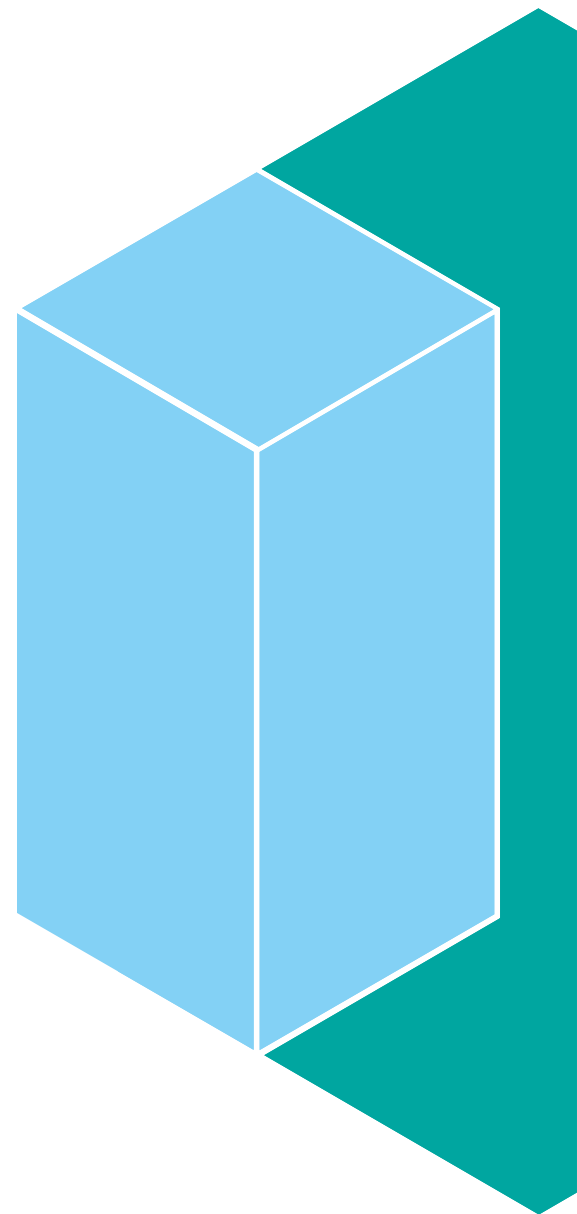


# Enhancing the IMS catalog with better metadata and improvements to HALDB Session B07

Richard Tran, [richtran@us.ibm.com](mailto:richtran@us.ibm.com)



Sharpen your competitive edge  
**2016 IMS Technical Symposium**  
March 7 – 10, 2016  
Wiesbaden, Germany

[www.ims-symposium.com](http://www.ims-symposium.com)

## Please Note

- **IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.**
- **Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.**
- **The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.**
- **The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.**

# Agenda

- **IMS Catalog**

- What's new with V14
- Enablement programs
- Metadata tips

- **HALDB**

- New feature for V13 and V14
- Enablement tips

# Agenda

- **IMS Catalog**

- **What's new with V14**
- Enablement programs
- Metadata tips

- **HALDB**

- New feature for V13 and V14
- Enablement tips

## The IMS Catalog at a glance

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



# What's new with V14

## ▪ **Catalog features**

- IMS managed ACBs
- Data definition language (DDL) for IMS
- IMS catalog activity record
- GSAM metadata support

## ▪ **Catalog enabled features**

- Extended SQL support for COBOL and .NET
  - Support for GROUP BY
  - Support for aggregates (e.g., SUM, AVG, etc.)

# Agenda

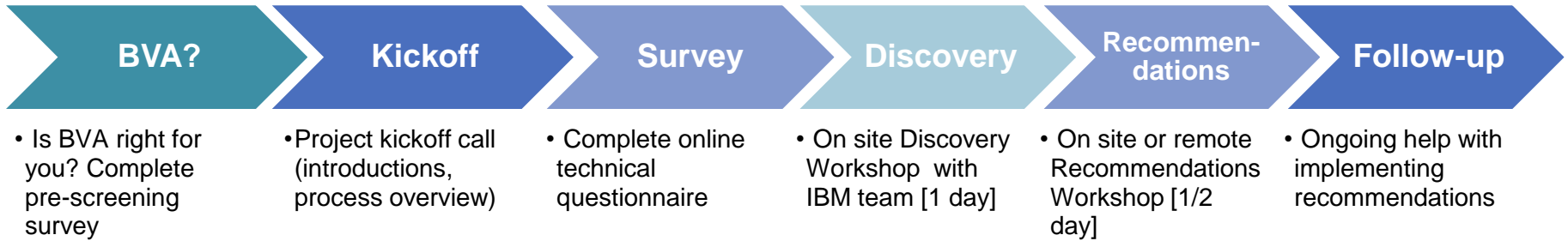
- **IMS Catalog**

- What's new with V14
- Enablement programs**
- Metadata tips

- **HALDB**

- New feature for V13 and V14
- Enablement tips

# IMS Business Value Assessment (BVA) Workshop



BVA helps identify ways to get the most out of your IMS investment

- Tactical and strategic recommendations to:
  - Increase benefits while lowering costs
  - Improve processing efficiencies for IMS applications
  - Modernize access to IMS applications and data

