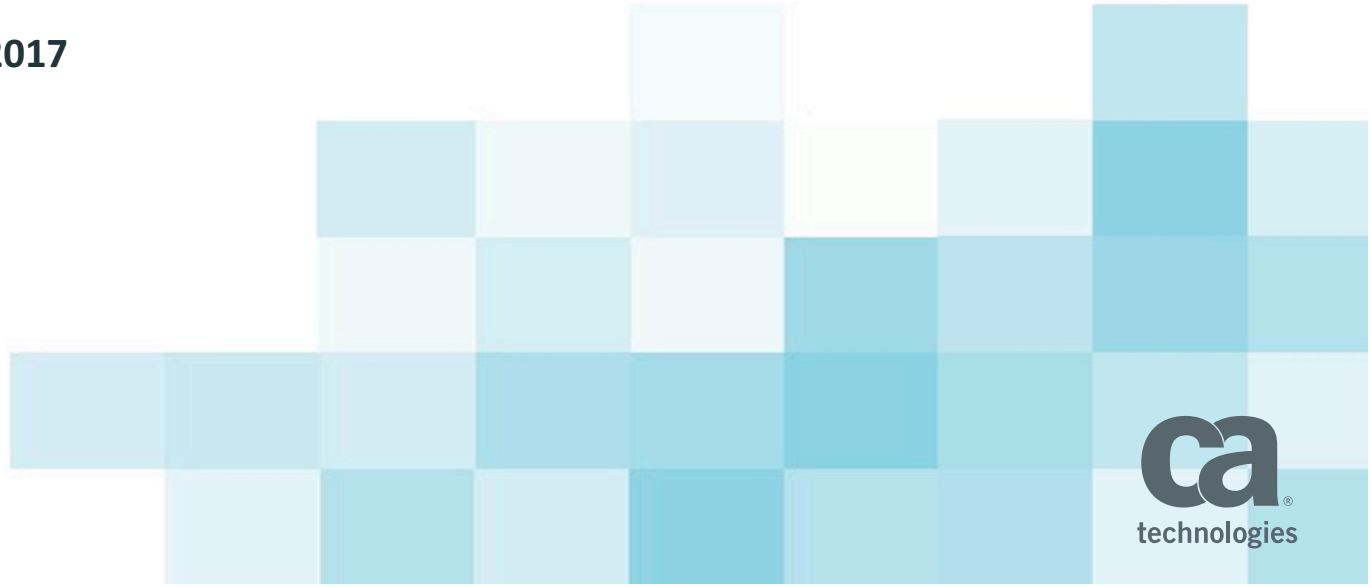
PBR RPN - Removing Partitioning restrictions in Db2 12 for z/OS

Steve Thomas - CA Technologies

Belgian GSE Meeting – December 7th 2017



GSE GUIDE SHARE EUROPE

ca technologies

Agenda

- Current Limitations in Db2 for z/OS Partitioning
- Evolution of partitioned tablespaces
- Relative Page Numbering in Db2 12 for z/OS
- Creating and Altering PBR RPN objects
- Migration, Utilities and Operational Considerations
- Summary and references

Why were changes needed?

- Planning for future at design time is hard
  - Max No. of partitions depends on partition size and page size
- DSSIZE is at a tablespace level and changing it is disruptive
- Cannot add a partition in the middle
- REBALANCE must run against multiple partitions
- Index and XML DSSIZE dependence
- Need for more or larger partitions?
- Maximum table size not comparable to other RDBMS

Comparison with other RDBMS

- Maximum Table Sizes in other databases:
  - MS SQL - 524,272Tb
  - Oracle – 4Gb * block size (with BIGFILE tablespace)
  - Db2 11 for z/OS – 16Tb (4k pages), 128Tb (32k pages)
  - Db2 (was Db2 for LUW) – 2 Zb
  - As of Wikipedia October 2016 – link in notes
- Db2 12 for z/OS - 4Pb (1Tb/partition * 4096 partitions)
  - 256 trillion rows in a single table (4k pages)
  - Theoretical limit of Architecture is much higher

Comparing Partition limits

Db2 11 for z/OS

- Max no of partitions = 4096
- Max partition size = 256Gb
  - Only powers of 2
  - Only 64 partitions for 4kB pages
- Max table size = 128Tb
  - Using 32k pages & 32Gb DSSIZE
- Number of partitions, partition size, and page size are bound

Db2 12 for z/OS

- Max no of partitions = 4096
- Max partition size = 1Tb
  - Any number of Gigabytes
  - Partitions can have different sizes
- Max table size = 4Pb
  - Independent of Page size
- Less disruptive partition management

A little history – Db2 for z/OS V1 in 1983

- Index Controlled Partitioning (ICP) – now Classic Partitioning
- 4 bytes RID (Record ID)
  - 3 bytes page number + 1 byte ID entry
- Maximum partitions = 64, Largest object = 64Gb
- Relation between number of partitions and Maximum size

Number of partitions	Max partition size (GB)
1 – 16	4
17 - 32	2
33 - 64	1

First changes not until Version 5 (1997)

- RID expanded to 5 bytes
  - 4 bytes page number + 1 byte ID entry
- Largest table or tablespace = 1,016Gb
- Introduces LARGE tablespaces
  - LARGE implied if NUMPARTS > 64
- New limits for LARGE tablespaces
  - Maximum number of partitions = 254
  - Maximum partition size 4Gb
- Non-LARGE tablespaces retained old limits

