



IBM Software Group

DB2 UDB for z/OS Version 8 Overview

Roger Miller, DB2 UDB for z/OS

ibm.com/software/db2zos

DB2 Information Management Software

@business on demand software

This is an overview of 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.

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



The 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.

Continuous Availability



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

Scalability and Very Large Database

- Add partitions
- Separate partitioning & clustering
- Index improvements
- 4096 Partitions
- Rotate partitions
- Extend columns
- Optimization improvements
- Memory and scale increased



4

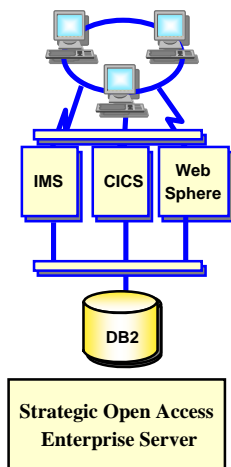
Very large databases face the combined challenge of very high performance needs, continuous availability and complexity. Improvements in scale and flexibility are more important in this area. Being able to have more partitions and to add them with ALTER are a big improvement.

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.

Java and the web



- **Application Connectivity for JAVA**
- **Improve data sharing member routing**
- **LOCATION aliases at requester & server**
- **Extended function, standards**
- **Reduced processing**
- **Stored Procedure Enhancements**
- **Java Universal Driver**

DB2 V8 also provides many improvements for Java and the web: security, availability, usability and performance. The improved data sharing member routine is more robust. Having the ability to use multiple names for a server adds flexibility. Increased levels of standards are implemented. Performance improvements will reduce cpu costs substantially. Improved granularity for stored procedures help with managing your work load, using the WLM.

The Java Universal Driver is used across the DB2 family.

Queries and data warehouses

- Optimization Improvements
 - ❑ Improved techniques
 - ❑ Enhanced data
 - ❑ Visual Explain
- Enhanced index options
- Materialized Query Tables
- New Partitioning options
- QMF improvements
- SQL enhancements



Queries and data warehousing are improved a lot in V8. Optimization improvements provide a performance boost and make the job simpler. Improved optimization techniques like ability to use indexes more, star join and scale improvements allow reduced work for computers and for people. Enhanced data helps get the best access path. Visual Explain improves the ability to analyze and resolve any problems.

The many improvements for indexes, materialized query tables and partitioning can save space and add new options for improved performance and availability, even while simplifying the process. Not padded, clustering, longer and backward scans help indexes. Being able to add, rotate and rebalance partitions improve partitioning options. SQL enhancements on this page and the next improve portability of the SQL, improve the ability to express queries, and help with performance. QMF enhancements build upon these strengths and add new function to reporting, dash boards, and a new platform in WebSphere.

Migrating and porting applications



- Multi-row INSERT, FETCH & UPDATE
- GET DIAGNOSTICS
- INSERT within SELECT
- IDENTITY Column enhancements
- SEQUENCES
- CURRENT PACKAGE PATH
- SQL Procedure Language
- Dynamic Scrollable Cursor, Common Table Expression, Recursion
- Scalar Fullselect
- Materialized Query Table
- UNICODE SQL, Multiple CCSIDs
- XML Publishing
- Long names, long statements...




7

Improvements in the SQL have made migrating from other platforms, such as Unix and Windows much easier. Many of the early customers moved applications.

Version 8 is a breakthrough in SQL, with too many new functions to list them all. We will discuss a few of them. Add support for volatile tables, group by expression, multiple DISTINCT clauses, and qualified names for INSERT and UPDATE and many more to the list above.

Table function improvements, trigger performance, star join sparse index improvements, cost-based parallel sort, better ability to use indexes, longer statements, longer names, longer literals and predicates, session variables, new special registers and SQL procedures improvements, provide a giant leap for SQL.

Enterprise Applications & : DB2 UDB for z/OS

 4.6 certified	<ul style="list-style-type: none"> -64 bit virtual storage -Unicode -Schema evolution -System-level backup and recovery 	<ul style="list-style-type: none"> -Longer Table Names & Column Names -Additional statistics -Convert Column Type -Altering CLUSTER Option
	 8.45 certified	<ul style="list-style-type: none"> -Multi-row fetch & insert -Multiple DISTINCT Clauses -Lock Contention on volatile Tables -Fast Retrieval of Most Recent Value
 7.8 certified		

Most of the key items in this version help our key enterprise application partners: SAP, PeopleSoft and Siebel, but also many other applications and customers. The 2 GB address space limit is a major problem for customers who have large dynamic statement cache, many concurrently open data sets and long running units of work. Unicode helps vendors support multinational companies and improve character conversions. These applications often insert and fetch many rows. Using a technique called array fetch & insert or multirow fetch & insert can be as much as 50% less cpu time locally, but improves remote access more. SQL flexibility improvements allow DB2 to be efficient in performance and in productivity for our partners. There are roughly 50 items, with some differences, for each partner. See the book DB2 for z/OS V8: Through the Looking Glass & What SAP Found, SG24-7088 for more detail. SAP R/3 4.6 and PeopleSoft PeopleTools 8.45 certified for V8 in May 2004. Siebel 7.8 was certified in 2005.

Scaling DB2 UDB for z/OS



Image of Earth from Moon, Source: NASA (Public Domain)

Breaking through limitations: scaling

- **Virtual Storage 2 GB** 2^{31} to 2^{64}
- **Partitions** 254 to 4096
- **Open data sets** 32 K to 64 K
- **Active logs** 31 to 93
- **Archive logs** 1000 to 10,000
- **Maximum table size** 16 TB to 128 TB
(partitioned, 32K page)

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.

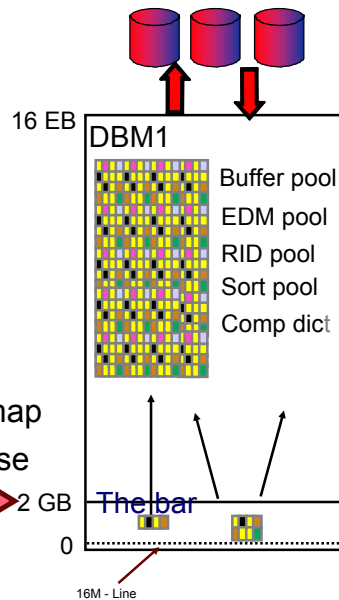
Increasing the amount of virtual storage we can address directly can help with the ability to scale and simplify management for virtual storage. It will require more real memory, but permit increased scalability and availability.

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.

Doubling the number of concurrent open data sets provides the needed ability to handle more data sets. This change needs z/OS 1.5. See PQ96189.

64 bit evolution

- zSeries, z/OS, z/Architecture & large real storage
 - ▶ V6 data space advantages
 - zSeries & 64 bit addressing
- Version 8 64 bit virtual
 - ▶ z/OS 64-bit Virtual Storage Roadmap
 - ▶ Improve scalability, availability, ease
 - ▶ Move above the bar
 - ▶ Hiperpool/dataspace no longer needed, no ECSA for locks



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 room will you get in DBM1?

It depends.

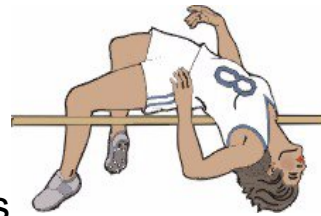
Any virtual pools?

Buffer pool control blocks

Castout pool

EDM DBDs pool, Sort Pool

RID Pool, compression dictionaries



Some growth in thread storage

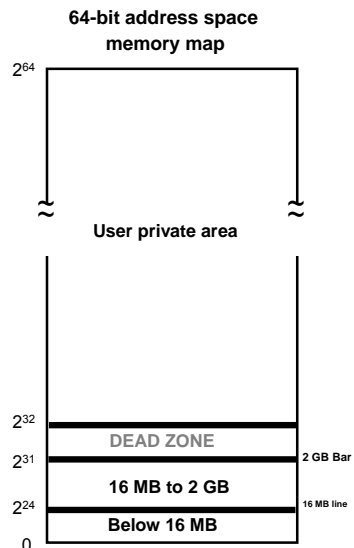
You need to calculate, monitor and manage

Some customers get big improvements in virtual storage, but others get very little. If you use little or no compression, have already minimized sort pools, RID pools, castout (no data sharing) and have moved the dynamic statement cache and buffers to data spaces, then the usable storage can shrink. With the new sizes and options in V8, some of the space grows. On the other hand, in about 80% of the cases we looked at, there are some virtual buffer pools left or substantial savings are available from the buffer pool control blocks (estimate 5% of data space and hiperspace buffer pools), data sharing castout, DBDs, sort pool, RID pool (estimate 90%), and compression dictionaries.

You need to do the calculations, monitor and manage storage in V8.

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



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 (e)logo)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.

Limits: DB2 for z/OS



Image of Earth from Moon, Source:
NASA (Public Domain)

Breaking through limitations - SQL

- **Table name sizes** 18 to 128
- **VIEW & ALIAS names** 18 to 128
- **Column name sizes** 18 to 30
- **SQL statement length** 32K to 2MB
- **Tables in a join** 15 to 225
- **Character Literals** 255 to 32704
- **Hex literal digits** 255 to 32704
- **Predicates** 255 to 32704
- **Index key** 255 to 2000

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 name sizes & SQL statement lengths makes porting from other DBMS much easier and improves DB2 family compatibility. Increasing the amount of virtual storage allows the longer names, larger SQL statements and increased sizes. DB2 V8 will often require a little more real memory (1% to 10% for large subsystems), but permit increased scalability and availability.

Limits: DB2 for z/OS



Image of Earth from Moon, Source:
NASA (Public Domain)

Breaking through limitations SQL ...

- **Current optimization** 8 to 128
- **CURRENT PACKAGESET** 18 to 128
- **CURRENT PATH** 254 to 2048
- **SCHEMA** 8 to 128
 - **Authorization ids still 8 bytes**

The larger number of tables in a join helps with porting applications and improves our ability to handle vendor applications.

