

Implicitly or Explicitly Defined Db2 Objects – the Good, the Bad and the Ugly

Steen Rasmussen, Customer Services Consultant



Abstract

Across the many releases of Db2 for z/OS, the database engine has introduced several significant changes to which and how objects can be created implicitly. This session will dive into the type of objects Db2 can create automatically, describing advantages and pitfalls for their respective usage. In the end, how this is accomplished will depend heavily on the SQL syntax and keywords used.

Agenda

- What can be defined implicitly & how to “semi control” behavior
- CURRENT RULES special register impact
- Challenges having implicitly/explicitly defined objects
- Advantages using implicitly/explicitly defined objects
- Real life scenarios to illustrate differences
 - Starting with the most simple use case gradually increasing complexity

Why Bother Using One or The Other(1)

- Do you have an object naming convention in place ?
 - It will be violated using Implicit objects
- Are you using LISTDEF processing and wildcarding ?
 - You might find this a challenge
- Are you using catalog queries based on your naming convention
 - This might get a lot harder – wildcarding based on naming standard probably not possible (unless you are very lucky)
- If implicitly defined database names used
 - There's a LIMIT
 - You can't explicitly create DBs prefixed **DSNnnnnn**
- Think about PRIQTY/SECQTY – you might have to ALTER unless (*MGEXTSZ*)

Why Bother Using One or The Other(2)

- You can create certain DSN* databases

```
.CONNECT M10A
BPA0198I: CURRENT FUNCTION LEVEL IS V12R1M508
RETCODE =      0

.OPTION NOERRORS NOSQLERRORS NOLOG SQLFORMAT(SQL)
RETCODE =      0

CREATE DATABASE DSNSTEEN;
DSNT400I  SQLCODE = 000,    SUCCESSFUL EXECUTION
CREATE DATABASE DSNT0000;
DSNT400I  SQLCODE = 000,    SUCCESSFUL EXECUTION
CREATE DATABASE DSN88888;
DSNT408I  SQLCODE = -20074, ERROR:  THE OBJECT DSN88888 CANNOT BE
CREATED BECAUSE THE FIRST THREE CHARACTERS ARE RESERVED FOR
SYSTEM OBJECTS

DSNT418I  SQLSTATE      = 42939 SQLSTATE RETURN CODE
DSNT415I  SQLERRP      = DSNXICDB SQL PROCEDURE DETECTING ERROR
DSNT416I  SQLERRD      = 2  0  0  -1  0  0 SQL DIAGNOSTIC INFORMATION
DSNT416I  SQLERRD      = X'00000002' X'00000000' X'00000000'
X'FFFFFFFF' X'00000000' X'00000000' SQL DIAGNOSTIC
INFORMATION
BPA0012E: DB2 SQL/DDL ERROR HAS OCCURRED - ROLLBACK ISSUED.
RETCODE =      8
```

What can be created IMPLICITLY

- Database for Table if not specified.
- Tablespace for Table if not specified.
- Index for Uniqueness / Constraint if the Table's Tablespace Implicitly created.
- ROWID GENERATED BY DEFAULT when RULES=STD or Tablespace Implicitly created.
- xLOB objects depends on RULES and whether Tablespace Implicitly created (use cases to follow).
- For XML the DOCID

What can be created IMPLICITLY

- *Are you happy with the DEFAULT Tablespace/Indexspace attributes ?*
- *If not - talk to your favorite Db2 SYSPROG to modify ZPARM attributes/values (those you can modify)*

```
CREATE TABLESPACE IDUGEU23
  USING STOGROUP SYSDEFLT
    PRIQTY -1
    SECQTY -1
    ERASE NO
  BUFFERPOOL BP1
  DSSIZE 4G
  CLOSE YES
  LOCKMAX SYSTEM
  SEGSIZE 32
  INSERT ALGORITHM 0
  FREEPAGE 0
  PCTFREE 5 FOR UPDATE 0
  GBPCACHE CHANGED
  DEFINE NO
  LOGGED
  TRACKMOD YES
  MAXPARTITIONS 256
  COMPRESS NO
  LOCKSIZE ROW
  MAXROWS 255
  CCSID EBCDIC
  NUMPARTS 1;
```

```
----- Thread Terminator DSNZPARM Display ----- 23/07/10 17:25
Command ==>> Scroll ==> CSR
LINE 319 OF 363

DB2 SSID ==> D12A      Lmod Name ==> D12APARM      Assem Date ==> 04/19/23

-----
DESCRIPTION                                VALUE      KEYWORD
-----
Default 16K buffer pool for implicit tablespaces BP16K1  TBSBP16K
Default 32K buffer pool for implicit tablespace BP32K1  TBSBP32K
Default 8K buffer pool for implicit tablespaces. BP8K1  TBSBP8K
```