Number of partitions	Max partition size (GB)
1 – 16	4
17 - 32	2
33 - 64	1
65 - 254	4

This didn’t last long - until Version 6 (1998)

- DSSIZE replaced LARGE
  - LARGE deprecated – supported but DSSIZE preferred
- Partition size up to 64Gb
  - Size >4Gb requires extended format and extended addressing
- Up to 254 partitions of 64 GB
  - Largest tablespace = 16 TB

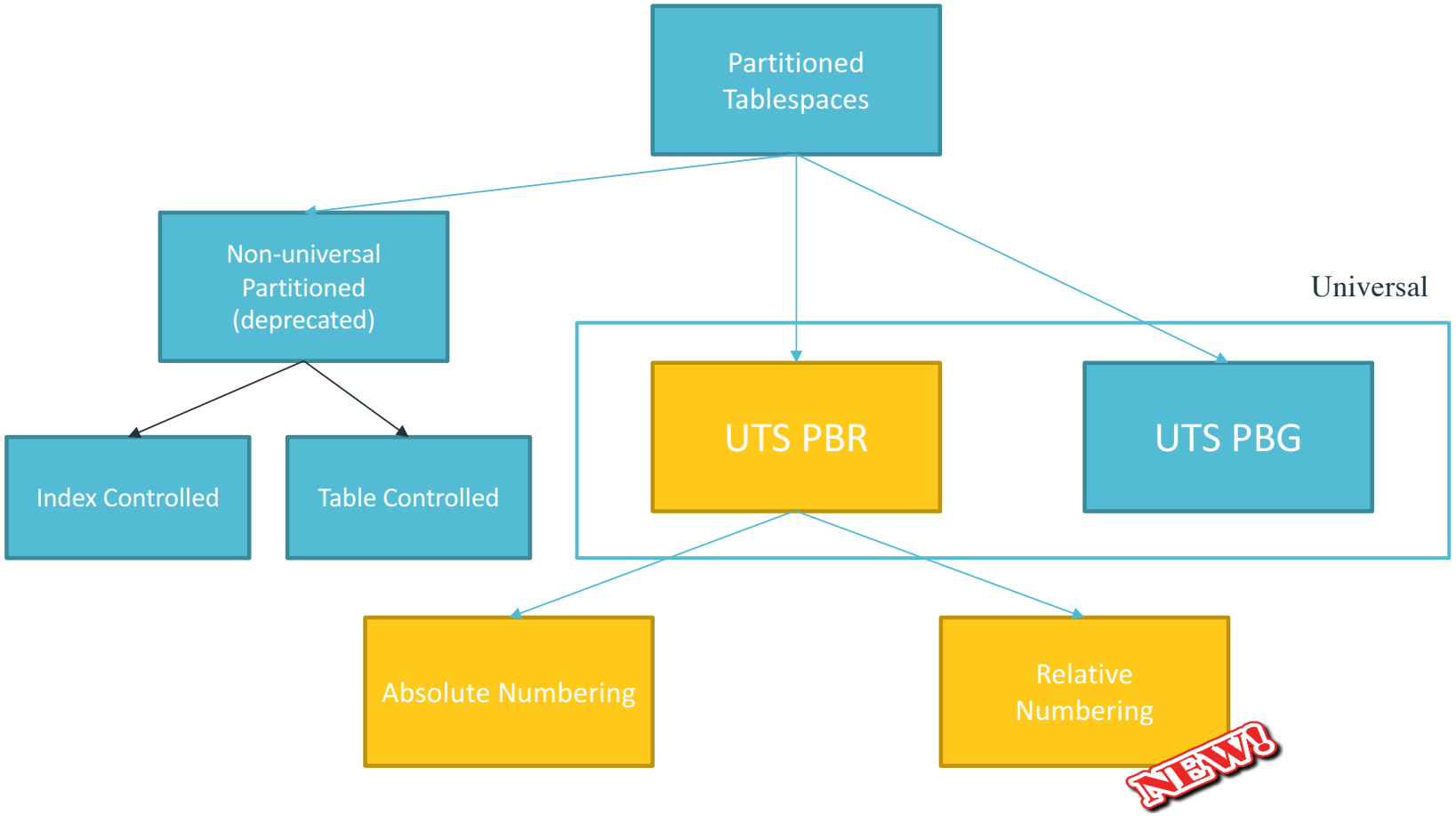
Pace of change then accelerated

- Version 8 (2004)
  - Table Controlled partitioning (TCP)
  - Data partitioning secondary indexes
  - Up to 4,096 partitions, Largest tablespace = 128Tb
  - If NUMPARTS > 254, then max DSSIZE depends on the page size
- Version 9 (2007)
  - UTS – Partitioned By Range (PBR) or by Growth (PBG)
  - No changes in limits (except for NPI)

And finally we have Db2 12 for z/OS (2016)

- New type of UTS PBR – **Relative Page Numbering (RPN)**
- Record ID (RID) increased from 5 bytes to 7 bytes
- Partition size in PBR RPN can be up to 1Tb
  - Total size of PBR RPN up to 4Pb and 256 trillion rows
  - Should keep us happy for a while!
- Removes many other limitations
  - DSSIZE no longer at the table space level
  - Max No. of partitions does not depend on DSSIZE and page size
  - Ability to insert a partition in the middle of a tablespace
  - Index and XML DSSIZE independence