The additional active and archive logs provides larger volumes and better flexibility for the amount of log data. Customers can keep up to 372 GB of active log data and 40 terabytes of archive log data.

Maximum table sizes increase for partitioned table spaces with 8K (to 32 TB), 16K (to 64 TB) and 32K pages (to 128 TB).

Increasing the special register lengths allows more flexibility for applications.

SQL and DB2 Family



- Multi-row INSERT, FETCH & UPDATE
- GET DIAGNOSTICS
- INSERT within SELECT
- IDENTITY Column enhancements
- SEQUENCES
- CURRENT PACKAGE PATH
- SQL Procedure Language
- Dynamic Scrollable Cursors
- Common Table Expressions, Recursion
- Scalar Fullselect
- Materialized Query Tables
- UNICODE SQL, Multiple CCSIDs
- XML Publishing
- Long names, long statements...

Version 8 is also a breakthrough in SQL, with too many new functions to list them all. We will discuss a few of them on the next pages. Add support for volatile tables, group by expression, multiple DISTINCT clauses, and qualified names for INSERT and UPDATE.

Table function improvements, trigger performance, star join sparse index improvements, cost-based parallel sort, better ability to use indexes, longer statements, longer names, longer literals and predicates, session variables, new special registers and SQL procedures improvements, provide a giant leap for SQL.

Multi-row INSERT



- Inserts multiple rows on one API call
- Can be ATOMIC or NOT ATOMIC
- Can be static or dynamic SQL
- Significant performance boost

```
INSERT INTO T1
VALUES( :ARRAY1, :ARRAY2)
FOR :hv ROWS
NOT ATOMIC CONTINUE ON SQLEXCEPTION;
```

Being able to insert multiple rows helps in application portability and can improve performance, particularly across a network.

You can have atomicity (all rows must be successfully inserted or all are rolled back) or insert and then resolve any errors.

This change supports host language arrays, adding another in a range of customer enhancement requests.

Multi-row INSERT - continued



```
STMT = 'INSERT INTO T1
        VALUES( ?, ?)
        FOR MULTIPLE ROWS ATOMIC';

PREPARE S1 FROM STMT;

EXECUTE S1 FOR :HV ROWS
        USING :ARRAY1, :ARRAY2;
```

Here is another example of multi-row insert. This example is using dynamic SQL to insert multiple rows with a single SQL statement using host variable arrays. At run time, the number of rows to be inserted is retrieved from HV, and that number of values is obtained from ARRAY1 and ARRAY2. The application could continue to execute that statement.

Multi-row FETCH



- Returns multiple rows on one API crossing
- "wide" cursor with locks on multiple rows
- Supports scrollable and non-scrollable, static and dynamic SQL
- Significant performance boost

```
DECLARE C1 CURSOR
  WITH ROWSET POSITIONING
  FOR SELECT COL1, COL2 FROM T1;
OPEN C1;
FETCH NEXT ROWSET FROM C1
  FOR :hv ROWS INTO :ARRAY1, :ARRAY2;
```

Multiple row FETCH also helps with application portability and performance. It provides a new concept called a "wide" cursor, which contains multiple rows rather than just one. A rowset is a set of rows that is retrieved through a multiple-row fetch.

Being able to FETCH multiple rows at once can make a larger percentage improvement in performance, compared to INSERT. The FETCH statement requires less processing than INSERT, generally.

Positioned UPDATE and DELETE of Multi-row FETCH



- Allows positioned UPDATE or DELETE to be used on a "wide" cursor
- Update or delete all rows
- Or update or delete a specific row

```
UPDATE T1 SET COL1='ABC'  
WHERE CURRENT OF CURSOR C1  
FOR ROW :hv OF ROWSET
```

The concept of the wide cursor was extended to the UPDATE and DELETE. This shows how to update a specific row of the rowset in a wide cursor. If the cursor is positioned on a rowset, and the UPDATE is WHERE CURRENT OF cursor-name, then all rows corresponding to the rows of the current rowset are updated.

GET DIAGNOSTICS



- Returns SQL much more error information
 - ▶ for overall statement
 - ▶ for each condition (when multiple errors occur)
- Supports SQL error message tokens greater than 70 bytes (SQLCA limitation)

```
INSERT INTO T1 FOR 5 ROWS VALUES(:ARRAY);  
GET DIAGNOSTICS :ERR_COUNT = NUMBER;  
DO II = 1 TO ERR_COUNT;  
    GET DIAGNOSTICS FOR CONDITION :II  
        :RC = RETURNED_SQLSTATE;  
END;
```

The new GET DIAGNOSTICS statement is important to provide the information from all of the extended names and new function. Most programmers will need to switch from using the less standard SQLCA and use this more standard, more capable facility for diagnostic information when their applications include long names or multi-row operations.

The GET DIAGNOSTICS statement can return a lot more information about the statement and / or about conditions and connections. It can return the longer names. It can return multiple conditions for the multi-row statements. It can return the error message associated with an error.

GET DIAGNOSTICS information



- **Statement** LAST ROW, PARAMETER MARKERS, NUMBER, RESULT SETS, NUMBER ROWS, RETURN STATUS, CURSOR ATTRIBUTES, MORE, NUMBER, ROW COUNT
- **Condition** CATALOG NAME, CONDITION NUMBER, CURSOR NAME, ERROR CODEs, MESSAGE ID, TOKENs, REASON CODE, SQLCODE, ROW NUMBER, SQLERRD, TOKEN COUNT, MESSAGE TEXT, SQLSTATE, SERVER NAME
- **Connection** AUTHENTICATION TYPE, AUTHORIZATION ID, CONNECTION STATE, STATUS, ENCRYPTION TYPE, SERVER CLASS NAME

In addition to the information that can only be returned using GET DIAGNOSTICS, such as for multiple rows or for long names, there are many items that now have a standard technique for returning them to the application. There is a more standard technique for getting information about the statements, more information about any warnings or error conditions and information about the connection.

INSERT within SELECT

- Elegant technique for retrieving values created / modified by DB2 during INSERT
 - ▶ identity columns, sequence values
 - ▶ user-defined defaults, expressions
 - ▶ columns modified by triggers
 - ▶ ROWIDs, CURRENT TIMESTAMP, ...
 - ▶ INSERT with return or SELECT from INSERT



EXAMPLE:

```
SELECT C1, C2, C3, C4, C5 FROM  
INSERT (C1, C5) INTO T1  
VALUES('ABC', CURRENT DATE);
```

Have you wanted to return values from the row you just inserted? Applications often need the value of the identity column, the defaults, expression results, current timestamp or the effect of a trigger. Now you can SELECT FROM the inserted row. While there have been some specialized techniques, this is more general and more elegant.

Some programmers think of this as being able to return the generated values from an insert or insert with returns.

New Sequence Options

- Identity columns
 - ✓ Column of a table
 - ✓ Alterable & new attributes
 - ✓ SELECT FROM INSERT
- Sequence
 - ✓ Separate object, new data definition, grant, ...
 - ✓ Alterable & similar attributes
 - ✓ SELECT FROM INSERT
- GENERATE_UNIQUE value unique across Sysplex



Identity columns were introduced in DB2 UDB for OS/390 Version 6, but Version 8 provides needed enhancements for usability. New attributes are added for sequences and the attributes can be altered, so that operations like LOAD are easier. A new SELECT FROM INSERT statement provides a standard way to retrieve the result of inserting an identity column.

Sequences, like identity columns, provide an incremented counter within the DBMS. While the identity column is in a table, the sequence is a separate, standalone object. Most of the attributes of a sequence are very similar to those of an identity column. New data definition SQL statements are added to CREATE, ALTER and DROP sequences. Sequences are helpful in porting Oracle applications to DB2, while identity columns are more like the SQL Server construct.

A new GENERATE_UNIQUE function is also added to generate a CHAR(13) FOR BIT DATA that is unique across a data sharing group or Parallel Sysplex.

Identity Column Improvements

- ALTER support for Identity Columns
 - ▶ GENERATED ALWAYS | BY DEFAULT
 - ▶ **RESTART WITH value**
 - ▶ INCREMENT BY
 - ▶ MINVALUE | **NO MINVALUE**
 - ▶ MAXVALUE | **NO MAXVALUE**
 - ▶ CYCLE | NO CYCLE
 - ▶ CACHE | NO CACHE
 - ▶ **ORDER | NO ORDER**



(**Bold** keywords are new in V8)

When identity columns were provided with Version 6, customers identified some important enhancements that were needed. Many customers asked for the ability to ALTER the identity column attributes, especially the ability to change the GENERATED option. This added flexibility will allow identity columns to be used in many more situations.

New identity column attributes can be specified to aid porting from other vendor implementations: RESTART WITH, NO MINVALUE, NO MAXVALUE, NO ORDER, ORDER. The identity column is a style often used by those who use Sybase.

Sequences

- Useful for porting Oracle applications
- New SQL support:
 - ▶ CREATE SEQUENCE
 - ▶ ALTER SEQUENCE
 - RESTART WITH value
 - INCREMENT BY
 - MINVALUE, MAXVALUE
 - CYCLE/NO CYCLE
 - CACHE/NO CACHE
 - ▶ GRANT/REVOKE privileges for sequence object
 - ▶ NEXT VALUE FOR expression
 - ▶ PREVIOUS VALUE FOR expression



Sequences, like identity columns, provide an incremented counter within the DBMS. While the identity column is in a table, the sequence is a separate, standalone object. Most of the attributes of a sequence are very similar to those of an identity column. New data definition SQL statements are added to CREATE, ALTER and DROP sequences.

Sequences are helpful in porting Oracle applications to DB2. The language to use a sequence is to specify NEXT VALUE FOR expression or PREVIOUS VALUE FOR expression in your SQL statement.

