Demystifying Online DB2 for z/OS Migration

Jay Yothers
IBM

Wednesday, March 12
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 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.

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.

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious, and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Trademarks: The following terms are trademarks or registered trademarks of other companies and have been used in at least one of the pages of the presentation:

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both: AIX, AS/400, DataJoiner, DataPropagator, DB2, DB2 Connect, DB2 Extenders, DB2 OLAP Server, DB2 Universal Database, Distributed Relational Database Architecture, DRDA, eServer, IBM, IMS, iSeries, MVS, Net.Data, OS/390, OS/400, PowerPC, pSeries, RS/6000, SQL/400, SQL/DS, Tivoli, VisualAge, VM/ESA, VSE/ESA, WebSphere, z/OS, zSeries

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

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

- Define Online Migration
- Why can Migration be Online?
- But … What if it fails?
- ACCESS(MAINT)
- Migration Serialization
- Finding Interference
- Stretching out ENFM
- Practice
- DSNTIJTC
- DSNTIJEN
What is Online?

- What is Online --- What can be running concurrently with migration
  - The business application set
    - Business applications do not directly access the Catalog or Directory
What is Online?

- What is Online --- What can be running concurrently with migration
  - The business application set
    - Business applications do not directly access the Catalog or Directory

- What is NOT Online --- What can NOT be running concurrently
  - Utilities – ours or anyone else's
  - Binds – automatic or otherwise
  - DDL
  - GRANTs or REVOKEs
  - Monitors
  - Your queries of the Catalog – directly or through tools
What is Migration?

- **DSNTIJTC** - (CATMAINT)
  - Must be run the first time V11 is started
  - DB2 will not accept any work until DSNTIJTC completes
  - DDL against the Catalog

- **DSNTIJEN** - (CATENFM)
  - The ENFM process
  - A series of job steps
    - DDL against the Catalog and Directory
    - SHRLEVEL REFERENCE/CHANGE REORGs of Catalog and Directory table spaces

- **DSNTIJNF**
  - Rebuilds the SCA
  - “Flips” the NFM bit on
What is Migration

- Other jobs
  - DSNTIJNH
    - Causes DSNTIJEN to stop after the current table space
  - DSNTIJCS
    - “Flips” the CMn* bit on
  - DSNTIJES
    - “Flips” the ENFMn* bit on
Why can Migration be Online?

- No Magic! (almost)
- Normal DDL
  - Acquires usual locks associated with normal DDL
- Normal SHRLEVEL REFERENCE/CHANGE REORGs
  - Acquires usual claims and drains associated with normal REORG
- Most application objects are cached in memory
  - EDM pool contains DBDs and Packages
  - Dynamic Statement Cache contains bound dynamic SQL
- Locking associated with caching something is fleeting
  - EDM uses a separate task to read from DBD01 and SPT01
  - PREPARE locks are released prior to returning to the application
But ... What if it fails?

- The Catalog and Directory will be left completely operational
  - No recovery is necessary

- DSNTIJTC
  - All changes are in a single commit scope
    - Either all changes are successful or none of them are
    - A failure here would be as though it had never happened at all

- DSNTIJEN
  - A series of job steps work through table spaces one at a time
    - DDL associated with tables in the table space, single commit scope
      - A failure here would be as though it had never happened at all
      - REORG SHRLEVEL REFERENCE/CHANGE or “Reset” the table space
        - A failure here would be as though it had never happened at all

- These jobs are completely restartable unchanged
  - After the reason for failure has been dealt with
  - Run the job again, it will pick up where it left off and continue on
ACCESS (MAINT)

- There is only 1 Catalog and Directory for the entire Group
- For DSNTIJTC, this is one way to avoid -904 00C900A6 until it completes successfully
- For DSNTIJEN, this doesn't have much of an effect
Migration Serialization

- **DDL**
  - Exclusive lock on DSNDB06
  - Read and Write Claims on various Catalog and Directory table spaces
  - Shared and Exclusive locks on various rows/pages in the Catalog and Directory

