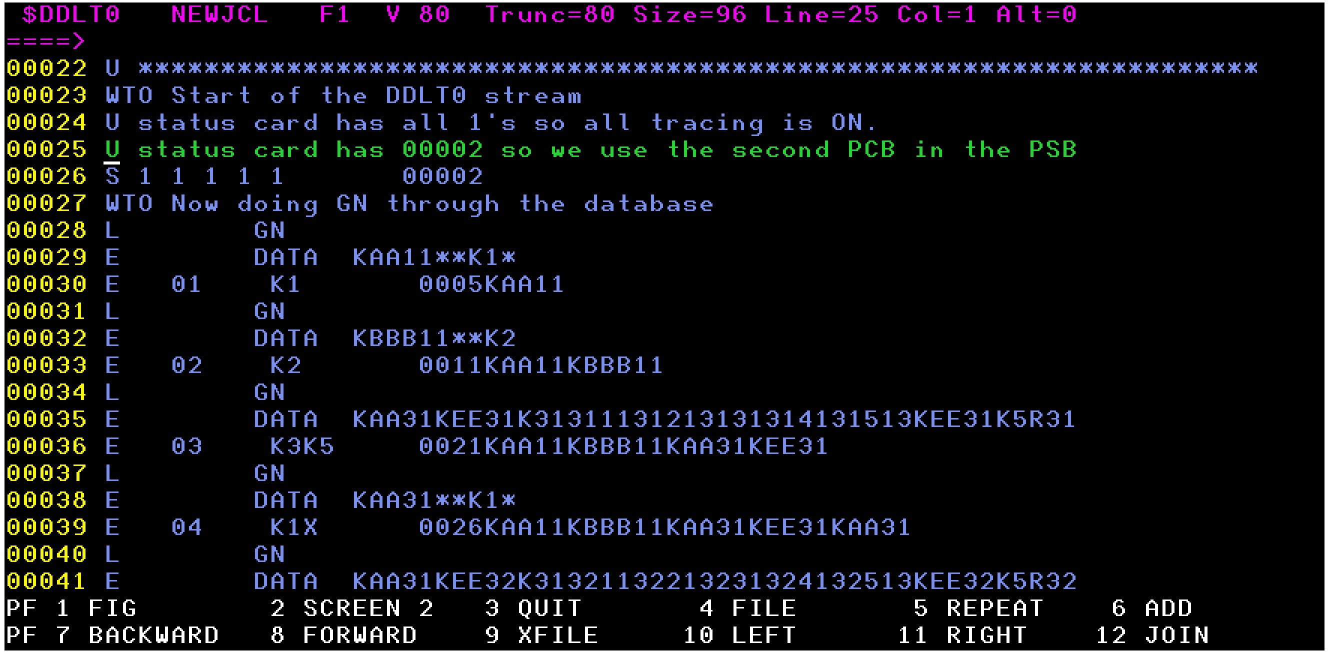
Point and click to select Sensegs

**Automatic Generation**

```
************************************************************
*         PCB NUMBER 5       DB    DEDBJN21
************************************************************
  PCB         TYPE=DB,DBDNAME=DEDBJN21,POS=M,PROCOPT=A,KEYLE
                 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
************************************************************
```

