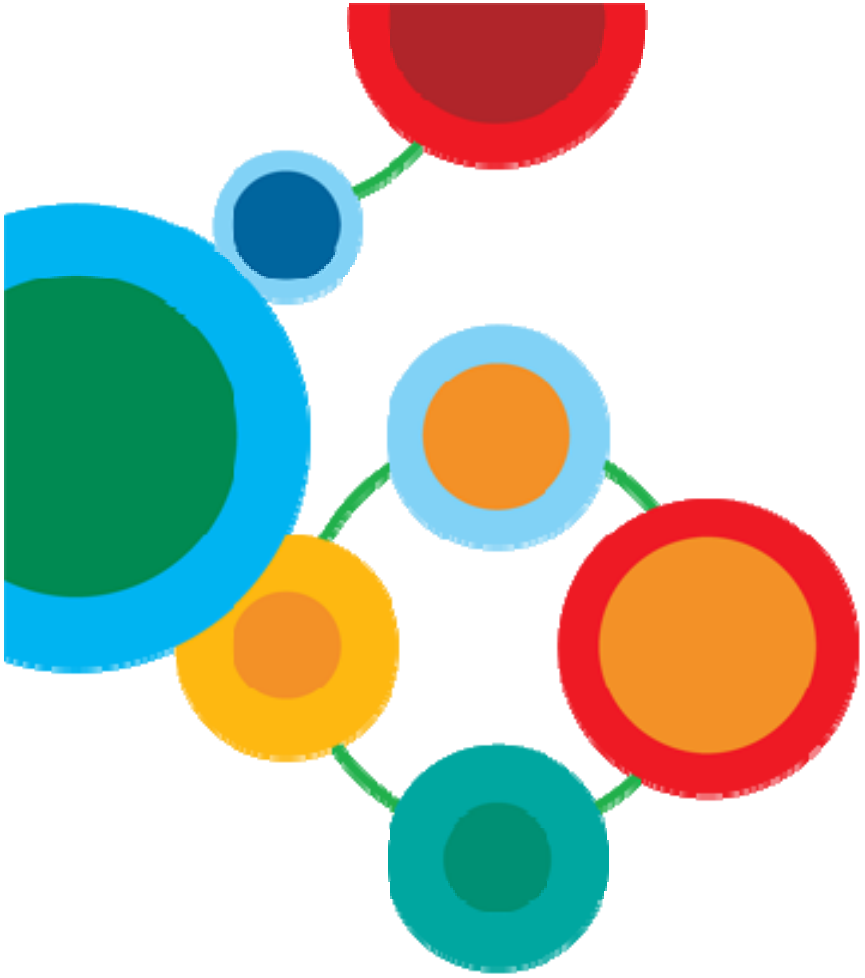


What's New in IBM DB2 10 for z/OS for Large Objects (LOBs)

Session Number 3109

Jay Yothers, IBM

IBM Software
Information On Demand 2011

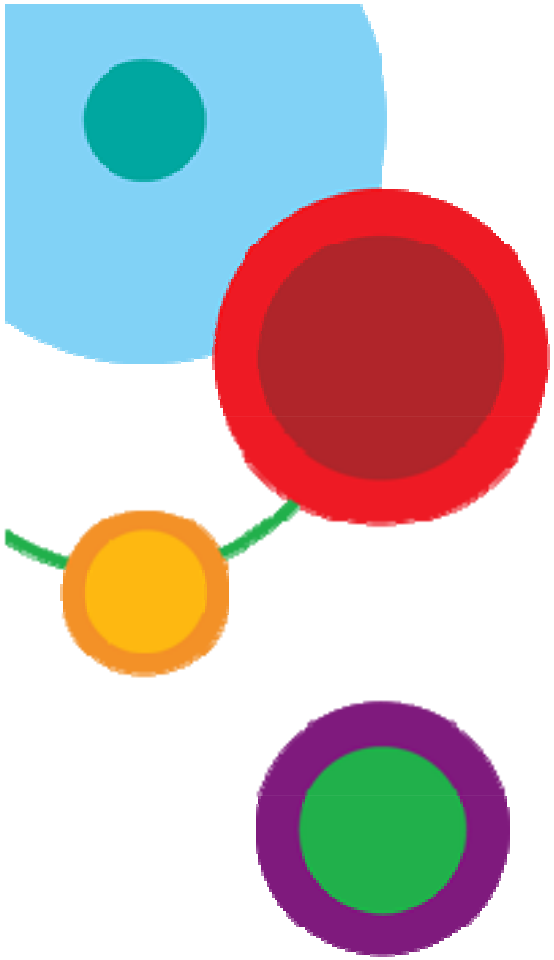
Please Note:

