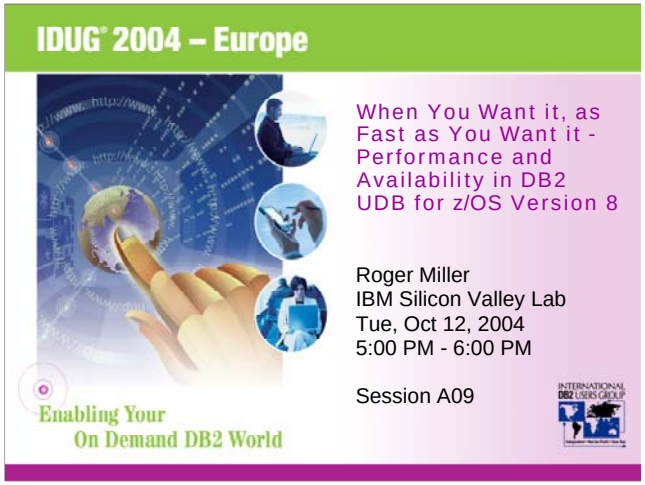


A09: This is an overview of improved performance & availability in DB2 ® UDB for z/OS Version 8 (V8). DB2 V8 is the twelfth release and delivers more function than any release of DB2 for MVS. DB2 V8 became generally available March 26, 2004. This version brings extensive integration and synergy with zSeries hardware, with middleware and with applications. Data support, application development and query enhancements are added for e-business, building upon the traditional enterprise of choice characteristics of availability, exceptional scalability, and performance.
DB2 Version 8 has been re-engineered for e-business on demand, with many fundamental changes in architecture and structure. Key improvements enhance scalability, application porting, security, and continuous availability. Management for very large databases is made much easier, 64-bit virtual storage support makes management simpler and improves scalability and availability. This new version breaks through many old limitations in the definition of DB2 objects. These enhancements include SQL improvements, schema evolution, longer names for tables and columns, longer SQL statements, enhanced Java and Unicode support, enhanced utilities, more log data sets, and a lot more.

DB2 V8 makes fundamental changes in many areas, reengineering much of DB2 to improve e-business and to help customers break through limitations. Key improvements include scalability, ability to port applications, and to deliver continuous availability. The ability to manage very large databases is substantially improved. Key vendor applications receive a wide range of improvements. Other highlights:

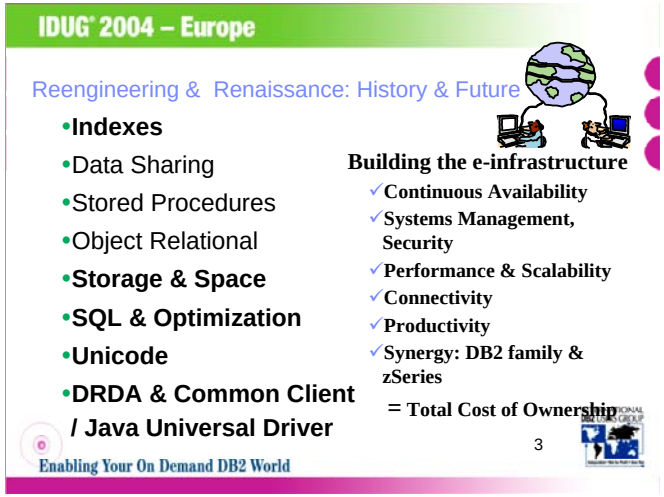
Integration - WebSphere & Java with Type 4 driver - DB2 delivers improved ease of application development with Java on zSeries. QMF provides enhanced data visualization, with multiple interfaces for robust secured access to key business data across the enterprise. zSeries - DB2 leverages the hardware and software to provide a highly reliable and secure DBMS.

High Availability - Online Schema Evolution - make critical DB schema (structure) changes without having to first DROP and then CREATE it. Point in Time Recovery - faster recovery / less costly outages - back in production mode more quickly.

Flexible growth - Increased number of new / ported key ISV applications through SQL enhancements, Long Name support, and UNICODE for continued application enablement.

Incremental Scalability with 64 bit virtual addressing: Use more storage, more effectively, with less complexity. Eliminate system constraints as more applications and users are added. Open and scalable architecture.

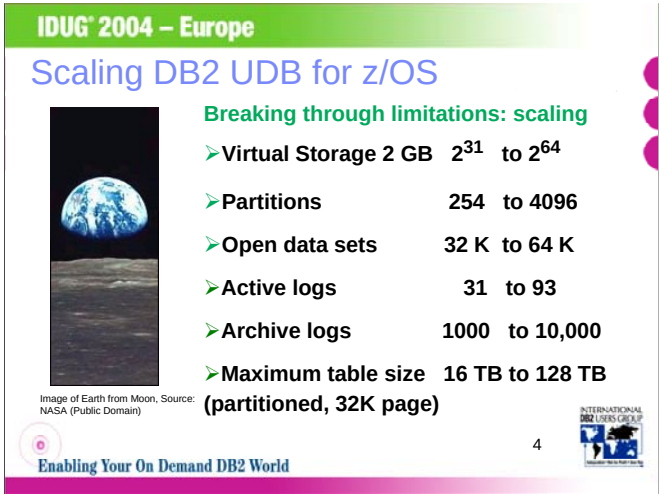
http://ibm.com/software/db2zos/db2zosv8.html



Renaissance of DB2 continues as an ongoing process, refreshing some components, updating work during each release. Some of the architecture has made more dramatic changes, such as changing to type 2 indexes, data sharing, stored procedures, object relational and moving to larger address spaces. Unicode changes our definition & handling of characters, changing the foundation. A common client across the DB2 family provides better family consistency and a stronger DRDA.

Rebuilding, rearchitecting & renewing continues in V8, with major changes in indexes, storage, SQL, optimization, Unicode, Common Client & Java Universal Driver.

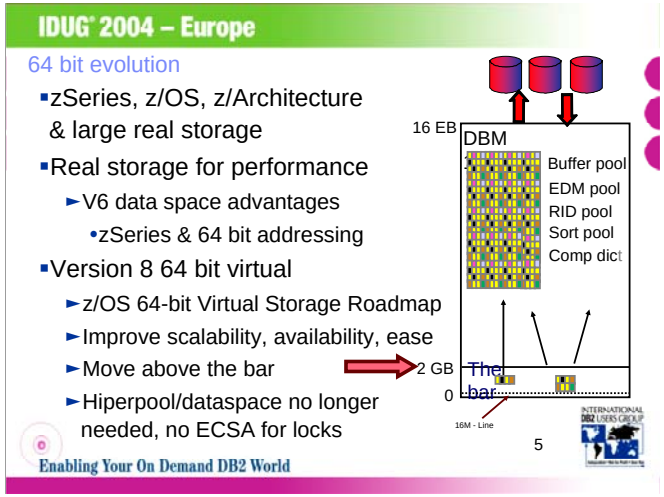
Building the e-infrastructure shows the value proposition of DB2 for z/OS: leadership in continuous availability, systems management, security, performance and scalability with connectivity to everything. Consistency with the DB2 UDB family on all platforms delivers portability and improves productivity. Integration and synergy with zSeries & z/OS helps with all of the infrastructure values.



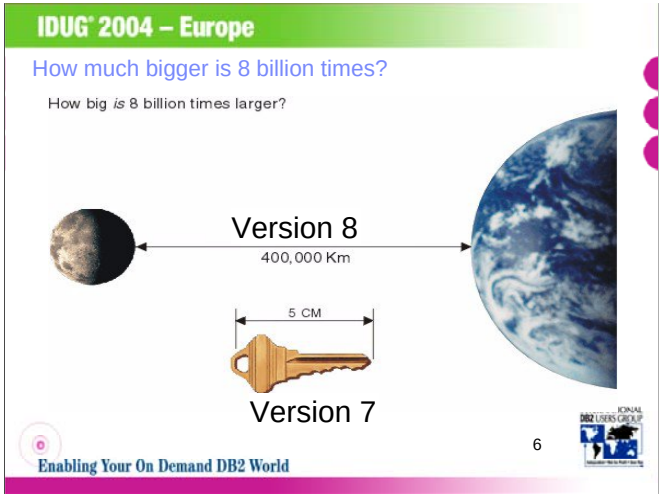
One of the keys to reengineering is breaking through the limits of the current architecture. Increasing some limits improves scalability. Increasing other limits improves productivity, portability & family consistency.

Increasing the amount of virtual storage we can address directly can help with the ability to scale and simplify management for virtual storage. It will require more real memory, but permit increased scalability and availability.

Increasing name sizes & SQL statement lengths makes porting from other DBMS much easier and improves DB2 family compatibility. Increasing the maximum number of partitions helps DB2 scale farther and makes management much easier when you can have one partition per day for 11 years.



The biggest impact of the zSeries architecture on DB2 is the ability to use more memory more effectively. Prior to the zSeries, customers were limited to 2 GB real storage due to the 31-bit addressing of the S/390 architecture. The real storage limit of 2 GB is a leading performance inhibitor for many high end customers. Another performance inhibitor is the 2 GB virtual storage limit for the main DB2 (DBM1) address space. If you have zSeries & OS/390 V2R10 64-bit mode or z/OS, use V6 buffer pools in data spaces, but not otherwise. See V7 Performance Topics red book & the web. See What's New? for V8 use of 64 bit virtual storage.
There are more steps as real & virtual memory sizes increase, moving more above the line and above the bar. See the Roadmap, GM13-0076-01 updated June 2002.
ibm.com/servers/eserver/zseries/library/whitepapers/gm130076.html



How much bigger is a factor of 8 billion?