Generated PSB source

# Querying an IMS Database with DFSDDLT0

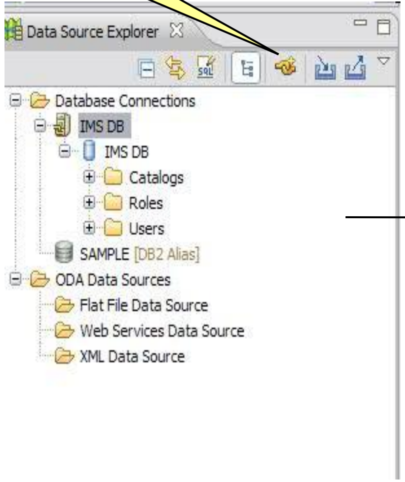


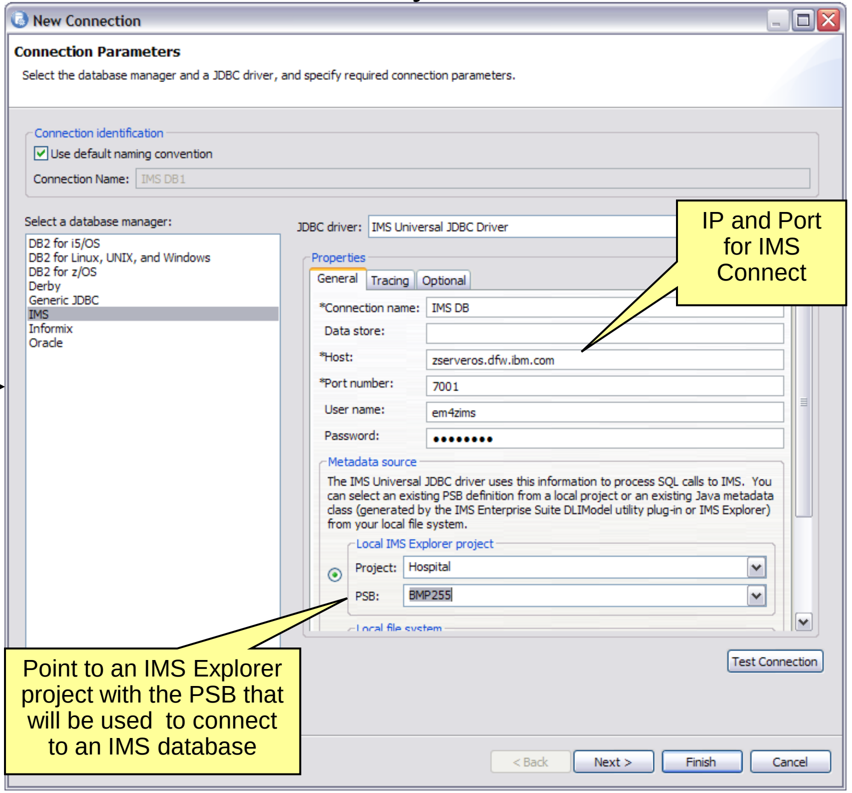
```
 $DDLT0    NEWJCL    F1   V 80   Trunc=80 Size=96 Line=25 Col=1 Alt=0
====>
U *********************************************************************
WTO Start of the DDLT0 stream
U status card has all 1's so all tracing is ON.
U status card has 00002 so we use the second PCB in the PSB
S 1 1 1 1 1       00002
WTO Now doing GN through the database
L        GN
E        DATA  KAA11**K1*
E   01    K1       0005KAA11
L        GN
E        DATA  KBBB11**K2
E   02    K2       0011KAA11KBBB11
L        GN
E        DATA  KAA31KEE31K3131113121313131413151 3KEE31K5R31
E   03    K3K5     0021KAA11KBBB11KAA31KEE31
L        GN
E        DATA  KAA31**K1*
E   04    K1X      0026KAA11KBBB11KAA31KEE31KAA31
L        GN
E        DATA  KAA31KEE32K3132113221323132413251 3KEE32K5R32
PF 1 FIG          2 SCREEN 2    3 QUIT        4 FILE        5 REPEAT    6 ADD
PF 7 BACKWARD     8 FORWARD     9 XFILE      10 LEFT       11 RIGHT    12 JOIN
```

# Querying an IMS Database with IMS Explorer

1 – Start by establishing  a connection to an IMS system ….

Create a new connection…




New Connection

**Connection Parameters**

Select the database manager and a JDBC driver, and specify required connection parameters.