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

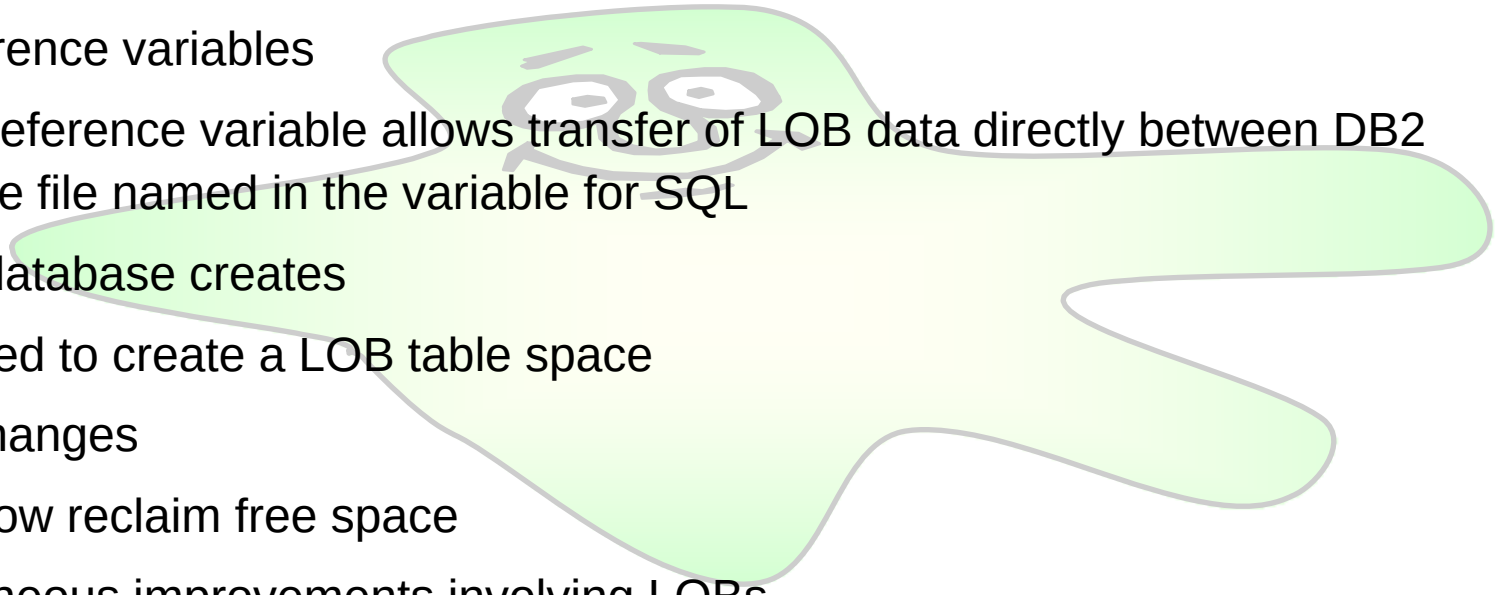
Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream,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