It's the factor between 5 centimeters and the distance from the earth to the moon.

For time, consider 8 billion seconds
60 seconds per minute, 60 minutes per hour

24 hours per day, 365 days per year

250 years

The difference between a finger snap and back to the declaration of independence or French revolution.
Start now with 2 GB. Double it every year. It will take 33 years to reach 16 exabytes, but many customers are using more than 2 GB today, so this change provides about 25 years at this growth rate.

IDUG 2004 – Europe

How do we create room?

Move over the bar

Below 16 MB line 2^{24}

Below 2 GB bar 2^{31}

Above 2 GB bar 2^{64}

No hiperpools or data spaces - not needed

Most large storage areas moved above the bar

Management of virtual storage is much simpler

Allows increase in number of threads

7

Enabling Your On Demand DB2 World

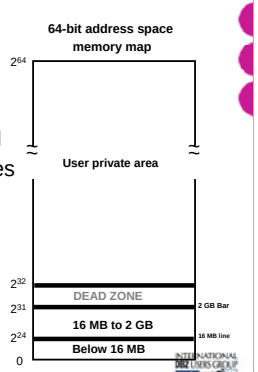
We are still working to move a few z/OS and DFSMS control blocks above the 16 MB line, but below the 16M line is a relatively minor concern after z/OS R3 and R5. The area below the 2 GB bar is 2^7 or roughly 128 times larger, at 2 GB. Large customers have filled this space in DBM1, so it's time to move the large data areas above the bar.

The area over the bar is 2^{33} or roughly 8 billion times larger. It is not infinite, but the virtual address space will not be the limiting factor for some time. Our concerns will shift to real storage. We can restructure to remove the complexity and overhead of hiperspaces and data spaces. Moving most large data areas above the bar is how we spell relief for virtual storage constraints.

IDUG 2004 – Europe

64 bit virtual storage

- Line 2^{24} Bar 2^{31} 2^{64}
- Above the bar
  - Buffer pools, EDM pool, Sort Pool
  - Rid Pool, Compression dictionaries
  - IRLM locks, PC=YES
- Real storage requirements probably increase
- Improve scalability & availability
- Much reduced management for virtual storage



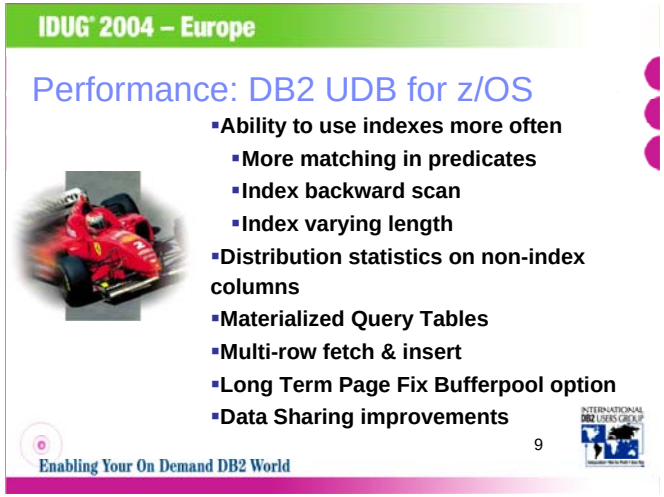
8

Enabling Your On Demand DB2 World

A statement of direction was included in the September 11, 2001 announcement, IBM z/OS Version 1 Release 2: Enabling and Protecting Your e-business and Preview: z/OS Version 1 Release 3

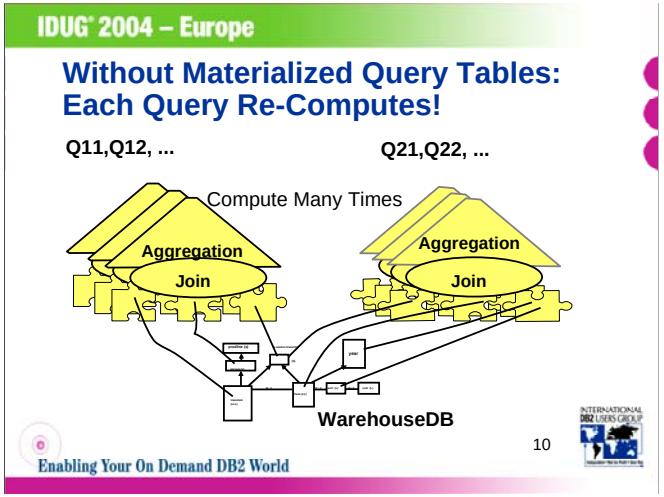
IBM plans to deliver 64-bit virtual storage addressing for the DB2® for z/OS product in a future release. The future release of DB2 for z/OS, with 64-bit virtual address support, can only execute on IBM (elogo)server zSeries 900 (z900), or equivalent, running z/OS V1R3, or later.

Instead of hiperspaces or data spaces, the single large address space can allow easier management of storage. We expect real storage needs to increase as scalability & availability are addressed.

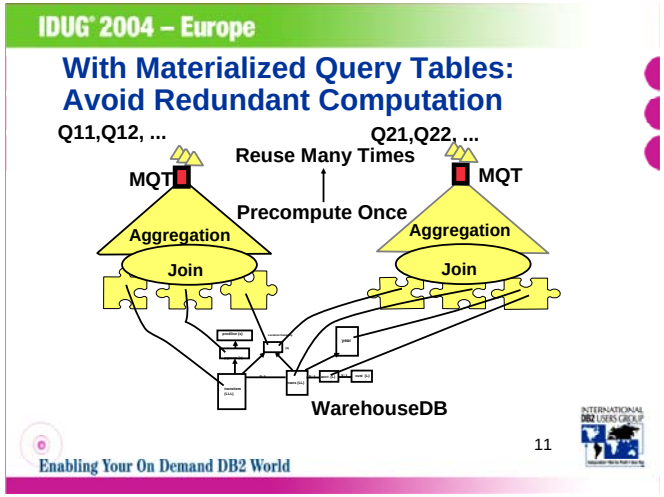


Many of the performance enhancements come from the enhanced index capabilities. Comparing a decimal column to a floating point number could not be done in the past. Being able to compare these values can mean choosing a better index or even an index access instead of a table space scan. An index can be used for a backward scan, so some indexes may not be required, reducing the overhead for inserts & deletes. Support for varying length indexes can save space and can use index-only access with a varchar. Having distribution statistics for columns which are not part of the index can give the optimizer better information so it can perform better optimization. Gathering these statistics was a separate DSTATS program, and is now part of RUNSTATS.

Materialized query tables can provide a one or two order of magnitude performance improvement by rewriting queries to use the precalculated information.

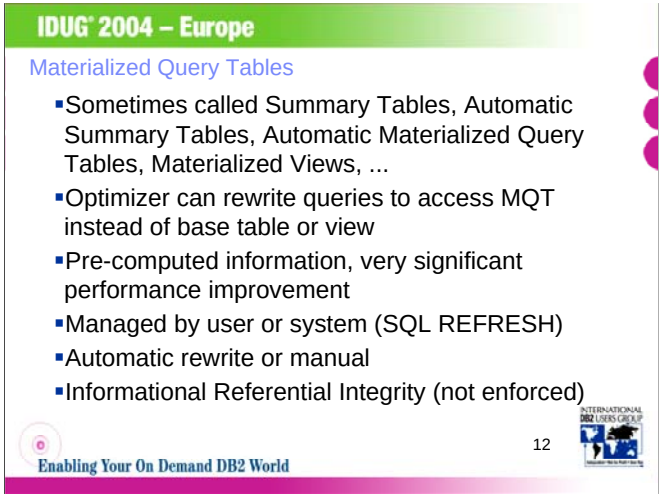


A materialized query table (MQT) can avoid redundant work of scanning, aggregation and joins. Multiple levels of summary tables have been used in warehouses and complex applications for years. One of the major issues is communicating the summaries to the users. In some cases, the users want to query the base data. With MQTs, the query users do not have to be aware of the MQT.



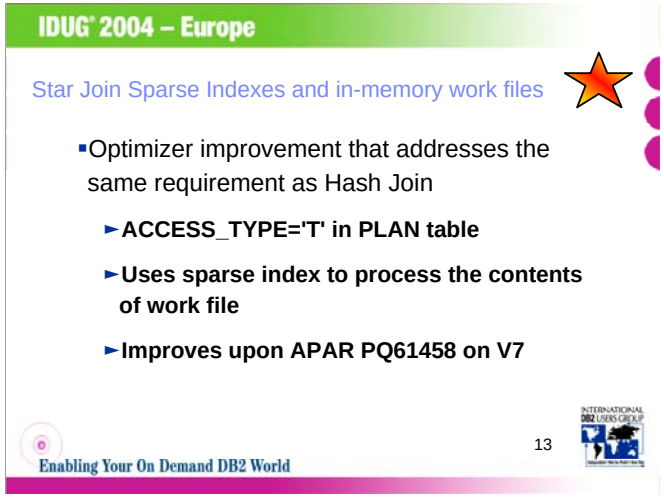
Even though the query is submitted for the base table, the optimizer can rewrite the query to use the MQT. Using the precalculated information can improve subsequent queries by as much as two or three orders of magnitude. Materialization or precalculation and parallelism resolve the long response times.

A database administrator can use an MQT much as she or he would use an index for optimization. Controls for usage, initial loading and refresh are part of the definition.



Many names are used for the MQTs in various implementations, including automatic summary tables, aggregate summary tables, automatic materialized query tables and materialized views. They all have a common objective, to precompute information for later use by other statements.