- **REORG SHRLEVEL REFERENCE/CHANGE** of Catalog and Directory table spaces
  - Drain Writers if SHRLEVEL REFERENCE
  - Define Shadows, Unload, Sort, Reload (Rebuild) into Shadows
  - Drain All
  - Rename data sets
  - Switch phase DDL (Essentially materializing “deferred” alter)
  - Dedrain All
Finding Interference

- Ensure sufficient space in Catalog and Directory SMS data class
  - For newly defined table spaces and indexes
  - For output of DSNTIJEN REORGs
  - Recommend same space currently occupied by Catalog and Directory

- Run REORG SHRLEVEL REFERENCE/CHANGE prior to running DSNTIJEN
  - Failure to get the Drain All will not feel as bad
    - Will help to identify activities that will interfere with DSNTIJEN
  - Will give you a general idea of how long it takes
  - Will give you an idea how much SORTWORK and Image Copy space
  - Will leave table spaces in clustered order

SYSUTILX  (Reset)   SYSTSIXS  (CHANGE)
SYSLGRNX  (CHANGE)   SYSTSTAB  (CHANGE)
SYSCOPY  (REFERENCE)   SYSSTR  (REFERENCE)
SYSRRTSTS  (REFERENCE)
Stretching out ENFM

- If ENFM doesn't finish in allotted window, it can be stopped
  - Catalog and Directory can be left as is until next window

- Use DSNTIJNH to cause DSNTIJEN to stop
  - When the end of the maintenance window is approaching
    - DSNTIJEN will stop after REORG of the current table space

- Do not modify or “split” the DSNTIJEN job
  - It is not necessary and could be error prone
  - DSNTIJEN will pick up where it left off
    - Regardless of why it stopped
Practice

- Most successful customers
  - Make a clone of the production system
    • Including the workload, monitors and all
    • IBM InfoSphere OWR can help
  - Practice running DSNTIJTC and DSNTIJEN on the clone
    • Builds confidence in ability to migrate online
    • Establishes points when interfering processes must be terminated
    And when they can be restarted
    • Sets realistic time lines
    • Sets expectations on space requirements
DSNTIJTC

Alter existing Catalog objects
Create new Catalog objects

DSNTIJEN

ENFM DDL
SYSUTILX
SYSLGRNWX
SYSCOPY
SYSRTSTS
SYSTSIXS
SYSTSTAB
SYSSTR
DSNTIJTC - New columns

- All new columns are NOT NULL WITH DEFAULT unless otherwise indicated

- **SYSCOPY** MODECREATED CHAR(2)
- **SYSDATATYPES** ARRAYLENGTH BIGINT
  ARRAYINDEXTYPEID INTEGER
  ARRAYINDEXTYPELEN BIGINT
  ARRAYINDEXSUBTYPE CHAR(1)
- **SYSDEPENDENCIES** BAUTH SMALLINT
  DVERSION VARCHAR(122)
- **SYSINDEXPART** RBA_FORMAT CHAR(1)
- **SYSOBDS** RBA ROWID WITH DEFAULT X'00000000000000000000000000000000'
  OBD_IMAGE BLOB(1G)
  RELCREATED CHAR(1)
# DSNTIJTC - New columns