V9 LOB Improvements

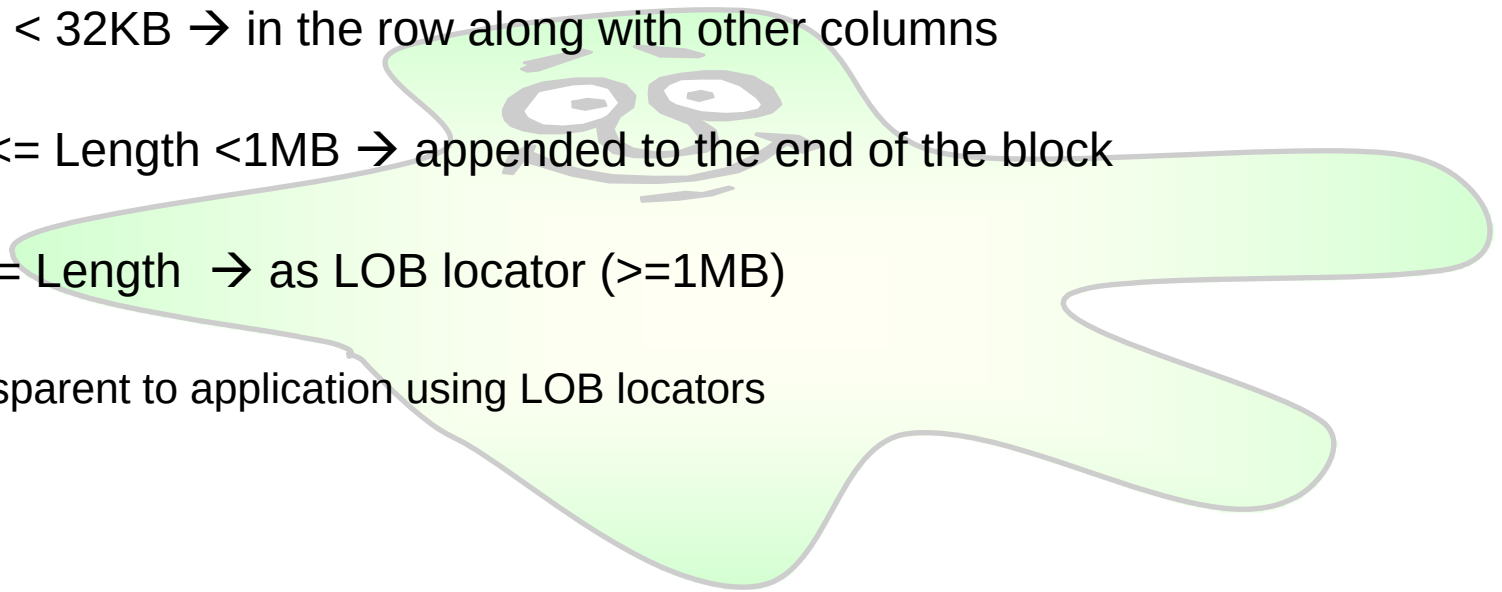
V9 LOB Improvements

- Progressive Streaming for LOB Values
  - Transparent to application using LOBs
- FETCH CONTINUE
  - Allows applications to retrieve LOB/XML data in pieces without the use of locators
- File reference variables
  - A file reference variable allows transfer of LOB data directly between DB2 and the file named in the variable for SQL
- Implicit database creates
  - No need to create a LOB table space
- Utility Changes
  - Can now reclaim free space
- Miscellaneous improvements involving LOBs



Progressive Streaming for LOB values

- Progressive Streaming for (output) LOB Locator Values
  - LOB size 10/90 rule?
  - DB2 uses LOB size to determine whether to send LOB data to Java or DB2 CLI clients
  - Length < 32KB → in the row along with other columns
  - 32KB <= Length <1MB → appended to the end of the block
  - 1MB <= Length → as LOB locator (>=1MB)
    - Transparent to application using LOB locators



FETCH CONTINUE

Allows applications to retrieve LOB/XML data in pieces without the use of locators

```
DECLARE t1 TABLE (c1 CLOB(10M))
DCL hv1 SQL TYPE IS CLOB(32767)
DECLARE cur1 CURSOR FOR SELECT c1 FROM t1;
OPEN cur1;

FETCH WITH CONTINUE cur1 INTO :hv1;
IF hv1_LENGTH > 32767 THEN
    Acquire storage hv1_LENGTH in length
    FETCH CURRENT CONTINUE cur1 INTO :acquired_storage
    Concatenate data in hv1 and acquired_storage host variables
```

FETCH CONTINUE

Allows applications to retrieve LOB/XML data in pieces without the use of locators

```
DECLARE t1 TABLE (c1 CLOB(10M))
DCL hv1 SQL TYPE IS CLOB(32767)
DECLARE cur1 CURSOR FOR SELECT c1 FROM t1;
OPEN cur1;

Empty result
FETCH WITH CONTINUE cur1 INTO :hv1;
IF hv1_LENGTH > 32767 THEN
      DO UNTIL hv1_LENGTH <= 32767
            Concatenate result and hv1
            FETCH CURRENT CONTINUE cur1 INTO :acquired_storage
      END
Concatenate result and hv1
```

File Reference Variables

A file reference variable allows direct transfer of LOB data between DB2 and the file named in the variable

```
DCL hv1 SQL TYPE IS CLOB_FILE
    hv1_NAME_LENGTH      INTEGER
    hv1_DATA_LENGTH      INTEGER
    hv1_FILE_OPTIONS     INTEGER
    hv1_NAME             CHAR(255)
Length of hv is 267 bytes

DCL hv2 SQL TYPE IS BLOB_FILE
DCL hv3 SQL TYPE IS DBCLOB_FILE
```