Connection identification

[✓] Use default naming convention

Connection Name: IMS DB1

Select a database manager:

DB2 for i5/OS
DB2 for Linux, UNIX, and Windows
DB2 for z/OS
Derby
Generic JDBC
IMS
Informix
Oracle

JDBC driver: IMS Universal JDBC Driver

Properties

General | Tracing | Optional

| | |
|---|---|
| *Connection name: | IMS DB |
| Data store: | |
| *Host: | zserveros.dfw.ibm.com |
| *Port number: | 7001 |
| User name: | em4zims |
| Password: | ●●●●●●●● |

Metadata source

The IMS Universal JDBC driver uses this information to process SQL calls to IMS. You can select an existing PSB definition from a local project or an existing Java metadata class (generated by the IMS Enterprise Suite DLIModel utility plug-in or IMS Explorer) from your local file system.

● Local IMS Explorer project

Project: Hospital

PSB: BMP255

Local file system

Test Connection

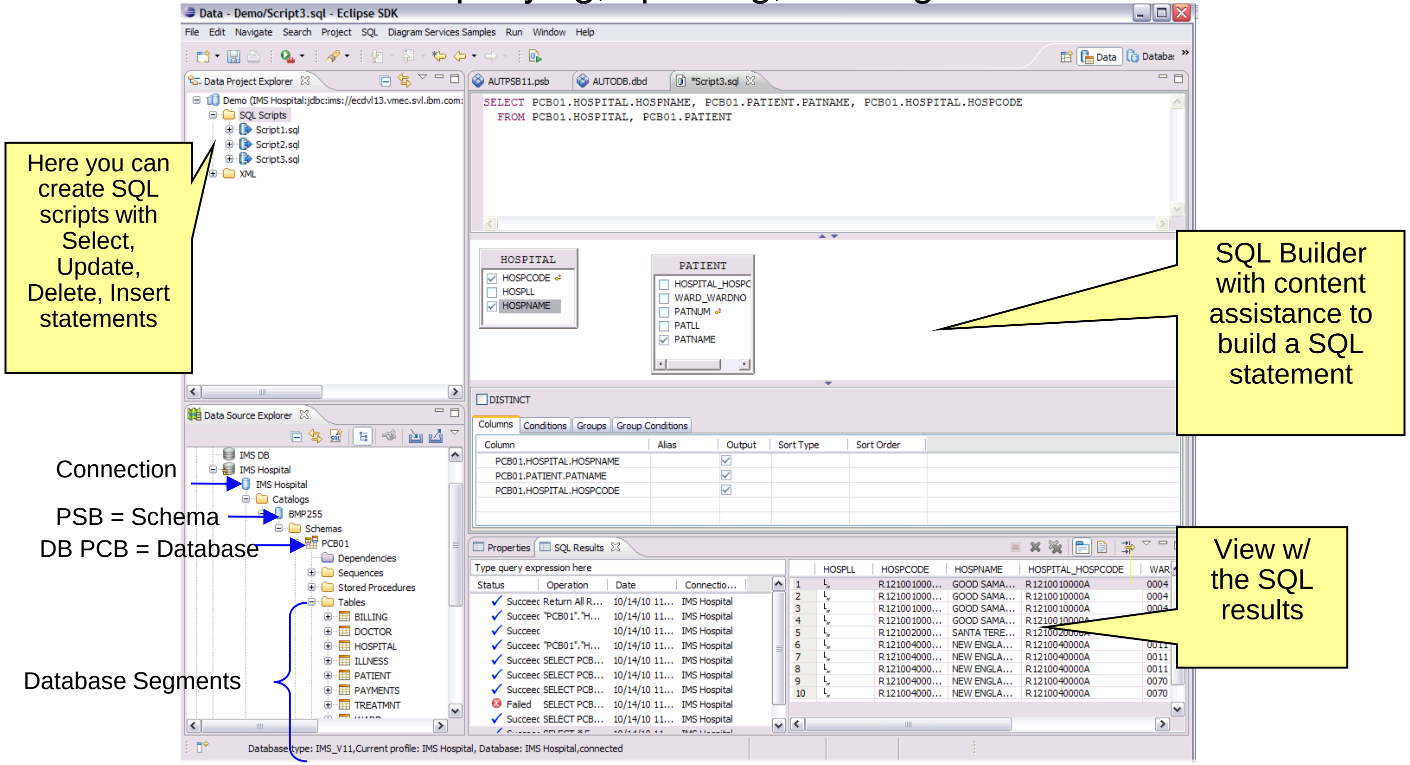
< Back | Next > | Finish | Cancel

IP and Port for IMS Connect

Point to an IMS Explorer project with the PSB that will be used to connect to an IMS database

# Querying an IMS Database with IMS

## 2 – Connect … and start querying, updating, deleting IMS data



```
SELECT PCB01.HOSPITAL.HOSPNAME, PCB01.PATIENT.PATNAME, PCB01.HOSPITAL.HOSPCODE
  FROM PCB01.HOSPITAL, PCB01.PATIENT
```

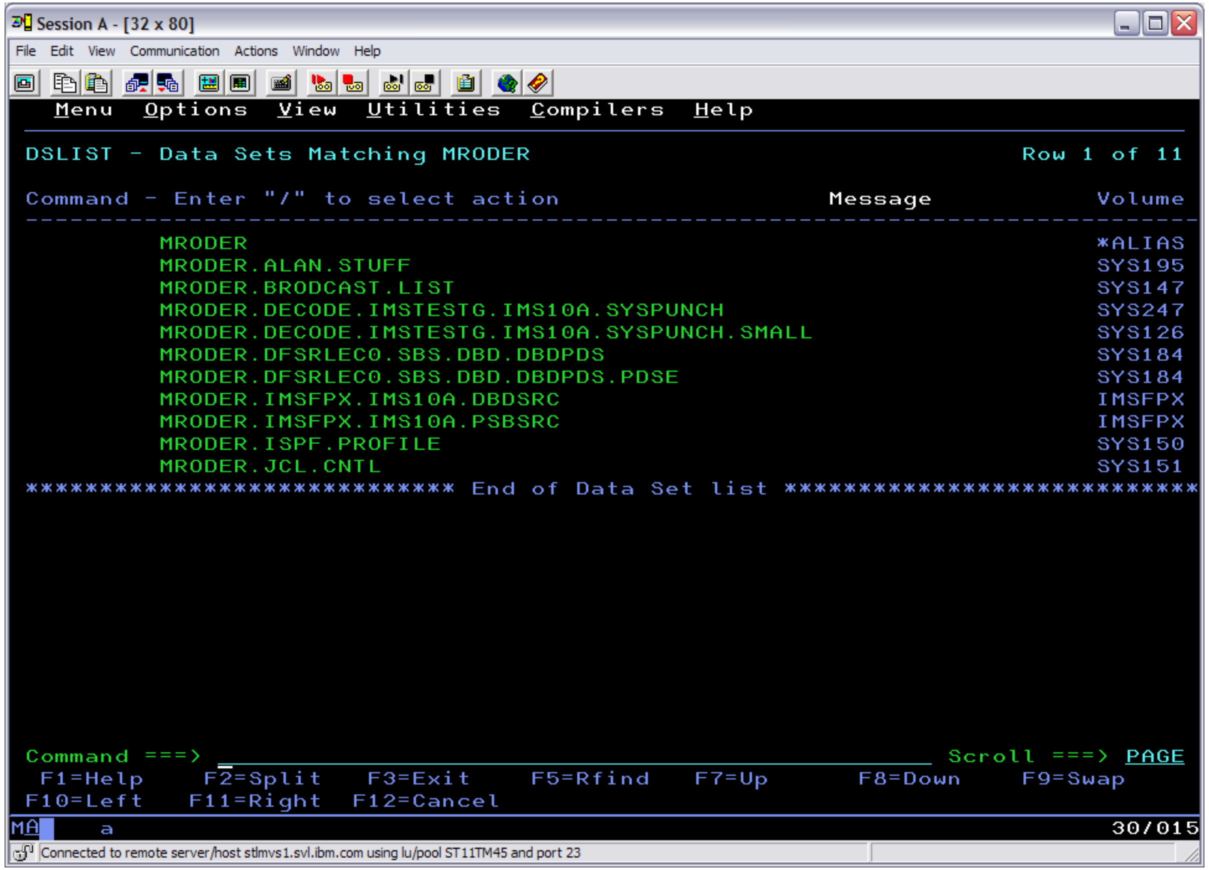
| Column | Alias | Output | Sort Type | Sort Order |
|---|---|---|---|---|
| PCB01.HOSPITAL.HOSPNAME | | ✓ | | |
| PCB01.PATIENT.PATNAME | | ✓ | | |
| PCB01.HOSPITAL.HOSPCODE | | ✓ | | |