<table>
<thead>
<tr>
<th>Table</th>
<th>Column</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSPACKAGE</td>
<td>BUSTIMESENSITIVE</td>
<td>CHAR(1)</td>
<td>DEFAULT 'N'</td>
</tr>
<tr>
<td></td>
<td>APPLCOMPAT</td>
<td>VARCHAR(10)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ARCHIVESENSITIVE</td>
<td>CHAR(1)</td>
<td>DEFAULT 'N'</td>
</tr>
<tr>
<td></td>
<td>EXTSEQNO</td>
<td>INTEGER</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DESCSTAT</td>
<td>CHAR(1)</td>
<td></td>
</tr>
<tr>
<td>SYSPACKCOPY</td>
<td>BUSTIMESENSITIVE</td>
<td>CHAR(1)</td>
<td>DEFAULT 'N'</td>
</tr>
<tr>
<td></td>
<td>APPLCOMPAT</td>
<td>VARCHAR(10)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ARCHIVESENSITIVE</td>
<td>CHAR(1)</td>
<td>DEFAULT 'N'</td>
</tr>
<tr>
<td></td>
<td>EXTSEQNO</td>
<td>INTEGER</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DESCSTAT</td>
<td>CHAR(1)</td>
<td></td>
</tr>
<tr>
<td>SYSPACKSTMT</td>
<td>EXPANSION_REASON</td>
<td>CHAR(2)</td>
<td></td>
</tr>
<tr>
<td>SYSPENDINGDDL</td>
<td>COLNAME</td>
<td>VARCHAR(128)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PARTITION</td>
<td>SMALLINT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PARTITION_KEYWORD</td>
<td>VARCHAR(18)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>COLUMN_KEYWORD</td>
<td>VARCHAR(18)</td>
<td></td>
</tr>
<tr>
<td>SYSPLAN</td>
<td>PROGAUTH</td>
<td>CHAR(1)</td>
<td>DEFAULT 'D'</td>
</tr>
</tbody>
</table>
**DSNTIJTC - New columns**

- **SYSQUERY**
  - SELECTVTY_OVERRIDE: CHAR(1)
  - ACCESSPATH_HINT: CHAR(1)
  - OPTION_OVERRIDE: CHAR(1)
  - SELECTIVITY_VALID: CHAR(1)

- **SYSQUERYPLAN**
  - EXPANSION_REASON: CHAR(2)

- **SYSSEQUENCES**
  - SEQSCHEMA: VARCHAR(128)
  - SEQNAME: VARCHAR(128)

- **SYSTABLES**
  - ARCHIVING_SCHEMA: VARCHAR(128)
  - ARCHIVING_TABLE: VARCHAR(128)
  - STATS_FEEDBACK: CHAR(1) DEFAULT 'Y'

- **SYSTABLEPART**
  - RBA_FORMAT: CHAR(1)
  - PCTFREE_UPD: SMALLINT
  - PCTFREE_UPD_CALC: SMALLINT DEFAULT -1

- **SYSTABLESPACESTATS**
  - UPDATESIZE: BIGINT
  - LASTDATACHANGE: TIMESTAMP
DSNTIJTC - New/Altered indexes

- All new indexes BUFFERPOOL BP0 NOT PADDED with Basic 6-byte RBA format

- DSNSQX03 ON SYSSEQUENCES (SEQSCHEMA, SEQNAME)
- DSNTTX02 ON SYSTABSTATS (DBNAME, TSNAME, PARTITION)

ALTER INDEX
- DSNDOB02 ON SYSOBDSDS ADD COLUMN(RBA)
DSNTIJTC - New table spaces

- All table spaces BUFFERPOOL BP0 CCSID UNICODE LOCKSIZE ROW with Basic 6-byte RBA format unless otherwise indicated

- SYSTsixc
- SYSTsqre
- SYSTsqrs
- SYSTssfb
- SYSTsvar BUFFERPOOL BP16K0
- SYSTsvau
## DSNTIJTC - New tables

### SYSINDEXCLEANUP
- **DBNAME**: VARCHAR(24), nullable
- **INDEXSPACE**: VARCHAR(24), nullable
- **ENABLE_DISABLE**: CHAR(1)
- **MONTH_WEEK**: CHAR(1)
- **MONTH**: SMALLINT, nullable
- **DAY**: SMALLINT, nullable
- **START_TIME**: TIME, nullable
- **END_TIME**: TIME, nullable