The objective is to make the use as transparent as possible. Customers may choose to have the MQTs managed and refreshed by the system or manually. Most customers will have the optimizer rewrite queries, but MQTs can be names in SQL statements.



Performance of star join is critical to data warehousing applications where the star schema is the main database design principle. The star join implementation in DB2 for z/OS must deal with a large number of work files, especially for a highly normalized star schema that can involve many snowflakes. Currently, if one or more snowflake work files are joined after the fact table, the sort merge joins tend to be selected as the join method because the work files do not have indexes. However, the cost of sorting can be large both in time and space.

The star join implementation in DB2 UDB for z/OS potentially has to deal with a large number of work files, especially for a highly normalized star schema that can involve many snowflakes, and the cost of the sorting of these workfiles can be very expensive. DB2 V8 extends the use of a sparse index (a dynamically built index pointing to a range of values) to the star join work files and adds a new optional function of data caching on star join workfiles. The decision to use the sparse index is done based on the estimation of the costs of the access paths available.

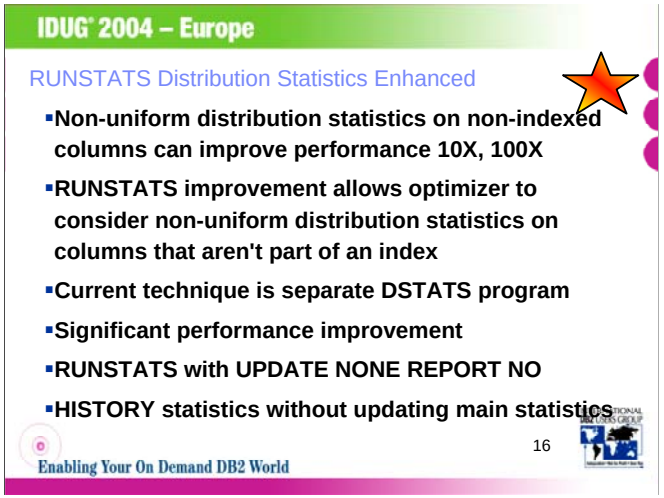
Each time a trigger is invoked, a work file is created for the old and new transition variables. For a conditional trigger (one with a WHEN clause), the work file is created for all changes, even for those for which the trigger is not activated. In earlier versions, the work file is created to evaluate the condition and then deleted after deciding not to fire the trigger. Storage is used for small numbers of rows to avoid creating and deleting the work files.

A cardinality option for a user-defined table function reference is added to the SQL language. This is a nonstandard SQL feature, specific to IBM DB2 for z/OS implementation.

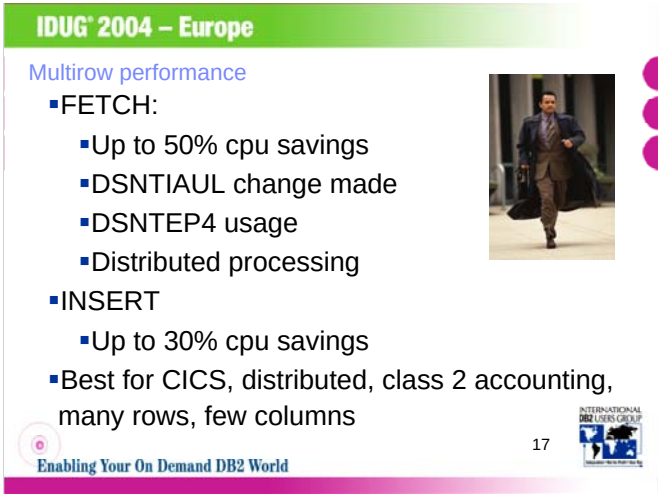
With this option, users can tune the performance of queries that contain user-defined table functions.

A join predicate between a user-defined function and a base table will be stage 1, and possibly indexable, if the base table is the inner table of the join and the other stage 1 conditions are met.

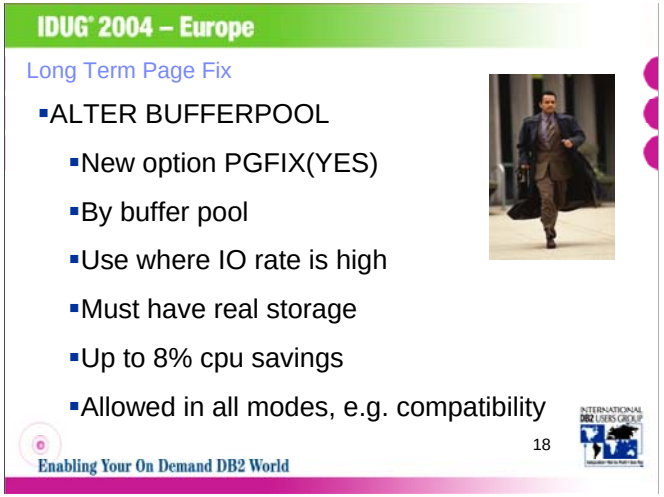
Rows returned from a user defined function can be pre-fetched into a work file in its first invocation, based on the access cost estimation.



Skewed data distributions are responsible for a high proportion of performance problems with DB2 queries, especially in ad hoc queries. Symptoms can be less than optimal join sequences, too much synchronous I/O, and long response times. When there is asymmetrical distribution of data, not having distribution statistics on non-leading indexed columns and/or non-indexed columns can cause DB2 to make sub-optimal table join order and table join method decisions. Collecting distribution statistics for non-leading indexed columns and/or non-indexed columns allows DB2 to use these statistics for better access path selection. Better index selections can be made, when there are screening predicates or there are matching in-list / in-subq predicates which break up matching equals predicates. RUNSTATS with keywords REPORT NO and UPDATE NONE allows users to invalidate dynamic SQL caching for the table space and/or index space without the overhead of collecting statistical information and without generating reports or updating catalog tables.



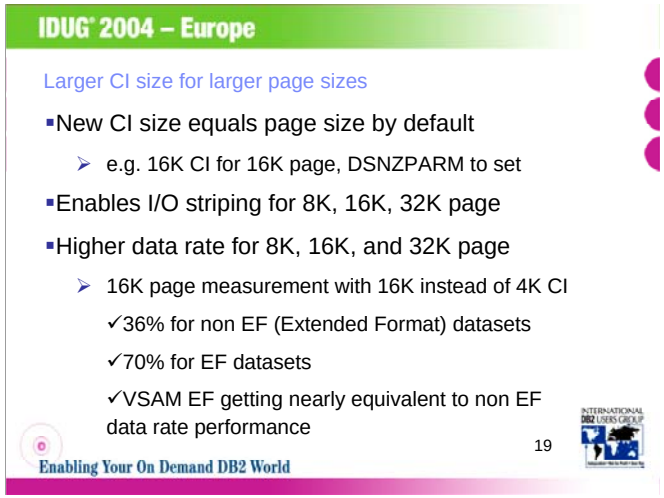
Here are some program changes already implemented to save cpu time. If you use DSNTIAD, then the changes are included. If you use DSNTEP2, then changing to DSNTEP4 can save a lot of cpu time if you fetch large numbers of rows. Multirow fetch is used by distributed processing. If you have coded ODBC with the array interface, multirow fetch is used. The improvements can vary a lot, but the biggest savings will be where the processing is simple, in the CICS (non OTE) environment, with distributed processing where latency and TCP/IP processing can be avoided, with class 2 accounting on, with large numbers of rows, but few columns. For example, one customer measurement was a 76% improvement.



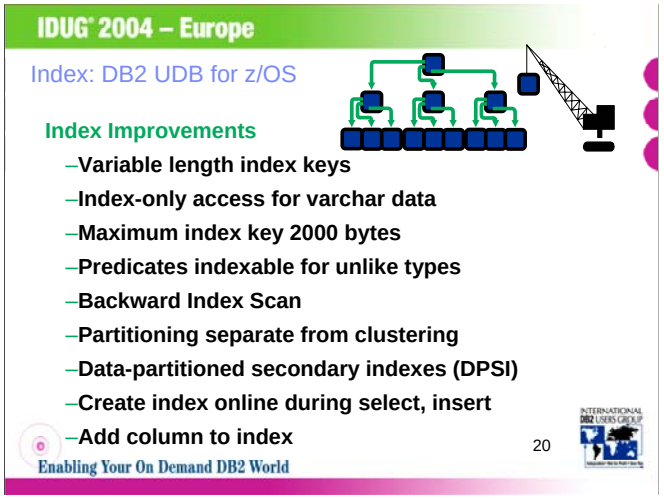
ALTER BUFFERPOOL has a new option that most customers should use for subsystems which read or write frequently. Recommendation: Alter your DB2 Version 8 buffer pools which have frequent page reads or writes to use PGFIX YES if you have sufficient real storage available for these buffer pools. Fixing the buffer page **once and keeping them fixed** in real storage avoids the processing time that DB2 needs to fix and free pages **each time there is an I/O**. In some cases, this processing time can be as much as 10% for I/O intensive workloads. To use this option, issue the following command:

ALTER BPOOL(*bpname*) VPSIZE(*vpsize*) PGFIX(YES)

where *bpname* is the name of the buffer pool and *vpsize* is the size of the virtual pool.



A new option in DB2 V8 allows use of larger CI sizes with 8K, 16K and 32K pages, rather than using 4 K pages on disk. There are very substantial performance improvements in disk performance for this change. Note that 32K CI size results in 16K block size, so space is not wasted on disk.

