Modern IMS Database Management with DDL and IMS Explorer for Development

GSE BeNeLux IMS User Group

Clive Harriss
*IBM Early Experience Program Manager (eXp) – IMS*

clive_harriss@uk.ibm.com

With acknowledgements to Deepak Kohli

© Copyright IBM Corporation 2018
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.

- Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user’s job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.
Statement of Direction

• IBM IMS has evolved into a more dynamic system, based on continuous redesign that will ultimately eliminate the need for system generation. This redesign currently includes dynamic definition for resources such as application programs, databases, routing codes, transactions, OSAM buffer pools, VSAM share pools, MSC physical links, logical links, logical link paths, and remote logical terminals.

• IMS 14 delivered optional support allowing IMS to dynamically manage application control blocks (ACBs). IMS-managed ACBs allows for the use of DDL to replace DBD and PSB generation processes. Databases and program views can be dynamically defined using SQL DDL statements instead of generation utilities, such as ACBGEN.

• IBM intends to require IMS management of ACBs in the future. IMS and the IMS catalog must be set up to support ACB management. IMS provides a utility for this.

• At a later date, after the requirement for IMS-managed ACBs is in place, IBM also intends to remove the generation processes for PSBLIB, DBDLIB and ACBLIB. At that time, the IMS catalog, SQL, and DDL become the interface to IMS database management.

• These planned changes to IBM IMS will enhance availability, free up DBA resources, improve productivity, simplify operations, testing, and debugging, and accelerate time-to-value for new business solutions.

• Statements regarding IBM’s future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
TOPICS

• Data Definition Language (DDL)
  • Requirements to use DDL
  • Value ADD

• DDL: CREATE, ALTER, DROP

• Ways to submit DDL:
  • DDL Batch Utility
  • IMS Explorer for development
WHY USE DDL?
The Problem (AS IS …)

- Coding PSBs, DBDs
  - Assembler Macros (not trivial)
  - PSBGEN, ACBGEN, ACBGEN
  - Proprietary to IMS

- Skills Issue
  - Harder for IMS to play in digital age
  - How do you “Provision an IMS database”
The Solution: Use DDL

- DDL
  - IMS Explorer
    - For Development
- DDL
  - Batch Utility

Diagram:
- DDL to Catalog
- Catalog to IMS
- Staging Directory to IMS
- IMS to Directory

Red X:
- AC-MIB
The Solution

• Use DDL

  • Leverage the Industry standard Data Definition Language (DDL) to affect database and schema changes

  • Exploit DDL authoring tools such as the IMS Enterprise Suite Explorer for Development (E4D) to model database changes and create DDL
    • DDL-authoring tools are prevalent in the market

  • Reduced time and complexity of creating IMS databases

  • Align with industry practices and expectations

  • Provide an audit trail capturing changes made
    • SMF type 29 sub-type 3 for Catalog activity.
Software requirements

- IMS Managed ACBs implemented & enabled
  - IMS Catalog
  - Directory
  - BSDS
  - SCI & OM must be started

- For DDL Tooling to submit DDL:
  - IMS Connect & ODBM must be started

- The User must have Security access to use the IMS Catalog PSB DFSCP001.
IMS specific parameters in DDL

- The DDL standard does not contain all of the options for IMS
  - It is not IMS specific
  - Other DBMSs have similar specific requirements

- Many options existing in the PSBGEN and DBDGEN macros are unique to IMS.
  - e.g., DB Access Types: PHIDAM, HIDAM, PHDAM, etc.

- IMS DDL includes extensions specific to IMS structures to allow more detailed database definitions.
DDL defaults and Enhanced IMS syntax

- All parameters that can be specified in the DBDGEN or PSBGEN macros are **optional parameters** in the IMS Enhanced DDL syntax

- The IMS Enhanced **DDL syntax can be used with existing defaults**
  - Defaults values may be overridden when specified
Some IMS DDL defaults

- CREATE DATABASE
  - DB Access Type will be defaulted to PHIDAM, OSAM
- CREATE TABLE
  - Field positioning is based on the order of the fields defined in the
  - e.g., CREATE TABLE tbl ( flda SHORT, fldb INT, fldc SHORT)

- Decimal field types are defaulted to Packed Decimal
- Defaults are based on existing IMS default values
Standard DDL syntax

**Multiple DDL statements** are needed in order to *describe an IMS database*, such as: database organization type, access method, record mapping and record relationships.