IN DSNDB06.SYSTSIXC

### INDEX DSNICX01
- (DBNAME, INDEXSPACE)
### DSNTIJTC - New tables

- **SYSQUERYPREDICATE**
  - QUERYID: BIGINT
  - QUERYNO: INTEGER
  - QBLOCKNO: SMALLINT
  - APPLNAME: VARCHAR(24)
  - PROGNAME: VARCHAR(128)
  - PREDNO: INTEGER
  - TYPE: CHAR(8)
  - LEFT_HAND_SIDE: VARCHAR(128)
  - LEFT_HAND_PNO: INTEGER
  - LHS_TABNO: SMALLINT
  - LHS_QBNO: SMALLINT
  - RIGHT_HAND_SIDE: VARCHAR(128)
  - RIGHT_HAND_PNO: INTEGER
  - RHS_TABNO: SMALLINT
  - RHS_QBNO: SMALLINT
  - FILTER_FACTOR: FLOAT
  - BOOLEAN_TERM: CHAR(1)
  - SEARCHARG: CHAR(1)
  - JOIN: CHAR(1)
  - AFTER_JOIN: CHAR(1)
## DSNTIJTC - New tables

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDED_PRED</td>
<td>CHAR(1)</td>
</tr>
<tr>
<td>REDUNDANT_PRED</td>
<td>CHAR(1)</td>
</tr>
<tr>
<td>DIRECT_ACCESS</td>
<td>CHAR(1)</td>
</tr>
<tr>
<td>KEYFIELD</td>
<td>CHAR(1)</td>
</tr>
<tr>
<td>EXPLAIN_TIME</td>
<td>TIMESTAMP</td>
</tr>
<tr>
<td>CATEGORY</td>
<td>SMALLINT</td>
</tr>
<tr>
<td>CATEGORY_B</td>
<td>SMALLINT</td>
</tr>
<tr>
<td>TEXT</td>
<td>VARCHAR(2000)</td>
</tr>
<tr>
<td>PRED_ENCODE</td>
<td>CHAR(1)</td>
</tr>
<tr>
<td>PRED_CCSID</td>
<td>SMALLINT</td>
</tr>
<tr>
<td>PRED_MCCSID</td>
<td>SMALLINT</td>
</tr>
<tr>
<td>MARKER</td>
<td>CHAR(1)</td>
</tr>
<tr>
<td>PARENT_PNO</td>
<td>INTEGER</td>
</tr>
<tr>
<td>NEGATION</td>
<td>CHAR(1)</td>
</tr>
<tr>
<td>LITERALS</td>
<td>VARCHAR(128)</td>
</tr>
<tr>
<td>CLAUSE</td>
<td>CHAR(8)</td>
</tr>
<tr>
<td>GROUP_MEMBER</td>
<td>VARCHAR(24)</td>
</tr>
<tr>
<td>ORIGIN</td>
<td>CHAR(1)</td>
</tr>
<tr>
<td>UNCERTAINTY</td>
<td>FLOAT(4)</td>
</tr>
<tr>
<td>SECTNOI</td>
<td>INTEGER</td>
</tr>
</tbody>
</table>
DSNTIJTC - New tables

- SYSQUERYPREDICATE... COLLID VARCHAR(128)
  VERSION VARCHAR(122)
  PRIMARY KEY (QUERYID,PREDNO)
  FOREIGN KEY DSNQY@QE (QUERYID)
  REFERENCES SYSQUERY (QUERYID)
  ON DELETE CASCADE
  IN DSNDB06.SYSTSQRE

- INDEX DSNQEX01 (QUERYID, PREDNO) UNIQUE
- INDEX DSNQEX02 (QUERYID)
DSNTIJTC - New tables

- **SYSQUERYSEL**
  - QUERYID: BIGINT
  - QUERYNO: INTEGER
  - QBLOCKNO: SMALLINT
  - APPLNAME: VARCHAR(24)
  - PROGNAME: VARCHAR(128)
  - SECTNOI: INTEGER
  - COLLID: VARCHAR(128)
  - VERSION: VARCHAR(122)
  - PREDNO: INTEGER
  - INSTANCE: SMALLINT
  - SELECTIVITY: FLOAT
  - WEIGHT: FLOAT
  - ASSUMPTION: VARCHAR(128)
  - INSERT_TIME: TIMESTAMP
    - GENERATED ALWAYS FOR EACH ROW ON UPDATE
    - AS ROW CHANGE TIMESTAMP
  - EXPLAIN_TIME: TIMESTAMP
  - REMARKS: VARCHAR(762)
DSNTIJTC - New tables

