



Introducing Data Definition Language (DDL) for IMS Metadata #17764

Dennis Eichelberger
deichel@us.ibm.com

IMS Support – zGrowth Washington System Center



#SHAREorg



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

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



Acknowledgements and Disclaimers

Availability. References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates.

The workshops, sessions and materials have been prepared by IBM or the session speakers and reflect their own views. They are provided for informational purposes only, and are neither intended to, nor shall have the effect of being, legal or other guidance or advice to any participant. While efforts were made to verify the completeness and accuracy of the information contained in this presentation, it is provided AS-IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this presentation or any other materials. Nothing contained in this presentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

© **Copyright IBM Corporation 2015. All rights reserved.**

— ***U.S. Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.***

IBM, the IBM logo, ibm.com, z/OS, IMS, ODM, DB2, CICS, WMQ, WAS, WOLA, RAA, RAD, RDz, RTC, are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or TM), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at

•“Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml

•Other company, product, or service names may be trademarks or service marks of others.

Dynamic Database Definition

- Data Definition Language – DDL

Generically, Data Definition Language refers to any formalized language that describes data, information structures of a data construct or access to those structures.

The language uses a collection of command verbs to manipulate control schemas. Schemas can be manipulated: they may be added, changed or deleted during the life of the formalized data construct.

DDL is often considered a subset of SQL.

Data Definition Language - DDL

- Agenda
- Usage
- Implementation
- Samples

Dynamic Database Definition

- **Data Definition Language – DDL**

All database systems have a uniqueness to the data structures required in their specific environment. While each system uses the industry standard DDL to define its structures, because of this uniqueness, each system also has its own syntax extensions that are specialized to their own DBMS environment.

IMS is no different in this respect.

IMS 14 provides for using the standard DDL.

IMS 14 also includes extensions specific to IMS structures to allow more detailed database definitions. These extensions closely match the current non DDL definitions used in IMS.

Dynamic Database Definition

- IMS 14 provides for the IMS Catalog to be the central focal point for IMS database and program changes. The trusted Repository.
 - Leverage the Industry standard Data Definition Language (DDL) to affect database and schema changes
 - Provide an audit trail capturing changes made
 - Log record x'2A' for DDL Information
 - Log record x'2B' for IMS Catalog changes
 - SMF 29 type 3 for Catalog activity
 - IMS will load from the catalog where changes are made through DDL
 - Dynamic implementation of IMS control blocks
 - Reduced time and complexity of creating IMS databases
 - An alternative to proprietary PSBGEN, DBDGEN, and ACBGEN processes
 - Use industry standard DDL for database and schema implementations
 - 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
 - DDL – authoring tools use IMS Universal drivers or supporting tooling
 - Align with industry practices and expectations

Dynamic Database Definition

- Simplify database creation and modifications
- Use DDL to manipulate IMS databases
 - Define
 - Modify
 - Delete
- Bypass the database generation related processes
 - DBDGEN
 - PSBGEN
 - ACBGEN
- Simplify activation of resource changes in IMS
 - Changes may be activated when committed

IMS and DDL value additions

- Simplifies the process of adding a new Database resource
 - Without DDL:
 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. Perform ACBGEN
 6. Allocate database data sets
 7. Define DBRC definitions
 8. Define IMS database for dynamic allocation (MDA)
 9. Define database and program online resource definitions:
 10. Issue CREATE DB/PGM to create database and program (DRD)
 11. Define database and program online resource definitions:
 - CREATE DB / PGM command to create the database and program (DRD)
 - Specify system definition macros and Online Change
 12. Perform online change to load IMS application-related definitions (MOLC)
 13. Initialize and Load IMS databases
 14. Establish database recovery point by taking an image copy (IC)
 15. Start IMS resources - Databases, Programs, Transactions

IMS and DDL value additions

- Simplifies the process of adding a new Database resource
 - With DDL:
 1. Generate the DDL statements for the database and program views.
 2. Submit DDL statements
 - DDL changes will be held in IMS a pending state.
 3. Allocate your database data sets.
 4. Define DBRC definitions.
 5. Define IMS database for dynamic allocation (MDA).
 6. Initialize and Load IMS databases as needed.
 7. Establish database recovery point by taking an image copy (IC).
 8. Define database and program online resource definitions:
 - CREATE DB / PGM command to create the database and program (DRD).
 - Specify system definition macros and Online Change
 9. Activate the database and program definitions.
 - IMPORT DEFN SOURCE(CATALOG)
 10. Start IMS resources - Databases, Programs, Transactions, Routing Codes.