Dynamic Scrollable Cursors



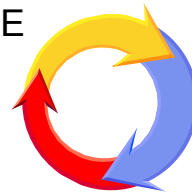
- Scrollable cursor provides access to base table rather than work file -- allows viewing of updates by others.
- Defaults to single row fetch, so DDF applications should use:
 - ▶ multi-row FETCH
 - ▶ positioned update/delete for multi-row FETCH

```
DECLARE C1 SENSITIVE DYNAMIC SCROLL  
CURSOR FOR SELECT C1, C2 FROM T1;
```

Static scrollable cursors came in V7, with the ability to use a work file for scrolling. Dynamic cursor scrolling is performed directly on the base tables. You can use embedded SQL for scrolling, and this work complements the multi-row SQL statements.

Common Table Expressions (CTE) & Recursion

- Refer to table-name later in SQL
- Allow in SELECT, CREATE VIEW, INSERT
- WITH table-name [column-list] AS fullselect
- For recursion fullselect of CTE refers to itself
fullselect that does not refer to CTE
UNION ALL
fullselect that refers to CTE



Common Table Expressions provide improved usability and DB2 family consistency. In some cases they can be used to improve performance as well, computing a value once, rather than several times.

The common table expression is defined and used within an SQL statement. Each common table expression can be referenced many times in the statement, and all references to a common table expression in an SQL statement share the same result table. This is unlike regular views or nested table expressions which are derived each time they are referenced.

Common Table Expression & Recursion Example

```
WITH RPL (PART, SUBPART, QUANTITY) AS  
  ( SELECT R.PART, R.SUBPART, R.QUANTITY  
    FROM PARTLIST AS R WHERE R.PART = '01'  
    UNION ALL  
    SELECT C.PART, C.SUBPART, C.QUANTITY  
      FROM RPL AS PARENT, PARTLIST AS C  
      WHERE PARENT.SUBPART = C.PART    )  
  SELECT DISTINCT PART, SUBPART, QUANTITY  
  FROM RPL  
  ORDER BY PART, SUBPART, QUANTITY;
```

This example shows the syntax for using a common table expression (WITH to the end of the column list, then the fullselect. The fullselect includes the initialization SELECT clause, UNION ALL, and the SELECT clause which refers to the CTE.

The SELECT statement is from the CTE, and then specifies the sort order.

CURRENT PACKAGE PATH



- Important for SQLJ and DDF apps
 - Less need for SET CURRENT PACKAGESET
 - Less network traffic for multiple PKLIST values
 - Easier to switch to/from JDBC and SQLJ
 - Better handling of nested stored procedure packages
- Allows application to specify search list of package schemas (similar to PKLIST)

```
SET CURRENT PACKAGE PATH =  
ALPHA, BETA, PROD
```

Package switching and versioning for static SQL applications is critical to DB2 for z/OS customers. SQLJ access will increase the need for these types of control. A new special register, CURRENT PACKAGE PATH, provides a means to specify a list of collections to search for the appropriate package.

The semantics are similar to the PKLIST Bind Option, except that the PACKAGE PATH list is processed at the server. This new special register will provide control for applications that do not run under a DB2 plan.

Scalar fullselect

- Allows scalar fullselect where expressions were previously supported

```
SELECT C1 FROM T1 WHERE  
T2>(SELECT COL1 FROM T2 ...)
```

```
SELECT C1,  
      (SELECT COL1 FROM T2...),  
      C3  
FROM T1
```

The scalar fullselect or a fullselect that results in a single scalar value to be used where an expression is allowed today. The scalar fullselect can be used in the SELECT clause, in the WHERE clause, or in a CASE expression, for instance. This improves DB2 function, the ability to port applications and DB2 family compatibility.

A scalar fullselect, as supported in an expression, is a fullselect, enclosed in parentheses, that returns a single row consisting of a single column value. If the fullselect does not return a row, the result of the expression is the null value.

Scalar fullselect - 2

```
UPDATE NEW_PARTPRICE N
SET PRICE =
CASE
  WHEN( (SELECT ONHAND#
        FROM INVENTORY
        WHERE PART=N.PART) < 7 )
    THEN 1.1 * PRICE
  WHEN( (SELECT ONHAND#
        FROM INVENTORY
        WHERE PART=N.PART) > 20 )
    THEN .8 * PRICE
  ELSE PRICE
END;
```

This is an example of a scalar fullselect in a CASE expression. You can see that this construct increases the power of the CASE expression.

VOLATILE Table Support

- Encourages index access for tables that have unpredictable cardinality
- Significant performance improvement for some SAP applications

```
CREATE TABLE XYZ ..... VOLATILE;  
ALTER TABLE XYZ ..... VOLATILE;
```

Where applications have tables that are volatile, it is difficult or perhaps impossible to find a good time to gather statistics. Volatile tables have a wide range of cardinality. For example, a table might have no rows part of the time and a million rows at other times.

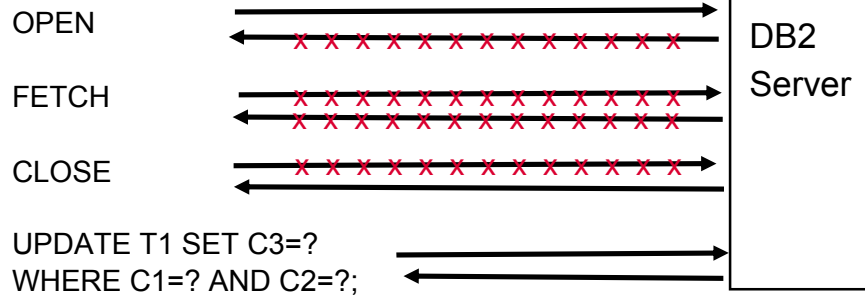
Customers or vendors can use this option on CREATE TABLE or ALTER TABLE to indicate that the table is volatile, and index access should be encouraged.



READ ONLY KEEP UPDATE LOCKS

- Allows WebSphere persistence layer to minimize network traffic when using searched update and pessimistic locking.

```
PREPARE "SELECT C1, C2, C3 FROM T1
WHERE C1=? AND C2=? FOR READ ONLY
KEEP UPDATE LOCKS"
```



The WebSphere persistence layer currently uses ISOLATION(RS) to retrieve one or more rows with the FOR UPDATE clause when loading the WebSphere entity beans. The Java application is then allowed to perform updates on those beans, and the updates are subsequently sent to DB2 as searched UPDATE statements. WebSphere uses this approach to minimize the number of open cursors at runtime.

With DB2 V8, we will be able to obtain exclusive locks with a FOR READ ONLY query. This will allow the JDBC driver and DDF to use block fetch for the SELECT (eliminating the extra network messages required with a FOR UPDATE query), while still obtaining and holding the locks WebSphere needs for the searched UPDATE statement. This will provide significant CPU and elapsed time improvements.



IS NOT DISTINCT FROM

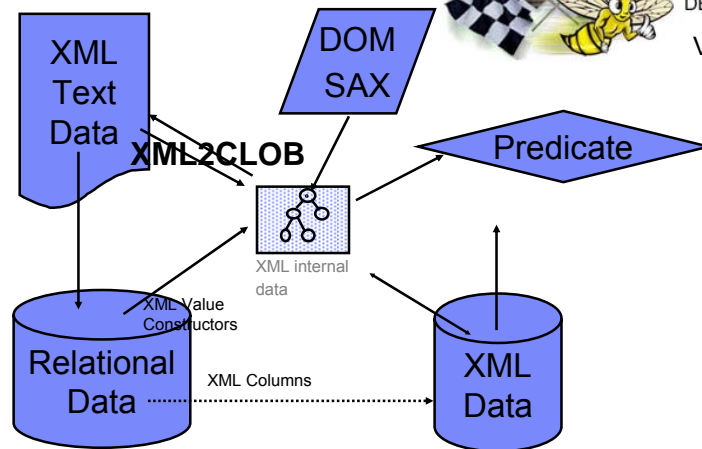
- SQL uses three-valued logic where any given comparison can return: TRUE, FALSE, or NULL
- Applications can use IS NOT DISTINCT FROM to obtain a TRUE result instead of NULL when comparing NULL values

```
SELECT C1 FROM T1 WHERE  
C1 IS NOT DISTINCT FROM :hv;
```

C1 value	:hv value	RESULT
NULL	'ABC'	FALSE
NULL	NULL	TRUE
'ABC'	'ABC'	TRUE
'ABC'	NULL	FALSE
'ABC'	'DEF'	FALSE

This is often called the distinct predicate. By definition a null value is unknown and this makes it unequal to all other values, including other null values. The only way to test for null values is to use the IS NULL predicate, as in "WHERE col IS NULL". A predicate of the form "WHERE col = :hv :nullind" will never match a null value in "col", even if the host variable "nullind" contains a null indicator. Unfortunately, this is not intuitively obvious. The IS NOT DISTINCT FROM predicate provides an explicit technique to compare and treat two NULL values as equal.

SQL/XML - XML extensions to SQL

DB2 UDB
V8.2**XML value constructors:**

**XMLELEMENT (xmlattributes), XMLFOREST,
XMLCONCAT, XMLAGG, XMLGEN, XMLNAMESPACES**

XML Publishing Functions provide a set of SQL built-in functions that allow applications to generate XML data from relational data with high performance. These functions reduce application development efforts in generating XML data for data integration, information exchange, and web services, thus enhance the leadership position of DB2 for z/OS as an enterprise database server.

XML Publishing Functions SQL/XML

- New built-in functions for generating XML
 - ▶ **XMLELEMENT** generates XML element from arguments
 - ▶ **XMLATTRIBUTES** specify attributes for the XML element within XMLELEMENT
 - ▶ **XMLFOREST** creates forest of XML elements that share a specific pattern
 - ▶ **XMLCONCAT** concatenates arguments to generate a forest of XML elements
 - ▶ **XMLAGG** aggregate produces forest of XML elements from a collection of XML elements
 - ▶ **XML2CLOB** converts transient XML data type into UTF-8 CLOB
 - ▶ **XMLNAMESPACES** generates XML namespace declarations
- Augmenting XML generation function of XML Extender

The DB2 XML Extender is supported on z/OS. In V8, we are pushing more XML support into the engine by providing support for some of the built-in XML publishing functions that are currently in the process of being standardized in the ANSI SQL committee, such as XMLELEMENT, XMLATTRIBUTES, XMLFOREST, XMLCONCAT, XMLAGG, XML2CLOB and XMLNAMESPACES.