Complete this short survey to get started:

- <https://ibm.biz/BdHAFG>



*A no-charge workshop!*



*Contact*

• [ibmims@us.ibm.com](mailto:ibmims@us.ibm.com)



# IMS “Deployment Project Office” (DPO) Program

## IMS “Deployment Project

### Objective

- **Drive client enablement and related POCs to accelerate deployment of select capabilities into production**
- **Deliver signature client experience and gain customer references with IMS key solutions**
- **Reduce support cost by assisting customers with deployments.**
- **Learn and drive feedback into development and product management.**

### Success

- **Completed POCs**
- **POCs converted to production deployment**
- **# of References**

### Approach

Execute a “sales” approach - qualify client opportunities and align resources accordingly

- Capped # of hours (~50 hrs) of technical guidance (web/conference) w/ workshop option
- No charge – production deployment + reference is “payment”
- IBM DPO project team consists of (tech lead, project manager, deployment team)

# IMS DPO Open Database Activities (sample)

Activity	Tasks	Client
Deployment agreement	Complete a formal deployment contract	Executive Sponsor
Project Planning	<ul style="list-style-type: none"> <li>• Survey client POC team on current production usage scenarios</li> <li>• Best Practices</li> <li>• Basic Training/Education</li> </ul>	<ul style="list-style-type: none"> <li>• Sponsor</li> <li>• Project Manager</li> <li>• POC Lead</li> </ul>
Implementation and Integration Planning & Preparation	<ul style="list-style-type: none"> <li>• Design session</li> <li>• Workshop (1 day)</li> </ul>	<ul style="list-style-type: none"> <li>• Project Manager</li> <li>• POC Lead/Team</li> </ul>
	Enable Open Database Access by setting up CSL and ICON and catalog (optional)	Sys Prog
Database modeling	Model Databases in E4D and generate appropriate metadata	<ul style="list-style-type: none"> <li>• DBA</li> <li>• Sys Prog</li> </ul>
Infrastructure setup	Create fast performing queries specifically for their use case	<ul style="list-style-type: none"> <li>• App Dev</li> <li>• DBA</li> <li>• Sys Prog (optional)</li> </ul>
Sample applications	Create a functionally performing application that meets the customer use case	App Dev
Optimization	Check to see that the IMS application meets performance requirements	<ul style="list-style-type: none"> <li>• App Dev</li> <li>• DBA</li> <li>• SysProg</li> </ul>
Project Close-Out	<ul style="list-style-type: none"> <li>• POC analysis &amp; write-up</li> <li>• DPO survey &amp; feedback</li> <li>• Schedule follow-up checkpoints for production deployment</li> </ul>	<ul style="list-style-type: none"> <li>• Sponsor</li> <li>• Project Manager</li> <li>• POC team lead</li> </ul>

# Agenda

- **IMS Catalog**

- What's new with V14
- Enablement programs
- Metadata tips**

- **HALDB**

- New feature for V13 and V14
- Enablement tips

## Modeling data using a pseudo catalog

- **The IMS JDBC driver access metadata from the IMS catalog using a Get Unique Record (GUR) call**
  - The GUR call returns back an XML representation of the PSB or DBD resource
  
- **The IMS Explorer for Development generates both PSB and DBD source as well as GUR XML equivalents**
  - Provides the ability to test metadata changes prior to hardening those changes to the IMS catalog

## How to access the XML pseudo catalog representation

- When creating an IMS database connection, make sure to select a local project as the Metadata source