File Reference Variables

hv1_NAME = ‘MY.SELECTED.LOB.FILE’

hv1_NAME_LENGTH = LENGTH(‘MY.SELECTED.LOB.FILE’)

hv1_FILE_OPTIONS = SQL_FILE_CREATE

SELECT c1 FROM t1 INTO :hv1

hv1_FILE_OPTIONS = SQL_FILE_OVERWRITE

hv1_FILE_OPTIONS = SQL_FILE_APPEND

hv1_NAME = ‘MY.LOB.TOINSERT.FILE’

hv1_NAME_LENGTH = LENGTH(‘MY.LOB.TOINSERT.FILE’)

hv1_FILE_OPTIONS = SQL_FILE_READ

INSERT INTO t1(c1) VALUES(:hv1)

Cannot be passed to or from Stored Procedures or User Defined Functions

Implicit Database Creates

- In V8, unless CURRENT RULES is 'STD'

  CREATE TABLE t1 (c1 CLOB(1M), … r1 ROWID NOT NULL) IN db1
  - Creates base table space implicitly

  CREATE LOB TABLESPACE lts1 IN db1 …

  CREATE AUX TABLE axt1 IN db1.lts1 STORES t1 COLUMN c1

  CREATE INDEX axix1 ON axt1

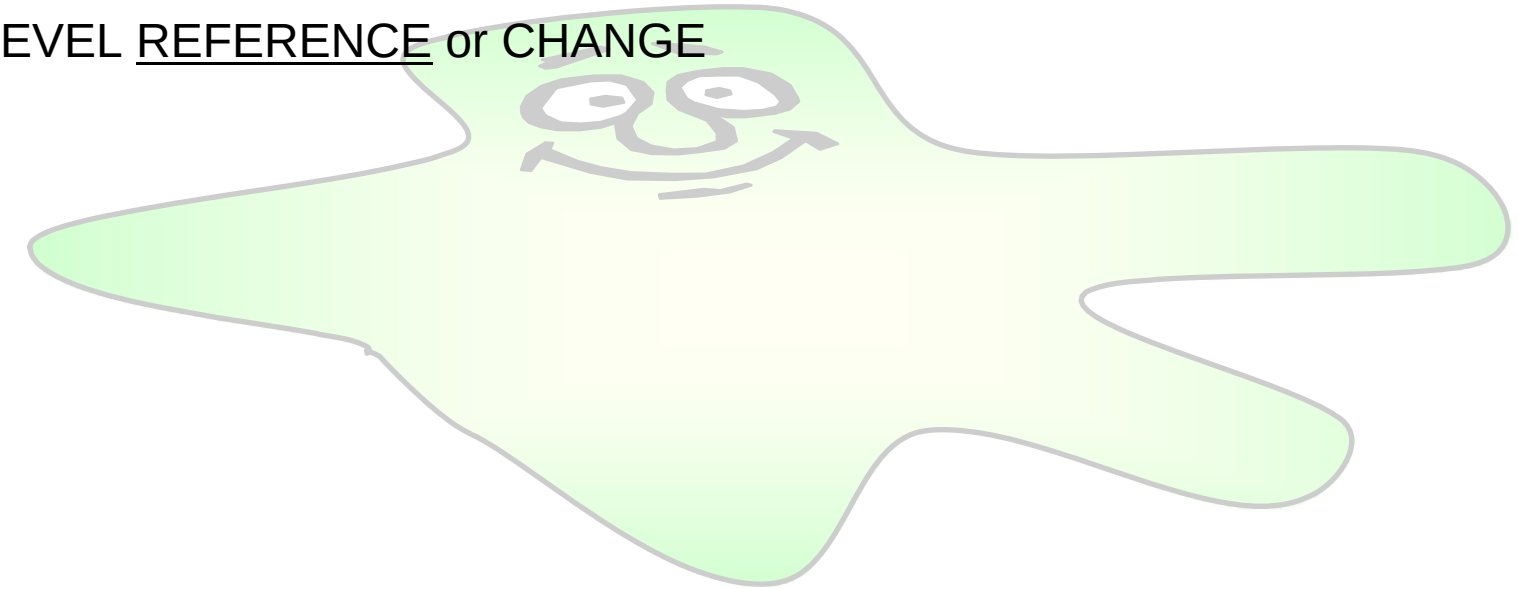
  … for each LOB column in the table

- In V9

  CREATE TABLE t1 (c1 CLOB(1M), …) IN db1
  - Creates base table space implicitly
  - Creates LOB table spaces implicitly for all LOB columns
  - Creates auxiliary tables implicitly for all LOB columns
  - Creates auxiliary indexes implicitly for all LOB columns
  - "Sliding" secondaries