The new XML publishing functions replace the XML generation function of XML Extender.

Transparent ROWID column

- Eliminates the need to explicitly declare a ROWID column in tables that include LOBs
- DB2 will generate a "hidden" ROWID column, which is not visible on SELECT *
- Simplifies porting of LOB applications from other platforms



DB2 UDB for z/OS currently requires a ROWID column to be included in tables which have LOBs. In V8, DB2 will generate a ROWID column if one is not present, and will not include this column in a SELECT *. These changes make application porting simpler with LOBs.

ON COMMIT DROP enhancement



- Declared Temporary Table improvement
- Temp table dropped automatically at COMMIT if no HELD cursors open on the temp table
- Significant improvement for DDF threads that use CMTSTAT=INACTIVE
 - Allows thread to go INACTIVE

This change is a significant change for distributed work that can be switched from active to inactive. Temporary tables that do not have any HELD cursors open are dropped automatically at COMMIT. This will allow the threads to be switched to INACTIVE. The ON COMMIT DROP TABLE clause is added to the DECLARE GLOBAL TEMPORARY TABLE SQL statement. The clause indicates that the declared global temporary table should be implicitly dropped at COMMIT if there are no open cursors on the table that are defined as WITH HOLD.

SQL statements up to 2M bytes



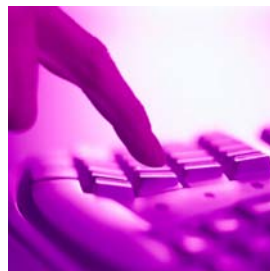
DB2 UDB
V8.2

- SQL statements can be up to 2M bytes in length
- Parse tree has been completely rearchitected to reduce SQL too complex SQLCODE -101 scenarios due to SQL statement length
- Important for SQL Procedure Language applications
- Needed for generated SQL, long names, increased numbers of partitions, ...

In addition to other limit-breaking support in Version 8, you can now have SQL statements that are up to 2 MB in length. A number of the Version 8 capabilities stretch the limit on the size of an SQL statement. Long names and 4096 partitions take much more space. An SQL Procedure must be completely stated in a single SQL statement, and was limited to 32K. Other changes in DB2 allow much larger structures and thus much larger statements. SQL statements that are too large or too complex should be very unusual, but still possible, with the larger statements.

SQL Procedure Language extensions

- RETURN statement
- SIGNAL/RESIGNAL support
- CREATE PROCEDURE up to 2M bytes
- Integrated debugger
- GET DIAGNOSTICS for all fields in SQLCA
- Get values from RETURN
- Enhanced support for labels
- Iterate statement



Several of the changes to SQL are very important for SQL procedures. The limit of 32 kilobytes for an SQL statement was for the entire SQL procedure, and that limit is now 2 megabytes. The ability to return information to the caller of an SQL procedure is improved with the RETURN statement issued from within an SQL procedure. The GET DIAGNOSTICS statement (already valid within SQL procedures) is extended to support returning the status information from a RETURN statement of an SQL procedure (the new RETURN_STATUS keyword). Other DB2 platforms support the RETURN statement and the GET DIAGNOSTICS enhancement within SQL procedures, so these changes improved DB2 family consistency.

Other SQL Improvements and DB2 Family Compatibility

- GROUP BY expression
 - SELECT A+B, ... FROM T ... **GROUP BY A+B**
- SELECT statement with ORDER BY
 - SELECT ... FETCH FIRST ROW ONLY **ORDER BY A**
- Qualified column names on UPDATE SET clause
 - UPDATE T1 SET **T1.COL1**...
- Qualified column names on INSERT
 - INSERT **T1.COL1** INTO T1 VALUES...
- Multiple DISTINCT clauses
 - SELECT COUNT(**DISTINCT**(A1)), AVG(**DISTINCT**(A2)) ...
- Column functions → more general aggregate functions

We are making big strides in SQL improvements, including: GET DIAGNOSTICS, SEQUENCE, Dynamic scrollable cursors, Scalar fullselect, Multiple DISTINCT clauses, GROUP BY expression, Qualified column names on INSERT and UPDATE SET clause, Unicode for SQL, intermixing EBCDIC, ASCII, or Unicode columns in a single SQL statement and SELECT from an INSERT statement. In addition, we are making major changes in our SQL system limits by extending support for long names in SQL objects to 30 characters for column names and to 128 characters for most other SQL objects; expanding support for SQL statements up to 2 megabytes in length; increasing the length of literals and predicates to 32 kilobytes; and extending JOIN to allow up to 225 tables in a single statement.

www7b.boulder.ibm.com/dmdd/library/techarticle/0209cotner/0209cotner.html

Improved Security

- Multilevel security with row level granularity
- Session variables
- Special registers for identification
- RACF access control & DB2 operator commands
- Encryption and decryption built-in functions
- Encryption for DRDA on the wire



Everyone seems to be more aware of security today. Improving integration and making security more robust and easier to manage are very important.

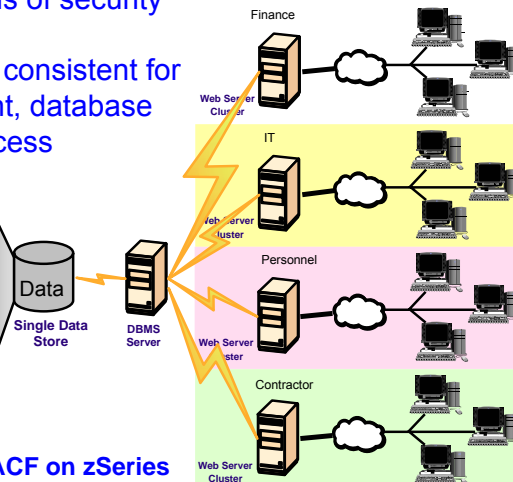
Customers asked for a wide range of enhancements for security. DB2 V8 provides new options for higher security, more granularity, and more information for additional flexibility in applications and SQL.

Improvements include improvement for RACF access control, encryption and decryption built-in functions and encryption for data on the DRDA wire.

Multilevel Security and DB2 UDB for z/OS V8

- Labeled security allows sharing of resources with mixed levels of security in a single image
- Integrated access control, consistent for files, communications, print, database
- Control SQL and utility access

SECURITY LABEL	Col 1	Col 2	Col 3
Personnel	234	USA	50%
Finance	198	France	23%
Personnel	2	UK	9%
Finance	234	USA	11%
Personnel	22	Germany	9%
IT	87	USA	14%
Contractor	23	UK	20%
Personnel	34	Germany	43%
Finance	98	USA	12%
IT	223	USA	10%
Contractor	45	Canada	29%



Multilevel Security: z/OS 1.5 & RACF on zSeries

43

Architecture

z/OS 1.5 and RACF 1.5 or Security Server add another type of security, called multilevel security, labeled security or mandatory access control (MAC) to our capabilities. The only option in the past with a high degree of separation has been physical separation. In the database world that might mean another machine or LPAR or perhaps another subsystem, another database or another table. With multilevel security, we still have a high degree of security even with data in the same table.

Access control is consistent across many types of resources using RACF, so that multilevel controls apply for data sets, for communications, for print and for database access – both objects and now with row level granularity. The DB2 controls are for both SQL access and for utility access.

For an more on multilevel security, see **Planning for Multilevel Security and Common Criteria (GA22-7509)**

<http://publibz.boulder.ibm.com/epubs/pdf/e0z2e100.pdf>

<http://publibz.boulder.ibm.com/epubs/pdf/e0z2e111.pdf>

Multilevel Security and DB2 Row-Level Security Revealed, SG24-6480

<http://www.redbooks.ibm.com/redpieces/pdfs/sg246480.pdf>

Multilevel Security by Row



Sally 
SECLABEL='RAINBOW'

Joe 
SECLABEL='PASTEL'

Sam 
SECLABEL='SUNSET'

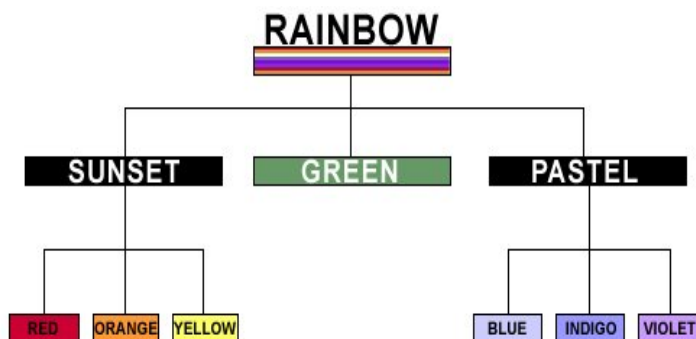
DB2 SECURITY LABEL_EXT	COL1	COL2	COL2
RAINBOW	56	7	76
RAINBOW	24	56	65
RAINBOW	42	6	45
BLUE	3	456	7
INDIGO	113	456	56
VIOLET	3	456	4
BLUE	4	456	7
RED	4	76	567
ORANGE	33	7	567
RED	5455	76	567
YELLOW	999	65	45

- Table column defined AS SECURITY LABEL
- Check each new seclabel value SQL & utilities
- Mandatory access control: run time user to data
- Integration with z/OS V1R5 and RACF

DB2 V8 adds row-level security for applications that need more granularity in their security. For example, in organizational hierarchies, it is desirable to set up a corresponding security hierarchical scheme in which employees can see their own payroll data, a first line manager can see payroll information on all of the reporting employees, and so on. Security schemes often include a security hierarchy such as TOP SECRET, SECRET, or UNCLASSIFIED. When companies need to store multiple customers data in a single DBMS, security and privacy may demand row level security. The granularity must be extended from table level to row level for individual user access to be restricted to a specific set of rows. Traditionally, views and joins have been the application solution to limit access to selected rows and columns, but they can be cumbersome to construct with the desired level of granularity and not very effective for update / insert / delete. Triggers, database constraints, and stored procedures are often needed for update control. DB2 V8 provides mandatory access control with multilevel security (MLS) with row granularity. This support combines with new z/OS and RACF access control functions available with z/OS V1R5.