Metadata source:	Local IMS Explorer project
Project:	SolutionKit
PSB:	DEMO01

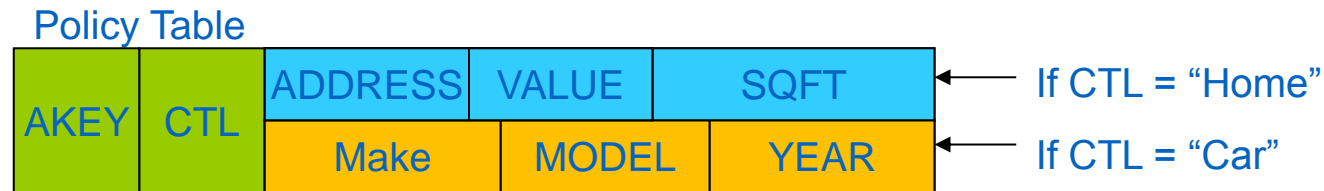
- This will generate a unique IMS JDBC connect URL to reference the XML files directly

```
URL: jdbc:ims://mytest.com:5555/xml://DEMO01.psb;dpsbOnCommit=true;treatInvalidDecimalAsNull=true;  
xmlMetadataLocation=C:/Users/IBM_ADMIN/IBM/rationalsdp/workspace/SolutionKit;
```

- The IMS catalog is still recommended for production usage and changes should be hardened there

## IMS dynamic record mapping

- An IMS record can be mapped multiple ways depending on a control field
- For example an insurance policy table can be interpreted as multiple types of policies depending on the value of the control field



- By defining maps in the IMS catalog, the IMS JDBC driver will automatically apply the correct overlays **instead of** the application developer

## Defining dynamic mappings

- The IMS Explorer for Development allows a DBA to define maps and the various possible overlays (cases)
- First define the segment maps and cases through the **Add or Edit Segment Maps wizard**

The screenshot shows a configuration window with two sections: "Segment Maps" and "Map Cases".

**Segment Maps**  
Segment maps enable the alternate mapping of fields within a segment. [Learn more...](#)

Map Name	Control Field Name
POLICY	TYPE

Buttons: Add..., Edit..., Remove

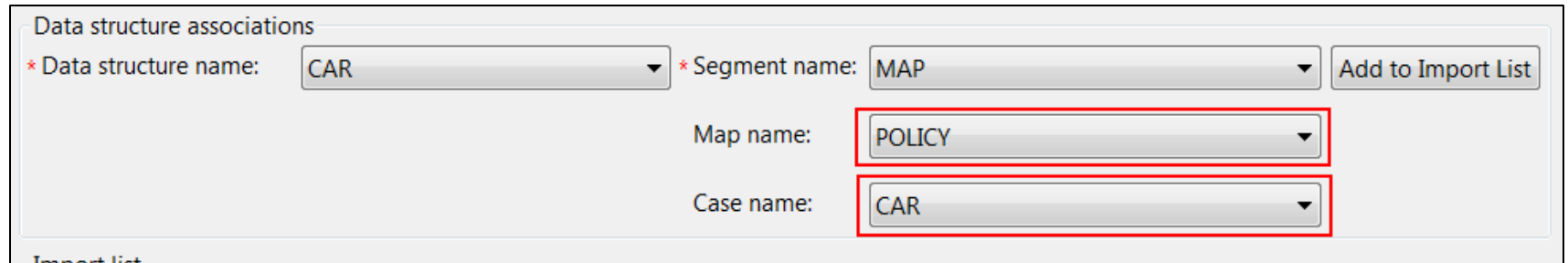
**Map Cases**  
Map cases define alternate mappings of fields within a segment. [Learn more...](#)

Case Name	Control Field Value	Map Name
LIFE	L	POLICY
HOUSE	H	POLICY
CAR	C	POLICY

Buttons: Add..., Edit..., Remove

## Import the individual overlays for each case

- **The IMS Explorer for Development allows for copybook imports to overlay each case through the Import Data Structure Wizard**



The screenshot shows a window titled "Data structure associations" with the following fields and controls:

- \* Data structure name: CAR (dropdown menu)
- \* Segment name: MAP (dropdown menu)
- Add to Import List (button)
- Map name: POLICY (dropdown menu, highlighted with a red box)
- Case name: CAR (dropdown menu, highlighted with a red box)

Import list

- **Make sure to provide an overlay for each **case** that was previously defined**