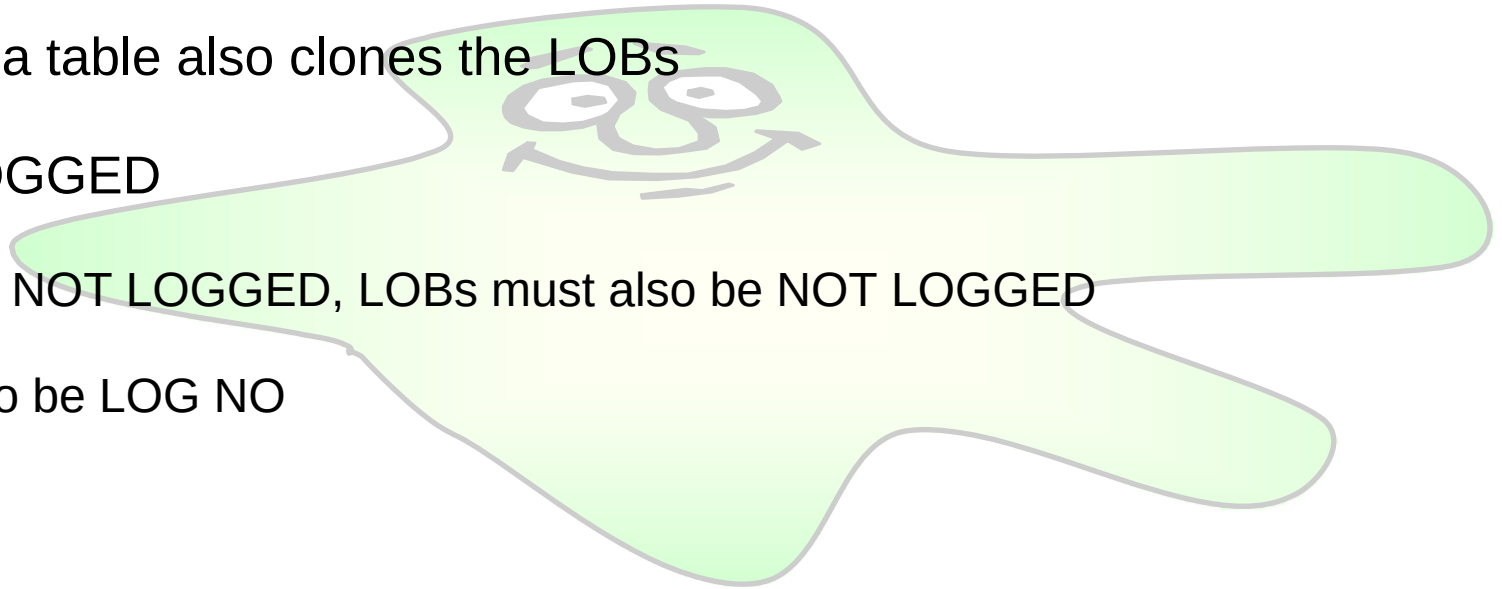
Utilities

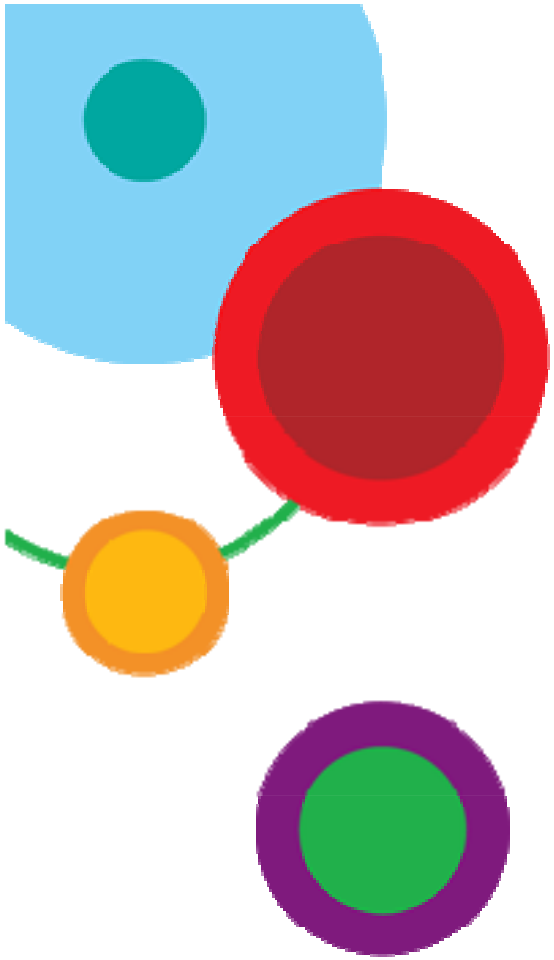
- LOAD / Cross load LOB column lengths > 32KB supported
- REORG LOB reclaim space
  - SHRLEVEL REFERENCE
- Online CHECK LOB and DATA
  - SHRLEVEL REFERENCE or CHANGE