<ftp://ftp.software.ibm.com/software/db2storedprocedure/db2zos390/techdocs/Z15.pdf>

Multilevel Security by Row ...



45

With the hierarchy established in the security manager layer, the system would understand that users with authority to access RAINBOW can access anything. Someone with authority to access PASTEL information can access any row associated with BLUE, INDIGO, VIOLET, or PASTEL. Someone with SUNSET can access SUNSET, RED, ORANGE, YELLOW. This is a lot more powerful than just having an exact match on security label (i.e., user's label must exactly match the data's label), since it has the notion of "groups" that make security administration easier to manage.

With this additional capability, we'll be able to implement that type of security scheme without requiring the application to access the data using special views or predicates. See the z/OS 1.5 book, Planning for Multilevel Security and the DB2 UDB for z/OS V8 Release Guide for more.

<http://publibz.boulder.ibm.com/epubs/pdf/e0z2e100.pdf>

<http://publib.boulder.ibm.com/epubs/pdf/dsnrgj10.pdf>

Row Granularity Multilevel Security



Table has column defined AS SECURITY LABEL

Each row value has a specific security label

Get security labels from RACF

Save in rows for INSERT, UPDATE, LOAD, ...

Check for each new seclabel value accessed

If access is allowed, then normal access

If access is not allowed, data not returned

Runtime user to data checking

Seclabel values are cached to minimize cpu

Requires z/OS V1R5 and Security Server (RACF)

We have had many requests for row-level security for applications that need more granular security schemes. For example, in organizational hierarchies, it is desirable to set up a hierarchy in which employees can see their own payroll data, a first line manager can see his or her payroll information and all of the employees reporting to that manager, and so on. In addition, government security schemes often include a security hierarchy such as TOP SECRET, SECRET, or UNCLASSIFIED. We are currently developing support for row-level security that includes support for these types of hierarchical security schemes. You can use this support by adding a column that acts as the security label.

Session Variables



Variables set by DB2, connection or signon exit

Built in function to retrieve value for a variable

Use function in views, triggers, stored procedures
& constraints to enforce security policy

Can have more general, flexible access checks

Multiple columns, AND/OR logic, ...

PLAN, PACKAGE, SECLABEL, VERSION, ...

Complements other security mechanisms

```
CREATE VIEW V1 AS SELECT * FROM T1 WHERE  
COL5 = GETVARIABLE('SYSIBM.SECLABEL');
```

Session Variables provide another way to provide information to applications. Some variables will be set by DB2. Others can be set in the connection and signon exits to set these session variables

A new built-in function GETVARIABLE is added to retrieve the values of a session variable. This function can be used in views, triggers, stored procedures and constraints to help enforce a security policy. If your primary security need is more general, flexible controls, this information complements other security mechanisms.

For example, you can have a view which provides data that is at the current security label.

New Special Registers

Client information for this connection

Provided by sqleseti, Java methods, RRS
SIGNON & SET_CLIENT_ID

- CLIENT_ACCTNG accounting string
- CLIENT_APPLNAME value of application name
- CLIENT_USERID client user ID
- CLIENT_WRKSTNNAME workstation name

Four new SPECIAL REGISTERS are added to the product. These special registers are CLIENT_ACCTNG, CLIENT_APPLNAME, CLIENT_USERID, and CLIENT_WRKSTNNAME. The information is provided through a number of application programming interfaces.

These special registers were added to DB2 for Linux, UNIX & Windows V8.

New Built-in Functions

- ENCRYPT_TDES: encrypt column in a table with a user-provided encryption password
- ENCRYPTION PASSWORD special register
- DECRYPT_BIT, DECRYPT_CHAR, DECRYPT_DB
- GET_HINT: obtain hint to help remember ENCRYPTION PASSWORD
- GENERATE_UNIQUE creates CHAR(13) FOR BIT DATA value that is unique across Sysplex
- DRDA encryption on the wire

Functions ENCRYPT_TDES (triple DES), DECRYPT_BIN, DECRYPT_CHAR, and GETHINT are added. The SET ENCRYPTION PASSWORD statement allows the application to specify a password. The ability to generate a unique value is also included. These changes came in DB2 for Linux, UNIX and Windows V8, so this change improves DB2 family consistency.

DRDA is extended to allow encryption of the data being sent.

DB2 UDB Family SQL Reference



Version 2
September 2004

IBM DB2 Universal Database

SQL Reference for Cross Platform Development

z/OS OS/400 AIX HP-UX Solaris Linux Windows

SQL Reference for DB2 UDB family

<http://www.ibm.com/developerworks/db2/library/techarticle/0206sqlref/0206sqlref.html>

Version 2 adds DB2 UDB for z/OS V8

DB2 UDB for iSeries V5R3

Do you want to design or write applications for the entire DB2 family, rather than for just one of the platforms? If so, you need the IBM DB2 Universal Database SQL Reference for Cross-Platform Development.

Version 2 covers these product versions:

- DB2 UDB for Linux, UNIX & Windows V8

- DB2 UDB for z/OS V8

- DB2 UDB for iSeries V5 R3

SQL consistency across the DB2 family has improved substantially in the past few versions, while significant new common function has been added. SQL that is common to the DB2 UDB relational database products and the SQL 1999 Core standard is much more comprehensive. The 852 page book describes the rules and limits for preparing portable programs for these versions. More is coming.

<http://ibm.com/developerworks/db2/library/techarticle/0206sqlref/0206sqlref.html>

DB2 UDB Family SQL Reference



New features:

- Sequences
- CURRENT SCHEMA special register
- Scalar subselect
- Built-in functions, encryption support
- Common table expressions
- Lateral correlation
- Group by expressions
- Materialized query tables
- Alter identity columns
- ITERATE statement
- Increased SQL limits, enhanced long string support
- Other portability enhancements

IBM DB2 Universal Database SQL Reference for Cross-Platform Development. Version 1 & (1.1) covers these product versions:
DB2 UDB for Linux, UNIX & Windows V7 R2 (V8.1)

DB2 UDB for z/OS V7

DB2 UDB for iSeries V5 R1 (V5 R2)


These are the primary enhancements in the Version 2 Reference over the 1.1 version. Many more SQL statements will port across the three platforms. These enhancements also improve the ability to port from other DBMS to the DB2 family.

<http://ibm.com/developerworks/db2/library/techarticle/0206sqlref/0206sqlref.html>

IBM Software Group | DB2 Information Management Software IBM

DB2 Family SQL

z z/OS V7
common
UW Linux, Unix & Windows V8.2



z {

c { Inner and Outer Joins, Table Expressions, Subqueries, GROUP BY, Complex Correlation,
o Global Temporary Tables, CASE, 100+ Built-in Functions, Limited Fetch, Insensitive Scroll
m Cursors, UNION Everywhere, MIN/MAX Single Index Support, Self Referencing Updates with
n Subqueries, Sort Avoidance for ORDER BY, and Row Expressions, Call from trigger,
 statement isolation

U { Updateable UNION in Views, ORDER BY/FETCH FIRST in subselects & table expressions,
W GROUPING SETS, ROLLUP, CUBE, INSTEAD OF TRIGGER, EXCEPT, INTERSECT,
 16 Built-in Functions, MERGE, Native SQL Procedure Language, SET CURRENT
 ISOLATION, BIGINT data type, file reference variables, SELECT FROM UPDATE, DELETE
 & MERGE, multi-site join, 2M Statement Length, GROUP BY Expression, Sequences, Scalar
 Fullselect, Materialized Query Tables, Common Table Expressions, Recursive SQL,
 CURRENT PACKAGE PATH, VOLATILE Tables, Star Join Sparse Index, Qualified Column
 names, Multiple DISTINCT clauses, ON COMMIT DROP, Transparent ROWID Column, FOR
 READ ONLY KEEP UPDATE LOCKS, SET CURRENT SCHEMA, Client special registers,
 long SQL object names, SELECT from INSERT


52

This text just shows the relationship of DB2 for Linux, Unix & Windows with DB2 for z/OS and OS/390 Version 7, comparing a March 2001 z/OS version with an October 2004 LUW version. The next step in the process is DB2 UDB for z/OS Version 8 and then the next release of DB2 for Linux, Unix & Windows, code named Stinger at this point. There are three sets of SQL noted above, with none that is unique to DB2 UDB for z/OS in the first group, SQL that is common across DB2 UDB for Linux, Unix, Windows and z/OS in the large group in the middle, then SQL that is unique to DB2 UDB for Linux, Unix and Windows in the bottom group. Sheryl M. Larsen, Inc. provided the base for this information. www.smlsql.com

IBM Software Group | DB2 Information Management Software IBM

DB2 Family SQL

z z/OS V8
common
UW Linux, Unix & Windows V8.2



Z	{	Multi-row INSERT, FETCH & multi-row cursor UPDATE, Dynamic Scrollable Cursors, GET DIAGNOSTICS, Enhanced UNICODE for SQL, join across encoding schemes, IS NOT DISTINCT FROM, Session variables
c o m m o n	{	Inner and Outer Joins, Table Expressions, Subqueries, GROUP BY, Complex Correlation, Global Temporary Tables, CASE, 100+ Built-in Functions including SQL/XML, Limited Fetch, Insensitive Scroll Cursors, UNION Everywhere, MIN/MAX Single Index Support, Self Referencing Updates with Subqueries, Sort Avoidance for ORDER BY, and Row Expressions, 2M Statement Length, GROUP BY Expression, Sequences, Scalar Fullselect, Materialized Query Tables, Common Table Expressions, Recursive SQL, CURRENT PACKAGE PATH, VOLATILE Tables, Star Join Sparse Index, Qualified Column names, Multiple DISTINCT clauses, ON COMMIT DROP, Transparent ROWID Column, Call from trigger, statement isolation, FOR READ ONLY KEEP UPDATE LOCKS, SET CURRENT SCHEMA, Client special registers, long SQL object names, SELECT from INSERT
U W	{	Updateable UNION in Views, ORDER BY/FETCH FIRST in subselects & table expressions, GROUPING SETS, ROLLUP, CUBE, INSTEAD OF TRIGGER, EXCEPT, INTERSECT, 16 Built-in Functions, MERGE, Native SQL Procedure Language, SET CURRENT ISOLATION, BIGINT data type, file reference variables, SELECT FROM UPDATE, DELETE & MERGE, multi-site join

53

This text just shows the relationship of DB2 for Linux, Unix & Windows with DB2 for z/OS, comparing the z/OS version from March 2004 with the LUW version from October 2004. The next step in the process is the next release of DB2 for Linux, Unix & Windows, code named Stinger at this point. There are three sets of SQL noted above, with some that is unique to DB2 UDB for z/OS in the first group, SQL that is common across DB2 UDB for Linux, Unix, Windows and z/OS in the large group in the middle, then SQL that is unique to DB2 UDB for Linux, Unix and Windows in the bottom group. Sheryl M. Larsen Inc provided the base for this information. www.smlsql.com

In earlier releases, there was nothing unique to z, much more that was unique to LUW, and much less in common.