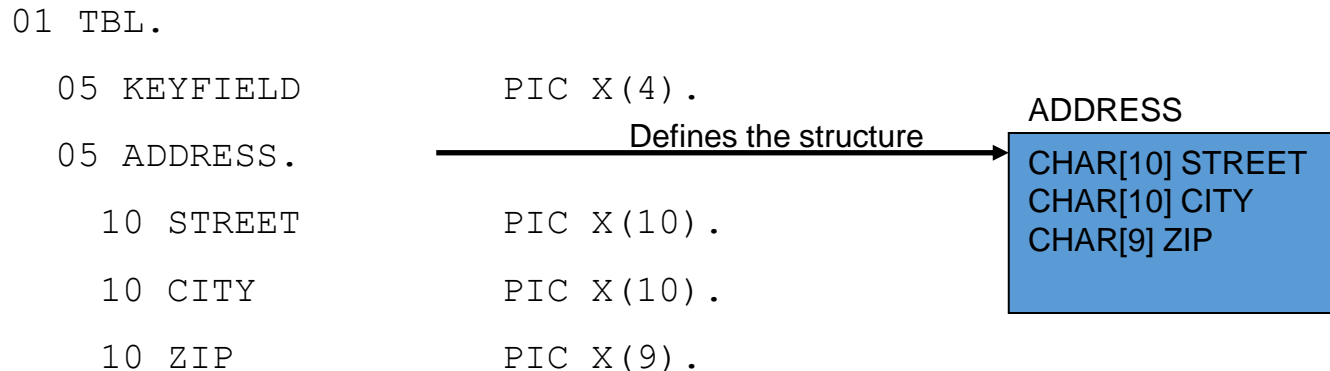
## The DBD source for a dynamic map definition

- The IMS Explorer for Development will generate source containing new DBD gen macros (DFSMAP, DFSCASE) that will allow dynamic maps to be added to the IMS catalog

```
DFSMAP NAME=POLICY,  
        DEPENDONGON=TYPE  
DFSCASE NAME=LIFE,  
        CASEID=L,  
        CASEIDTYPE=C,  
        MAPNAME=POLICY  
DFSCASE NAME=HOUSE,  
        CASEID=H,  
        CASEIDTYPE=C,  
        MAPNAME=POLICY  
DFSCASE NAME=CAR,  
        CASEID=C,  
        CASEIDTYPE=C,  
        MAPNAME=POLICY
```

# IMS STRUCT datatype support

- Most existing IMS field definitions are based on COBOL copybooks or PL/I include files
- STRUCTs are an inherent part of most copybooks
- Example of a STRUCT



## STRUCT limitations

- The IMS JDBC driver fully supports the JDBC standard in regards to the **STRUCT** datatype
  - However, many JDBC tools do not
- Sub-fields of a **STRUCT** cannot be referenced directly in a **SQL** statement
  - Valid SQL
    - SELECT **ADDRESS** FROM TBL
  - Invalid SQL
    - SELECT **STREET**, **CITY**, **ZIP** FROM TBL
    - SELECT \* FROM TBL WHERE **ZIP** = ?

```
ADDRESS
CHAR[10] STREET
CHAR[10] CITY
CHAR[9] ZIP
```

# IMS flatten structure capabilities

- **There are two ways to flatten these types of structures**
  - Modify the copybook
  - Use the IMS JDBC flattenTables connection property
- **Conceptual example of the flattenTables property**

01 TBL.			01 TBL.	
05 KEYFIELD	PIC X(4).		05 KEYFIELD	PIC X(4).
05 ADDRESS.			05 STREET	PIC X(10).
10 STREET	PIC X(10).	→ flattenTables=true	05 CITY	PIC X(10).
10 CITY	PIC X(10).		05 ZIP	PIC X(9).
10 ZIP	PIC X(9).			

# IMS flatten structure capabilities when applied to ARRAY types

- The `flattenTables` property also applies to ARRAY datatypes
- IMS does create a naming convention when looking at ARRAY types
  - `<ARRAYNAME>_<ELEMENT#>_<FIELDNAME>`
- Conceptual example of the `flattenTables` property

01 STUDENT.		01 STUDENT.
05 COURSES OCCURS 2 TIMES.		05 COURSES_1_COURSENAME PIC X(15).
10 COURSENAME PIC X(15).	→ <code>flattenTables=true</code>	05 COURSES_1_INSTRUCTOR PIC X(15).
10 INSTRUCTOR PIC X(25).		05 COURSES_2_COURSENAME PIC X(15).
		05 COURSES_2_INSTRUCTOR PIC X(15).

# Determining where to flatten complex structures

## ▪ **Flattening in the copybook**

- Preserves the flatten model with the IMS catalog
- Provides the same flatten view to all users of the IMS catalog
  - e.g., Java, COBOL, .NET
- Allows for custom names versus generated names such as in the ARRAY case

## ▪ **Flattening through the JDBC driver**

- Preserves the original copybook structure in the IMS catalog
- Does not require manual editing of copybooks during copybook import

## Custom data type support

- **IMS data is stored on disk as a BLOB, so interpretation of that BLOB is left to the application to decide**
- **IMS supports the use of custom data types in order to represent the data as an equivalent Java data type**
- **A few examples:**
  - A date value that is based on the number of days since Jan 1, 1950
  - A date value that is stored as a packed decimal number:  
0x19500101c