- Database
- Table spaces
- Tables (segments) and columns (field) mappings in a database record.
- Relationships with other databases Primary keys and foreign keys.
- Programview (PSB) describing a program's characteristics.
  - Table columns and fields to be sensitive to
Standard DDL syntax - continued

- Basic DDL statements
  - CREATE resource
    - Define & generate
  - ALTER resource
    - Redefine & regenerate
  - DROP resource
    - Delete

<table>
<thead>
<tr>
<th>IMS</th>
<th>GEN statement</th>
<th>DDL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>DBD</td>
<td>DATABASE</td>
</tr>
<tr>
<td>Segment</td>
<td>SEGM</td>
<td>TABLE</td>
</tr>
<tr>
<td>Dataset</td>
<td>DATASET</td>
<td>TABLESPACE</td>
</tr>
<tr>
<td>Area</td>
<td>AREA</td>
<td>TABLESPACE</td>
</tr>
<tr>
<td>Program</td>
<td>PSBGEN</td>
<td>PROGRAMVIEW</td>
</tr>
<tr>
<td>PCB</td>
<td>PCB</td>
<td>SCHEMA</td>
</tr>
<tr>
<td>Senseg</td>
<td>SENSEG</td>
<td>SENSEGVIEW</td>
</tr>
</tbody>
</table>
IMS DDL syntax – CREATE DATABASE & TABLESPACE Example

CREATE DATABASE cogdbd
ACCESS HDAM OSAM
RMNAME(DFSHDC40 3 3 25);

CREATE TABLESPACE cogdata IN cogdbd
SIZE PRIMARY 8192;
COMMENT ON TABLESPACE cogdata IN cogdbd IS 'Dataset Group 1';
**IMS and DDL value additions**

Adding a new Fast Path DEDB to IMS before CREATE DDL extensions

1. Define your database characteristics (DBD)
2. Compile/link database definitions (DBDGEN)
3. Define your program specifications (PSB)
4. Compile/link your program specifications (PSBGEN)
5. Create ACBs (ACBGEN)
6. Allocate database data sets (VSAM)
7. Define DBRC definitions (INIT)
8. Perform online change to load DEDB related control blocks (MOLC)
9. Format IMS FP databases (Utility)
10. Establish database recovery point by taking an image copy (IC)
11. Issue CREATE PGM|TRAN to create program /transaction for online application (DRD)
12. Start IMS databases, programs (COMMAND)
Adding a new DEDB to IMS with **DDL CREATE DB** Extensons

1. Define your database characteristics (DBD)
2. Compile/link database definitions (DBDGEN)
3. Define your program specifications (PSB)
4. Compile/link your program specifications (PSBGEN)
5. Create ACBs (ACBGEN)
6. Allocate database data sets (VSAM)
7. Define DBRC definitions (INIT)
8. Perform online change to load DEDB-related control blocks (MOLC)
9. Format IMS FP databases (Utility)
10. Start IMS databases, programs (COMMAND)

1. DDL CREATE DATABASE and PROGRAMVIEW for new DEDBs and their PSBs (Replaces 1–9, 12)
2. If database and program runtime attribute definitions do not exist and DRD is enabled, DDL CREATE DATABASE and PROGRAMVIEW will **automatically create** the associated runtime definitions for the database and the program as BMP
3. Issue CREATE PGM|TRAN to create program/transaction for online application
4. Establish database recovery point by taking an image copy (IC)
5. Start IMS databases, programs (COMMAND)
DDL Extensions for Create DBs and PSBs

DFSDFxxx member

*--------------------------------------------------------------------*
* CATALOG SECTION                                                    *
*--------------------------------------------------------------------*
*SECTION=CATALOG
CATALOG=Y
ACBMGMT=CATALOG
ALIAS=DFSC
RETENTION=(VERSIONS=5, DAYS=365)
STORCLAS=BASE
MGMTCLAS=STANDARD
SMSVOLCT=1
SPACEALLOC=(PRIMARY=500 SECONDARY=50)
GURCACHE=5
AUTOCREATE=YES AUTOIMPORT(CREATE)
<SECTION DDL>
DDLHLQ=IMSP
SHARELVL=1
CYLINDERS=250
DATACLAS=NORMIMS
MGMTCLAS=STANDARD
STORCLAS=BASE
VOLUME=NORM01