SET CURRENT RULES = 'xxx'

- DB2 is the default
 - Will eliminate many of the Implicit Object definitions
- STD
 - Will enable more Implicit Object definitions

(will be covered in detail later using a PBR use case with a CLOB)

Use Case 1 : The Simple Scenario

Most simple use case where implicit objects can be introduced (1)

- Table created in 4 different ways:
 - DB and TS Explicit (1) and DB and/or TS Implicit (2+3+4)

```
CREATE TABLE RASST02.IDUGEU23TB1
  (DEPTNO CHARACTER(3) FOR SBCS DATA NOT NULL
  ,CONSTRAINT DEPTNO PRIMARY KEY
  (DEPTNO) ) IN IDUGEU24.IDUGEUTS;
=====
CREATE TABLE RASST02.IDUGEU23TB2
  (DEPTNO CHARACTER(3) FOR SBCS DATA NOT NULL
  ,CONSTRAINT DEPTNO PRIMARY KEY
  (DEPTNO) ) IN DATABASE IDUGEU24;
=====
CREATE TABLE RASST02.IDUGEU23TB3
  (DEPTNO CHARACTER(3) FOR SBCS DATA NOT NULL
  ,CONSTRAINT DEPTNO PRIMARY KEY
  (DEPTNO) ) ;
=====
SET CURRENT RULES = 'STD' ;
CREATE TABLE RASST02.IDUGEU23TB4
  (DEPTNO CHARACTER(3) FOR SBCS DATA NOT NULL
  ,CONSTRAINT DEPTNO PRIMARY KEY
  (DEPTNO) ) ;
```

The final
results on
next page

Most simple use case where implicit objects can be introduced (2)

- Explicit DB.TS requires Explicit Unique Index.
 - DB or TS Implicitly created - Unique Index Implicitly created
 - CURRENT RULES no impact

Necessary to explicitly create UNIQUE INDEX

TABLE NAME	SCHEMA	DEPENDENT TS/IX	SCHEMA/DB
IDUGEU23TB1	RASST02	IDUGEU23TB1 * TABLE SPACE IDUGEUTS	RASST02 1 IDUGEU24
IDUGEU23TB2	RASST02	IDUGEU23TB2 * TABLE SPACE * INDEX IDUGEU23 IDUGEU23_#_5KN	RASST02 1 1 IDUGEU24 RASST02
IDUGEU23TB3	RASST02	IDUGEU23TB3 * TABLE SPACE * INDEX IDUGEU23 IDUGEU23_#_AXN	RASST02 1 1 DSN01572 RASST02
IDUGEU23TB4	RASST02	IDUGEU23TB4 * TABLE SPACE * INDEX IDUGEU23 IDUGEU23_#_37N	RASST02 1 1 DSN01573 RASST02

Now include CLOB (1)

- Let's look at another object type and implicit create options.

```
CREATE TABLE RASST02.IDUGNA24TB1
  (DEPTNO CHARACTER(3) FOR SBCS DATA NOT NULL
  ,LOBDATA CLOB(1M) WITH DEFAULT NULL
  ,LOB_ROWID ROWID NOT NULL GENERATED ALWAYS ) IN IDUGNA24.IDUGNATS;
=====
CREATE TABLE RASST02.IDU24T23TB2
  (DEPTNO CHARACTER(3) FOR SBCS DATA NOT NULL
  ,LOBDATA CLOB(1M) WITH DEFAULT NULL
  ,LOB_ROWID ROWID NOT NULL GENERATED ALWAYS ) IN DATABASE IDUGNA24;
=====
CREATE TABLE RASST02.IDUGNA24TB3
  (DEPTNO CHARACTER(3) FOR SBCS DATA NOT NULL
  ,LOBDATA CLOB(1M) WITH DEFAULT NULL
  ,LOB_ROWID ROWID NOT NULL GENERATED ALWAYS ) ;
=====
SET CURRENT RULES = 'STD' ;
CREATE TABLE RASST02.IDUGNA24TB4
  (DEPTNO CHARACTER(3) FOR SBCS DATA NOT NULL
  ,LOBDATA CLOB(1M) WITH DEFAULT NULL
  ,LOB_ROWID ROWID NOT NULL GENERATED ALWAYS ) ;
***** Bottom of Data *****
```