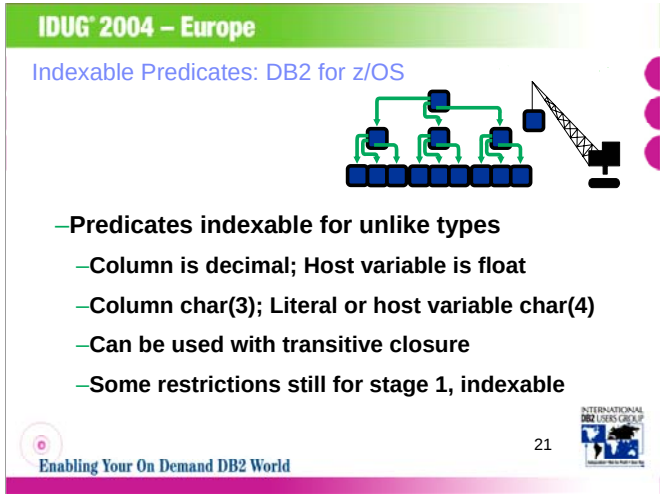


DB2 V8 provides many new opportunities for improving index processing, rebuilding the architecture for indexes.

We are able to use indexes more effectively, reducing the space in variable-length indexes, being able to have index-only access with variable-length data and being able to use the index when the predicates do not match.

In some cases, such as backward index scans or partitioning, we will be able to work as efficiently with one less index. Being able to eliminate an index will improve the insert, delete, LOAD, REORG and update processing.

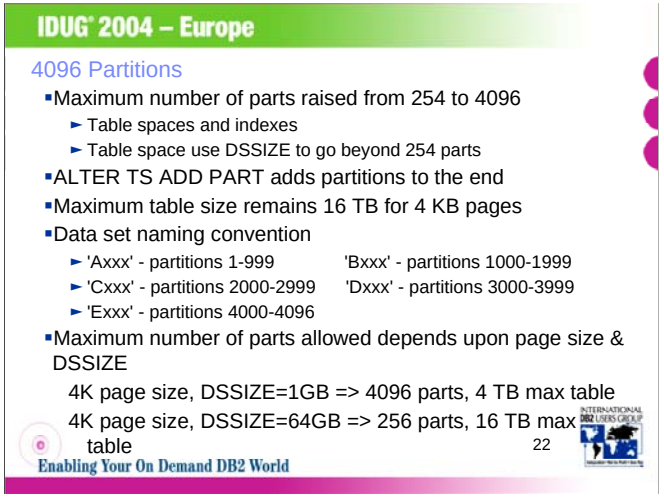
We have more flexibility in indexes, with longer index keys, the ability to partition secondary indexes and the ability to have more effective clustering.



The most common mismatches for data types come with languages like Java, C++ and C and decimal data. Often the comparison is from a floating point host variable to a decimal column.

A second type of mismatch that is very common is to have a literal or host variable with a character column length greater than that of the column.

For both of these cases, the result was often poor performance because of the inability to use an index. While there are still some restrictions, performance is expected to improve substantially for many customers.



The maximum number of partitions goes from 254 to 4096, so that you can have one partition per day for more than 11 years. Do not define all of the partitions, since you can add new partitions at the end.

While the maximum size remains 16 terabytes for 4K pages, larger page sizes can exceed 16 terabytes for a single table. DSSIZE is recommended, although LARGE also allows more than 254 parts.

This change requires changes in the data set naming convention.

Availability: DB2 UDB for z/OS

Continuous Availability

- **Online Schema Evolution: database changes with ALTER instead of DROP / CREATE e.g. ADD partition**
- **Data Partitioned Secondary Indexes**
- **System-Level Log Point Recovery**
- **Improved LPL Recovery**
- **Additional online zparms**

The most important change for many customers, especially database administrators, is the ability to use ALTER in many places instead of needing to drop and redefine. We call this schema evolution, and it can reduce outages by hours or days for a major structure change on an application.

The ability to have secondary indexes that are partitioned with the data can improve recovery times by an order of magnitude. It can also eliminate the outage for online reorganizing a single partition or BUILD2 phase.

We have some additional cases where subsystem parameters can be changed while the subsystem is running.

ftp://ftp.software.ibm.com/software/db2storedprocedure/db2zos390/techdocs/Z03.pdf

New ALTERs

- **Add a partition to the end of a table**
- **Rotate partitions**
- **Extend CHAR(n) column lengths**
- **Change type within character data types (CHAR, VARCHAR)**
- **Change type within numeric data types (small integer, integer, float, real, float8, double, decimal).**
- **Change type graphic data types (GRAPHIC, VARGRAPHIC)**
- **Includes column data type changes for columns that are referenced within a view**
- **Includes column changes for indexed columns**

This is part one of a list of the changes in the ability to ALTER instead of needing to DROP and recreate. As we looked at the number of possible changes, it became clear that schema evolution would have to evolve. We tried to include the most important changes in this first delivery.

New ALTERs ...

- **Add a column to an index**
- **Drop partitioning index (or create table without one)**
- **Alter to clustering index or to not cluster**
- **Alter an index to have true varying length character columns within a key (not padded) or to padded**
- **Alter identity column attributes**
- **Alter sequence attributes**
- **Alter VOLATILE attribute**
- **Add column AS SECURITY LABEL**
- **Alter function & procedure failure options**
- **Alter view regenerate**

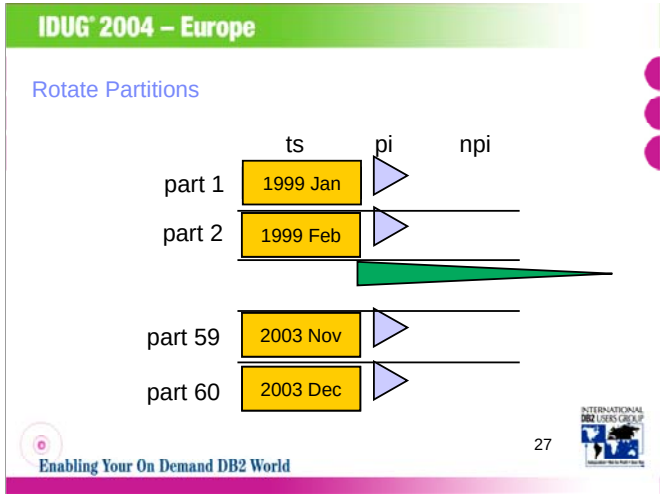
This is the second part of a list of the changes in the ability to ALTER instead of needing to DROP and recreate. On the next foils, we'll discuss the most important changes. There are many more attributes to alter. If you don't see attributes that are important to alter, be sure to indicate your priorities to your IBMer. We expect to see online schema evolution evolve in versions of DB2 after V8.

Online Schema Evolution 1

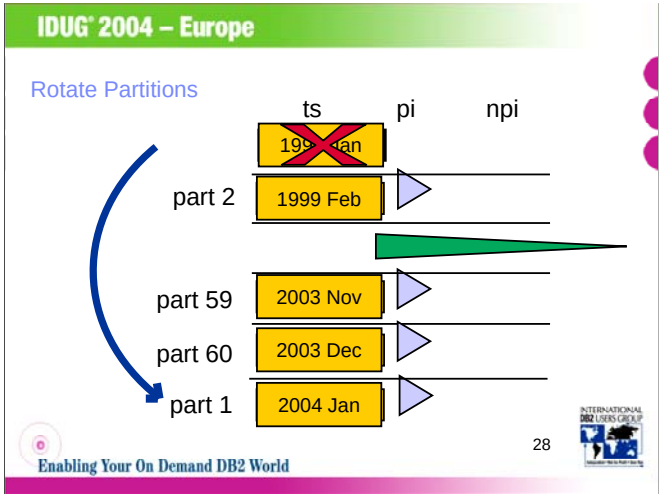
- Changing Partition Definitions
  - Add Partition
  - Rotate Partition
    - Move first part to last
    - E.g. keep rolling 36 months
- ALTER INDEX : ADD COLUMN, PADDED or NOT, CLUSTER or NOT
- CREATE INDEX while online

Three types of changes are very high on our priority list: changing partitions, changing table attributes and unbundling partitioning and clustering. This is the first category, partition changes.
Adding a new partition to an existing partitioned tables space is very important. Rotating the partitions, such as keeping a rolling 36 months of data is also key.
Creating an index has been very disruptive, but V8 provides changes that will allow creation of an index while the work continues.

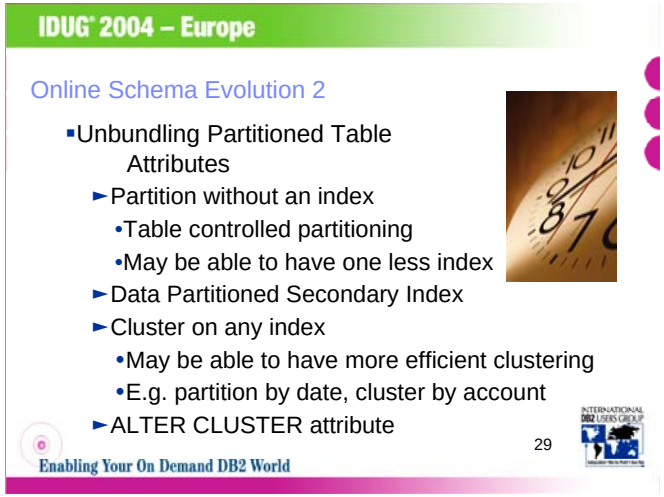
ftp://ftp.software.ibm.com/software/db2storedprocedure/db2zos390/techdocs/Z21m.pdf
ftp://ftp.software.ibm.com/software/db2storedprocedure/db2zos390/techdocs/Z22m.pdf



This is a picture for rotating a partition, to keep the most current 60 partitions or five years by month. As we reach the end of December 2003, we need to get a new partition for 2004.