Miscellaneous

- Elimination of LOB locks for improved availability and performance
- Logging for > 1GB LOBs
- TRUNCATE also deletes LOBs
- VARBINARY compatible with BLOB
- Cloning a table also clones the LOBs
- NOT LOGGED
  - If base NOT LOGGED, LOBs must also be NOT LOGGED
  - Used to be LOG NO

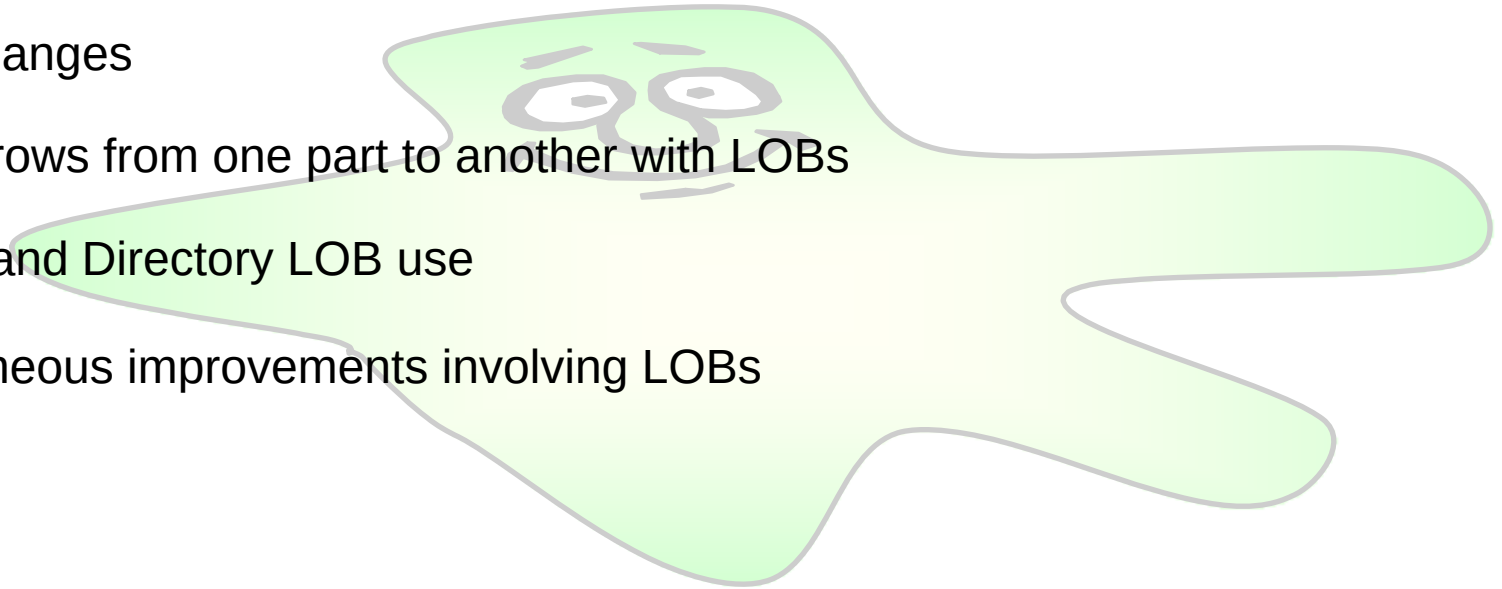




V10 LOB Improvements

V10 LOB Improvements

- LOB streaming between DDF and the rest of DB2
  - For LOB data flowing into DB2
- Inline LOBs
  - LOB data in the base table row
- Utility Changes
  - Move rows from one part to another with LOBs
- Catalog and Directory LOB use
- Miscellaneous improvements involving LOBs



LOB Streaming

- Input LOB length is unknown
- V8 implemented input LOB streaming between clients and DDF
  - Inserts and Updates
  - Entire LOB materialized on server before being placed in the table
  - Repeated requests for storage to accommodate materialization
- V10 implements input LOB streaming between DDF and the DB2 engine
  - Inserts and Updates
  - Only 1 chunk materialized in the engine before being placed in the table
  - A single request for storage, reused for succeeding chunks