Partitioned TS options in Db2 12 for z/OS

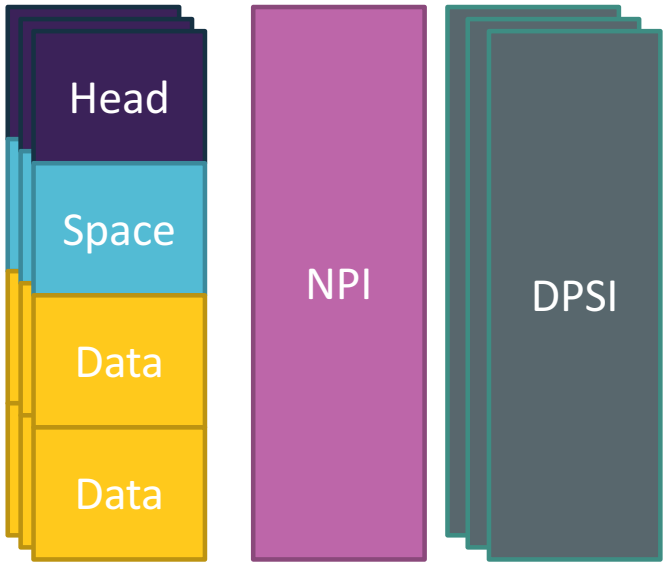


Partitioned Tablespaces in Version 12

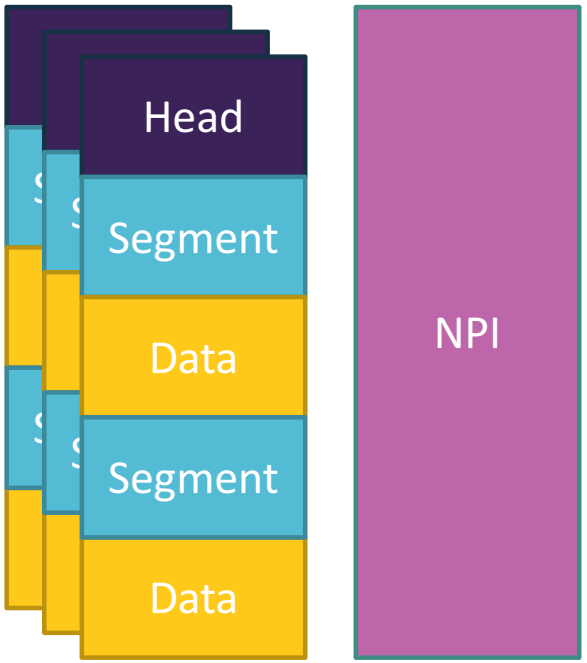
- Classic partitioned (Non-UTS) tablespaces deprecated
  - Covers both Index Controlled (ICP) and Table Controlled (TCP)
    - DSNZPARM DPSEGSZ controls whether they can be created at all
    - DSNZPARM PREVENT_NEW_IXCTRL_PART controls ICP creation
  - Default Partitioned object is now a PBR with SEGSIZE 32
- Partition by growth (PBG) universal table spaces
  - Defined using both MAXPARTITIONS and SEGSIZE
  - Can grow up to 128TB
  - Partitioned indexes are not supported
  - A non-partitioning index (NPI) always uses a 5 byte RID
  - Use absolute numbering

TCP vs PBG visually

Classic (TCP)