- SYSQUERYSEL...
  PRIMARY KEY
  (QUERYID,PREDNO,INSTANCE)
  FOREIGN KEY DSNQY@QL
  (QUERYID,PREDNO)
  REFERENCES SYSQUERYPREDICATE
  (QUERYID,PREDNO)
  ON DELETE CASCADE
  IN DSNDB06.SYSTSQRS

- INDEX DSNQLX01
  (QUERYID,PREDNO,INSTANCE) UNIQUE
- INDEX DSNQLX02
  (QUERYID,PREDNO)
DSNTIJTC - New tables

- **SYSSTATFEEDBACK**
  - TBCREATOR: VARCHAR(128)
  - TBNAME: VARCHAR(128)
  - IXCREATOR: VARCHAR(128)
  - IXNAME: VARCHAR(128)
  - COLNAME: VARCHAR(128)
  - NUMCOLUMNNS: SMALLINT
  - COLGROUPCOLNO: VARCHAR(254)
  - TYPE: CHAR(1)
  - DBNAME: VARCHAR(24)
  - TSNAME: VARCHAR(24)
  - REASON: CHAR(8)
  - BLOCK_RUNSTATS: CHAR(1)
  - REMARKS: VARCHAR(762)
  - LASTDATE: DATE

  IN DSNDB06.SYSTSSFB

- **INDEX DSNSFX01**
  - (TBCREATOR, TBNAME, IXCREATOR, IXNAME, COLNAME, COLGROUPCOLNO, NUMCOLUMNNS, TYPE)  CLUSTER

- **INDEX DSNSFX02**
  - (TBCREATOR, TBNAME)

- **INDEX DSNSFX03**
  - (IXCREATOR, IXNAME)
## DSNTIJTC - New tables

- **SYSVARIABLES**
  - **VARID**: BIGINT, GENERATED ALWAYS AS IDENTITY

### Column Details
- **SCHEMA**: VARCHAR(128)
- **NAME**: VARCHAR(128)
- **OWNER**: VARCHAR(128)
- **OWNERTYPE**: CHAR(1)
- **RELCREATED**: CHAR(1)
- **CREATEDTS**: TIMESTAMP
- **TYPESCHEMA**: VARCHAR(128)
- **TYPENAME**: VARCHAR(128)
- **DATATYPEID**: INTEGER
- **SOURCETYPEID**: INTEGER
- **LENGTH**: INTEGER
- **SCALE**: SMALLINT
- **CCSID**: INTEGER
- **DEFAULT**: CHAR(3)
- **ROWID**: ROWID, GENERATED ALWAYS
DSNTIJTC - New tables

- SYSVARIABLES...
  - DEFAULTTEXT: CLOB(2M) INLINE LENGTH 2000
  - DESCRIPTOR: BLOB(2M) INLINE LENGTH 2000
  - ENVID: INTEGER
  - REMARKS: VARCHAR(762)
  - IBMREQD: CHAR(1)

- PRIMARY KEY (SCHEMA,NAME)

- IN DSNDB06.SYSTSVAR

- INDEX DSNOVX01 (SCHEMA,NAME) UNIQUE
DSNTIJTC - New tables

- SYSVARIABLEAUTH
  - GRANTOR VARCHAR(128)
  - GRANTORTYPE CHAR(1)
  - GRANTEE VARCHAR(128)
  - GRANTEETYPE CHAR(1)
  - SCHEMA VARCHAR(128)
  - NAME VARCHAR(128)
  - COLLID VARCHAR(128)
  - CONTOKEN CHAR(8)
  - FOR BIT DATA
  - READAUTH CHAR(1)
  - WRITEAUTH CHAR(1)
  - AUTHHOWGOT CHAR(1)
  - GRANTEDTS TIMESTAMP
  - IBMREQD CHAR(1)
  - FOREIGN KEY DSNOV@VA
    (SCHEMA,NAME)
    REFERENCES SYSVARIABLES
    (SCHEMA,NAME)
    ON DELETE CASCADE
  IN DSNDB06.SYSTSSVAU