Dynamic Database Definition - Restrictions

- Restrictions
 - Manual steps will be required for some database changes
 - User makes DDL change to the catalog
 - Perform manual steps such as reorganization, image copy
 - IMPORT the changes from the catalog into IMS
 - User makes DDL change to the catalog
 - Manual steps will be required for some database changes
 - The User must have Security access to DDL PSB DFSCP001

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 may automatically add the information to the catalog and make it available online
- DDL allows IMS to conform to industry standards for database definitions

Implementing DDL

- New IMS Catalog user:
 - Build the IMS catalog database.
 - Build the IMS management of ACBs.
 - Tell IMS that the catalog should be used instead of the existing ACBLIB
 - Tell IMS to use the catalog and directory by specifying the catalog is enabled in the DFSDFxxx PROCLIB member or the equivalent user exit (DFS3CDX0).

- Existing IMS Catalog user:
 - Build the IMS management of ACBs.
 - Tell IMS that the catalog should be used instead of the existing ACBLIB
 - Tell IMS to use the catalog and directory by specifying the catalog is enabled in the DFSDFxxx PROCLIB member or the equivalent user exit (DFS3CDX0).

Update the existing records in the IMS catalog.

DDL Activation

- DFSDFxxx syntax

```
>>-----+-----+-----<<
      |                               | -Y- |
      | -DROPAUTOCLEANUP= (-+-N-+-----) - |
```

```
>>-----+-----+-----<<
      |                               | -Y- |
      | -AUTOCREATE= (-----+-N-+-----) -- |
```

```
>>-----+-----+-----<<
      |                               | -ACBLIB-- |
      | -ACBMGMT= (-+-CATALOG-+-----) - |
```

- DROPAUTOCLEANUP= N|Y|NO|YES
- AUTOCREATE= N|Y|NO|YES
- The Catalog Database access default is changed from READ to UPDATE
ACCESS=UPDATE

DDL Activation

- DROPAUTOCLEANUP= N|Y|NO|YES
 - Specifies that IMS will automatically perform the removal of databases and related resources at the commit of DDL DROP statement. When this parameter is omitted, the default is to not perform automatic removal of the database and related resources.
 - The steps of automatic database definition include, but are not limited to the following:
 - Deletion of the database data set.
 - Deletion of the database from the RECON registration.
 - Deleting the database and related resources to IMS.

DDL Activation

- **AUTOCREATE= N|Y|NO|YES**
 - Specifies that IMS will automatically perform database definition steps during IMPORT. When this parameter is omitted, the default is to not perform automatic database definition.
 - The steps of automatic database definition include, but are not limited to the following:
 - Database data set creation
 - Database registration with DBRC
 - Dynamic data set allocation member (DFSMDA) creation
 - Database load operation
 - Database initialization
 - Defining the database and related resources to IMS

DDL Activation

- Upon receiving the DDL statements
 - If directed, IMS can activate certain changes to database or program view definitions automatically
 - Changes not activated automatically are held pending activation and can be activated via the IMPORT command
 - `IMPORT DEFN TYPE(DIRECTORY)`
- No need to use Online Change or recycle IMS system to activate an ACB

DDL Activation

- During the IMPORT from the IMS catalog IMS will do the following:
 - Automatically quiesce and stop all activity on resources impacted by any catalog changes.
 - Remove all old copies of the resources from memory.
 - This will cast out DMBs and PSBs from their respective pools.
 - Coordinate the change across the IMSplex for sharing IMS catalogs.
 - Apply the pending changes from the IMS catalog into the directory.
 - Make all of the previously quiesced resources available after the change to the catalog has been activated.
- IMS Resources are now available

DDL Activation - Impact

- Some DBRC commands are enhanced to either specify a catalog name or display the current default IMS catalog
- New messages may need to be monitored
- IMS Utility changes to access the catalog
- Exit DFS3CXD0 may need modification

Some DDL usage scenarios

- Tools can generate DDL based on Database metadata retrieved through standard discovery mechanisms like JDBC Database Metadata.
 - e.g., IMS Explorer for Development and Optim Data Studio can generate DDL for an IMS database using JDBC discovery.
- Tools can generate DDL for storing its own metadata repository
 - e.g., Cognos generates DDL for its own content store that holds information on created reports.
- Business Analytics can take existing data and enhance that data through analytics. The enhanced data can then be written back into the database for later use.

Resource Name Comparison

- IMS will support the standard DDL syntax for CREATE, ALTER and DROP of Databases and Tables
 - Consume the standard DDL generated without IMS affinity
- The equivalent IMS to DDL statements are shown in the Table here.