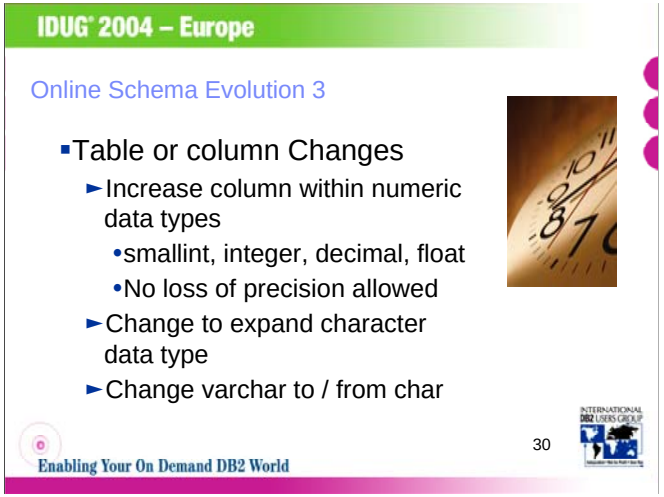
Rather than just create a new partition, we empty the first logical partition and rotate it to be the last one. In many cases, one additional partition is needed.



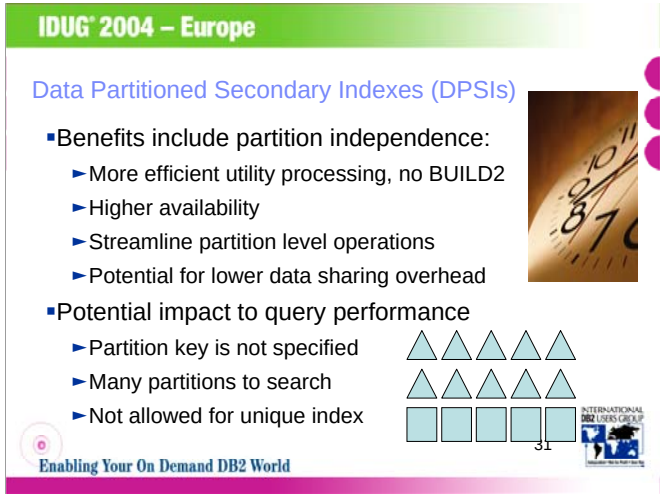
Partitioning and clustering are bundled in current DB2. Some of the time we are required to make a difficult choice. We also want to partition without an index and be able to cluster on any index. These changes will allow us to have one less index and less random IO in some cases.

ftp://ftp.software.ibm.com/software/db2storedprocedure/db2zos390/techdocs/Z21m.pdf

ftp://ftp.software.ibm.com/software/db2storedprocedure/db2zos390/techdocs/Z22m.pdf

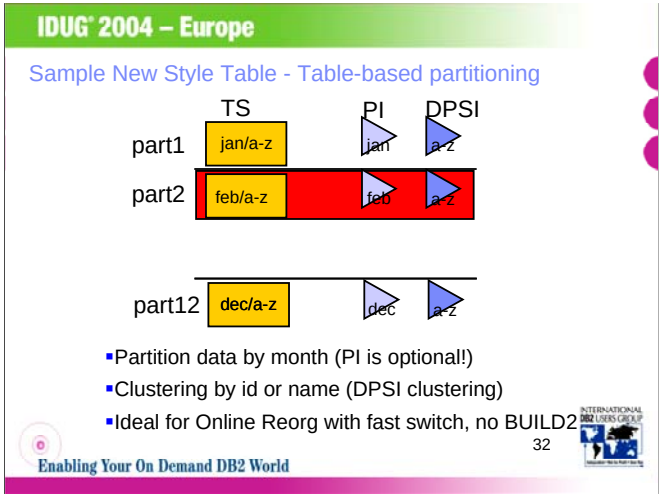


We are able to change the data type for columns. In V5 we could increase the size of varchar columns, but this change allows us to extend numeric and character columns and to change between char and varchar.

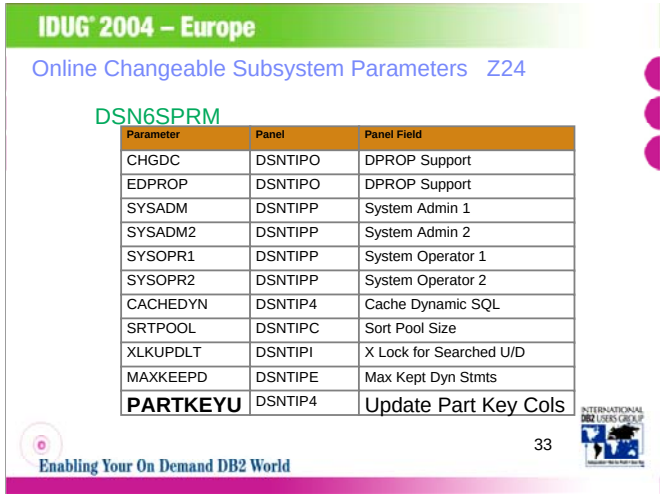


DPSI benefits are substantial in terms of the ability to reduce contention on indexes in SQL and utilities that process a range of partitions. The recovery time can be improved by an order of magnitude or more. Eliminating the BUILD2 phase of online reorg is a big improvement.

This option does not fit some situations, such as when we must search many partitions or if the index must be unique.



This is an example of the new style table, with table-based partitioning, rather than index-based partitioning. Note that the data is partitioned by month. An index is not required for the partitioning. Clustering for the data is by the id or name within each partition of the DPSI. This an ideal organization for online reorg of a single partition. The BUILD2 phase is not required. If the month is not provided, a name search using the DPSI may need to search in every partition.



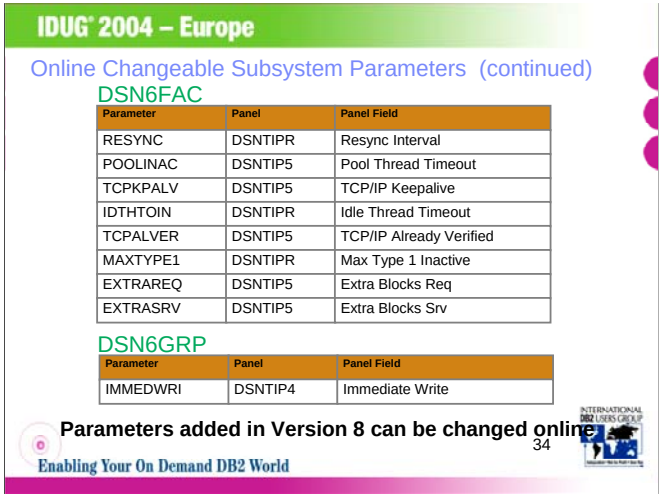
SYSADM / SYSOPR Parms: Install SYSADM auth required to change these online.

PARTKEYU : After online change, update of value in a partitioning key column may result in -904 SQL code.
PARTKEYU Avoid drain when updating values in partitioning key columns. Currently, if the update requires moving the data row from one partition to another, DB2 tries to take exclusive control of the objects to perform the update by acquiring DRAIN locks. Because of this, no other application can access the range of partition affected by update of values in partitioning key columns.

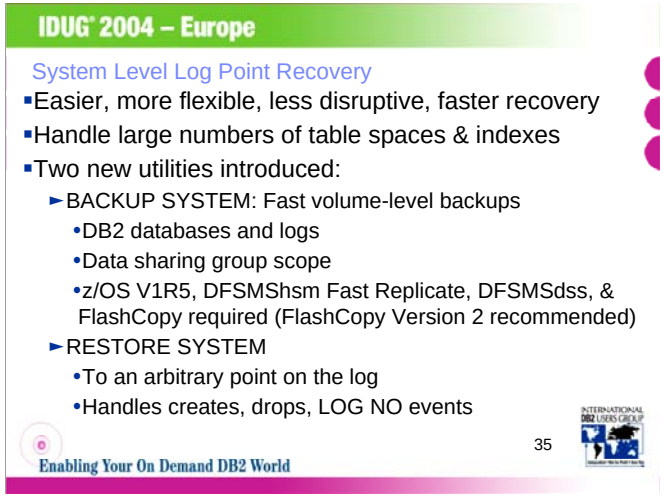
XLKUPDLT: Changes don't affect currently running statements.

MAXKEEPD Changes take effect after next COMMIT.

CACHEDYN Changes don't affect currently running statements.



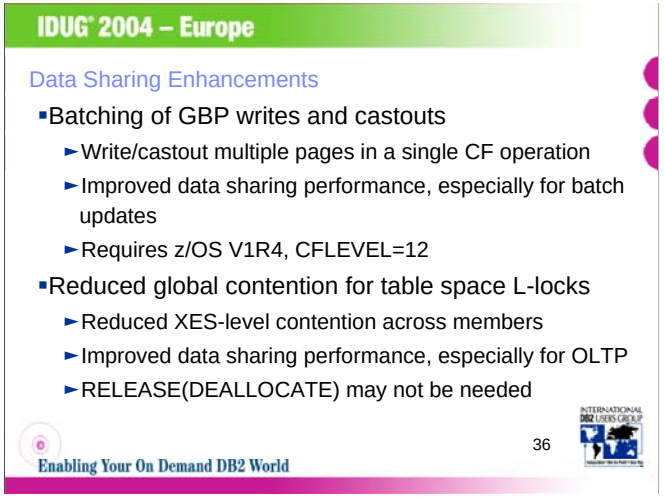
About 20 additional parameters can be changed now, in addition to the roughly 60 which could be changed in Version 7 and about 25 new parameters added in Version 8 which can be changed online.
The process for changing parameters is the same as in Version 7, except that these additional parameters are changed if the new set of system parameters differ.