- INDEX DSNVAX01
  (GRANTEE,GRANTEETYPE,SCHEMA,NAME)

- INDEX DSNVAX02
  (GRANTOR,GRANTORTYPE,SCHEMA,NAME)

- INDEX DSNVAX03
  (SCHEMA,NAME)
DSNTIJTC - New RI

- SYSTABSTATS FOREIGN KEY DSNDP@TT
  (DBNAME,TSNAME,PARTITION)
  REFERENCES SYSTABLEPART
  (DBNAME,TSNAME,PARTITION)
  ON DELETE CASCADE
DSNTIJEN – 0A-0B – Image Copies

- Image copy anything in DSNDB01 or DSNDB06 that is in ICOPY or COPYP
DSNTIJEN – 01 – ENFM START

- CREATE TABLESPACE SYSTSCPY CCSID EBCDIC
  SYSTSISS
  SYSTSTSS
  SYSTSTSS
  SYSTSTSS
  SYSTSTSS
  SYSTSTSS
  SYSTSTSS
  SYSTSTSS
  SYSTSTSS
  SYSTSTSS
  SYSTSTSS
  SYSTSTSS
  SYSTSTSS
  SYSTSTSS
  SYSTSTSS
  SYSTSTSS
  SYSTSTSS
  SYSTSTSS
  SYSTSTSS
  SYSTSTSS
  SYSTSTSS

- ALTER TABLE SYSPENDINGDDL
  ALTER COLUMN CREATEDTS TIMESTAMP(12)
DSNTIJEN – 01 – ENFM START

- SYSCOLAUTH FOREIGN KEY DSNDF@AC
  (CREATOR, TNAME, COLNAME)
  REFERENCES SYSCOLUMNS
  (TBCREATOR, TBNAME, NAME)
  ON DELETE CASCADE

- VARIABLE SYSIBMADM.GET_ARCHIVE CHAR(1) DEFAULT "N"
- VARIABLE SYSIBMADM.MOVE_TO_ARCHIVE CHAR(1) DEFAULT "N"
- VARIABLE SYSIBM.CLIENT_IPADDR CHAR(39)

- DELETE FROM SYSDATABASE WHERE DBID=1 (Scan - BP0)
- DELETE FROM SYSTABLESPACE WHERE DBID=1 (Scan - BP0)
DSNTIJEN – 01 – ENFM START

- "CREATE DATABASE" DSNDB01

- "CREATE TABLESPACE" SYSLGRNX SYSUTILX BUFFERPOOL BP32K
DBD01
SCT02
SPT01

- "CREATE TABLESPACE" SYSUTILX BUFFERPOOL BP32K
“CREATE TABLE” SYSLGRNX

LGRDBID CHAR(2)
LGRPSID CHAR(2)
LGRUCDT CHAR(6)
LGRUCTM CHAR(8)
LGRSRBA CHAR(6)
LGRSPBA CHAR(6)
LGRPART SMALLINT
LGRSLRSN CHAR(6)
LGRELRSN CHAR(6)
LGRMEMB CHAR(2)

IN DSNDB01.SYSLGRNX

INDEX DSNLLX01 (LGRDBID,LGRPSID,LGRPART,LGRMEMB,LGRSLRSN)
UNIQUE

INDEX DSNLLX02 (LGRDBID,LGRPSID,LGRSLRSN)

Note the RBA and LRSN columns are still CHAR(6) because that's the way the data currently exists in SYSLGRNX.
“CREATE TABLE” SYSUTIL

- USUUUID: CHAR(16)
- USUJOBNM: CHAR(8)
- USUAUUID: CHAR(8)
- USURDATE: CHAR(4)
- USUREL: CHAR(3)
- USUIRQD: CHAR(1)
- USULSIZEx: CHAR(4)
- USULCURM: CHAR(4)
- USUUTNAM: CHAR(8)
- USUPHASE: CHAR(8)
- USUDSNU: CHAR(8)
- USUDSNU2: CHAR(2)
- USUSTATU: CHAR(1)
- USUTREQ: CHAR(1)
- USUFORCE: CHAR(1)
- USURLOK: CHAR(1)
- USUCMPOK: CHAR(1)
- USURSFLG: CHAR(1)
- USURTFLG: CHAR(1)
- USURTFLG2: CHAR(1)
“CREATE TABLE” SYSUTIL ...