Now include CLOB (2)

- Only using Implicit tablespace (but Explicit DB) – at least all objects are in the same DB (*think LISTDEF, SQL etc.*)

TABLE NAME	TABLE/INDEX	SCHEMA/DB	TYPE
IDUGNA24TB1	IDUGNA24TB1	RASST02	TABLE
	* TABLE SPACE	1	
	IDUGNATS	IDUGNA24	TABLE SPACE
IDUGNA24TB2	IDUGNA24TB2	RASST02	TABLE
	* TABLE SPACE	2	
	* TABLE	1	
	* INDEX	1	
IDUGNA24TB3	IDUGNA24	IDUGNA24	TABLE SPACE
	L0NRFS DJ	IDUGNA24	TABLE SPACE
	IDUGNLOBDA0NRFVHWL	RASST02	TABLE
	I IDUGNLOBDA0NRFWS1	RASST02	INDEX
	IDUGNA24TB3	RASST02	TABLE
	* TABLE SPACE	2	
IDUGNA24TB4	* TABLE	1	
	* INDEX	1	
	IDUGNA24	DSN01574	TABLE SPACE
	L0MLQ QY5	DSN01574	TABLE SPACE
	IDUGNLOBDA0MLQUIC2	RASST02	TABLE
	I IDUGNLOBDA0MLQUQK	RASST02	INDEX
	IDUGNA24TB4	RASST02	TABLE
	* TABLE SPACE	2	
* TABLE	1		
IDUGNA24TB4	* INDEX	1	
	IDUGNA24	DSN01575	TABLE SPACE
	L0MLQ5EI	DSN01575	TABLE SPACE
	IDUGNLOBDA0MLQ9P3X	RASST02	TABLE
	I IDUGNLOBDA0MLQXZI	RASST02	INDEX

• Explicitly referenced DB.TS incomplete until all AUX objects created explicitly.

• All other use cases are complete – downside is the provided object names.



Leading the **Db2** User Community for **35** Years

A large, semi-transparent blue circle on the left side of the slide. Inside the circle, there is a faint, stylized illustration of a server rack with various components and cables.

IDUG

2023 EMEA Db2 Tech Conference

Use Case 2 : Things Can Get Messy

Things can get messy (1)

- Object details
 - PBG MAXPARTITIONS 2 NUMPARTS 2
 - Two CLOB columns
 - Explicitly defined :
 - 1 DB
 - 1 PBG TS
 - 4 LOB TS (2 LOBS x 2 PARTITIONS)
 - 1 BASE TB (and potentially base indexes)
 - 4 AUX tables
 - 4 AUX indexes
 - (Lots of typing – but naming convention maintained – so far)

Things can get messy (2)

- Next step is to ADD a PBG partition (ALTER TABLE ADD PART)
 - ALTER MAXPARTITIONS from 2 -> 3
 - This will cause 2 LOB tablespaces to be added IMPLICITLY
 - Naming convention goes South
- Next ADD CLOB column to the base table
 - Explicitly definition:
 - 3 LOB tablespaces (3 partitions)
 - 3 AUX tables
 - 3 AUX indexes
- IMPLICITLY or EXPLICITLY defined depends on SET CURRENT RULES

Things can get messy (3)

- Status after initial create (naming convention maintained) :

CMD	NAME	CREATOR	DNAME	DCREATOR	TYPE
_____	IDUGEU23	RASST02	IDUGEU23	RASST02	DATA BASE
			* TABLE SPACE	5	
			* TABLE	5	
			* INDEX	5	
_____			BASELOBA	IDUGEU23	TABLE SPACE
_____			BASELOBB	IDUGEU23	TABLE SPACE
_____			BASELOBC	IDUGEU23	TABLE SPACE
_____			BASELOBD	IDUGEU23	TABLE SPACE
_____			BASEPBG	IDUGEU23	TABLE SPACE
_____			BASELOBA_PART1_CLO>	RASST02	TABLE
_____			BASELOBB_PART2_CLO>	RASST02	TABLE
_____			BASELOBC_PART1_CLO>	RASST02	TABLE
_____			BASELOBD_PART2_CLO>	RASST02	TABLE
_____			BASEPBG_TAB	RASST02	TABLE
_____			BASELOBA_PART1_IX1	RASST02	INDEX
_____			BASELOBA_PART2_IX1	RASST02	INDEX
_____			BASELOBC_PART1_IX2	RASST02	INDEX
_____			BASELOBC_PART2_IX2	RASST02	INDEX
_____			BASEPBG_TAB_IX	RASST02	INDEX
			***** BOTTOM OF DATA *****		