IMS	GEN statement	DDL
Database	DBD	DATABASE
Segment	SEGM	TABLE
Dataset	DATASET	TABLESPACE
Area	AREA	TABLESPACE
Program	PSBGEN	PROGRAMVIEW
PCB	PCB	SCHEMA
Senseg	SENSEG	SENSEGVIEW

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 provides defaults through a descriptor stored in the IMS catalog
- This descriptor is based on IMS recommended defaults
 - Modifiable to user requirements
 - The recommended defaults can be seen in the syntax train tracks

DDL defaults and Enhanced IMS syntax

- Overriding the IMS system defaults
 - Enhanced DDL 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
- DDL syntax has also been Enhanced to fully specify their PSB definitions
 - If a PSB is defined in the same commit scope as a CREATE or ALTER database call, no default PSB will be generated

Standard DDL syntax - continued

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

IMS	GEN statement	DDL
Database	DBD	DATABASE
Segment	SEGM	TABLE
Dataset	DATASET	TABLESPACE
Area	AREA	TABLESPACE
Program	PSBGEN	PROGRAMVIEW
PCB	PCB	SCHEMA
Senseg	SENSEG	SENSEGVIEW

Standard DDL syntax - continued

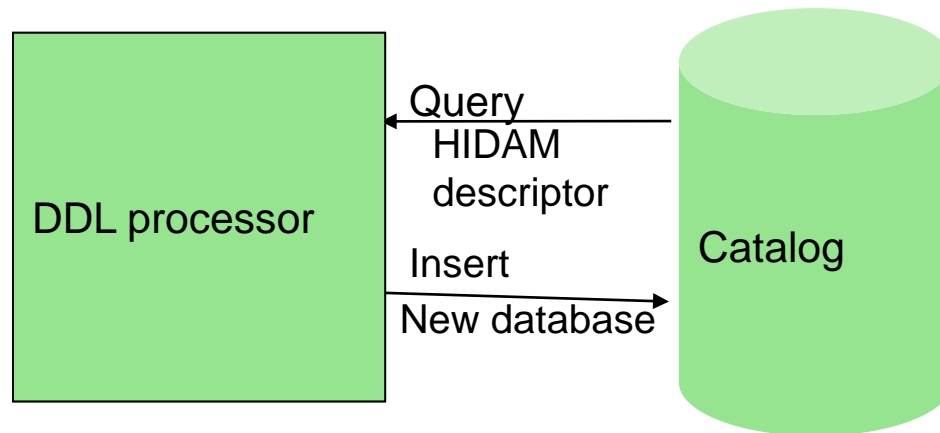
Multiple DDL statements are needed in order to describe the make-up of a database, such as: databases organization and access method

- Database
- Table spaces
- Tables and columns mappings in a database record
- Relationships with other databases Primary keys and foreign keys
- Programview (PSB) describing a programs characteristics
 - Tables columns and fields

Creating resources based on default descriptor

- A resource can be created with values stored in the Catalog descriptors.
- For example, when defining a database of a specific type, the catalog is queried for the descriptor of that type to obtain and use the defaults contained in the descriptor. The user can issue the following statement:

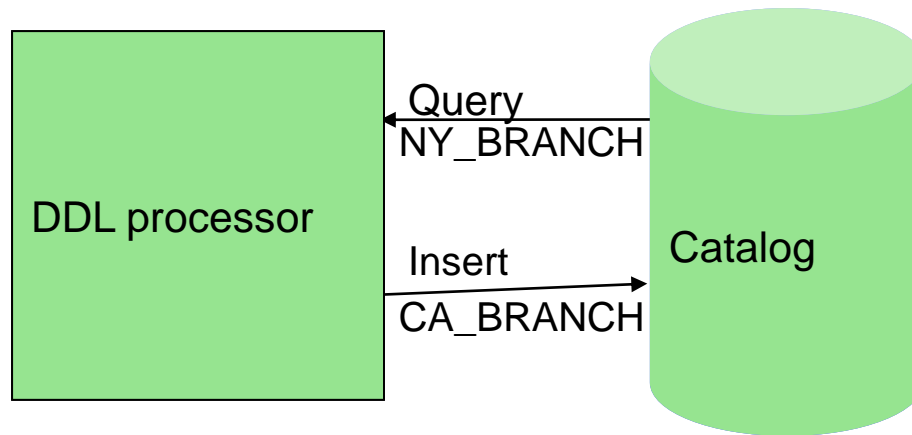
– CREATE ...new_database.. ACCESS HIDAM



Creating resources based on another resource

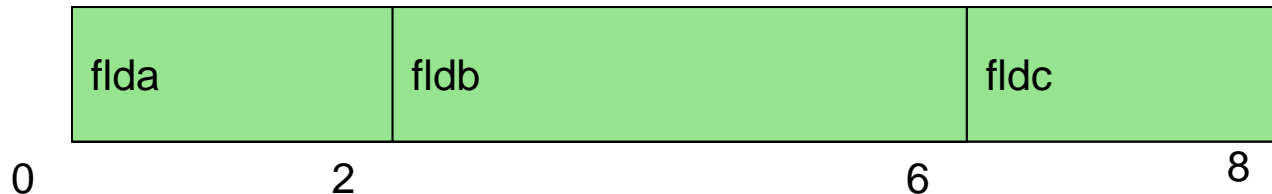
- A resource can be created with defaulted values from another resource
- For example, if a bank wants to create a database for their California branches similar to their existing New York branches they can issue the following statement:

– `CREATE ...resource. CA_BRANCH LIKE NY_BRANCH`



Some IMS DDL default values

- 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

Defaults for defining new Resources

Pre-defined defaults

- A pre-defined set of system defaults for IMS resources are defined as descriptors and stored in the IMS catalog.
- Keywords not explicitly defined for individual resources during a CREATE will default to the descriptor values.
- These defaults can be overwritten by the user.
- Defaulted descriptors
 - DATABASE
 - TABLESPACE
 - TABLE
 - PROGRAMVIEW
 - SENSEGVIEW

Defaults for defining new Resources

Resources

- Resource Definition defaults are as defined in the IMS Database Administration manual.
- A resource can be created based on a default descriptor residing in the IMS catalog.
 - Descriptor is user modifiable
- A resource can be created by using an existing resource as a descriptor. This is performed by using the LIKE keyword in conjunction with the CREATE **resource** statement. The resource values are copied from the existing resource if not explicitly defined on the CREATE statement. The new resource inherits the values from the named 'like' resource.
- A resource can be created from scratch by specifying all attributes for the resource being defined.

Creating resources

- *CREATE resource_name*
- A resource can be created based on a default descriptor residing in the IMS catalog.
- A resource can be created by using an existing resource as a descriptor. This is performed by using the LIKE keyword in conjunction with the **CREATE resource** statement. The resource values are copied from the existing resource if not explicitly defined on the CREATE statement. The new resource inherits the values from the named 'like' resource. Any explicitly specified values override the resource referenced as 'like' in the CREATE.
- A resource can be created from scratch specifying all the needed attributes for the resource being defined.
- A Created resource may be automatically or manually Imported to Active status in the IMS.
- A Created resource is always a new resource. DRD or Sysgen needed to complete.

Altering resources

- ALTER *resource_name*
 - Resource must currently exist in the catalog.
 - Resource defaults do not apply.
 - The alter is applied only to the value(s) being altered.
 - Values that are not specifically altered remain the same.
 - An Altered resource must be manually Imported to Active status in the IMS.

Dropping resources

- DROP *resource_name*
 - Resource must currently exist in the catalog
 - Dropping some resources may cause an ALTER to occur to an existing resource.
 - e.g. the Drop of a TABLESPACE within a DATABASE
 - A Dropped resource may be automatically or manually Imported to remove its Active status in the IMS.
 - Dependent upon the type of resource being dropped

Data Definition Language - DDL

- Examples

CREATE DATABASE

- The CREATE DATABASE statement defines a database. This creates a definition of the database only.
 - Options will vary depending on database type.
- The Creation of database is always a new resource. The database will always be created at Version 0.
- 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

DDL syntax

■ CREATE DATABASE – HDAM

```
>>-CREATE--DATABASE--database_name--+-----+----->
                                     '-LIKE--resource_name-'

                                     .-VSAM-.

>--ACCESS--HDAM--+-----+-----+-----+----->
                '-OSAM-' |           .-Cp1047---. |
                '-CCSID--+encoding+--'

                                     .-PASSWDNO--.   .-DATXEXITNO--.
>--+-----+-----+-----+-----+-----+----->
    '-VERSION--version_identifier-' '-PASSWDYES-'   '-DATXEXITYES-'

>--+-----+-----+-----+-----+-----+----->
    '-RMNAME (-mod--+-----+-----+-----+-----+--)-'
    '-RMANCH-anch-' '-RMRBN-rbn-' '-RMBYTES-bytes-'

>--+-----+-----+-----+-----+-----+-----><
    |           .-NONE------. |
    |           |           | |
    '-DATA CAPTURE--+CHANGES-(-exit_changes)-+--'
```

IMS DDL - CREATE DATABASE

■ HDAM

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

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

IMS DDL syntax – ALTER DATABASE

- 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 are 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.

IMS DDL – Alter with Database Versioning

Original DATABASE DBD (generates version 0):

```


DBD      NAME=COGDBD,                                C
          ENCODING=Cp1047,                            C
          ACCESS=(HDAM,OSAM) ,                        C
          RMNAME=(DFSHDC40,3,3,25) ,                  C
          PASSWD=NO
DATASET  DD1=COGDATA,                                C
          DEVICE=3390,                                C
          SIZE=(8192)
SEGM     NAME=ROOT,                                  C
          PARENT=0,                                    C
          BYTES=(20) ,                                C
          RULES=(LLL,HERE)
FIELD    NAME=(ROOTKEY,SEQ,U) ,                      C
          BYTES=12,                                    C
          START=1,                                     C
          TYPE=C,                                      C
          DATATYPE=CHAR
FIELD    NAME=TABTYPE,                               C
          BYTES=8,                                     C
          START=13,                                    C
          TYPE=C,                                      C
          DATATYPE=CHAR
  