© Copyright IBM Corporation 2018
IMS and DDL value additions

Simplifies the process of adding new application metadata to the Catalog

- **Without DDL:**
  - The DBD source would have to be enhanced with the **COBOL copybook or PL/I include information** for each segment overlay
  - DBDGEN/ACBGEN needs to be performed
  - Online Change needs to be completed

- **With DDL:**
  - DDL ALTER TABLE statement will automatically add the information to the catalog
  - IMPORT DEFN SOURCE(CATALOG) makes it available online

© Copyright IBM Corporation 2018
RESOURCE: DATABASE
CREATE DATABASE

- The **CREATE DATABASE** statement defines a database. This creates a definition of the database only.
  - Options will vary depending on database type.
- A Created database may be automatically or manually Imported to Active status in the IMS.
- Other definitions usually entered on the DBDGEN are created with separate CREATE resource_name statements.
  - Dataset
  - Area
  - Segment
  - Field
IMS DDL - CREATE DATABASE

- HDAM

```sql
CREATE DATABASE COGDBD
  ACCESS HDAM OSAM
  RMNAME (DFSHDC40 RMANCH 3 RMRBN 3 RMBYTES 25)
  VERSION 'Latest version of COGDBD'
  PASSWDNO
  CCsid 'Cp1047';
```

```sql
DBD NAME=COGDBD,
    ENCODING=Cp1047,
    ACCESS=(HDAM,OSAM),
    RMNAME=(DFSHDC40,3,3,25),
    PASSWD=NO,
    VERSION='Latest version of COGDBD'
```

© Copyright IBM Corporation 2018
The **ALTER DATABASE** statement changes attributes of the database.

The **ALTER DATABASE** keywords are the same as the keywords of **CREATE DATABASE**.

There are no defaults for an **ALTER DATABASE**.
- Any keywords entered will override the existing values.
- Keyword values not entered will remain the same value.

- An Altered database must be manually IMPORTed to Active status in the IMS.
DROP DATABASE

- DATABASE database_name
  - Identifies the database to drop. The name must identify a database that exists to IMS. When a database is dropped, all of its tables and indexes are also dropped.

- A dropped database may be automatically Imported to remove it from IMS.

>>-DROP--DATABASE--database_name-----------------------------<<
RESOURCE: TABLESPACE
CREATE TABLESPACE

- The CREATE TABLESPACE statement defines a group dataset within the database or an area for a DEDB. This is equivalent to the DATASET or AREA statements used in the IMS DBDGEN source.

- The ddnames used on the CREATE TABLESPACE statement must be unique within an IMS system. Non-unique ddnames in two or more DBDs may result in destruction of the database.

- Options vary depending on dataset type.

  - Note:
    - A Tablespace created with a Database may be automatically Imported with the Database to Active status in the IMS.
    - A Tablespace added to an existing Database must be manually Imported.

\[\text{CREATE TABLESPACE} \quad \text{ddname} \quad \text{IN} \quad \text{database_name} \quad | \quad \text{Options} \quad | \quad \text{----} \]
IMS DDL syntax – CREATE TABLESPACE Example

CREATE DATABASE cogdbd
  ACCESS HDAM OSAM
  RMNAME(DFSHDC40 3 3 25);

CREATE TABLESPACE cogdata
  IN cogdbd
  SIZE PRIMARY 8192;
COMMENT ON TABLESPACE cogdata IN cogdbd IS 'Dataset Group 1';
ALTER TABLESPACE

• The ALTER TABLESPACE statement changes attributes of the data set group within the database or an area for a DEDB. This is equivalent to the DATASET or AREA statements as defined in IMS DBDGEN source.

• The ALTER TABLESPACE keywords are the same as the keywords of CREATE TABLESPACE.

• There are no defaults for an ALTER TABLESPACE.
  • Any keywords entered are overrides to the existing values.
  • Keyword values not entered will remain the same value.

• An Altered Tablespace must be manually Imported.

```sql
>>-ALTER--TABLESPACE--ddname---IN--database_name---| Options |----->
```
The **DROP TABLESPACE** statement removes a data set group from the database or an area from a DEDB.