Enhancements to system-level log point recovery for DB2 provide improved usability, more flexibility, and faster recovery. You can now recover your data to any point on the log, regardless of whether you have uncommitted units of work. As a result, data recovery time improves significantly for large DB2 systems that contain many thousands of objects. Two new utilities provide system-level point-in-time recovery:

The BACKUP SYSTEM utility provides fast volume-level copies of DB2 databases and logs. It relies on new DFSMShsm services in z/OS Version 1 Release 5 that automatically keep track of the volumes that need to be copied. BACKUP SYSTEM is less disruptive than using the SET LOG SUSPEND command for copy procedures. An advantage for data sharing is that BACKUP SYSTEM is group scope.

The RESTORE SYSTEM utility recovers a DB2 system to an arbitrary point in on the log. RESTORE SYSTEM automatically handles any creates, drops, and LOG NO events that might have occurred between the backup and the recovery point.



Batching of GBP writes and castouts

Write/castout multiple pages in a single CF operation

Improved data sharing performance, especially for batch updates

Requires z/OS R4, CFLEVEL=12

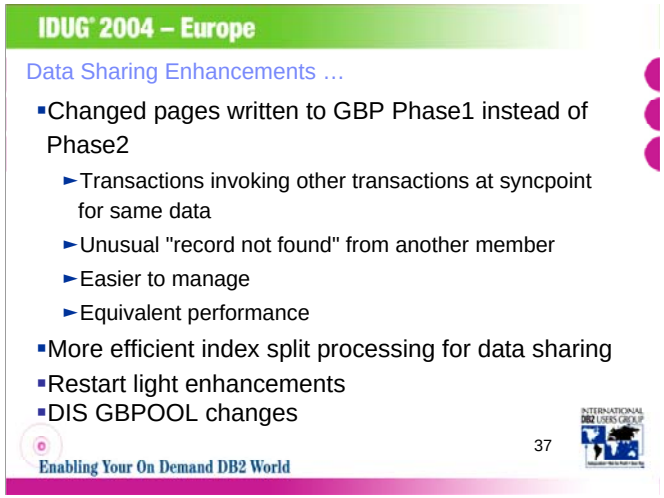
Reduced global contention for table space L-locks

IX/IX and IX/IS TS locks no longer hit XES-level contention across members

Improved data sharing performance, especially for OLTP

Recommendation for RELEASE(DEALLOCATE) can be softened

New locking protocol enacted only with New Function Mode, and requires quiesce of data sharing group



Changed pages written to GBP at Phase1 instead of Phase2

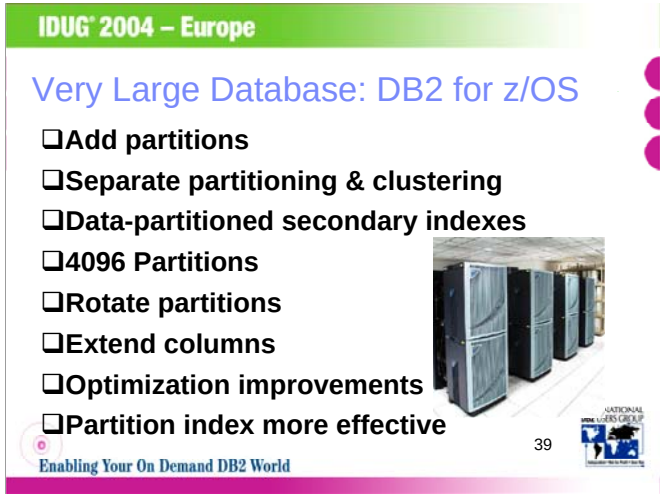
Some Tx Managers spawn other transactions at syncpoint

Spawned tx can encounter "record not found" if it tries to read originating tx's update from another member (rare, but a few customers have reported it)

Moving writes up to Phase1 by default removes need to monitor for this and to set IMMEDWRITE PH1 Zparm or Bind option if needed

Equivalent performance for Ph1 vs. Ph2 writes
More efficient index split processing for data sharing

Currently, database administrators need to specify primary and secondary space quantity. Then they need to monitor the data set sizes and extents to avoid an outage. Since the secondary extents are a fixed size, they must make the tradeoff between wasted space for many small tables and not having adequate space for rapidly growing tables. We can manage the extent size based upon the growth pattern of the table or index.

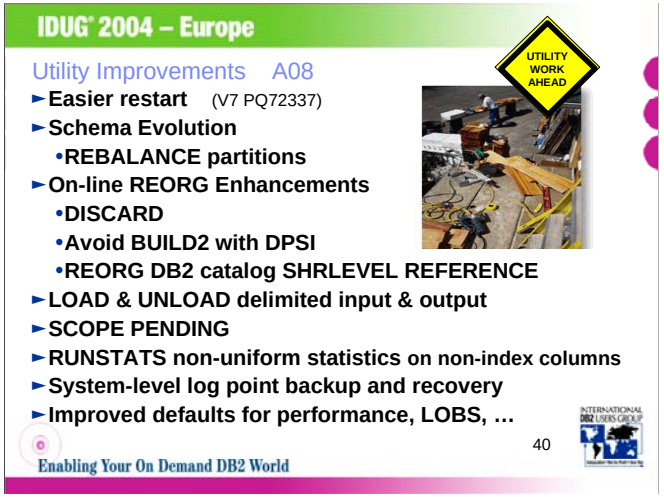


Very large databases face the combined challenge of very high performance needs, continuous availability and complexity. Improvements in scale and flexibility are more important in this area. Being able to have more partitions and to add them with ALTER are a big improvement.

Often it is useful to partition by date, so that we can archive or delete an entire partition, but processing will be much more efficient with another clustering order, such as by customer. Before this change, the clustering order was the same as the partitioning. This flexibility offers many opportunities for improved performance and availability.

Some customers have an index that is used only for partitioning the data or have extra columns at the beginning of the index. Being able to avoid the extra index or columns can improve our efficiency a lot.

For these very large tables, the ability to have more partitions, to add new partitions and to be able to rotate partitions is crucial.



Many utility enhancements are part of the base changes in this version, supporting long names, Unicode, 64 bit addressing, DPSIs, system backup and recovery, multilevel security and schema evolution. These utility enhancements improve value for the money.

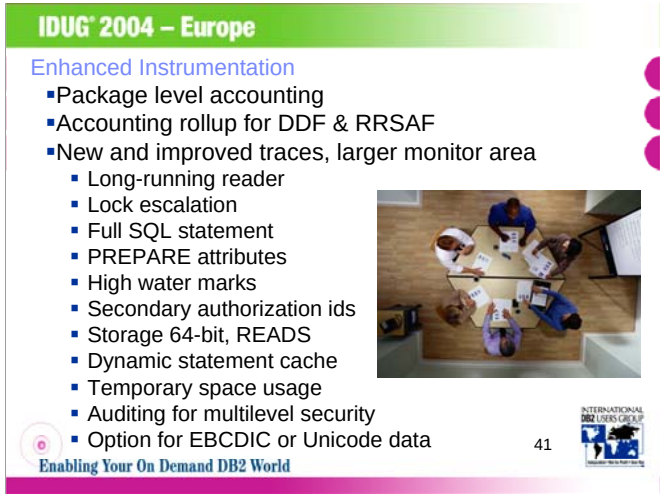
Schema evolution uses utility support to rotate the first partition to the last partition. The new REBALANCE function can balance the sizes of a partition range or of all partitions.

The REORG DISCARD can be performed with SHRLEVEL CHANGE. DPSIs can be reorganized without a BUILD2 phase. DB2 catalog tables can all be reorganized in SHRLEVEL REFERENCE or read only mode.

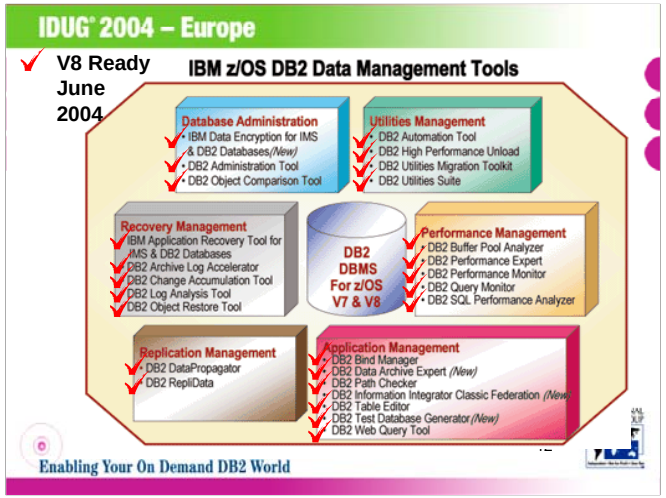
Delimited files can be input to LOAD or output from UNLOAD.