```

IMS DDL – Alter with Database Versioning

Original DATABASE DBD (generated version 0):

```

DBD      NAME=COGDBD,                                C
          ENCODING=Cp1047,                            C
          ACCESS=(HDAM,OSAM) ,                        C
          RMNAME=(DFSHDC40,3,3,25) ,                  C
          PASSWD=NO
DATASET  DD1=COGDATA,                                C
          DEVICE=3390,                                C
          SIZE=(8192)
SEGM     NAME=ROOT,                                  C
          PARENT=0,                                    C
          BYTES=(28) ,                                C
          RULES=(LLL,HERE)
FIELD    NAME=(ROOTKEY,SEQ,U) ,                      C
          BYTES=12,                                    C
          START=1,                                    C
          TYPE=C,                                      C
          DATATYPE=CHAR
FIELD    NAME=TABTYPE,                              C
          BYTES=8,                                    C
          START=13,                                   C
          TYPE=C,                                      C
          DATATYPE=CHAR
  
```



IMS DDL – Alter DATABASE

- Database Version specified

```
ALTER DATABASE COGDBD
  DBVER 6;
ALTER TABLE root
  IN COGDBD
  MAXBYTES 28
  ADD COLUMN New_Field_01 INTERNALNAME newfld01 DOUBLE START 21 ;
```

- Database Version is generated as next in sequence

```
ALTER DATABASE COGDBD
  DBVER AUTO;
ALTER TABLE root
  IN COGDBD
  MAXBYTES 28
  ADD COLUMN New_Field_01 INTERNALNAME newfld01 DOUBLE START 21 ;
```

- Current Database Version is updated

```
ALTER DATABASE COGDBD
  DBVER CURRENT;
ALTER TABLE root
  IN COGDBD
  MAXBYTES 28
  ADD COLUMN New_Field_01 INTERNALNAME newfld01 DOUBLE START 21 ;
```


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.
 - **AUTODROPTRANS**
 - Optional flag to drop transactions and program views of the database.
 - The PSBs are only be dropped only if all PCBs referenced the Database resource being dropped. Otherwise the PSB is altered so only the affected PCBs are removed.
- An dropped database may be automatically Imported to remove it from IMS.

```
>>-DROP--DATABASE--database_name---+-----+-----><  
                                     '-AUTODROPTRANS-'
```

- Resources may need to be removed from DRD or Sysgen

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

DDL syntax

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:
 - 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.

```
>>-CREATE--TABLESPACE--ddname---IN--database_name---| Options |---->
```

IMS DDL syntax – CREATE TABLESPACE Example

```

DBD      NAME=COGDBD,                                C
          ENCODING=Cp1047,                            C
          ACCESS=(HDAM,OSAM),                          C
          RMNAME=(DFSHDC40,3,3,25),                    C
          PASSWD=NO
DATASET  DD1=COGDATA,                                C
          DEVICE=3390,                                C
          SIZE=(8192),                                C
          REMARKS='Dataset Group 1'
SEGM     NAME=ROOT,                                  C
          PARENT=0,                                    C
          BYTES=(20),                                  C
          RULES=(LLL,HERE)
FIELD    NAME=(ROOTKEY,SEQ,U),                        C
          BYTES=12,                                    C
          START=1,                                     C
          TYPE=C,                                       C
          DATATYPE=CHAR
FIELD    NAME=TABTYPE,                                C
          BYTES=8,                                     C
          START=13,                                    C
          TYPE=C,                                       C
          DATATYPE=CHAR

```

```

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';

```

DDL syntax

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.

```
>>-ALTER--TABLESPACE--ddname---IN--database_name---| Options |---->
```

IMS DDL syntax – CREATE TABLE Example

SEGM	NAME=TDEC,	C
	PARENT=ROOT,	C
	BYTES=(10,6),	C
	REMARKS='This describes table TDEC.',	C
	RULES=(LLL,HERE)	
FIELD	NAME=RNUM,	C
	BYTES=4,	C
	START=3,	C
	DATATYPE=INT	
FIELD	NAME=LL,	C
	BYTES=2,	C
	START=1,	C
	DATATYPE=SHORT	
FIELD	NAME=CDEC,	C
	EXTERNALNAME=CDECIMAL,	C
	BYTES=4,	C
	START=7,	C
	DATATYPE=DECIMAL(7,2)	

```

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.';

```

IMS DDL syntax – ALTER TABLE

- The ALTER TABLE statement changes attributes of the table 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 is manually Imported.