Things can get messy (4)

- Status after ADD CLOB column.
 - You can control the EXPLICIT/IMPLICIT (via CURRENT RULES)
 - The cost is sacrificing the naming convention – but a lot easier (think about 100 partitions)

<u>DNAME</u>	<u>DCREATOR</u>	<u>TYPE</u>
DAVETCOL05SEOEGCFT	BLADA08	TABLE
DAVETCOL07SEOES8Y1	BLADA08	TABLE
DAVETS1_TAB	BLADA08	TABLE
DAVETS1A_PART1_CLO>	BLADA08	TABLE
DAVETS1A_PART2_CLO>	BLADA08	TABLE
DAVETS1B_PART1_CLO>	BLADA08	TABLE
DAVETS1B_PART2_CLO>	BLADA08	TABLE
DAVETS1C_PART1_CLO>	BLADA08	TABLE
DAVETS2C_PART2_CLO>	BLADA08	TABLE
DAVETS3C_PART3_CLO>	BLADA08	TABLE
DAVETS1_TAB_IX	BLADA08	INDEX
DAVETS1A_PART1_IX1	BLADA08	INDEX
DAVETS1A_PART2_IX1	BLADA08	INDEX
DAVETS1B_PART1_IX2	BLADA08	INDEX
DAVETS1B_PART2_IX2	BLADA08	INDEX
DAVETS1C_PART1_IX3	BLADA08	INDEX
DAVETS1C_PART2_IX3	BLADA08	INDEX
DAVETS1C_PART3_IX3	BLADA08	INDEX
IDAVETCOL05SEOEGB8	BLADA08	INDEX
IDAVETCOL07SEOESWT	BLADA08	INDEX

<u>DNAME</u>	<u>DCREATOR</u>	<u>TYPE</u>
BASELOBA_PART1_CLO>	RASST02	TABLE
BASELOBB_PART2_CLO>	RASST02	TABLE
BASELOBC_PART1_CLO>	RASST02	TABLE
BASELOBD_PART2_CLO>	RASST02	TABLE
BASEPBG_TAB	RASST02	TABLE
BASEPCOL050HIMIBKY	RASST02	TABLE
BASEPCOL070HIMQ8AD	RASST02	TABLE
BASEPCOL080HINJRGD	RASST02	TABLE
BASEPCOL080HINRZBY	RASST02	TABLE
BASEPCOL080HINYG6N	RASST02	TABLE
BASELOBA_PART1_IX1	RASST02	INDEX
BASELOBA_PART2_IX1	RASST02	INDEX
BASELOBC_PART1_IX2	RASST02	INDEX
BASELOBC_PART2_IX2	RASST02	INDEX
BASEPBG_TAB_IX	RASST02	INDEX
IBASEPCOL050HIMI8T	RASST02	INDEX
IBASEPCOL070HIMQ6C	RASST02	INDEX
IBASEPCOL080HINJGU	RASST02	INDEX
IBASEPCOL080HINR9N	RASST02	INDEX
IBASEPCOL080HINZXS	RASST02	INDEX



IDUG

2023 EMEA Db2 Tech Conference

Use Case 3 : Things Can Also Be Easy – Looking at PBR's

Things can also be easy (1)

- PBR table with one LOB and 6 partitions.
 - Implicit DB and TS created (violating our naming convention (*DSN05572.HSBCIMP1*) chosen by Db2).
 - All AUX objects created – object creation complete and ready to use.

```
CREATE TABLE RASST02.HSBCIMP1
  (COUNTRY VARCHAR(50)
  ,CITY VARCHAR(30)
  ,ZIPCODE CHARACTER(10)
  ,RESIDENTS INTEGER
  ,LOBDATA CLOB(2M) INLINE LENGTH 0
  ,LOB_ROWID ROWID GENERATED ALWAYS )
PARTITION BY RANGE
  (COUNTRY NULLS LAST ASC)
  ( PARTITION 1 ENDING ('G') INCLUSIVE
  , PARTITION 2 ENDING ('J') INCLUSIVE
  , PARTITION 3 ENDING ('M') INCLUSIVE
  , PARTITION 4 ENDING ('ST') INCLUSIVE
  , PARTITION 5 ENDING ('T') INCLUSIVE
  , PARTITION 6 ENDING ('ZZZ') INCLUSIVE );
```