USUPOS CHAR(4)
USUDONE CHAR(8)
USUCKSUM CHAR(4)
USUDBOB CHAR(2)
USUPSID CHAR(2)
USUPSDD CHAR(2)
USUCATMGFRM CHAR(1)
USOOFLAG CHAR(1)
USUBDBNAM CHAR(8)
USUSPNAM CHAR(8)
USUMEMBR CHAR(8)
USUOCATR CHAR(1)
USUOCATV CHAR(1)
USUCATCV CHAR(1)
USUOCATCV CHAR(1)
USUUDA CHAR(150)
USURTIME CHAR(4)
USURLSN CHAR(6)
USURDATO CHAR(4)
USURTIMO CHAR(4)
“CREATE TABLE” SYSUTIL ...

USURLSNO CHAR(6)
USUR5 CHAR(10)
USUCNTR CHAR(4)
USUR6 CHAR(92)
USUUSTRN VARCHAR(27000)

IN DSNDB01.SYSUTILX

INDEX DSNLUX01 (USUUUID) UNIQUE
INDEX DSNLUX02 (UTILID,SEQNO) UNIQUE

Note the RBA and LRSN columns are still CHAR(6) because that's the way the data currently exists in SYSLUTIL.
### DSNTIJEN – 01 – ENFM START

- **“CREATE TABLE” SYSUTILX**

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTILID</td>
<td>CHAR(16)</td>
</tr>
<tr>
<td>SEQNO</td>
<td>CHAR(2)</td>
</tr>
<tr>
<td>RESV1</td>
<td>CHAR(12)</td>
</tr>
<tr>
<td>CHECKPOINT</td>
<td>VARCHAR(32000)</td>
</tr>
<tr>
<td>FOREIGN KEY</td>
<td>DSN1T@1T</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>SYSUTIL</td>
</tr>
<tr>
<td>ON DELETE</td>
<td>CASCADE</td>
</tr>
<tr>
<td>IN</td>
<td>DSNDB01.SYSUTILX</td>
</tr>
</tbody>
</table>

  FOREIGN KEY DSN1T@1T
  (UTILID)
  REFERENCES SYSUTIL
  (USUUUID)
  ON DELETE CASCADE
  IN DSNDB01.SYSUTILX
### DSNTIJEN – 01 – ENFM START

- "CREATE TABLE" DBDR
  
<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SECLEN</td>
<td>INTEGER</td>
</tr>
<tr>
<td>DBID</td>
<td>SMALLINT</td>
</tr>
<tr>
<td>SECTION</td>
<td>SMALLINT</td>
</tr>
<tr>
<td>DBD_ROWID</td>
<td>ROWID</td>
</tr>
<tr>
<td>DBD_DATA</td>
<td>BLOB(2G)</td>
</tr>
</tbody>
</table>

  In DSNDB01.DBD01

- INDEX DSNDB01X (DBID,SECTION) UNIQUE

- INDEX DSNDB01X (DBID,SECTION) UNIQUE

- INDEX DSNDB01X (DBID,SECTION) UNIQUE

- INDEX DSNDB01X (DBID,SECTION) UNIQUE
“CREATE TABLE” SCTR
SCTLL INTEGER
SCTNAME CHAR(14)
SCTDAT VARCHAR(4028)
IN DSNDB01.SCT02