IMS DDL syntax – DROP TABLE

- All the tables (segments) in an hierarchy can be dropped by a single DROP TABLE statement.
- The DROP TABLE statement will drop child segments.
- A Dropped Table may be automatically Imported to activate the removal of the segments from the active IMS database.

```
>>-DROP--TABLE--table_name-----><
```

- Resources may need to be removed from DRD or Sysgen

Program View Generation

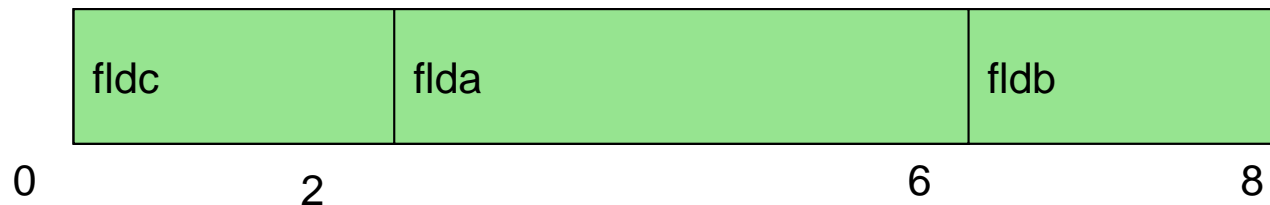
- On a CREATE or ALTER *resource* command
 - A new PSB is generated for the user in the IMS catalog
- This PSB is only be authenticated to the user that issued the CREATE or ALTER command
 - Additional users can be added through the normal Security procedures (SAF)
- PSB naming convention, possible generation options:
 - Name may be based off of the DBD resource name
 - DBD: MYDBD
 - PSB: MYDBDP
 - Name is based off of a user supplied prefix
 - PREFIX: PSBPFX
 - PSB: PSBPFX1

CREATE PROGRAMVIEW Syntax

- 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.

Program View Generation

- 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 Syntax

- PROGRAMVIEW is equivalent of a PSB

```

                                .- ,----- .
                                V             |
>>---CREATE--PROGRAMVIEW--psb_name----- (---| schema_statement |---)--->

                                .-CMPATNO--.    .-GSROLBOKNO--.    .-OLICNO--.
>---+-----+-----+-----+-----+-----+-----+-----+----->
    | -DBLEVELCURR- |    ' -CMPATYES- '    ' -GSROLBOKYES- '    ' -OLICYES- '
    ' -DBLEVELBASE- '

>---+-----+-----+-----+-----+-----+-----+-----+----->
    ' -LOCKMAX-n- '    ' -IOASIZE-n- '    ' -MAXQ-n- '    ' -SSASIZE-n- '

>---+-----+-----+-----+-----+-----+-----+-----+-----><
    | -LANGASSEM-- |    |                                .-WTORNO--. |
    | -LANGCOBOL-- |    ' -IOEROPN-n-- +-----+----- '
    | -LANGPLI---- |    ' -WTORYES- '
    | -LANGPASCAL- |
    ' -LANGJAVA--- '

```


CREATE PROGRAMVIEW (PSB) Example

```
*      DB   PCB NUMBER 1      DB   DEDBJN21
*****
PCB      TYPE=DB,DBDNAME=DEDBJN21, POS=M, PROCOPT=A, KEYLEN=26,          C
          PCBNAME=PCB01, EXTERNALNAME=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
SENSEG   NAME=ARRAY, PARENT=HOSPITAL
SENSEG   NAME=STRUCT, PARENT=HOSPITAL
SENSEG   NAME=REDEFINE, PARENT=HOSPITAL
SENSEG   NAME=MAP, PARENT=HOSPITAL
SENSEG   NAME=EXFLDSEG, PARENT=HOSPITAL
SENSEG   NAME=NUMSEGM, PARENT=HOSPITAL
```

```
CREATE PROGRAMVIEW bmp255 (
  CREATE SCHEMA pcb01 USING dedbjn21 AS pcb01 (
    CREATE SENSEGVIEW hospital,
    CREATE SENSEGVIEW payments WITH PROCOPT 'GI',
    CREATE SENSEGVIEW ward,
    CREATE SENSEGVIEW patient,
    CREATE SENSEGVIEW illness,
    CREATE SENSEGVIEW treatmnt,
    CREATE SENSEGVIEW doctor,
    CREATE SENSEGVIEW billing,
    CREATE SENSEGVIEW array,
    CREATE SENSEGVIEW struct,
    CREATE SENSEGVIEW redefine,
    CREATE SENSEGVIEW map,
    CREATE SENSEGVIEW exfldseg,
    CREATE SENSEGVIEW numsegm
  ) PROCOPT 'A'
  POSMULTI
```