Things can also be easy (2)

- If we really like to obtain DB.TS naming convention
 - Explicitly create HSBCDB.HSBCIMP2 (no SET CURRENT RULES, so DB2 used)
 - We now have some work to do

```
CREATE TABLE RASST02.HSBCIMP2
  (COUNTRY VARCHAR(50)
  ,CITY VARCHAR(30)
  ,ZIPCODE CHARACTER(10)
  ,RESIDENTS INTEGER
  ,LOBDATA CLOB(2M) INLINE LENGTH 1000000
  ,LOB_ROWID ROWID GENERATED ALWAYS AS ROWID)
PARTITION BY RANGE
  (COUNTRY NULLS LAST ASC)
  ( PARTITION 1 ENDING ('G') INCLUSIVE
  , PARTITION 2 ENDING ('J') INCLUSIVE
  , PARTITION 3 ENDING ('M') INCLUSIVE
  , PARTITION 4 ENDING ('ST') INCLUSIVE
  , PARTITION 5 ENDING ('T') INCLUSIVE
  , PARTITION 6 ENDING ('ZZZ') INCLUSIVE )
IN HSBCDB.HSBCIMP2;
```

```
DB2 Object ==> T          Option ==> DI  Where => N
Table Name ==> HSBCIMP2  > Creator ==> RASST02  >
Qualifier ==> *          > N/A      ==> *          >
Loc: LOCAL ----- SSID: D12A -----RASST02 -          LINE 1 OF 3 >
CMD      NAME      CREATOR  DNAME      DCREATOR  TYPE
----- HSBCIMP2 RASST02  HSBCIMP2   RASST02   TABLE
* TABLE SPACE 1
HSBCIMP2      HSBCDB   TABLE SPACE
***** BOTTOM OF DATA *****
```

All AUX objects are missing – 18
CREATE statements to do

Things can also be easy (3)

- Same scenario (explicit DB.TS) but using CURRENT RULES='STD'

```
SET CURRENT RULES = 'STD' ;
CREATE TABLE RASST02.HSBCIMP3
  (COUNTRY VARCHAR(50)
  ,CITY VARCHAR(30)
  ,ZIPCODE CHARACTER(10)
  ,RESIDENTS INTEGER
  ,LOBDATA CLOB(2M) INLINE LENGTH 0
  ,LOB_ROWID ROWID GENERATED ALWAYS )
PARTITION BY RANGE
  (COUNTRY NULLS LAST ASC)
  ( PARTITION 1 ENDING ('G')    INCLUSIVE
  , PARTITION 2 ENDING ('J')    INCLUSIVE
  , PARTITION 3 ENDING ('M')    INCLUSIVE
  , PARTITION 4 ENDING ('ST')   INCLUSIVE
  , PARTITION 5 ENDING ('T')    INCLUSIVE
  , PARTITION 6 ENDING ('ZZZ') INCLUSIVE )
IN HSBCDB.HSBCIMP3;
```

Things can also be easy (4)

- Naming convention maintained for DB and base TS (*no DSNnnnnnn*)
 - Object status considered COMPLETE by Db2

```
RQTDI          ----- RC/Q Table Drop Impact ----- 2023/06/28 11:37
COMMAND ==>>>          SCROLL ==>>> CSR

Loc: LOCAL ----- SSID: D12A -----RASST02 -          LINE 01 OF 23  >
CMD      NAME      CREATOR  DNAME          DCREATOR  TYPE
-----  -
-----  HSBCIMP3  RASST02  HSBCIMP3      RASST02   TABLE
-----  * TABLE SPACE  7
-----  * TABLE        6
-----  * INDEX         6
-----  HSBCIMP3      HSBCDB    TABLE SPACE
-----  LXytyc4k      HSBCDB    TABLE SPACE
-----  LXytyjl3      HSBCDB    TABLE SPACE
-----  LXytyldj      HSBCDB    TABLE SPACE
-----  LXytySNZ      HSBCDB    TABLE SPACE
-----  LXyty05L      HSBCDB    TABLE SPACE
-----  LXyty8LX      HSBCDB    TABLE SPACE
-----  HSBCILOBDAXYTYGKGI  RASST02   TABLE
-----  HSBCILOBDAXYTYNEOF  RASST02   TABLE
-----  HSBCILOBDAXYTYOEEC  RASST02   TABLE
-----  HSBCILOBDAXYTYVWJD  RASST02   TABLE
-----  HSBCILOBDAXYTYYP3C  RASST02   TABLE
-----  HSBCILOBDAXYTY439C  RASST02   TABLE
-----  IHSBCILOBDAXYTYH20  RASST02   INDEX
-----  IHSBCILOBDAXYTYOTX  RASST02   INDEX
-----  IHSBCILOBDAXYTYPX0  RASST02   INDEX
-----  IHSBCILOBDAXYTYVPP  RASST02   INDEX
-----  IHSBCILOBDAXYTYZ09  RASST02   INDEX
-----  IHSBCILOBDAXYTY4IU  RASST02   INDEX
***** BOTTOM OF DATA *****
```