## How to write a custom user type converter

- In order to create a custom user type converter, the application developer will need to extend the `com.ibm.ims.dli.types.BaseTypeConverter` class
- The following methods will need to be implemented:
  - `readObject()`
    - For SQL SELECT statements
  - `writeObject()`
    - For SQL INSERT and DELETE statements
    - For SQL WHERE clauses



## Helper classes for writing a type converter

- **The IMS JDBC driver provides a ConverterFactory class that will allow users to instantiate basic converters**
  - DoubleTypeConverter
  - IntegerTypeConverter
  - UIntegerTypeConverter
  - PackedDecimalTypeConverter
  - etc.
- **It is recommended when building complex custom type converter to leverage the basic converters**
- **Converter classes can be found in com.ibm.ims.dli.converters package**

## Defining the custom type converter to the IMS catalog

- The IMS Explorer for Development allows DBAs to define when a custom type converter should be used through the Add or Edit Field Wizard

**Specify Data Type Information for the DBD Field**

The data type expected by the application program and the data converter type are used by the IMS catalog and the JDBC driver to specify how the application views and stores the data in a database.

Application data type  
Data type and data structure expected by application program.

\* Data type:  Length (BYTES):

Data conversion information

Converter type:  \* Converter class name:

- A Converter type of USER signals a custom converter
- The Converter class name is the fully qualified name
- The Data type parameter determines the application datatype representation of the custom type

## Defining converter properties

- **The IMS Catalog also allows converter properties to be set on a per field basis**
- **This allows a single type converter class to be used for more dynamic conversions depending on property values**
  - e.g., A date value stored as a packed decimal could have a property to tell if the packed decimal value is stored as signed or unsigned

Data conversion information

Converter type:  \* Converter class name:

Properties

Properties to pass to the converter class at run time.

Property name:	Property value:
isSigned = true	

## The DBD source for a custom type converter

- **The IMS Explorer for Development will generate new DBD source containing any custom type converter added through the Add or Edit Field Wizard**

```
FIELD EXTERNALNAME=MYCUSTOMFIELD,  
      BYTES=5,  
      START=32,  
      DATATYPE=DATE  
DFSMARSH ,  
      USERTYPECONVERTER=com.ibm.ims.SamplConv,  
      PROPERTIES=(isSigned=true),  
      PATTERN='yyyyMMdd'
```

# Agenda

- **IMS Catalog**

- What's new with V14
- Enablement programs
- Metadata tips

- **HALDB**

- New feature for V13 and V14**
- Enablement tips

## HALDB support for 8GB OSAM data sets

- **Allows for increased database scalability**
  - Increases overall database capacity
  - Reduces the number of data sets that need to be defined
  - Reduces risk of hitting the maximum number of open data sets
- **Support comes with V14**
- **V13 requires the following APAR PI23918**

## Requirements

- **DBRC is required for 8GB OSAM support**
- **Will not work with an unregistered IMS catalog**
  - Only IMS catalogs that are registered with DBRC can take advantage of this
- **New DBRC keyword, **OSAM8G**, used to specify the maximum data set size**
  - INIT.DB DBD(masterdbd) TYPHALDB OLRNOCAP **OSAM8G**
  - CHANGE.DB DBD(masterdbd) OLRNOCAP **OSAM8G**

# Agenda

- **IMS Catalog**

- What's new with V14
- Enablement programs
- Metadata tips

- **HALDB**

- New feature for V13 and V14
- Enablement tips**



## Data sharing environments

- **HALDB 8GB OSAM databases can be used in a data sharing environment if the following conditions are met:**
  - The IMS systems are all on either V14 or V13
  - All of the V13 systems must have APAR PI23918 applied
  - The **MINVERS** value in the RECON data set must be 14.1 or 13.1 with a CDSLID value set to 2
    - CHANGE.RECON MINVERS('14.1')
    - CHANGE.RECON MINVERS('13.1')
    - CHANGE.RECON CDSLID(0002)

## No HALDB Online Reorganization support

- **There is no online reorganization (OLR) support for HALDB 8GB OSAM**
- **Reorganizations should be done with the following offline utilities:**
  - HD Reorganization Unload utility (DFSURGU0)
  - HD Reorganization Reload utility (DFSURGL0)
- **INITIATE OLREORG will fail with a completion code of 1EF**

## No HALDB Alter support

- For similar reasons, there is also no HALDB Alter support with HALDB 8GB OSAM
- If HALDB Alter is important for your shop, then you'll want to stay with the 4GB size limit
- This can be defined with the new DBRC keyword **NOOSAM8G**

# Questions?