SCOPE PENDING provides improved usability. SCOPE PENDING indicates that only partitions in a REORP or AREO* state for a specified table space or partition range are to be reorganized.

ftp://ftp.software.ibm.com/software/db2storedprocedure/db2zos390/techdocs/Z06m.pdf



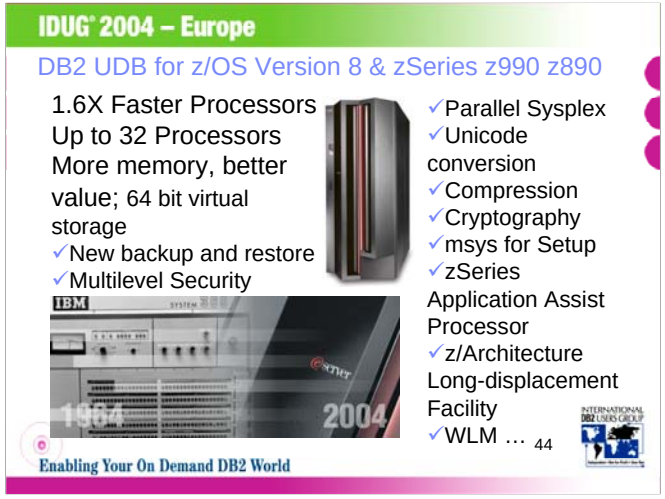
Many enhancements are made in instrumentation, helping to monitor and account for the larger and more varied workloads. Additional information is provided at a package level, if those traces are on. Accounting can roll up multiple trace records into one for DDF and RRSAF. A new IFCID is provided for lock escalation. The full SQL statement (not just 5000 bytes) can be traced with a new IFCID. The PREPARE statement attributes can be traced. The statement id is added to dynamic statement cache traces. Secondary ids can be retrieved with a synchronous read in an APAR that was added to V6 and V7. Additional fields were added to storage IFCIDs 225 and 217 for 64 bit addressing. Dynamic statement cache traces were improved to be more usable. A new IFCID 0342 was added for temporary space use by agents. Auditing was added for multilevel security. See the Release Guide, Appendix F for new and changed instrumentation.



Check marks indicate tools which are ready for DB2 UDB for z/OS by June 2004. All are ready now. With the exception of DB2 RepliData, ALL of these tools now exploit, not just tolerate, new DB2 V8 functions. See the tools web site for details and planning information. Click on Support to see exactly which levels are needed for V8.
http://www.ibm.com/software/data/db2imstools/

Fundamental to our tools strategy is to be able to extend and exploit the data base. To that end all of our tools exploit the features of DB2 V8 immediately at GA time.Our long term goal is to create tools that provide expert advice and automatic management features for DB2 to enable DB2 environments on all platforms to be easier to manage, require less administrative effort and less expertise to get outstanding performance and results. Tools are a long term and strategic initiative for IBM. We have increased our investment every year since we started in 2000. 2004 is no exception and we will be releasing new tools, release, versions and features every quarter, even as the portfolio is broadened. Candle tools have been added to our portfolio, and are ready for V8.

DB2 UDB for z/OS, Version 8 requires zSeries, 64-bit z/Architecture and z/OS V1.3 or later. V8 uses 64-bit virtual address spaces and also uses the changes in z/OS V1.4 and Coupling Facility CFLEVEL 12 for improvements in data sharing batch. z/OS V1.5, DFSMShsm™ Fast Replicate, DFSMSdss™, FlashCopy Version 2 and DB2 Utilities Suite for z/OS V8 are required for the new system-level backup and restore. Multilevel security is built upon z/OS V1.5 with Security Server RACF. V8 also provides additional uses for high performance cryptography, Unicode, Parallel Sysplex and data sharing.
See the Redbook Version 7 Performance Topics, chapter 10 for more discussion of synergy.

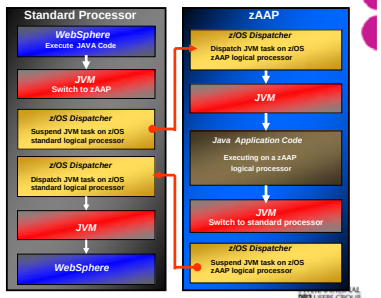


These are the key functions of the latest zSeries z990 and z890 that DB2 UDB for z/OS uses, almost everything to deliver **zSeries** ® and **z/OS**™ synergy**.** DB2 has used the function of the zSeries and z/OS platform extensively for many years. DB2 benefits from zSeries large real memory support, faster processors, and better hardware compression. DB2 uses Parallel Access Volume and Multiple Allegiance features of the IBM Enterprise Storage Server™ (ESS). ESS FlashCopy® is used for DB2 backup in combination with log suspend / resume. DB2 makes unique use of the z/Architecture™ instruction set, and a number of instructions provide improvement in reliability, performance and availability. DB2 continues to deliver synergy with hardware data compression, FICON™ (fiber connector) channels, disk storage, advanced networking function, and Workload Manager (WLM).

ibm.com/software/db2zos/ Click on Support, then on Frequently Asked Questions. Qualify the search with z990 to get the full page response.

zAAP Architecture and Workflow:
Executing Java under IBM JVM control

- Java work unit "eligible" to be dispatched on zAAP
- z/OS dispatcher attempts to dispatch eligible work on zAAP
- zAAP ineligible work only dispatched on standard processors
- May dispatch zAAP eligible work on standard processor
  - installation control



IBM JVM, parts of LE runtime, and z/OS Supervisor needed to support JVM execution can operate on zAAPs

IBM JVM communicates to z/OS dispatcher when Java code is to be executed

When Java is to be executed, the work unit is "eligible" to be dispatched on a zAAP

z/OS dispatcher attempts to dispatch zAAP eligible work on a zAAP (when present)

zAAP ineligible work only dispatched on standard processors

If there is insufficient zAAP capacity available, or standard processors are idle, the dispatcher may dispatch zAAP eligible work on a standard processor

There is an installation control to limit the use of standard processors to execute zAAP eligible work (see Java code execution options)

Greatest Hits: DB2 UDB for z/OS V8

- ✓High availability
- ✓Scalability or very large database
- ✓Java and the web
- ✓Queries and data warehouses
- ✓Migrating or porting applications
- ✓Application packages

webcast ... New DB2 V8 ROCKS

SAP PEOPLESOFT SIEBEL

Greatest hits are the situations where I would recommend looking at V8 soon. For these situations, V8 offers significant improvements and without adding much effort. Some cases will reduce the work. Value and effort vary widely for different customers and most of the improvements fit more than one category, so this list is not ordered.

Greatest Hit 1: High availability

One of the biggest steps for database administrators in continuous availability is online schema evolution, with the ability to add partitions and make about 20 changes with ALTER. New backup and recovery utilities are useful for disaster recovery and will be the primary backup technique for some customers. Improvements in utilities include more online performance and better usability.

Greatest Hit 2: Scalability or very large databases

Separate partitioning and clustering allows two dimensional clustering with more effective IO. New index options provide more efficient access. The maximum number of partitions is raised to 4096. The availability and optimization improvements are critical for very large databases. The ability to use more memory, more effectively is key for scalability.

Greatest Hit 3: Java and the web

Improvements in the SQLJ and JDBC support, a new Java Universal Driver, enhanced Unicode support, integration with WebSphere and new XML functions make Java and web applications more robust and more productive.

Greatest Hit 4: Queries and data warehouses

Optimization changes provide the best performance improvement opportunities in V8. Faster response and reduced processing time come from improved optimization and better information for the optimizer. New database design options for indexes, clustering and materialized query tables provide more gains. Warehouses often need to have the new rotate partition capability.

Greatest Hit 5: Migrating or porting applications from other platforms

Many SQL enhancements provide better compatibility with the DB2 family and with the industry. If customers develop on Windows, Unix or Linux, and then move to z/OS, the process is much easier. Early customers reported success at porting applications.

Greatest Hit 6: Application packages: SAP, PeopleSoft, Siebel, etc. …

About 50 improvements, including everything mentioned in the "Greatest Hits" section, are provided for most of the key vendor packages. SAP R/3 4.6 and PeopleSoft PeopleTools 8.45 are already certified for V8, less than four months after general availability, and more certifications are expected.

Version 8 is the twelfth and largest ever release of DB2 for z/OS. It brings new synergy with the zSeries hardware and uses the z/OS 64-bit virtual addressing capabilities. V8 improves data support, application development, and query function enhancements for e-business. It also builds on the traditional zSeries and DB2 characteristics of availability, exceptional scalability, and performance for the enterprise database management system of choice. V8 has been re-engineered for e-business on demand, with many fundamental changes in architecture and structure. Key improvements enhance scalability, application porting, security, architecture, and continuous availability. Management for very large databases is made much easier, while 64-bit virtual storage support makes management simpler and improves scalability and availability. This new version breaks through many old limitations in the definition of DB2 objects, including SQL improvements, online schema evolution, longer names for tables and columns, longer SQL statements, enhanced Java and Unicode support, enhanced utilities, more log data sets, more partitions, and many more advantages. Customers, vendors and consultants tell us that DB2 for z/OS Version 8 is exciting for them. It is a very important milestone.

Version 8 includes dozens of changes in SQL, improving family consistency in many cases, playing leapfrog in others, pushing DB2 SQL beyond current boundaries for enhanced application portability, open standards. Longer names for tables and columns mean that customers can use more meaningful names, matching standards. Longer SQL statements help with SQL that is generated or used in an SQL procedure. Here are some: multi-row INSERT, FETCH & UPDATE, GET DIAGNOSTICS, INSERT within SELECT, IDENTITY Column enhancements, SEQUENCES, CURRENT PACKAGE PATH, Dynamic Scrollable Cursors, Common Table Expressions, Scalar Fullselect, Materialized Query Tables, Unicode SQL, XML Publishing and much more. These changes improve our customer productivity, consistency across the DB2 family and ability to port applications. If you want to design or write applications for the entire DB2 family, then use the IBM DB2 Universal Database SQL Reference for Cross-Platform Development. Many barriers that limit our customers are removed: using 64 bit memory, providing consistent table and column name lengths, allowing 2 megabyte SQL statements, 4096 partitions, and three to ten times the log space. Customers encountered many limitations over the past 20 years, and lifting the limits required extensive reengineering for some. Other limits allow improved scalability and availability, such as tripling the active log size and ten times the archive logs. Allowing sixteen times the number of partitions allows a table to use one partition per day for eleven years.