PBG

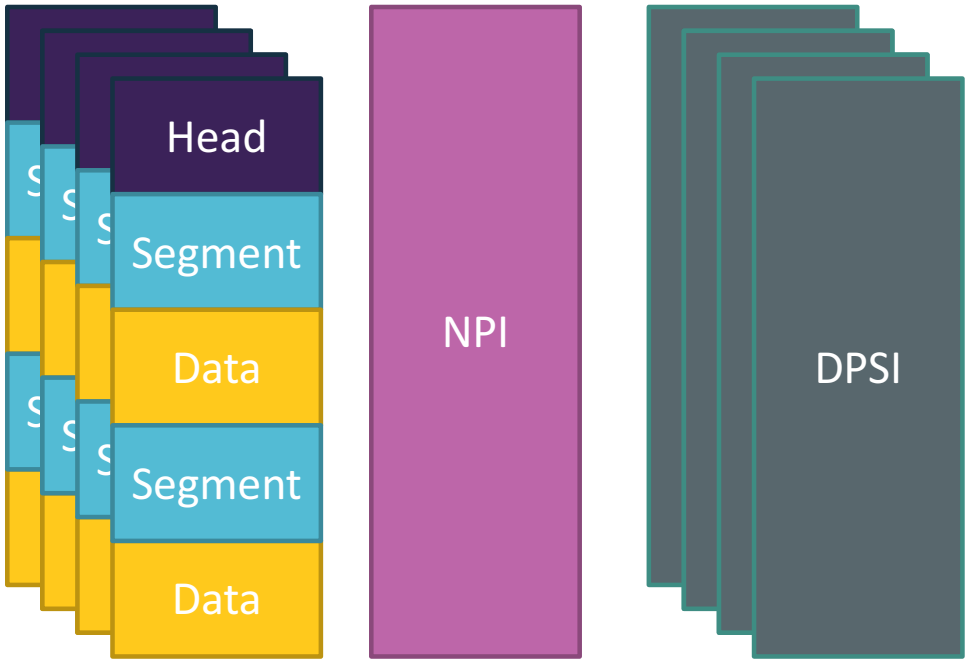


PBR changes in Db2 12 for z/OS

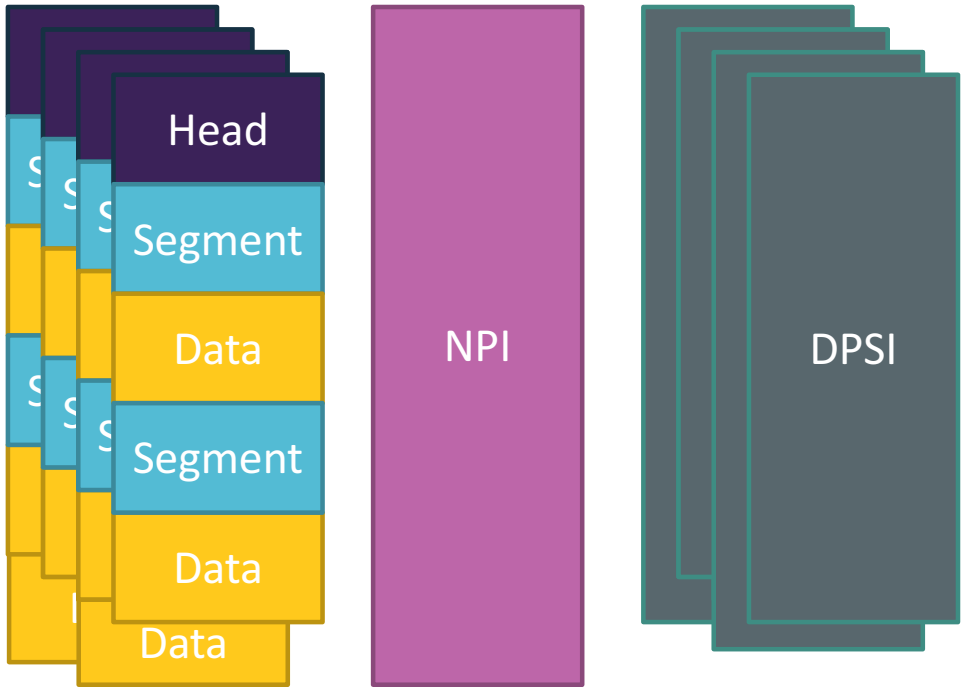
- Created using both NUMPARTS and SEGSIZE
- New PAGENUM ABSOLUTE | RELATIVE clause distinguishes between absolute and relative numbering
- Any type of index supported on a table in a range-partitioned table space
  - Partitioned or non-partitioned
  - Partitioning index is no longer required
- The rest of this presentation will focus on PBR only

PBR – Absolute vs Relative

Absolute Numbering



Relative Numbering



ca technologies

PBR RPN – Index considerations

- Index partitions now independent from base table partition size
  - Maximum index partition size is 1 TB
  - DSSIZE no longer needs to be in powers of 2
- Indexes on PBR RPN will still inherit default values
- NPIs on PBR RPN same characteristics as pre-Db2 12, except:
  - RID is 7 bytes
  - For 4K, 8K, 16K, or 32K index page size, maximum index space size will continue to be limited to 16Tb, 32Tb, 64T or 128 Tb
  - Maximum PIECESIZE limited to DSSIZE of table space, with maximum 256 GB; default is 4 GB