CREATE PROGRAMVIEW (PSB) Example

```
PCB      TYPE=TP, NAME=OUTPUT1
PCB      TYPE=TP, NAME=OUTPUT2
PCB      TYPE=DB, DBDNAME=PARTMSTR, PROCOPT=A, KEYLEN=100
SENSEG   NAME=PARTMAST, PARENT=0, PROCOPT=A
SENSEG   NAME=CPWS, PARENT=PARTMAST, PROCOPT=A
PCB      TYPE=GSAM, DBDNAME=REPORT, PROCOPT=LS
PSBGEN   LANG=COBOL, PSBNAME=APPLPGM3
END
```

```
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 pcb04.report PROCOPT 'LS',
) LANGCOBOL;
```

IMS DDL syntax – ALTER PROGRAMVIEW

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. Whenever a resource is deleted, its description is deleted from the catalog of the current IMS.
- AUTODROPTANS
 - Optional parameter supplied to drop all resources associated with the PSB.
 - The associated transactions and routing codes for that PSB will also be dropped

```
>>-DROP--PROGRAMVIEW--psb_name---+-----+-----><  
                                '-AUTODROPTANS-'
```

- Resources may need to be removed from DRD or Sysgen

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'

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;
```

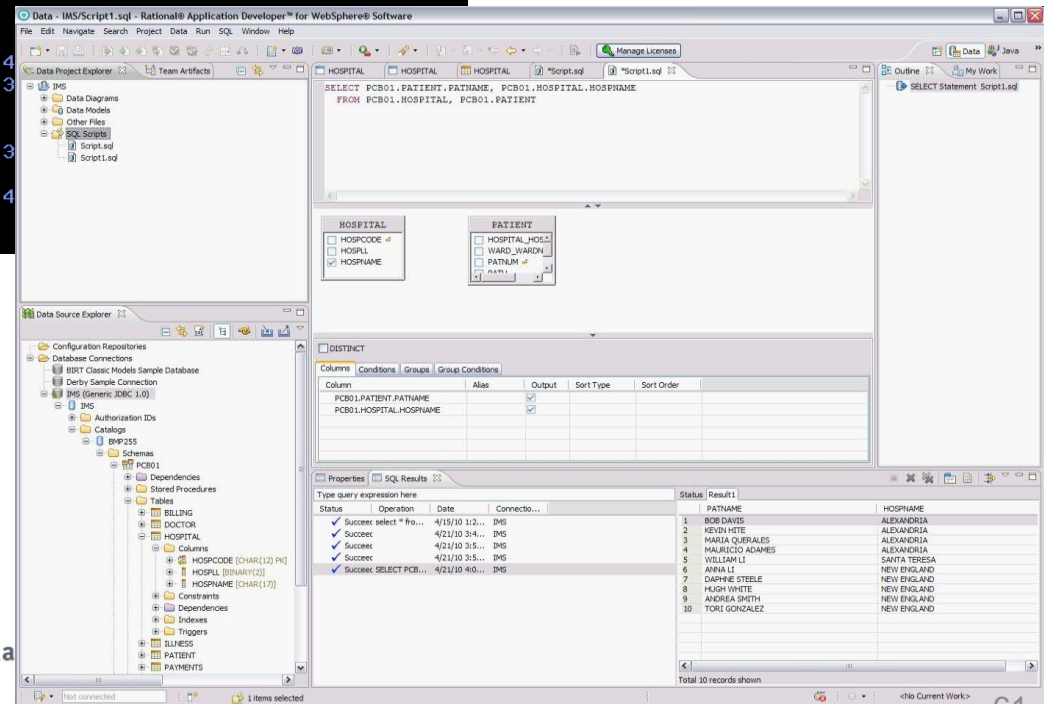
IMS Explorer for Development - DDL Input

- The IMS Explorer for Development
- The IMS DDL Batch utility

IMS Explorer for Development

- The IMS Explorer for Development is a tool to help with database visualization and querying. (E4D)

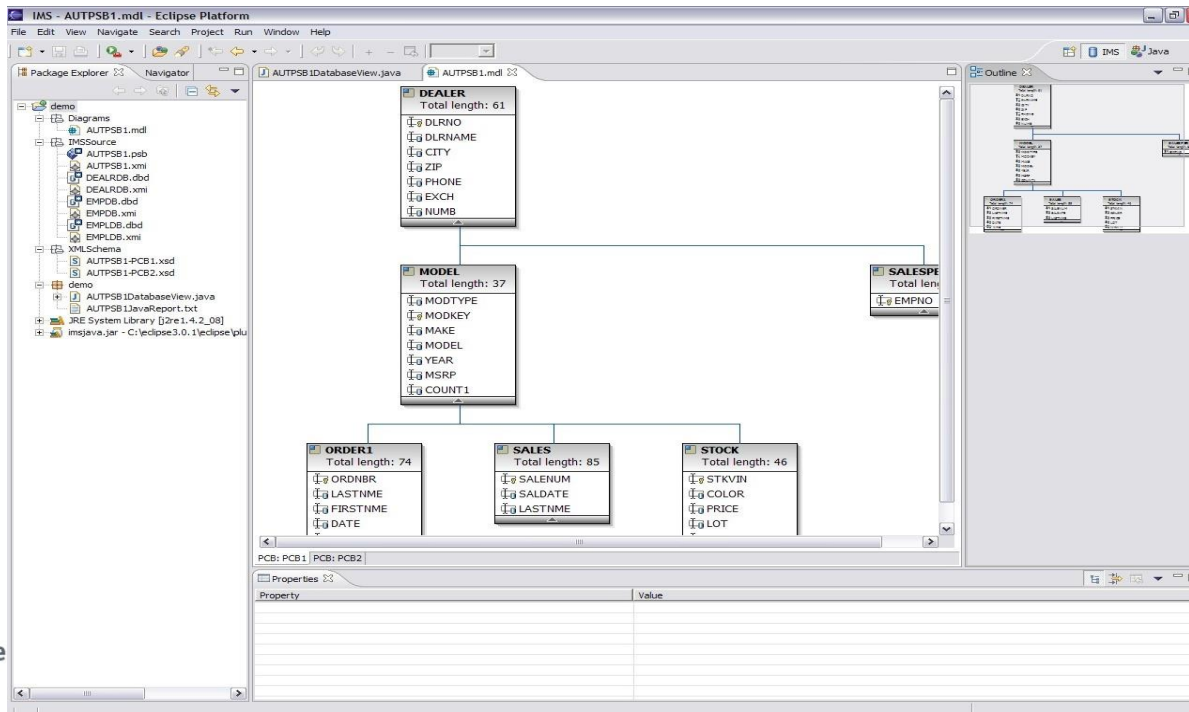
```
$DDLTO NEWJCL F1 V 80 Trunc=80 Size=96 Line=25 Col=1 Alt=0
====>
00022 U *****
00023 WTO Start of the DDLTO stream
00024 U status card has all 1's so all tracing is ON.
00025 U status card has 00002 so we use the second PCB in the PSB
00026 S 1 1 1 1 1 00002
00027 WTO Now doing GN through the database
00028 L
00029 E DATA KAA11**K1*
00030 E 01 K1 0005KAA11
00031 L
00032 E DATA KBBB11**K2
00033 E 02 K2 0011KAA11KBBB11
00034 L
00035 E DATA KAA31KEE31K31311131213131314
00036 E 03 K3K5 0021KAA11KBBB11KAA31KEE3
00037 L
00038 E DATA KAA31**K1*
00039 E 04 K1X 0026KAA11KBBB11KAA31KEE3
00040 L
00041 E DATA KAA31KEE32K31321132213231324
PF 1 FIG 2 SCREEN 2 3 QUIT 4 FILE