Inline LOBs

- Trade off: LOB performance and capability
  vs. number of base rows per page

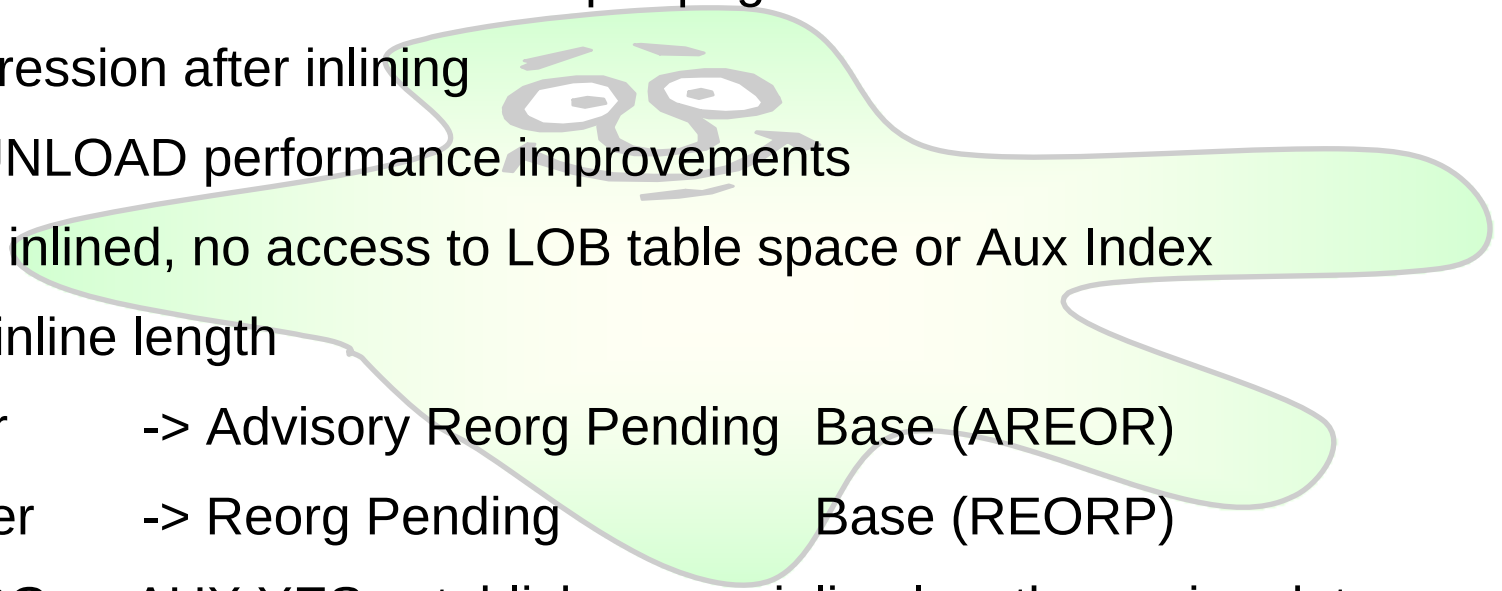
- c1 CLOB (max-length) INLINE LENGTH integer (0-32680)
- c2 BLOB (max-length) INLINE LENGTH integer (0-32680)
- c3 DBCLOB(max-length) INLINE LENGTH integer (0-16340)
  - Default is the LOB INLINE LENGTH zParm

- LOBs that exceed inline length **overflow** to LOB table space
  - Inline portion subject to compression and can be source for IOE
  - Out-of-line portion cannot be compressed or source of IOE

- Beta feedback: 80% CPU reduction on SELECT of a LOB

Inline LOBs …

- Non-null default values
  - Must be completely inlined
- Compression
  - For the inline portion
  - Increase number of base rows per page
  - Compression after inlining
- LOAD/UNLOAD performance improvements
  - If fully inlined, no access to LOB table space or Aux Index
- ALTER inline length
  - Larger -> Advisory Reorg Pending Base (AREOR)
  - Smaller -> Reorg Pending Base (REORP)
  - REORG … AUX YES establishes new inline length, moving data to/from LOB table space as necessary



Inline LOBs …

- Index on Expression
  - Must refer to only the inlined portion
  - CREATE INDEX ix1 ON t1 (expression-on-LOB-column)

    … WHERE (expression-on-LOB-column=:hv)
  - Resultant expression data type cannot be LOB
  - Expression examples
    - CREATE INDEX ioe1 ON s1.t1 **CHAR(SUBSTR(MYCLOB,8,23))**

      SELECT … WHERE **CHAR(SUBSTR(MYCLOB,8,23))** = :myhostvariable
    - CREATE INDEX ioe2 ON s2.t2 … **VARBINARY(SUBSTR(MYBLOB,48,17))**