Max Partitions – Absolute numbering

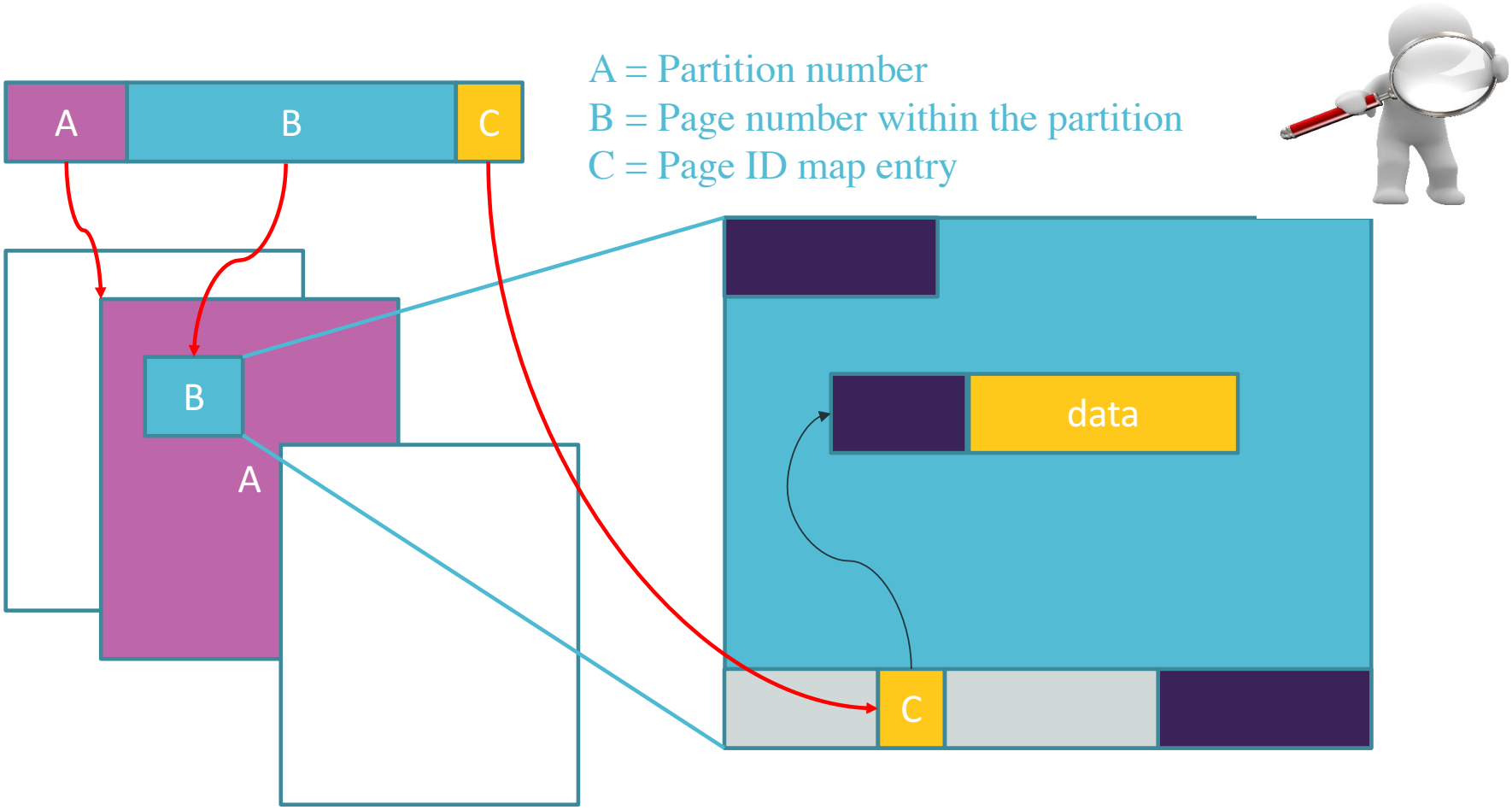
- Maximum number of partitions that a tablespace with absolute numbering can have is still driven by the table:

RID	Page size (KB)	DSSIZE (GB)	Max # of partitions	Total TS size (TB)
5 byte EA	4	1	4096	4
5 byte EA	4	256	64	16
5 byte EA	32	1	4096	4
5 byte EA	32	256	512	128
5 byte non EA (LARGE)	4	4	4096	16

Record Identifiers (RIDs)

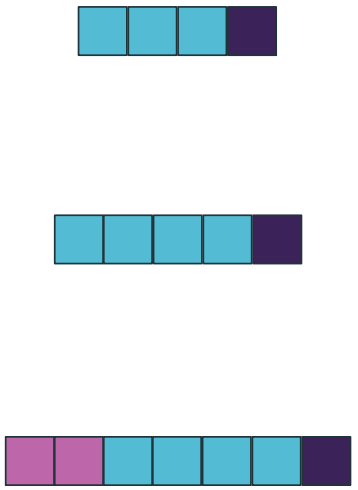
- A unique identifier that Db2 uses internally to identify a row
  - Data and Index Pages
  - Db2 Transaction Log
  - Utilities – e.g. RID Mapping Table in Online REORG
  - RIDPool during SQL Evaluation – List Prefetch
  - … and more
- A RID consists of the following elements:
  - A = Partition number (e.g. Partition 5)
  - B = Page number within the partition (e.g. Page 100)
  - C = Page ID map entry (e.g. 7th row in the Page)

How it all hooks together



RID Length changes

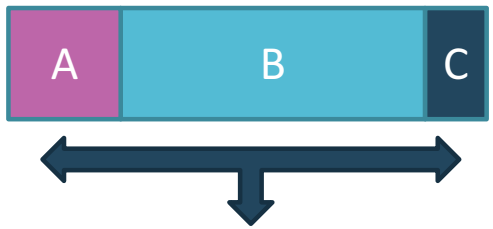
- 4 bytes in Db2 v1:
  - A and B share 3 bytes (24 bits)
  - C = 1 byte
- 5 bytes in Db2 v5:
  - A and B share 4 bytes (32 bits)
  - C = 1 byte
- And now 7 bytes in Db2 12:
  - A = 2 bytes,
  - B = 4 bytes,
  - C = 1 byte



A = Partition number
B = Page number
C = Page ID map entry

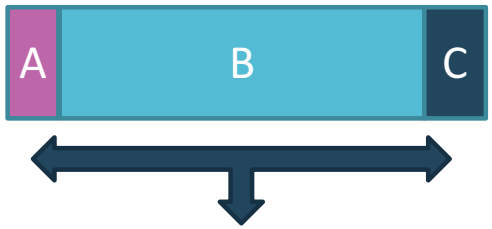
Explains link between partition numbers and sizes

- Db2 11 or Absolute page numbering in Db2 12
  - A and B share a fixed number of bits = 32 (5 byte RID)
  - A and B lengths are variable, resulting in many combinations



12 bits for Partitions (4,096)
20 bits left for Page Numbers

. . .