PF 7 BACKWARD 8 FORWARD 9 XFILE 10 LEFT
```



Complete your session evaluations online at www.SHARE.org/Orla

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



Complete

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: Database 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

IMS Batch DDL Job

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

DDL Creating an IMS database

The following scenario focuses on the elements of the database creation that apply to most database types.

- 1) User uses Explorer to model the database graphically
- 2) User uses Explorer to create the DDL
- 3) Create DDL for DBD(s)
 - 1) CREATE DATABASE, TABLESPACE, TABLE
- 4) Create DDL for PSB (user or dynamic)
 - 1) CREATE PROGRAMVIEW, SCHEMA
- 5) Explorer notifies user the impact of the DDL
- 6) Provide initial load data, if any
- 7) User submits the DDL to IMS (could be repeated)

DDL Altering an IMS database

The following scenario focuses on the elements of the database alter that apply to most database types.

- 1) User uses Explorer to model the database graphically
- 2) User uses Explorer to create the DDL
- 3) Alter DBD(s)
 - 1) ALTER DATABASE
- 4) Change PROGRAMVIEW
 - 1) DROP PROGRAMVIEW
 - 2) CREATE PROGRAMVIEW (new)
- 5) Explorer notifies user the impact of the DDL
- 6) User submits the DDL to IMS

DDL Dropping an IMS database

The following scenario focuses on the elements of the drop database that apply to all database types.

- 1) User uses Explorer to determine which database to remove
- 2) User uses Explorer to build the DDL
- 3) Drop DBD
 - 1) DROP DATABASE
- 4) Explorer notifies user the impact of the DDL
- 5) User submits the DDL to IMS

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 statements.
 - DDL- authoring tools are prevalent in the market.
 - Provide an audit trail capturing changes made.
 - Align with industry practices and expectations.

Dynamic Database Definition

- Thank you
- Please submit an evaluation

Dennis Eichelberger
deichel@us.ibm.com



Complete your session evaluations online at www.SHARE.org/Orlando-Eval