      SELECT … WHERE **VARBINARY(SUBSTR(MYBLOB,48,17))** = :myhostvariable
  - Index entry created at INSERT or UPDATE time
  - Index entry points to base row
  - Predicate causes **no** access to LOB table space or Aux Index

Utilities

- REORG SHRLEVEL(CHANGE) for LOBs
- REORG … REBALANCE … AUX YES rebalances even with LOBs
  - Rows moved from one part to another have their LOBs moved from LOB table space to another.
- REORG … AUX YES of a PBG table space with LOBs reorgs entire table space
  - Rows moved from one part to another have their LOBs moved from LOB table space to another.
- RECOVER to point in time
  - VERIFYSET to fail PIT recovery if entire set not included
  - ENFORCE to set CHKP/ACHKP for PIT recovery if entire set not included
  - Set includes Base, LOB, XML, and History table spaces (not RI)
- LOAD/UNLOAD with spanned records allows LOBs to appear among other non-LOB data in input/output file
  - SPANNED YES
- REORG SHRLEVEL(NONE) for LOB table spaces is **no longer supported**.

Catalog and Directory LOBs

DBD01.DBD_DATA	BLOB(2G)	
SPT01.SPTSEC_DATA	BLOB(2G)	INLINE(0-32138)
SPT01.SPTSEC_EXPL	BLOB(2G)	
SYSAUTOALERTS.OUTPUT	CLOB(2M)	
SYSAUTORUNS_HIST.OUTPUT	CLOB(2M)	
SYSCONTROLS.RULETEXT	CLOB(2M)	
SYSCONTROLS.DESCRIPTOR	BLOB(2M)	
SYSINDEXES.PARSETREE	BLOB(1G)	
SYSINDEXES.RTSECTION	BLOB(1G)	
SYSJARCONTENTS.CLASS_SOURCE	CLOB(10M)	
SYSJAROBJECTS.JARDATA	BLOB(100M)	
SYSPACKSTMT.STATEMENT	CLOB(2M)	INLINE(15360)
SYSPACKSTMT.STMTBLOB	BLOB(2M)	INLINE(7168)
SYSPENDINGDDL.STATEMENT_TEXT	CLOB(2M)	
SYSQUERY.STMTTEXT	CLOB(2M)	INLINE(2048)
SYSROUTINES.TEXT	CLOB(2M)	
SYSROUTINES.PTREE	BLOB(1G)	
SYSTABLES_PROFILES.PROFILE_TEXT	CLOB(1M)	
SYSTRIGGERS.STATEMENT	CLOB(2M)	
SYSVIEWS.STATEMENT	CLOB(2M)	
SYSVIEWS.PARSETREE	BLOB(1G)	INLINE(27670)
SYSXSROBJECTS.GRAMMAR	BLOB(250M)	
SYSXSROBJECTS.PROPERTIES	BLOB(5M)	
SYSXSROBJECTCOMPONENTS.COMPONENT	BLOB(30M)	
SYSXSROBJECTCOMPONENTS.PROPERTIES	BLOB(5M)	

Miscellaneous

- Online ALTER of PAGESIZE and DSSIZE for LOB table spaces
- ALTER LIMITKEY with LOBs
  - REORG AUX moves LOBs from LOB table space to LOB table space as necessary
- DEFINE NO for LOB table spaces
  - If no LOB values or all LOB values are completely inlined, the data set for the LOB table space does not get defined

    CREATE TABLE t1 (c1 CLOB(1M), …) IN db1

    - If base table space has DEFINE NO or IMPDSDEF=YES
      - » Creates LOB table spaces implicitly with DEFINE NO
      - » Creates auxiliary indexes implicitly with DEFINE NO
  - DEFINE NO is ignored for created auxiliary indexes

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.

Communities

- **On-line communities, User Groups, Technical Forums, Blogs, Social networks, and more**
  - Find the community that interests you …
    - **Information Management** **ibm.com**/software/data/community
    - **Business Analytics** **ibm.com**/software/analytics/community
    - **Enterprise Content Management** **ibm.com**/software/data/content-management/usernet.html

- **IBM Champions**
  - Recognizing individuals who have made the most outstanding contributions to Information Management, Business Analytics, and Enterprise Content Management communities
    - **ibm.com**/champion

Thank You!
Your Feedback is Important to Us

- Access your personal session survey list and complete via SmartSite
  - Your smart phone or web browser at: iodsmartsite.com
  - Any SmartSite kiosk onsite
  - Each completed session survey increases your chance to win an Apple iPod Touch with daily drawing sponsored by Alliance Tech