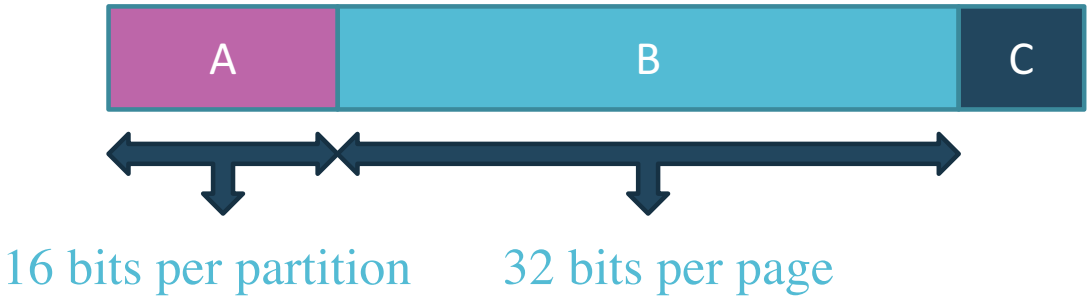
6 bits for Partitions (64)
26 bits left for Page Numbers

  - More Partitions results in fewer pages being available in each partition

A = Partition number
B = Page number
C = Page ID map entry

Solution is Relative Page Numbering

- A and B parts are both fixed, A = 16 bits, B = 32 bits
- Theoretical Architectural limit 65,536 partitions, 2^32 pages (8 ZB)
  - Note these have not been implemented yet!



A = Partition number
B = Page number
C = Page ID map entry

What is a Page Number?

- Absolute numbering: combination of partition & page numbers
  - Can determine the partition from the page number
- Relative numbering: just page number – within the partition
  - No way to determine partition number just from the page number
- Can be found in
  - Physical pages – see DSN1PRNT
  - Db2 log records – see DSN1LOGP
  - Control blocks – IFCIDs, internal control blocks, …
  - Can be specified for some utilities

How to create a PBR RPN

- CREATE TABLESPACE & CREATE TABLE PAGENUM clause
  - PAGENUM ABSOLUTE | RELATIVE
  - Explicit (CREATE TABLESPACE) vs implicit tablespace (CREATE TABLE)
  - Must also use Partition By Range specification
  - Supported from Function level V12R1M500

Some Notes

- Default defined by PAGESET_PAGENUM DSNZPARM
  - Db2 ships with a default of 'A' – works as before
- Data partition sizes can be up to 1Tb – default is 4Gb
  - No dependency on number of partitions
  - Supports any value from 1Gb to 1,024Gb - no need for DSSIZE to be powers of 2 as before
- Datasets must be Extended Format and Addressability
- Increased RID size has a knock on effect on Db2 log records
  - Tablespace records now 20 bytes larger, Index records 28 bytes larger
  - Applies to Db2 12 for z/OS regardless of whether you use RPN or not

Restrictions

- PAGENUM RELATIVE cannot be used for:
  - Tablespace organized by hash
  - Workfile tablespaces
  - LOB tablespaces
  - Clones
- 3 bytes minimum row data size
  - SQLCODE = -270, ERROR: FUNCTION NOT SUPPORTED

Tables with XML data

- Underlying XML table space implicitly created with the same PAGENUM attribute as the base table space.
  - You can ALTER the DSSIZE for a partition of an XML tablespace
- Default DSSIZEs for XML tablespaces:

Base table DSSIZE (GB)	4KB PGSIZE	8KB PGSIZE	16KB PGSIZE	32KB PGSIZE
1-4	4	4	4	4
…				
33-64	64	64	64	32
…				
513-1024	1024	1024	1024	512