| Status | Operation | Date | Connectio... |
|---|---|---|---|
| ✓ Succeec | Return All R... | 10/14/10 11... | IMS Hospital |
| ✓ Succeec | "PCB01"."H... | 10/14/10 11... | IMS Hospital |
| ✓ Succeec | | 10/14/10 11... | IMS Hospital |
| ✓ Succeec | "PCB01"."H... | 10/14/10 11... | IMS Hospital |
| ✓ Succeec | SELECT PCB... | 10/14/10 11... | IMS Hospital |
| ✓ Succeec | SELECT PCB... | 10/14/10 11... | IMS Hospital |
| ✓ Succeec | SELECT PCB... | 10/14/10 11... | IMS Hospital |
| ✓ Succeec | SELECT PCB... | 10/14/10 11... | IMS Hospital |
| ✗ Failed | SELECT PCB... | 10/14/10 11... | IMS Hospital |
| ✓ Succeec | SELECT PCB... | 10/14/10 11... | IMS Hospital |

| | HOSPLL | HOSPCODE | HOSPNAME | HOSPITAL_HOSPCODE | WAR |
|---|---|---|---|---|---|
| 1 | └ | R121001000... | GOOD SAMA... | R1210010000A | 0004 |
| 2 | └ | R121001000... | GOOD SAMA... | R1210010000A | 0004 |
| 3 | └ | R121001000... | GOOD SAMA... | R1210010000A | 0004 |
| 4 | └ | R121001000... | GOOD SAMA... | R1210010000A | [illegible] |
| 5 | └ | R121002000... | SANTA TERE... | R1210020000A | [illegible] |
| 6 | └ | R121004000... | NEW ENGLA... | R1210040000A | 0011 |
| 7 | └ | R121004000... | NEW ENGLA... | R1210040000A | 0011 |
| 8 | └ | R121004000... | NEW ENGLA... | R1210040000A | 0011 |
| 9 | └ | R121004000... | NEW ENGLA... | R1210040000A | 0070 |
| 10 | └ | R121004000... | NEW ENGLA... | R1210040000A | 0070 |

Database type: IMS_V11,Current profile: IMS Hospital, Database: IMS Hospital,connected