- The only supported keyword is the **IN** keyword to identify which database the tablespace will be removed from.
- A Dropped Tablespace must be manually Imported to remove it from IMS.

```
>>-DROP--TABLESPACE--ddname---IN--database_name----------------><
```
RESOURCE: TABLE
CREATE TABLE

- The **CREATE TABLE** statement defines a segment.

- The Creation of table is always a new resource.

- A Created table is manually **IMPORT**ed to Active status in the IMS.
IMS Segments are defined as a TABLE

- Up to 255 Table entries for each database
- Must be unique within database
- May not be SQL reserved word
- May not begin with DFS
- The order that they are entered is the order they will be defined.
  - e.g. CREATE TABLE tbl ( fldc SHORT, flda INT, fldb SHORT)

Import notes:

- A Table created with a Database may be automatically Imported with the Database to Active status in the IMS.
- A Table added to an existing Database must be manually Imported.
create table table_decimal
(r_number int internalname rnum,
  ll short internalname ll,
  c_decimal decimal(7,2) internalname cdec,
  foreign key references table_root
) in cogdbd.cogdata
internalname tdec
maxbytes 10
minbytes 6
insert logical
delete logical
replace logical
ambiguous insert here;

comment on table table_decimal in cogdbd is 'This describes table TDEC.';
The **ALTER TABLE** statement changes attributes of the data set group within the database or an area for a DEDB. This is equivalent of a SEGMENT as defined in IMS DBDGEN source.

- The **ALTER TABLE** keywords are the same as the keywords of CREATE TABLE.

- There are no defaults for an **ALTER TABLE**.
  - Any keywords entered are will override the existing values.
  - Keyword values not entered will remain the same value.

- An Altered Table manually Imported.
IMS DDL – Alter Table

Original DATABASE DBD

```
ALTER DATABASE COGDBD;
ALTER TABLE root
IN COGDBD
MAXBYTES 28
ADD COLUMN New_Field_01 INTERNALNAME newfld01 DOUBLE START 21;
```
IMS DDL syntax – DROP TABLE

• Segments in an hierarchy can be dropped by a single DROP TABLE statement.

• The DROP TABLE statement will drop child segments also.

>>DROP--TABLE--table_name-IN-database---------------------------------><
- CREATE PROGRAMVIEW is the equivalent of a PSB statement.
- SCHEMA is equivalent to a PCB.
- SENSEGVIEW is the equivalent of a SENSEG.
- DDL syntax has been Enhanced for users to write their own PSBs instead of relying on system defaulted ones.
• Nested in the CREATE PROGRAMVIEW must be one or more CREATE SCHEMA statements. SCHEMA statements describe the PCBs.

• Nested in each CREATE SCHEMA must be one or more CREATE SENSEGVIEW statements to describe the SENSEGs.

• Nested in each SENSEGVIEW may be SENFIELDs. The ordering in which the SCHEMA and SENSEGVIEW statements are specified matters. The order that they are entered is the order they will be defined.
  • e.g. CREATE SENSEGVIEW segment (fldc SHORT, flda INT, fldb SHORT)

• A Program view may be automatically Imported to be active in an IMS.
CREATE PROGRAMVIEW (PSB) Example

CREATE PROGRAMVIEW applpgm3 (  
   CREATE SCHEMA TP pcb01 USING output1,  
   CREATE SCHEMA TP pcb02 USING output2,  
   CREATE SCHEMA pcb03 USING partmstr AS pcb03 (  
      CREATE SENSEGVIEW partmast WITH PROCOPT 'A',  
      CREATE SENSEGVIEW cpws WITH PROCOPT 'A'  
   ) PROCOPT 'A',  
   CREATE SCHEMA GSAM pcb04.report PROCOPT 'LS',  
   ) LANGCOBOL;
CREATE PROGRAMVIEW - (PSB) Example

DFSIVP2  PSBSOR:
  PCB  TYPE=TP,MODIFY=YES
  PCB  TYPE=DB,DBDNAME=IVPDB2,PROCOPT=A,KEYLEN=10
  SENSEG NAME=A1111111,PARENT=0,PROCOPT=A
  PSBGEN LANG=ASSEM,PSBNAME=DFSIVP2
END

CREATE PROGRAMVIEW DFSIVP2 (  
  CREATE SCHEMA TP pcb01 MODIFYYES,
  CREATE SCHEMA pcb02 USING IVPDB2 (  
    CREATE SENSEGVIEW A1111111 WITH PROCOPT 'A',
    ) PROCOPT 'A',
  ) LANGASSEM;
The ALTER statement does not apply to a PROGRAMVIEW.