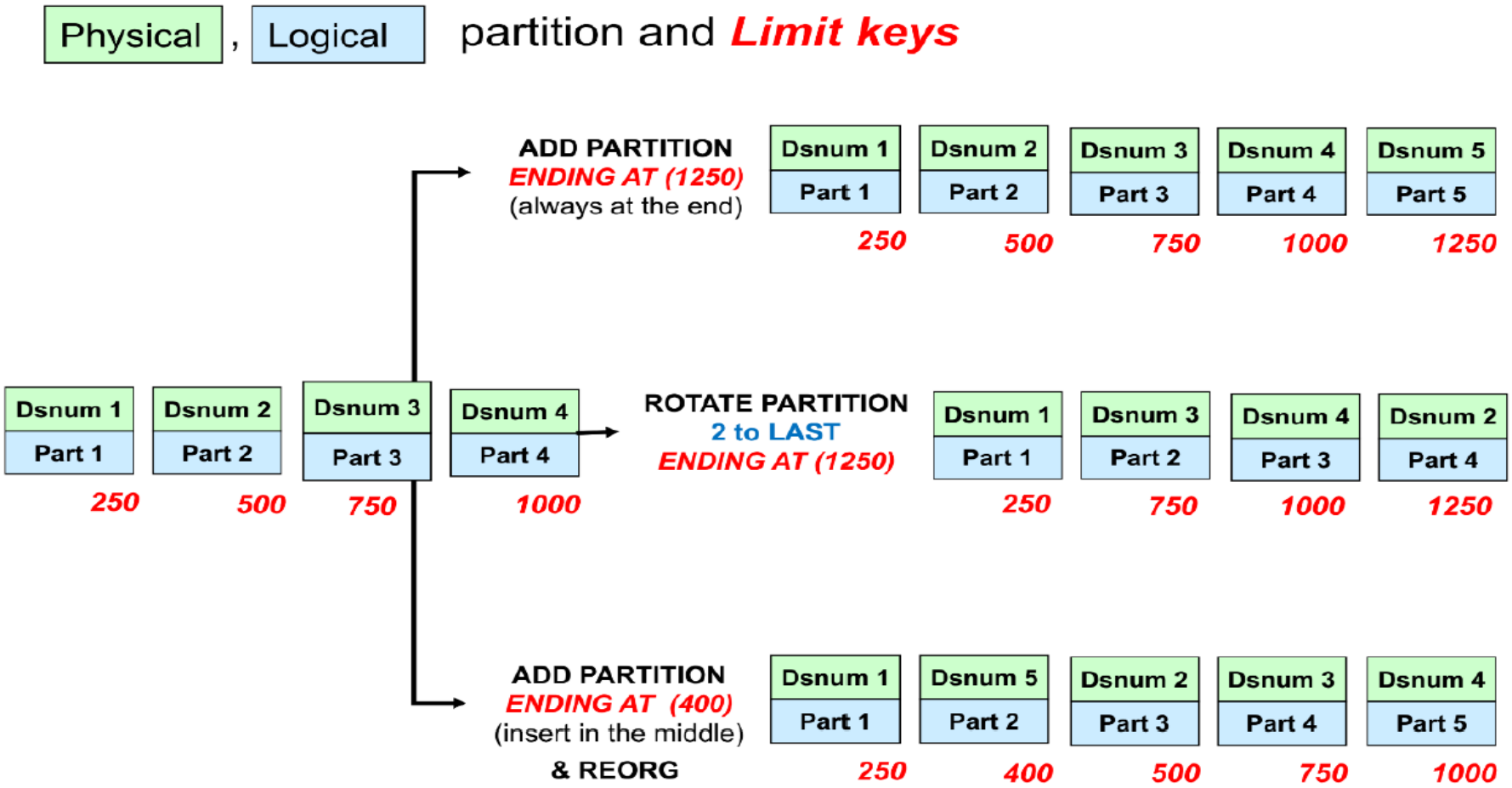
Indexes

- DSSIZE supported for a partitioned index on an RPN Tablespace
  - Specifies maximum size for each partition – any Integer from 1-1024Gb
- DSSIZE > 4Gb requires extended format and addressability
- The value is provided by:
  - DSSIZE value from the PARTITION clause for that partition
  - DSSIZE keyword not in any PARTITION clause
  - The default value is inherited from the base table space

Now for the real benefits…

- Altering DSSIZE supported at the partition level
  - Upwards is an immediate change – No REORG
  - Downwards still a Pending change at the Tablespace level using AREOR
  - Can be also applied to partitioned indexes and XML objects
- Insert a partition – split existing partition
  - Pending ALTER
    - Can REORG only the affected partition
    - See next slide for an example
  - Supported by all types of UTS PBR not just RPN
  - Not supported when LOB or XML objects involved

Inserting and Rotating Partitions



Source: IDUG Whitepaper, http://www.idug.org/db2v12whitepaper

Converting to PBR RPN

- How depends on your starting point
- From PBR – ALTER TABLESPACE PAGENUM RELATIVE
  - Conversion is a pending alter – tablespace level REORG required
    - Note BRF deprecation - after V12R1M500, REORG converts any rows in BRF to RRF
    - Also consider converting to extended RBA if this hasn't been done
- From Classic partitioned tablespace
  - Must convert to PBR first
  - Could mean ICP->TCP->UTS PBR Absolute->UTS PBR Relative
  - Note changes to PBR and to RPN can be implemented in 1 REORG
    - You will have 2 pending changes in SYSPENDINGDDL

Catalog changes for PBR RPN

- Catalog changes to reflect the new TS type and new possibilities at the partition level
  - PAGENUM column for tablespaces and indexes
  - DSSIZE for partitions and indexes
- How to determine PBR RPN objects in catalog
  - SELECT * FROM SYSIBM.SYSTABLESPACE
    WHERE PAGENUM = 'R';
- SYSPENDINGDDL
  - New columns to support Inserting a partition
  - REORG_SCOPE_LOWPART, REORG_SCOPE_HIGHPART

Some PBR RPN catalog changes

- **SYSIBM.SYSCOPY**
  - TTYPE for ICTYPE = A and STYPE=P
- **SYSIBM.SYSINDEXES**
  - PAGENUM – A, R, or NULL (A for NPI)
  - DSSIZE for partitioned indexes
- **SYSIBM.SYSINDEXPART**
  - PAGENUM – A, R, or NULL (A for NPI)
  - DSSIZE for partitioned indexes
- **SYSIBM.SYSTABLEPART**
  - TYPE – R for PBR
  - PAGENUM – A, R, or NULL
  - DSSIZE – maximum size of partition
- **SYSIBM.SYSTABLESPACE**
  - PAGENUM – A, R, or NULL

Support in utilities

- REPAIR CATALOG TABLESPACE xxx TEST
  - Check consistency of the object and the information in the catalog
  - Syntax changes due to 7 bytes RID and relative numbering
    - ROWID 2 bytes longer longer and new PART option for RPN objects used alongside PAGE
- CHECK DATA
  - Exception tables must contain 7 bytes RID (not required column)
- RECOVER
  - Recover before materializing PAGENUM RELATIVE not supported
  - RECOVER PAGE must be used with DSNUM for PBR RPN