Leading the **Db2** User Community for **35** Years

A large, faded IDUG logo and a stylized illustration of a server rack are visible on the left side of the slide.

IDUG

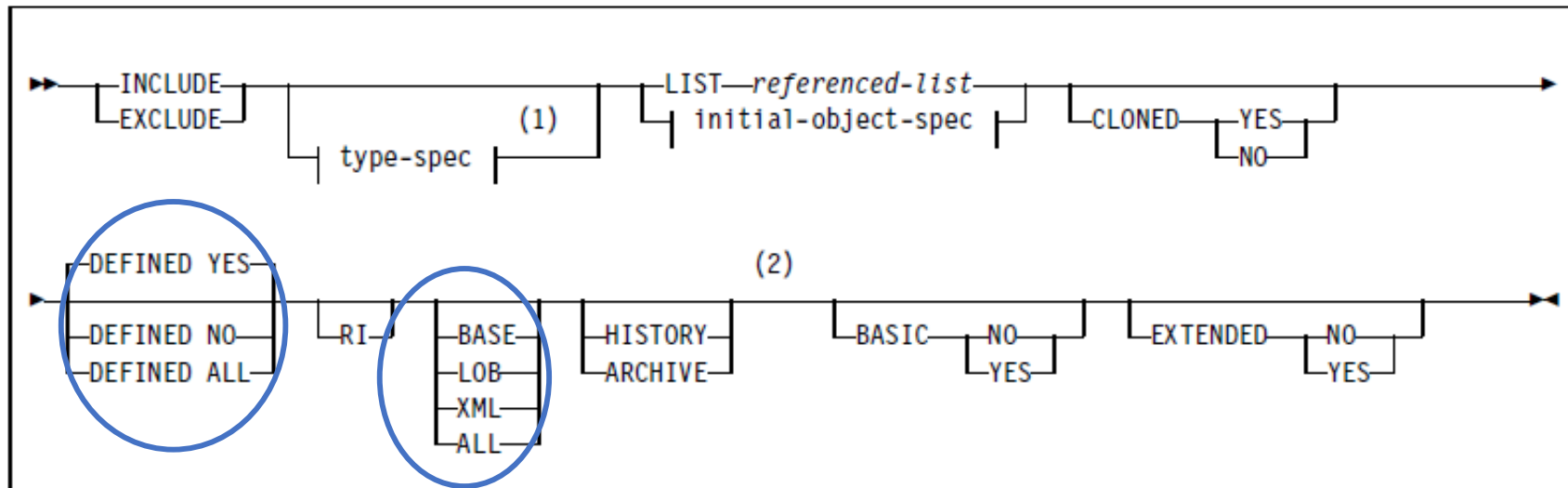
2023 EMEA Db2 Tech Conference

LISTDEF Processing Considerations

LISTDEF Processing(1)

- Syntax from IBM Db2 Utility Guide – focus on two keywords.

list-options:



Notes:

- 1 You must specify *type-spec* if you specify DATABASE.
- 2 HISTORY or ARCHIVE can be specified either before or after the BASE, LOB, XML, and ALL keywords. However, HISTORY and ARCHIVE are always processed last, after all other keywords are handled.

LISTDEF Processing(2)

- Partial wildcarding on Tablespace name can be challenging – no control of Implicit Tablespace names

TEN TABLESPACES :

PBG start = 2 parts.
2 LOBs = 4 pagesets.
ADD part = 1 pageset.
ADD LOB for 3 PBG
parts = 3 pagesets.

Initially PBG 2-PART
and 2x2 AUX explicitly
created.