If you want to improve DB2 family consistency, then DB2 UDB for z/OS Version 8 is a big step, changing the game from one of catch up to one of leapfrog.

Performance: DB2 UDB for z/OS

Performance Enhancements



- Ability to use indexes more often
 - More matching in predicates
 - Index backward scan
 - Index varying length
- Distribution statistics on non-index columns, improved optimization
- Materialized Query Tables
- Multi-row fetch & insert
- Long Term Page Fix Bufferpool option
- Data Sharing improvements

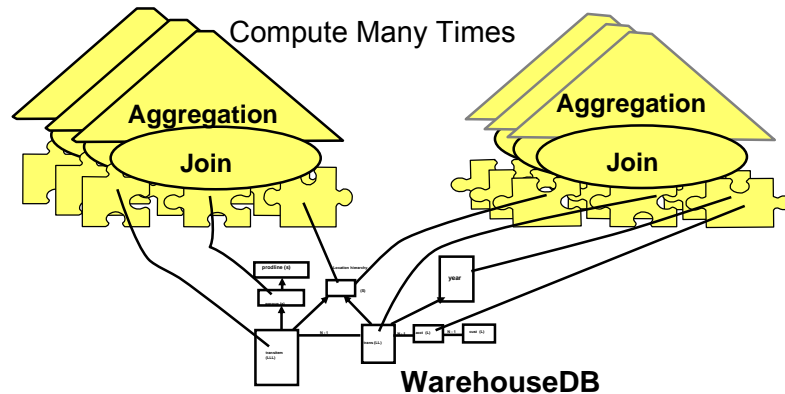
Many of the performance enhancements come from the enhanced index capabilities. Comparing a decimal column to a floating point number could not be done in the past. Being able to compare these values can mean choosing a better index or even an index access instead of a table space scan. An index can 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.

Without Materialized Query Tables: Each Query Re-Computes!

Q11,Q12, ...

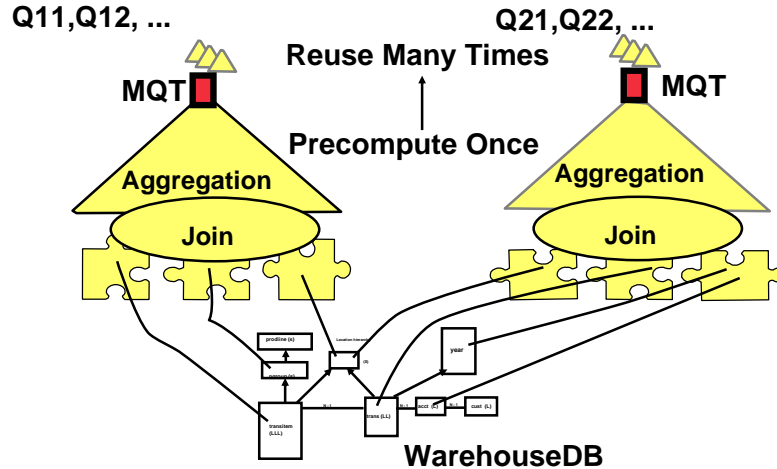
Q21,Q22, ...



55

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.

With Materialized Query Tables: Avoid Redundant Computation



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.

Materialized Query Tables

- Sometimes called Summary Tables, Automatic Summary Tables, Automatic Materialized Query Tables, Materialized Views, ...
- Optimizer can rewrite queries to access MQT instead of base table or view
- Pre-computed information, very significant performance improvement
- Managed by user or system (SQL REFRESH)
- Automatic rewrite or manual
- Informational Referential Integrity (not enforced)

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.

Star Join Sparse Indexes and in-memory work files



- Optimizer improvement that addresses the same requirement as Hash Join
 - ▶ **ACCESS_TYPE='T' in PLAN table**
 - ▶ **Uses sparse index to process the contents of work file**
 - ▶ **Improves upon APAR PQ61458 on V7**

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 the 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.

Parallel Sort



- Optimizer is enhanced to use parallel sort for multiple tables when cost tradeoff is advantageous

Currently "sort-for-composite" is not pushed down for parallelism if the composite involves more than one table. Elapsed time suffers because "sort" is not executed in parallel.

Due to the overhead of parallel sort, it may not be cost effective to execute sort in parallel for small sort data. The optimization is enhanced with costs for the parallel sort.

Trigger Performance

- TRIGGER work files are eliminated when old/new transition variables are returned for small number of rows
- Very significant performance enhancement when few or no triggers fired



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.

Table Function Improvements



- CARDINALITY clause
 - ▶ SELECT * FROM TABLE(TABUDF(C2)
CARDINALITY 30) AS ABC
- Performance improvement to move data between table functions and DB2 using block data movement

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.

RUNSTATS Distribution Statistics Enhanced



- **Non-uniform distribution statistics on non-indexed columns can improve performance 10X, 100X**
- **RUNSTATS improvement allows optimizer to consider non-uniform distribution statistics on columns that aren't part of an index**
- **Current technique is separate DSTATS program**
- **Significant performance improvement**
- **RUNSTATS with UPDATE NONE REPORT NO**
- **HISTORY statistics without updating main statistics**

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.

Multirow performance

- **FETCH:**
 - Up to 50% cpu savings
 - DSNTIAUL change made
 - DSNTEP4 instead of DSNTEP2
 - Distributed processing
 - QMF APAR PQ99482
- **INSERT**
 - Up to 30% cpu savings
 - Best for CICS, distributed, class 2 accounting, many rows, few columns



Here are some program changes already implemented to save cpu time. If you use DSNTIAUL, 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. QMF with APAR PQ99482 implements this change. 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.

Long Term Page Fix

- ALTER BUFFERPOOL
 - New option PGFIX(YES)
 - By buffer pool
 - Use where IO rate is high
 - Must have real storage
 - Up to 8% cpu savings
 - Allowed in all modes, e.g. compatibility mode



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.



REOPT(ONCE) BIND Option

- **Controls when DB2 builds access path for dynamic SQL**
- **Default, access path is calculated at PREPARE.**
- **REOPT(ONCE)**
 - **Defers access path selection until OPEN**
 - **Values of host variables on OPEN used to calculate access path**
 - **Resulting access path cached in global prepare cache**

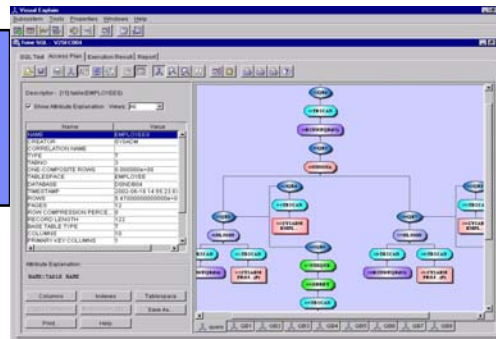


REOPT(ONCE) is a new bind option that tries to combine the benefits of REOPT(VARS) and dynamic statement caching. For an SQL statement with input host variables, no matter it is static or dynamic, the access path chosen by the optimizer during bind time (before the values of host variables are available) is not optimal in general. Bind option REOPT(VARS) solves this problem by (re)preparing the statement at run time when the input variable values are available, so that the optimizer can re-optimize the access path using the host variable values. However, for frequently called SQL statements that take very little time to execute, re-optimization using different input host variable values at each execution time is expensive, and it may affect the overall performance of applications. REOPT(ONCE) will re-optimize the access path only once (using one set of input variable values - currently the first set) no matter how many times the same statement is executed. The access path chosen based on the set of input variable values is stored in cache and will be used for all later executions. This solution is based on the assumption that the chosen set of host variable values at run time are better than the default ones chosen by optimizer at bind time.

EXPLAIN for global prepare cache

- Enhancements to the EXPLAIN statement allow you to obtain EXPLAIN information for entries in the DB2 global prepare cache
- Visual Explain uses this new function.

**EXPLAIN STMTCACHE
STMTID=integer
STMTTOKEN=string**



66

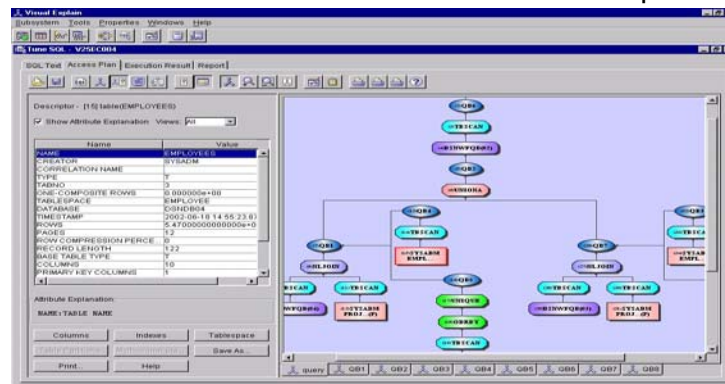
A new option of EXPLAIN is added to allow retrieving information for SQL statements in the Dynamic Statement Cache. This means that the current access path can be examined, not hoping that a current EXPLAIN will be the same. Visual Explain can use the new function.