Support in REORG

- Inline copies must be on partition level
- New RID size in mapping table for PBR RPN:

```
CREATE TABLE MYMAPPING_TABLE
   (TYPE CHAR( 01 ) NOT NULL,
    SOURCE_RID CHAR( 07 ) NOT NULL,
    TARGET_XRID CHAR( 11 ) NOT NULL,
    LRSN CHAR( 10 ) NOT NULL)
```

  - Originally DSNU366I if using wrong size but now DB2 auto-creates a new mapping table
- Tablespaces with 5 byte RIDs can use older mapping table
- Automatically created mapping tables use 7 bytes RID

Inline copies

- Inline copies used in Online REORG and LOAD may contain duplicate data pages and some might be out of sequence
  - Problem: how to determine the correct partition for each page?
  - Db2 cannot decide so partition level copies are required
  - New error message DSNU2922I
- Note this may cause problems if you take Tape based copies…
  - You now need a unit for each partition
  - Do you have enough?
  - Ideally move copies to disk and archive to tape after the job
  - IBM talking about providing some relief shortly – possible new TAPEUNITS option

Service Aids

- DSN1PRNT
  - New PART keyword for PBR RPN, ignored for absolute numbering.
  - Only one partition can be specified.
  - Always prints partition number (even for absolute numbering)
    - Example: PARTITION: # 0002 PAGE: # 00008000
- DSN1LOGP
  - New PART keyword – hex constant, 1-4 chars. Up to 10 PARTs
  - Prints the partition number if included in the log record
  - RID – hex constant, up to 10 characters
    - First 8 are page number, last two are page ID map entry
  - PAGE – hex constant up to 8 characters

Installation and migration

- BSDS conversion required
  - Db2 12 always writes 10 byte RBA/LRNS expanded log records
  - V12R1M500 and above uses Db2 12 log records for all objects
  - V12R1M100 internally uses longer log records, but writes Db2 11 compatible records
- New/updated DSNZPARMs related to PBR RPN
  - PREVENT_NEW_IXCTRL_PART - default is YES
  - PAGESET_PAGENUM – default is A
- Multiple IFCID changes – check documentation

Index considerations

- Indexes will consume more space due to extended RID
  - 2 bytes more per RID
- REORG of tablespace altered to RPN will also reformat the indexes to RPN
  - SYSCOPY will have entries for tablespaces and indexes
  - Affects only Indexes for PBR RPN tablespaces
  - RID stays 5 bytes in other types of tablespaces

Possibly affected SQL

- Do you use Direct row access?
  - PRIMARY_ACCESSTYPE='D'
  - Unless you do anything tricky, no changes needed
- RID(table-designator)
  - Returns a RID of a row – often used with Optimistic Locking
  - The result is a BIGINT value (8 bytes) even in prior versions
  - Do not change the returned value
- ROWID function and column type
  - Value is transparent to user

RID pool considerations

- Db2 uses RID pool for:
  - Enforcing unique keys for multi-row updates
  - List prefetch, including single index list prefetch access paths
  - Multiple index access paths and Hybrid Joins
- DSNZPARM MAXRBLK controls the maximum RID Pool size
  - Managing performance Guide:
    - No. of concurrent RID activities * average no. of RIDs * 2 * 5 (bytes per RID)
  - Installation guide formula:
    - No. of concurrent RID activities * average No. of RIDs * 2 * 8 (bytes per RID)
  - Safe to say it will go up in size!

Summary

- Universal Tablespaces strategic – convert as soon as possible
- PBR RPN allows more partitions and has new size limits
- PBR RPN allow data set size on partition level
- Can now insert partitions into a Tablespace
  - Via Splitting existing partition - also applies to absolute numbering
- Less disruptive partition management
- Conversion to PBR RPN requires tablespace level REORG
- Be aware of inline copies, new Db2 pages and log formats

References and Thanks

- Many thanks to my colleague Emil Kotrc producing much of the original material
- Db2 12 for z/OS, An IDUG User Technical Perspective Whitepaper
  - http://www.idug.org/db2v12whitepaper
- IBM Db2 12 for z/OS Technical Overview Redbook
  - http://www.redbooks.ibm.com/abstracts/sg248383.html?Open
- Db2 12 Documentation
  - http://www.ibm.com/support/knowledgecenter/SSEPEK_12.0.0/home/src/tpc/db2z_12_prodhome.html
- Db2 for z/OS: UTS Conversion Whitepaper and Blog
  - http://www-01.ibm.com/support/docview.wss?uid=swg27047046
  - https://www.ibm.com/developerworks/community/blogs/0399c6ff-7881-490a-a3e6-a65909a40085/entry/What_is_the_recommended_method_to_migrate_Classic_Partitioned_Table_Spaces_to_Partitioned_By_Range_Table_Spaces?lang=en
- VSAM demystified Redbook
  - http://www.redbooks.ibm.com/abstracts/sg246105.html?Open

Steve Thomas
Principal Engineering Services Architect
Steve.Thomas@ca.com

@Steve_db2
slideshare.net/CAInc
linkedin.com/company/ca-technologies