# Browsing Data Sets and Submitting JCL’s

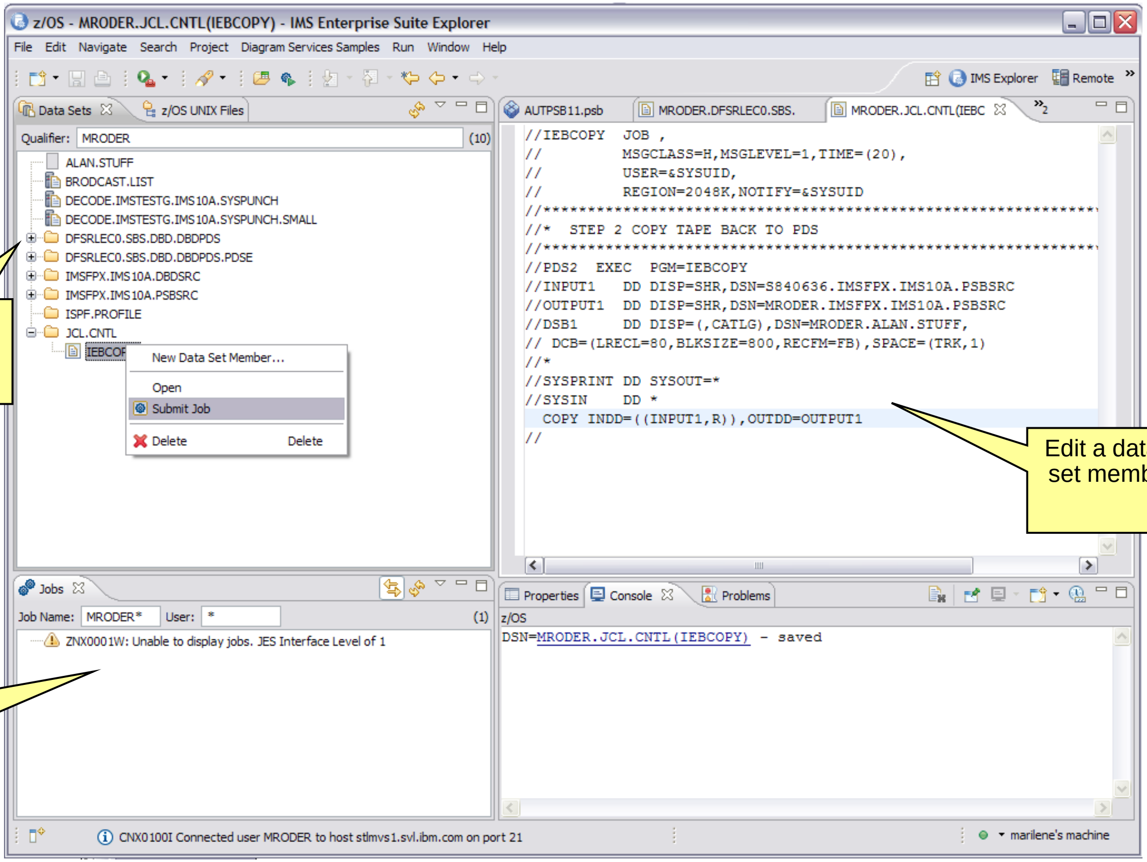


```
   Menu  Options  View  Utilities  Compilers  Help
 ------------------------------------------------------------------------------
 DSLIST - Data Sets Matching MRODER                                Row 1 of 11
 Command - Enter "/" to select action                  Message           Volume
 -------------------------------------------------------------------------------
          MRODER                                                          *ALIAS
          MRODER.ALAN.STUFF                                               SYS195
          MRODER.BRODCAST.LIST                                            SYS147
          MRODER.DECODE.IMSTESTG.IMS10A.SYSPUNCH                          SYS247
          MRODER.DECODE.IMSTESTG.IMS10A.SYSPUNCH.SMALL                    SYS126
          MRODER.DFSRLEC0.SBS.DBD.DBDPDS                                  SYS184
          MRODER.DFSRLEC0.SBS.DBD.DBDPDS.PDSE                             SYS184
          MRODER.IMSFPX.IMS10A.DBDSRC                                     IMSFPX
          MRODER.IMSFPX.IMS10A.PSBSRC                                     IMSFPX
          MRODER.ISPF.PROFILE                                             SYS150
          MRODER.JCL.CNTL                                                 SYS151
 ***************************** End of Data Set list ****************************

 Command ===> ________________________________________________ Scroll ===> PAGE
  F1=Help      F2=Split     F3=Exit      F5=Rfind     F7=Up        F8=Down      F9=Swap
 F10=Left     F11=Right    F12=Cancel
MA     a                                                                 30/015
```