For more on the new Visual Explain, see the next foil and also other Visual Explain and optimization information on the web:

<http://www.ibm.com/software/data/db2/zos/osc/ve/index.html>
<ftp://ftp.software.ibm.com/software/data/db2zos/Z34.pdf>

Visual Explain Enhancements: Version 7 and 8

- Significant improvements to Visual Explain tool:
 - ▶ Much more detailed information about access path
 - ▶ More statistical details for each node in the graph
 - ▶ XML document describing query access path selected
 - ▶ Easier collection of information to send for help



67

While optimization continues to improve, it has become more and more complex. SQL statements that are 2 MB long, allowing many more tables to be joined, materialized query tables, improved use of sparse indexes for star joins and making more predicates indexable are examples.

Explain table has been rearchitected to include filter factor estimation, cardinality estimation, cost estimation, parallelism, and system configuration. This information can be printed using XML structures, allowing the information to be collected more easily. We expect this process to make problem access path resolution easier.

Visual Explain can be downloaded now from the web and can be used on Version 7 or Version 8 subsystems, although some of the new function is only provided with Version 8.

<http://www.ibm.com/software/data/db2/zos/osc/ve/index.html>

<ftp://ftp.software.ibm.com/software/db2storedprocedure/db2zos390/techdocs/OA3.pdf>

<ftp://ftp.software.ibm.com/software/db2storedprocedure/db2zos390/techdocs/Z31m.pdf>

<ftp://ftp.software.ibm.com/software/db2storedprocedure/db2zos390/techdocs/Z27.pdf>

Enhanced Instrumentation

- Package level accounting
- Accounting rollup for DDF & RRSF
- New and improved traces, larger monitor area
 - Long-running reader
 - Lock escalation
 - Full SQL statement
 - PREPARE attributes
 - High water marks
 - Secondary authorization ids
 - Storage 64-bit, READS
 - Dynamic statement cache
 - Temporary space usage
 - Auditing for multilevel security
 - Option for EBCDIC or Unicode data



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 RRSF. 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.

Larger CI size for larger page sizes

- New CI size equals page size by default
 - e.g. 16K CI for 16K page, DSNZPARM to set
- Enables I/O striping for 8K, 16K, 32K page
- Higher data rate for 8K, 16K, and 32K page
 - 16K page measurement with 16K instead of 4K CI
 - ✓ +36% for non EF (Extended Format) data sets
 - ✓ +70% for EF data sets
 - ✓ VSAM EF getting nearly equivalent to non EF in data rate performance

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. The improvement in performance is better for Extended Format data sets, so that having a larger DSSIZE is more reasonable for performance.

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



73

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.

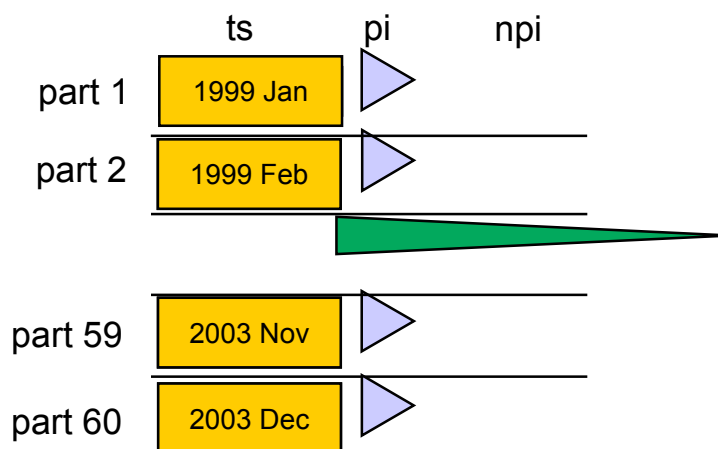
Adding a new partition to an existing partitioned tables space is very important. Rotating the partitions, such as keeping a rolling 36 months of data is also key.

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>

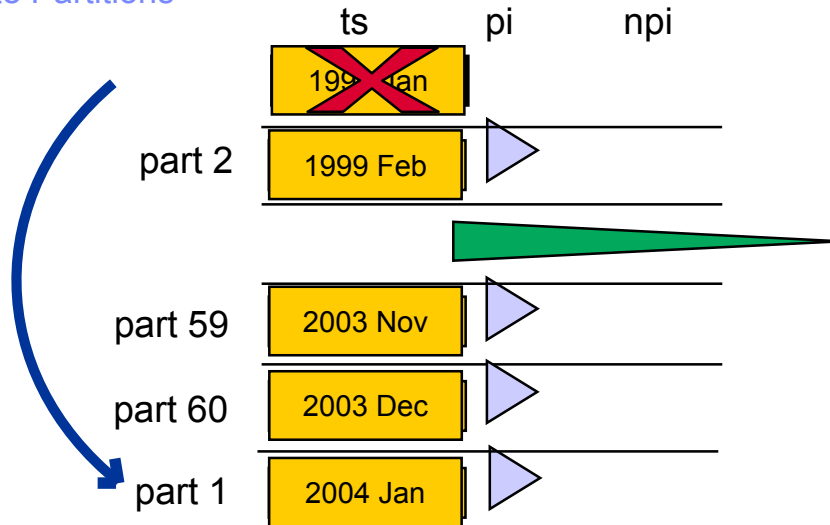
<ftp://ftp.software.ibm.com/software/db2storedprocedure/db2zos390/techdocs/Z22m.pdf>

Rotate Partitions



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.

Rotate Partitions



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.

Online Schema Evolution 2

- Table or column Changes
 - ▶ Increase column within numeric data types
 - smallint, integer, decimal, float
 - No loss of precision allowed
 - ▶ Change to expand character data type
 - ▶ Change varchar to / from char



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.

Online Schema Evolution 3

- Unbundling Partitioned Table Attributes
 - ▶ Partition without an index
 - Table controlled partitioning
 - May be able to have one less index
 - ▶ Data Partitioned Secondary Index
 - ▶ Cluster on any index
 - May be able to have more efficient clustering
 - E.g. partition by date, cluster by account
 - ▶ ALTER CLUSTER attribute



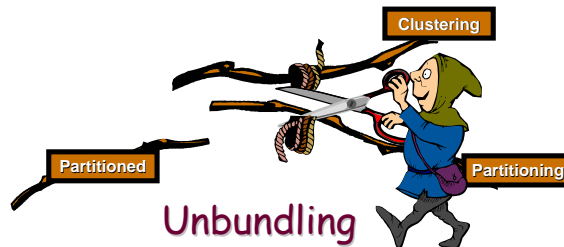
Partitioning and clustering are bundled in current DB2. Some of the time we are required to make a difficult choice. We also want to partition without an index and be able to cluster on any index. These changes will allow us to have one less index and less random IO in some cases.

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

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

Table-controlled partitioning

```
CREATE TABLE CUSTOMER (
  ACCOUNT_NUM
  INTEGER,
  CUST_LAST_NM
  CHAR(30),
  ...
  LAST_ACTIVITY_DT
  DATE,
  STATE_CD
  CHAR(2))
PARTITION BY ( ACCOUNT_NUM
ASC )
( PARTITION 1 ENDING AT (199),
PARTITION 2 ENDING AT (299),
PARTITION 3 ENDING AT (399),
PARTITION 4 ENDING AT (499) );
```



No indexes are required for partitioning!!

TB	101	201	301	401
	102	202	302	402
	103	203	303	403
	104	204	304	404
	105	205	305	405
	106	206	306	406

Partitioned table

Before V8, only **index-controlled partitioning** was supported. With index-controlled partitioning, the partition boundaries are specified on the CREATE INDEX statement, when creating a partitioning index on the table. This results in a table that is unusable or incomplete until the partitioning index is created.

DB2 V8 introduces **table-controlled partitioning**.

That is, you specify the partition boundaries and number of partitions in your CREATE TABLE statement.

PARTITION BY is the keyword which initiates table-controlled partitioning. Once the table definition has been completed successfully, the table is ready to use and you can insert your data. Indexes are no longer required for partitioned table spaces. If you attempt to create an index on this table using the VALUES keyword, an error is returned.

Both types of partitioning are supported with DB2 V8, but several new enhancements are only supported when using table-controlled partitioning.

You are encouraged to convert partitioned tables to use table-controlled partitioning. A non-disruptive method for doing this involves:
Creating a Data Partitioned Secondary Index (CREATE INDEX PARTITIONED) with DEFER YES. The CREATE INDEX PARTITIONED syntax triggers conversion to table-controlled partitioning, and the index is left in RBDP. There is no loss of availability.

Dropping the newly created (empty) index.

Alternatively, you can run the ALTER INDEX NOT CLUSTER statement on the partitioning index. That converts the partitioned table space to table-controlled partitioning, and ALTER it back to CLUSTER immediately after.

Version 8 classification of indexes

- An index may / may not be correlated with the partitioning columns of the table
 - Partitioning index (PI)
 - Secondary index
- An index may / may not be physically partitioned
 - Partitioned
 - Non-partitioned
- Clustering index:
 - Any index may be the clustering index
 - The clustering index can be non-unique

In DB2 V7 you only have index-controlled partitioning at your disposal to create a partitioned table. When using index-controlled partitioning, concepts such as `partitioned`, `partitioning` and `clustering` are intertwined. This occurs because the index that defines the key ranges for the different partitions is the partitioning index, it is necessarily partitioned (made up of different physical partitions), and it is also the clustering index.

Any other index on an index-controlled partitioned table is called a secondary index, and it is non-partitioned, that is, not split up into multiple partitions (although you can define pieces for those indexes to improve I/O performance).