Key performance enhancements deliver better family consistency and run many times faster. Being able to make database changes without an outage, such as adding a partition, is a breakthrough for availability. Alter your table and go, no need to drop and redefine. Online schema evolution is more resilient. The most important change for many customers is the ability to use ALTER in many places instead of needing to drop and redefine. We call this online schema evolution, and it can reduce outages by hours or days for a major database structure change. Database administrators can add a partition to an existing partitioned table space or rotate the partitions. Other changes in online schema evolution allow better partitioning and improved disk access, avoiding random access with more effective database designs. Many enhancements improve our integration with zSeries, z/OS and with key vendor applications like PeopleSoft, SAP and Siebel.

The DB2 UDB for z/OS web page has pointers to almost all available DB2 for z/OS information, including the complete library of books, an information center, announcements, papers, answers to frequently asked questions and presentations. You can get to the Library or to the Version 8 Information Center by clicking on Library, and then choosing the Information Center or one of the versions. Then you can download or view most of the books.

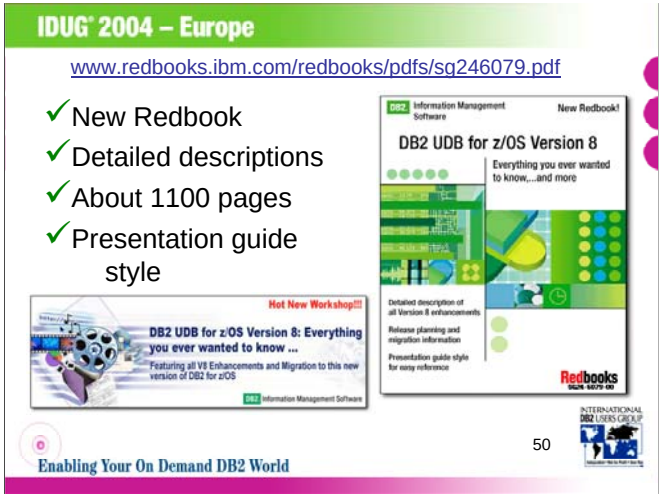
If you want to see the Version 8 materials, go to the V8 page. Many of the materials are on the Support page. You can get most DB2 and IMS tool information from the Tools page. Application developers and database administrators will be interested in the Applications or DeveloperWorks page. If you want information about conversion from other DBMS to DB2, see DeveloperWorks.

http://www.ibm.com/developerworks/db2/

http://www.ibm.com/developerworks/db2/zones/porting/

The easiest way to see new papers or presentations is directly on the ftp site. Sort by date to get the latest ones.

This is the main DB2 UDB for z/OS web page. You can get to the other DB2 for z/OS pages from here, so I often call this my home page. This page changes frequently, so look at the highlighted NEW items. Do you want to look in a DB2 book? Click on Library to see books on Version 8 (about 40), Version 7, 6, 5 or even 4. You can check the latest changes by looking at the Information Updates or go to the Information Center. From this page, you can look for conferences (Events), specific classes (Education), or services. If you want to see the latest on Version 8, click on the Version 8 link. If your primary concern is application development, the Developer Domain is for you. DB2 Magazine covers a broad range of topics about DB2. The latest machines z990 and z890 are on the zSeries page. Click DB2 and IMS Tools to see the wide range of help they provide.



This new IBM Redbook introduces the many enhancements made available with DB2 UDB for z/OS Version 8. It will help you understand the enhancements, and provides information to help you to evaluate the applicability to your environment. This book will help you plan for the installation of DB2 V8 or the migration from DB2 V7, with the Library.

http://www.redbooks.ibm.com/redbooks/pdfs/sg246079.pdf

http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg246079.html?Open

Bibliography: DB2 for z/OS

- DB2 for z/OS V8 Everything you ever wanted to know & more, SG24-6079
- DB2 for z/OS Stored Procedures: Through the CALL and Beyond, SG24-7083
- DB2 for z/OS V8: Through the Looking Glass & What SAP Found, SG24-7088
- SAP on DB2 for z/OS and OS/390: High Availability & Performance, SG24-6950
- DB2 for z/OS and OS/390: Ready for Java, SG24-6435
- PeopleSoft V8 on zSeries Using Data Sharing & Enterprise Storage, SG24-6093
- DB2 UDB for z/OS Version 8 Technical Preview, SG24-6871
- Distributed Functions of DB2 for z/OS and OS/390, SG24-6952
- Moving Data Across the DB2 Family, SG24-6905
- Implementing PeopleSoft 8.4 on zSeries, SG24-6549
- Siebel 7 with DB2 for z/OS: Database Implementation & Admin., SG24-6868
- SAP on DB2 UDB for z/OS: Multiple Components in One Database, SG24-6914
- SAP on DB2 UDB for z/OS: High Availability System Automation, SG24-6836
- A Practical Guide to DB2 UDB Data Replication V8, SG24-6828
- Squeezing the Most Out of Dynamic SQL, SG24-6418
- Large Objects with DB2 for z/OS and OS/390, SG24-6571
- DB2 for z/OS V7 Selected Performance Topics, SG24-6894
- DB2 for z/OS Application Programming Topics, SG24-6300
- DB2 for z/OS & OS/390 V7 Performance Topics, SG24-6129

If you want to look across the DB2 family, start at ibm.com/software/db2

Click Support or start at

www.ibm.com/software/data/support/

Then choose a product.
If you want to look at DB2 for z/OS, start at ibm.com/software/db2zos

Click Support or start at

www.ibm.com/software/data/db2/os390/support.html
If you want to look at DB2 for UNIX & Windows, start at ibm.com/software/db2/udb

Click Support or start at

www.ibm.com/software/data/db2/udb/support.html

Happy reading.

Disclaimer and Trademarks

Information contained in this material has not been submitted to any formal IBM review and is distributed on "as is" basis without any warranty either expressed or implied. Measurements data have been obtained in laboratory environment. Information in this presentation about IBM's future plans reflect current thinking and is subject to change at IBM's business discretion. You should not rely on such information to make business plans. The use of this information is a customer responsibility.

IBM MAY HAVE PATENTS OR PENDING PATENT APPLICATIONS COVERING SUBJECT MATTER IN THIS DOCUMENT. THE FURNISHING OF THIS DOCUMENT DOES NOT IMPLY GIVING LICENSE TO THESE PATENTS.

TRADEMARKS: THE FOLLOWING TERMS ARE TRADEMARKS OR ® REGISTERED TRADEMARKS OF THE IBM CORPORATION IN THE UNITED STATES AND/OR OTHER COUNTRIES: AIX, AS/400, DATABASE 2, DB2, e-business logo, Enterprise Storage Server, ESCON, FICON, OS/390, OS/400, ES/9000, MVS/ESA, Netfinity, RISC, RISC SYSTEM/6000, iSeries, pSeries, xSeries, SYSTEM/390, IBM, Lotus, NOTES, WebSphere, z/Architecture, z/OS, zSeries, @server

The FOLLOWING TERMS ARE TRADEMARKS OR REGISTERED TRADEMARKS OF THE MICROSOFT CORPORATION IN THE UNITED STATES AND/OR OTHER COUNTRIES: MICROSOFT, WINDOWS, WINDOWS NT, ODBC, WINDOWS 95

For additional information see ibm.com/legal/copytrade.phtml

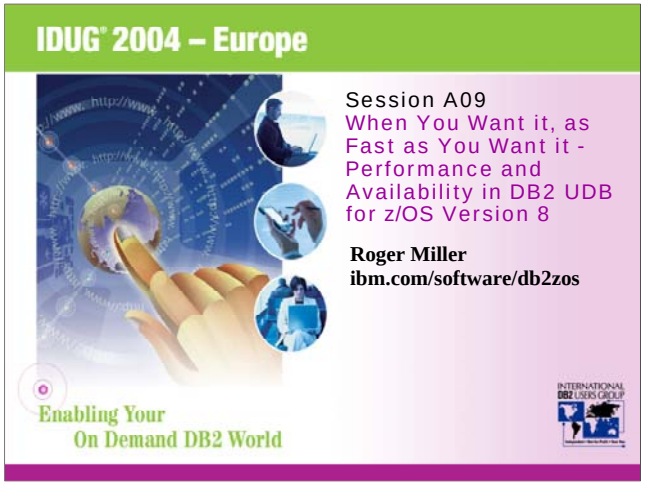
This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice. Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in the operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only. 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.



A09: This is an overview of improved performance & availability in DB2 ® UDB for z/OS Version 8 (V8). DB2 V8 is the twelfth release and delivers more function than any release of DB2 for MVS. DB2 V8 became generally available March 26, 2004. This version brings extensive integration and synergy with zSeries hardware, with middleware and with applications. Data support, application development and query enhancements are added for e-business, building upon the traditional enterprise of choice characteristics of availability, exceptional scalability, and performance.
DB2 Version 8 has been re-engineered for e-business on demand, with many fundamental changes in architecture and structure. Key improvements enhance scalability, application porting, security, and continuous availability. Management for very large databases is made much easier, 64-bit virtual storage support makes management simpler and improves scalability and availability. This new version breaks through many old limitations in the definition of DB2 objects. These enhancements include SQL improvements, schema evolution, longer names for tables and columns, longer SQL statements, enhanced Java and Unicode support, enhanced utilities, more log data sets, and a lot more.