DNAME	DCREATOR	TYPE
IDUGEU23	RASST02	DATA BASE
* TABLE SPACE	10	
* TABLE	10	
* INDEX	10	
BASELOBA	IDUGEU23	TABLE SPACE
BASELOBB	IDUGEU23	TABLE SPACE
BASELOBC	IDUGEU23	TABLE SPACE
BASELOBD	IDUGEU23	TABLE SPACE
BASEPBG	IDUGEU23	TABLE SPACE
L0HIMDZT	IDUGEU23	TABLE SPACE
L0HIMML4	IDUGEU23	TABLE SPACE
L0HINFZA	IDUGEU23	TABLE SPACE
L0HINNXU	IDUGEU23	TABLE SPACE
L0HINV69	IDUGEU23	TABLE SPACE
BASELOBA_PART1_CLO>	RASST02	TABLE
BASELOBB_PART2_CLO>	RASST02	TABLE

EXPLICIT
AUX DEFINE
YES

IMPLICIT AUX
DEFINE NO
and new part

LISTDEF Processing(3)

- Why do we have TEN tablespaces ?
 - PBG NUMPARTS 2 = 1 base tablespace
 - 2 LOB cols and 2 partitions = 4 AUX tablespaces
 - ADD PART – since 2 LOBs = 2 AUX tablespaces
 - ADD LOB col – since 3 partitions = 3 AUX tablespaces

LISTDEF Processing(4)

- Only DEFINED objects picked up
 - In this use case only explicitly and defined included in LISTDEF
 - Two LOB columns in this two part PBG are the only explicitly defined and DEFINE YES

```
OPTIONS PREVIEW
```

```
LISTDEF LIST1 INCLUDE TABLESPACE IDUGEU23.*
```

```
LISTDEF LIST1 -- 00000005 OBJECTS  
INCLUDE TABLESPACE IDUGEU23.BASELOBA  
INCLUDE TABLESPACE IDUGEU23.BASELOBB  
INCLUDE TABLESPACE IDUGEU23.BASELOBC  
INCLUDE TABLESPACE IDUGEU23.BASELOBD  
INCLUDE TABLESPACE IDUGEU23.BASEPBG
```

LISTDEF Processing(5)

- LISTDEF with keyword DEFINED NO illustrates this
 - Here instantiated pagesets NOT picked up

```
LISTDEF LIST1 INCLUDE TABLESPACE IDUGEU23.* ALL DEFINED NO
```

```
LISTDEF LIST1 -- 00000005 OBJECTS  
INCLUDE TABLESPACE IDUGEU23.L0HIMDZT  
INCLUDE TABLESPACE IDUGEU23.L0HIMML4  
INCLUDE TABLESPACE IDUGEU23.L0HINFZA  
INCLUDE TABLESPACE IDUGEU23.L0HINNXU  
INCLUDE TABLESPACE IDUGEU23.L0HINV69
```

LISTDEF Processing(6)

- Combining LOB & DEFINED ALL
 - Will process all AUX objects – but not BASE PBG

OPTIONS PREVIEW

```
LISTDEF LIST1 INCLUDE TABLESPACE IDUGEU23.* LOB DEFINED ALL
```

```
LISTDEF LIST1 -- 00000009 OBJECTS
INCLUDE TABLESPACE IDUGEU23.BASELOBA
INCLUDE TABLESPACE IDUGEU23.BASELOBB
INCLUDE TABLESPACE IDUGEU23.BASELOBC
INCLUDE TABLESPACE IDUGEU23.BASELOBD
INCLUDE TABLESPACE IDUGEU23.L0HIMDZT
INCLUDE TABLESPACE IDUGEU23.L0HIMML4
INCLUDE TABLESPACE IDUGEU23.L0HINFZA
INCLUDE TABLESPACE IDUGEU23.L0HINNXU
INCLUDE TABLESPACE IDUGEU23.L0HINV69
```

LISTDEF Processing(7)

- Two options to include everything:
 - You have to use ALL instead of LOB – then only one LISTDEF needed
 - OR use two INCLUDEs as below