Connected to remote server/host stlmvs1.svl.ibm.com using lu/pool ST11TM45 and port 23

# Browsing Data Sets and Submitting JCL’s



Qualifier: MRODER (10)

```
ALAN.STUFF
BRODCAST.LIST
DECODE.IMSTESTG.IMS10A.SYSPUNCH
DECODE.IMSTESTG.IMS10A.SYSPUNCH.SMALL
DFSRLEC0.SBS.DBD.DBDPDS
DFSRLEC0.SBS.DBD.DBDPDS.PDSE
IMSFPX.IMS10A.DBDSRC
IMSFPX.IMS10A.PSBSRC
ISPF.PROFILE
JCL.CNTL
  IEBCOP
```

```
//IEBCOPY  JOB ,
//           MSGCLASS=H,MSGLEVEL=1,TIME=(20),
//           USER=&SYSUID,
//           REGION=2048K,NOTIFY=&SYSUID
//*****************************************************************
//*  STEP 2 COPY TAPE BACK TO PDS
//*****************************************************************
//PDS2   EXEC  PGM=IEBCOPY
//INPUT1    DD DISP=SHR,DSN=S840636.IMSFPX.IMS10A.PSBSRC
//OUTPUT1   DD DISP=SHR,DSN=MRODER.IMSFPX.IMS10A.PSBSRC
//DSB1      DD DISP=(,CATLG),DSN=MRODER.ALAN.STUFF,
// DCB=(LRECL=80,BLKSIZE=800,RECFM=FB),SPACE=(TRK,1)
//*
//SYSPRINT DD SYSOUT=*
//SYSIN     DD *
  COPY INDD=((INPUT1,R)),OUTDD=OUTPUT1
//
```

Job Name: MRODER* User: * (1)

```
ZNX0001W: Unable to display jobs. JES Interface Level of 1
```

```
DSN=MRODER.JCL.CNTL(IEBCOPY) - saved
```

```
CNX0100I Connected user MRODER to host stlmvs1.svl.ibm.com on port 21
```