INDEX DSNSCT02 (SCTNAME) UNIQUE
“CREATE TABLE” SPTR

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPTLL</td>
<td>INTEGER</td>
</tr>
<tr>
<td>SPTLOCID</td>
<td>VARCHAR(128)</td>
</tr>
<tr>
<td>SPTCOLID</td>
<td>VARCHAR(128)</td>
</tr>
<tr>
<td>SPTNAME</td>
<td>VARCHAR(128)</td>
</tr>
<tr>
<td>SPTCONID</td>
<td>CHAR(8)</td>
</tr>
<tr>
<td>SPTRESV</td>
<td>CHAR(2)</td>
</tr>
<tr>
<td>SPTSEC</td>
<td>CHAR(4)</td>
</tr>
<tr>
<td>SPTSEQ</td>
<td>CHAR(2)</td>
</tr>
<tr>
<td>SPTBODY</td>
<td>VARCHAR(1)</td>
</tr>
<tr>
<td>SPTVER</td>
<td>VARCHAR(64)</td>
</tr>
<tr>
<td>SPT_ROWID</td>
<td>ROWID</td>
</tr>
<tr>
<td>SPT_DATA</td>
<td>BLOB(2G)</td>
</tr>
<tr>
<td>SPT_EXPLAIN</td>
<td>BLOB(2G)</td>
</tr>
</tbody>
</table>

IN DSNDB01.SPT01

INDEX DSNSPT01 (SPTLOCID,SPTCOLID,SPTNAME,SPTCONID, SPTRESV,SPTSEC,SPTSEQ) UNIQUE

INDEX DSNSPT02 (SPTVER,SPTLOCID,SPTCOLID,SPTNAME, SPTCONID,SPTRESV,SPTSEC,SPTSEQ) UNIQUE

Note the inline LOB length is irrelevant in this context and is always 32146
DSNTIJEN – 10 – ENFM SYSUTILX

- “ALTER TABLE” SYSUTIL ...

  ...  
  USUCNTR  CHAR(4)  
  USURLSNX  CHAR(10)  
  USURLSOX  CHAR(10)  
  USUR6  CHAR(72)  
  USUUSTRN  VARCHAR(27000)  
  IN DSNDB01.SYSUTILX  

- “Reset” SYSUTILX

- Note that this step requires that there be no utilities recorded in SYSUTILX
DSNTIJEN – 20-28 – ENFM SYSLGRNX

- **ALTER TABLE** SYSLGRNX
  - LGRSRBA CHAR(10)
  - LGRSPBBA CHAR(10)
  - LGRSLRNSN CHAR(10)
  - LGRELRSN CHAR(10)

- **REORG SYSLGRNX ==> SYSLGRNX**
  - SHRLEVEL CHANGE
DSNTIJEN – 30-38 – ENFM SYSCOPY

- ALTER TABLE SYSCOPY
  - START_RBA CHAR(10)
  - PIT_RBA CHAR(10)

- REORG SYSCOPY ==> SYSTSCPY SHRLLEVEL REFERENCE
DSNTIJEN – 40-48 – ENFM SYSRTSTS

- ALTER TABLE SYSINDEXSPACESTATS
  COPYUPDATELRSN CHAR(10)

- ALTER TABLE SYSTABLESPACESTATS
  COPYUPDATELRSN CHAR(10)

- REORG SYSRTSTS ==> SYSTSISS
  SYSTSTSS
  SHRLEVEL REFERENCE
DSNTIJEN – 50-57 – ENFM SYSTSIXS

- ALTER TABLE SYSINDEXES
  COPYLRSN CHAR(10)

- REORG SYSTSIXS ==> SYSTSIXS SHRLEVEL CHANGE
DSNTIJEN – 60-67 – ENFM SYSTSTAB

- ALTER TABLE SYSTABLES
  - RBA1 CHAR(10)
  - RBA2 CHAR(10)

- REORG SYSTSTAB ==> SYSTSTAB SHRLEVEL CHANGE
DSNTIJEN – 70-78 – ENFM SYSSTR

- ALTER TABLE SYSCHECKS
  RBA CHAR(10)

- REORG SYSSTR ==> SYSTSCHX SYSTSCKD SYSTSCKS SYSTSSRG SHRLEVEL REFERENCE
DSNTIJEN – ENFM CLEANUP

- DROP SYSCOOPY
- DROP SYSRTSTS
- DROP SYSSTR
DSNTIJNF – New Function Mode

- For data sharing, rebuild the SCA to handle large RBAs
- Group restart in ENFM will also rebuild the SCA
Questions and Answers

Thank You