Altering a PROGRAMVIEW is performed by:

- **DROP PROGRAMVIEW**
  - Must identify an existing defined program.

- **CREATE PROGRAMVIEW**
  - With the new information for the Application Program.
Drop PROGRAMVIEW

- The DROP statement removes a resource from IMS. Except for storage groups, any resources that are directly or indirectly dependent on that resource are deleted.

>>-DROP--PROGRAMVIEW--psb_name-----------------><
COMMENT ON Syntax

This statement provides optional user comments.

- Equivalent to the REMARKs keyword on the DBD / PSB source.
  - Optional user comments. A 1- to 256-character string enclosed in single quotation marks. The value specified cannot contain the following characters:
    - Less than (< ) symbols.
    - Greater than ( >) symbols.
    - Ampersands (&).
    - Double quotation marks.
    - Single quotation marks, except when they are used to enclose the full comment string. The following examples show correct and incorrect usages of single quotation marks:
      - CORRECT
        IS 'These remarks apply to the XYZ application'
      - INCORRECT
        IS 'These remarks apply to the 'XYZ' application'
Batch DDL Utility architecture
Batch DDL Utility Sample JCL

```plaintext
//IMSSAMPL JOB (999,XXX),'JAVA BPXBATCH',CLASS=A,MSGLEVEL=(1,1),
//  MSGCLASS=E,REGION=0M,NOTIFY=&SYSUID
// SET P1='com.ibm.ims.jdbc.batch.BatchUtil'
//JAVA JVM EXEC PGM=JVMLDMxx,REGION=0M,
//   PARM="/ &P1"
//STEPLIB DD DISP=SHR,
//   DSN=USER.CUSTOM.JZOS.LOADLIB
//SYSPRINT DD SYSPOUT=*
//SYSPUT DD SYSPOUT=*
//STDOUT DD SYSPOUT=*
//STDERR DD SYSPOUT=*
//CEEDUMP DD SYSPOUT=*
//ABNLIGNR DD DUMMY
//MAINARGS DD *
//IMSSQL DD DISP=SHR,
//   DSN=MYPDS(MYSCRIPT)
//STDENV DD *

export JAVA_HOME=myJavaHomePath
export PATH="/bin:"${JAVA_HOME}="/bin
LIBPATH="/lib:/usr/lib:"${JAVA_HOME}="/bin
export LIBPATH="$LIBPATH"
APP_HOME=${JAVA_HOME}
CLASSPATH="$APP_HOME:"${JAVA_HOME}="/lib:"${JAVA_HOME}="/lib/ext
CLASSPATH="$CLASSPATH":myLibPath/imsudb.jar
```
Batch DDL Utility syntax

• The input statements supported by the Batch DDL Utility include all supported SQL and DDL statements by the IMS JDBC driver.
• The following additional statements are supported:
  CONNECT [JDBC URL];
  This will create a JDBC connection to the IMS system using the specified JDBC URL.
  COMMIT;
  This will commit work on the open connection.
  ROLLBACK;
  This will rollback work on the open connection.
  DISCONNECT;
  This will disconnect the current connection.

• Statements must be delimited by a semi-colon.
CONNECT jdbc:ims://myConnectServer:5555/DFSCP001;

CREATE DATABASE DBFS9974 ACCESS HDAM VSAM CCSID 'Cp1047' DATA CAPTURE CHANGES (HELLO CNOPATH DATA INPOS PATH CKEY LOG, C12 CPATH DATA INPOS PATH CKEY LOG) RMNAME (DFSHDC40 RMBYTES 1 RMANCH 1 RMRBN 1) VERSION 'GOOD VERSION' PASSWDNO DATXEXITYES;

CREATE TABLE HOSP0000 (COLUMN1 DECIMAL(5,2) INTERNALNAME COLUMN1 TYPE C BYTES 10 START 1) AMBIGUOUS INSERT LAST IN DBFS9974;

CREATE TABLESPACE tb1 IN DBFS9974 SEARCHA 2 FREESPACE 99 FREEBLOCK 100 SIZE PRIMARY 28672 SCAN 0 BLOCK PRIMARY 32768;

COMMIT;
DISCONNECT;
## Batch DDL Utility error codes

The following error codes will be thrown by the JZOS Batch Launcher:

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Description</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>There was a connection error to IMS Connect</td>
<td>Verify that the connection parameters are correct</td>
</tr>
<tr>
<td>12</td>
<td>There was an error with the SQL statement execution. All work will be rollback to the prior commit point.</td>
<td>Verify that the SQL statement is valid</td>
</tr>
<tr>
<td>13</td>
<td>There was an issue with the commit</td>
<td>Check the JDBC error messages</td>
</tr>
<tr>
<td>14</td>
<td>There was an issue with the rollback</td>
<td>Check the JDBC error messages</td>
</tr>
<tr>
<td>15</td>
<td>There was an issue cleaning up the connection</td>
<td>Check the JDBC error messages</td>
</tr>
<tr>
<td>16</td>
<td>An invalid command was specified</td>
<td>Verify that valid commands were provided in the input file</td>
</tr>
</tbody>
</table>
Batch DDL Utility

The Batch DDL Utility is used to invoke DDL statements via JCL on the z platform

Requirements:
• The utility is bundled in the IMS JDBC driver (imsudb.jar)
  - APAR PI30848
• Uses IMS JDBC Type-4 connections to invoke SQL statements
  - IMS Connect, ODBM, SCI
• IBM Java for z/OS (JZOS) Batch Launcher
IMS Explorer for Development

- The IMS Explorer for Development is a tool to help with database visualization and querying.
IMS Explorer for Development – DDL creation

- The IMS Explorer for Development has Enhanced DDL editing and generation features
  - A full text DDL editor
    - users can manually write their own DDL scripts
  - A graphical interface for Creating, Altering & Dropping DDL resources
- The generated DDL uses the enhanced IMS DDL syntax
IMS Explorer for Development – DDL creation
IMS Explorer for Development – DDL creation
IMS Explorer for Development – DDL creation

CREATE TABLE PARTROOT (  
  PART_NO_EDIT CHAR(17) NOT NULL,  
  REJECT_CODE CHAR(1) NOT NULL,  
  PART_NO_EDIT_NEW CHAR(17) NOT NULL,  
  FILL_0 CHAR(4) NOT NULL  
);  

CREATE TABLE BACKORDR (  
  PARTROOT_PART_NO_EDIT CHAR(17) NOT NULL,  
  STOKSTAT_STOCKKEY CHAR(16) NOT NULL,  
  BACKKEY CHAR(10) NOT NULL,  
  WORK_ORDER CHAR(8) NOT NULL,  
  WO_QTY DECIMAL(8, 1) NOT NULL,  
  FILLER1 CHAR(2) NOT NULL,  
  FILLER2 CHAR(33) NOT NULL  
);  

CREATE TABLE STAINF (  
  <
IMS Explorer for Development - DDL validation

The IMS Explorer for Development provides additional validation prior to submitting a DDL script

- Speeds up database change development as syntax errors are detected sooner

- Warns the user if the DDL script has changes that requires user intervention
  - For example: Unload / Reload operations

- Alerts the user if the DDL script is long running vs short running
  - Long running: Creating a new database resource
  - Short running: Adding application metadata to an existing table

- Provides a way to view the IMS DDL audit trail for database change history
Dynamic Database Definition

Benefits

• Dynamic implementation of IMS control blocks.
  - An alternative to PSBGEN, DBDGEN, and ACBGEN processes.

• IMS will load from the catalog where changes are made through DDL.

• Use the industry standard DDL to define and modify database and schema creation and changes.

• Exploit DDL authoring tools such as the IMS Enterprise Suite Explorer for Development (E4D) to model database changes and create DDL.
  - DDL-authoring tools are prevalent in the market.

• Provide an audit trail capturing changes made.

• Align with industry practices and expectations.
Thank You!
Notice and disclaimers

- Copyright © 2018 by International Business Machines Corporation (IBM). No part of this document may be reproduced or transmitted in any form without written permission from IBM.
- U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.
- Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. This document is distributed “as is” without any warranty, either express or implied. In no event shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity. IBM products and services are warranted according to the terms and conditions of the agreements under which they are provided.
- IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply.
- Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.
- Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.
- References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.
- Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.
- It is the customer’s responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer’s business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer is in compliance with any law.

© Copyright IBM Corporation 2018
Notice and disclaimers cont.

- Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a particular, purpose.**

- The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.


- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both. Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.