marilene's machine

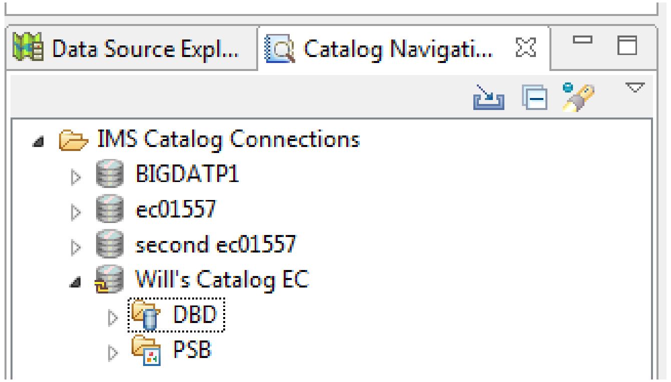
Browse data sets

Edit a data a set member

JES output is displayed here

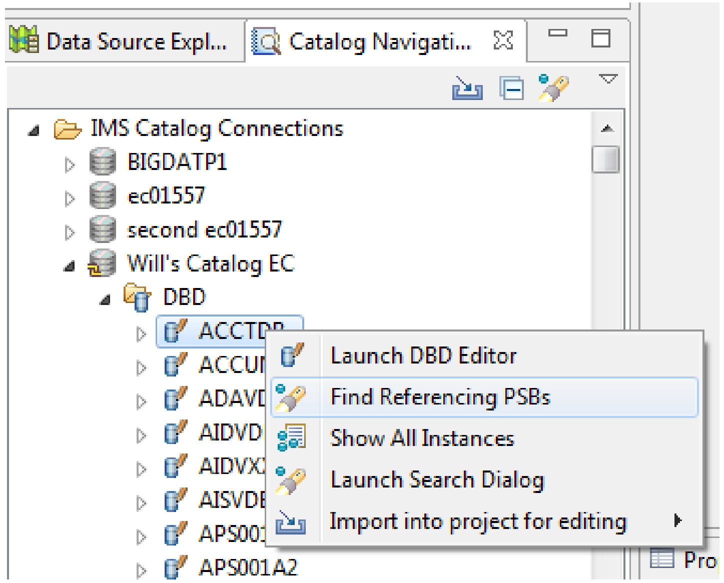
# IMS Catalog Navigation View

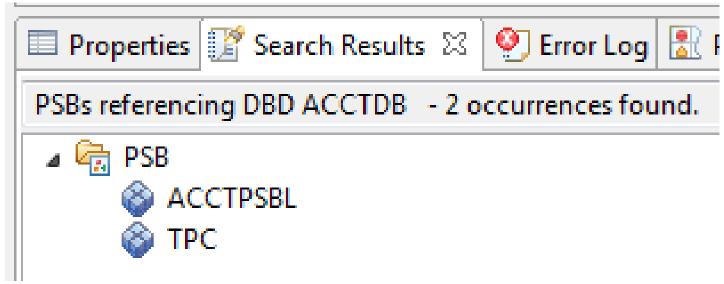
- Get a list of all the PSBs/DBDs in the system.



# Built-in queries

- **Several built-in queries have been added to assist with resource and relationship discovery**
  - "What are all the PSBs that reference this DBD?"



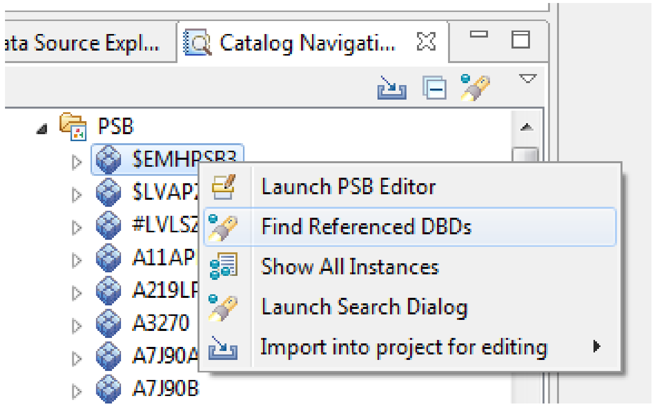


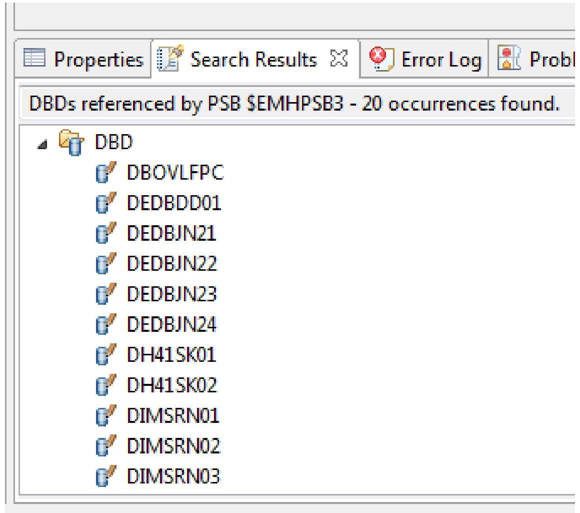
PSBs referencing DBD ACCTDB - 2 occurrences found.

PSB
ACCTPSBL
TPC

## Built-in queries continued

- “What are all the DBDs referenced by this PSB?”





DBDs referenced by PSB $EMHPSB3 - 20 occurrences found.

- DBD
  - DBOVLFPC
  - DEDBDD01
  - DEDBJN21
  - DEDBJN22
  - DEDBJN23
  - DEDBJN24
  - DH41SK01
  - DH41SK02
  - DIMSRN01
  - DIMSRN02
  - DIMSRN03