OPTIONS PREVIEW

```
LISTDEF LIST1 INCLUDE TABLESPACE IDUGEU23.* DEFINED ALL
INCLUDE TABLESPACE IDUGEU23.* LOB DEFINED ALL
```

```
LISTDEF LIST1 -- 00000010 OBJECTS
INCLUDE TABLESPACE IDUGEU23.BASELOBA
INCLUDE TABLESPACE IDUGEU23.BASELOBB
INCLUDE TABLESPACE IDUGEU23.BASELOBC
INCLUDE TABLESPACE IDUGEU23.BASELOBD
INCLUDE TABLESPACE IDUGEU23.BASEPBG
INCLUDE TABLESPACE IDUGEU23.L0HIMDZT
INCLUDE TABLESPACE IDUGEU23.L0HIMML4
INCLUDE TABLESPACE IDUGEU23.L0HINFZA
INCLUDE TABLESPACE IDUGEU23.L0HINNXU
INCLUDE TABLESPACE IDUGEU23.L0HINV69
```



Leading the **Db2** User Community for **35** Years

A large, semi-transparent blue circle on the left side of the slide. Inside the circle, there is a faint, stylized graphic of a server rack with arrows pointing up and down, suggesting data flow or technology.

IDUG

2023 EMEA Db2 Tech Conference

Challenges to Consider When Extracting DDL from the Catalog – Understand Your Tooling of choice

Extract / Generate DDL from the Catalog(1)

- Not an issue when everything explicitly defined.
- Challenge when implicitly or mixed implicitly/explicitly.
 - IDUGEU23TB3 table was created without IN DB / IN DB.TS
 - Not possible to “use the same object names”
 - Tooling (incl. your own) have to “THINK” – comment out implicit objects.
 - Depending on your environment – might be necessary to modify prior to execution.

```
--      CREATE DATABASE DSN01572
--                BUFFERPOOL BP1 INDEXBP      BP2
--                STOGROUP SYSDEFLT;
--
--      CREATE TABLESPACE IDUGEU23
--                USING STOGROUP SYSDEFLT
--                PRIQTY -1 SECQTY -1
--                MAXPARTITIONS 256 NUMPARTS 1;
--
--      CREATE TABLE RASST02.IDUGEU23TB3
--                (DEPTNO CHARACTER(3) FOR SBCS DATA      NOT NULL
--                ,DEPTNAME VARCHAR(36) FOR SBCS DATA    NOT NULL
--                ,MGRNO CHARACTER(6) FOR SBCS DATA
--                ,ADMRDEPT CHARACTER(3) FOR SBCS DATA  NOT NULL
--                ,LOCATION CHARACTER(16) FOR SBCS DATA
--                ,SDEPTNO CHARACTER(4) FOR SBCS DATA
--                ,CONSTRAINT DEPTNO PRIMARY KEY      (DEPTNO) );
--
--      CREATE UNIQUE INDEX RASST02.IDUGEU23_#_AXN
--                ON RASST02.IDUGEU23TB3
--                (DEPTNO ASC)
--                PIECESIZE 4194304K;
```

Extract / Generate DDL from the Catalog(2)

- Think of the PBG tablespace used earlier
 - Object details
 - PBG MAXPARTITIONS 2 NUMPARTS 2
 - Two CLOB columns
 - Explicitly defined :
 - 1 DB
 - 1 PBG TS
 - 4 LOB TS (2 LOBS x 2 PARTITIONS)
 - 1 BASE TB (and potentially base indexes)
 - 4 AUX tables
 - 4 AUX indexes
 - Then a PBG partition was added -> Two LOB tablespaces added IMPLICITLY.
 - Nice mix of Implicit/Explicit objects *(recommended solution next page)*

Extract / Generate DDL from the Catalog

- DDL is invalid if mixing IMPLICIT/EXPLICIT created objects.
- Don't specify NUMPARTS -> one is defined at creation time.
- When LOAD/INSERT needs another partition – Db2 will grow dynamically using implicit objects.
- Schema synchronization might be a challenge.
 - Naming convention mapping can't be done.
 - Table's tablespace mapping – you probably will have to live with different names (*often tablespace names are identical*).
- Why not use “profile” to specify object names ?
 - Maybe a promise from IBM at IDUG EMEA 2023

| Broadcom Mainframe Technical Exchanges

- ✓ European in-person event in Prague: April 16-18
- North American in-person event in Plano, TX: September 10-12 ([registration open](#))
- Global virtual event: October 8-10 ([save the date](#))

Make plans to attend

- Network with peers and Mainframe technical experts
- Technical education, product update, how-to and roundtable sessions
- No registration fee! Open to all Broadcom customers
- Learn more: <https://bit.ly/MainframeTechEx>





Thank You

**Any Additional Questions or
Comments ?**