Indexes on table-controlled partitioned tables can be classified as follows:

Based on whether or not an index is physically partitioned:

Partitioned index - The index is made up of multiple physical partitions (not just index pieces)

Non-partitioned index - The index is a single physical data set (or multiple pieces)

Based on whether or not the columns in the index correlate with the partitioning columns of the table (the partitioning columns are those specified in the `PARTITION BY` clause):

Partitioning index - The columns in the index are the same as (and have the same collating sequence), or start with the column(s) in the `PARTITION BY` clause of the `CREATE TABLE` statement.

Secondary index - Any index where the columns do not coincide with the partitioning columns of the table.

Based on whether or not the index determines the clustering of the data. Please note that when using table-controlled partitioning:

Clustering index - The index determines the order in which the rows are stored in the partitioned table.

Non-clustering index - The index does not determine the data order in the partitioned table.

The next few foils show some example to clarify the above terminology.

Partitioning indexes

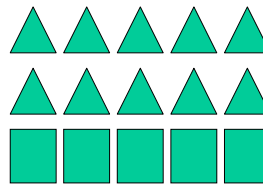
- A partitioning index
 - Has the same leftmost columns as the columns which partition the table
 - These columns have the same collating sequence (ASC / DESC)

```
CREATE TABLE CUSTOMER (  
    ACCOUNT_NUM                INTEGER,  
    LAST_ACTIVITY_DT           CHAR(3),  
    CCODE                      CHAR(2),  
    ...  
)  
PARTITION BY (ACCOUNT_NUM ASC)  
...  
CREATE ... INDEX part_ix_1 ON CUSTOMER (  
    ACCOUNT_NUM ASC)
```

As you can see in the example above, the table partitioning columns of table CUSTOMER is ACCOUNT_NUM. Therefore, every index on this table, whose left most column is ACCOUNT_NUM is considered to be partitioning.

Data Partitioned Secondary Indexes (DPSIs)

- Benefits include partition independence:
 - ▶ More efficient utility processing, no BUILD2
 - ▶ Higher availability
 - ▶ Streamline partition level operations
 - ▶ Potential for lower data sharing overhead
- Potential impact to query performance
 - ▶ Partition key is not specified
 - ▶ Many partitions to search
 - ▶ Not allowed for unique index



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.

Data Partitioned Secondary Indexes - Benefits

- Online REORG
 - No BUILD2 phase with DPSIs
- LOAD PART jobs
 - No shared pages with DPSIs
 - Removes page contention, and GBP flooding issues
 - More efficient "append" strategy can be used
- Easier partition level operations
 - Rolling in/out new partitions much easier, more efficient with DPSIs
- Media failure
 - Partition level recovery
- Data sharing overhead
 - Member <-> partition affinity routing is effective for DPSIs
- Allows query parallelism

Data partitioned secondary indexes have been 'invented' to be able to overcome some of the negative impacts former (non-partitioned) secondary indexes have. Therefore, if you use a DPSI instead of a NPSI (non-partitioned partitioning indexes) for your partitioned tables, you will encounter the following benefits: Using DPSIs eliminates the need for a BUILD2 phase. There is no BUILD2 phase processing for DPSIs. Because keys for a given data partition reside in a single DPSI partition, a simple substitution of the index partition newly built by REORG for the old partition is all that is needed. If all indexes on a table are partitioned (partitioned PI or DPSIs), the BUILD2 phase of REORG is eliminated.

Using DPSIs also eliminates LOAD PART job contention.. There is no contention between LOAD PART jobs during DPSI processing. This is because there are no shared pages between partitions on which to contend. Thus if all indexes on a table are partitioned, index page contention is eliminated.

Also note that during parallel LOAD PART job execution, each LOAD job inserts (loads) DPSI keys into a separate index structure, in key order. This allows the insertion logic to follow an efficient append strategy.

DPSIs also improve the recovery characteristics of your system. DPSIs can be copied and recovered at the partition level. Individual partitions can be rebuilt in parallel to achieve a fast rebuild of the entire index.

However, beware of queries that do not have partitioning column predicates. They have to scan all parts of the DPSI

DPSI -Design considerations

- DPSIs are a design choice and may not fit ALL situations
- Query predicates that solely reference columns in DPSIs must probe each partition - this will degrade query performance
 - `SELECT * FROM table WHERE DPSI_col = 'X'`
 - `SELECT * FROM table WHERE DPSI_col BETWEEN 'X' AND 'Y'`
`ORDER BY DSPI_col`
- If predicates also restricts the query to a single partition, query performance will benefit
 - "Partition pruning"
 - Bind time pruning
 - `P_COL = literal`
 - Runtime pruning with `REOPT(VARS)`
 - `P_COL = :hv , ? , special register`
- **Recommendation:** Code predicates explicitly in your applications to allow for partition pruning when a DPSI exists

Data Partitioned Secondary Indexes (DPSIs)

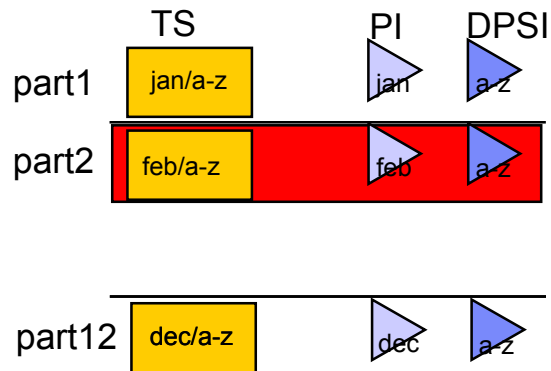
- DPSI = physically partitioned secondary index
 - ▶ Number of parts in DPSI = Number of parts in table
 - ▶ Keys in part 'n' of DPSI refer only to rows in part 'n' of table
- 3 kinds of indexes now:
 - ▶ Partitioning Index (PI).
 - As today, except optional in V8 and may or may not be partitioned
 - ▶ New Data Partitioned Secondary Index (DPSI).
 - ▶ Non Partitioned Secondary Index (NPSI) As today's NPI



For a DPSI, the same number of partitions are in the data and in the index. This is a new type of index that does not fit into the old categories. It is a secondary index, but it is partitioned. The partitioning is according to the table. The partitioning data might be columns that are not part of this index.

So the index partitions are partitioned just like the data. This provides many new options for combinations of partitioning and indexes.

Sample New Style Table - Table-based partitioning

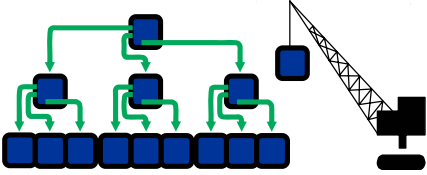


- Partition data by month (PI is optional!)
- Clustering by id or name (DPSI clustering)
- Ideal for Online Reorg with fast switch, no BUILD2

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 is 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.

Index: DB2 UDB for z/OS

Index Improvements

- 
- Variable length index keys
 - Index-only access for varchar data
 - Maximum index key 2000 bytes
 - Predicates indexable for unlike types
 - Backward Index Scan
 - Partitioning separate from clustering
 - Data-partitioned secondary indexes (DPSI)
 - Create index online during select, insert
 - Add column to index

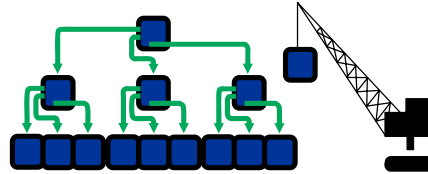
DB2 V8 provides many new opportunities for improving index processing, rebuilding the architecture for indexes.

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.

Indexable Predicates: DB2 for z/OS



- **Predicates indexable for unlike types**
 - **Column is decimal; Host variable is float**
 - **Column char(3); Literal or host variable char(4)**
 - **Can be used with transitive closure**
 - **Some restrictions still for stage 1, indexable**

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.

4096 Partitions

- Maximum number of parts raised from 254 to 4096
 - ▶ Table spaces and indexes
 - ▶ Table space use DSSIZE to go beyond 254 parts
- ALTER TS ADD PART adds partitions to the end
- Maximum table size remains 16 TB for 4 KB pages
- Data set naming convention
 - ▶ 'Axxx' - partitions 1-999 'Bxxx' - partitions 1000-1999
 - ▶ 'Cxxx' - partitions 2000-2999 'Dxxx' - partitions 3000-3999
 - ▶ 'Exxx' - partitions 4000-4096
- Maximum number of parts allowed depends upon page size & DSSIZE
 - 4K page size, DSSIZE=1GB => 4096 parts, 4 TB max table
 - 4K page size, DSSIZE=64GB => 256 parts, 16 TB max table

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.

Online Changeable Subsystem Parameters

DSN6SPRM

Parameter	Panel	Panel Field
CHGDC	DSNTIPO	DPROP Support
EDPROP	DSNTIPO	DPROP Support
SYSADM	DSNTIPP	System Admin 1
SYSADM2	DSNTIPP	System Admin 2
SYSOPR1	DSNTIPP	System Operator 1
SYSOPR2	DSNTIPP	System Operator 2
CACHEDYN	DSNTIP4	Cache Dynamic SQL
SRTPOOL	DSNTIPC	Sort Pool Size
XLKUPDLT	DSNTIPI	X Lock for Searched U/D
MAXKEEPD	DSNTIPE	Max Kept Dyn Stmts
PARTKEYU	DSNTIP4	Update Part Key Cols

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.

Online Changeable Subsystem Parameters (continued)

DSN6FAC

Parameter	Panel	Panel Field
RESYNC	DSNTIPR	Resync Interval
POOLINAC	DSNTIP5	Pool Thread Timeout
TCPKPALV	DSNTIP5	TCP/IP Keepalive
IDHTOIN	DSNTIPR	Idle Thread Timeout
TCPALVER	DSNTIP5	TCP/IP Already Verified
MAXTYPE1	DSNTIPR	Max Type 1 Inactive
EXTRAREQ	DSNTIP5	Extra Blocks Req
EXTRASRV	DSNTIP5	Extra Blocks Srv

DSN6GRP

Parameter	Panel	Panel Field
IMMEDWRI	DSNTIP4	Immediate Write

New parameters added in Version 8 can be changed online

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.

System Level Log Point Recovery

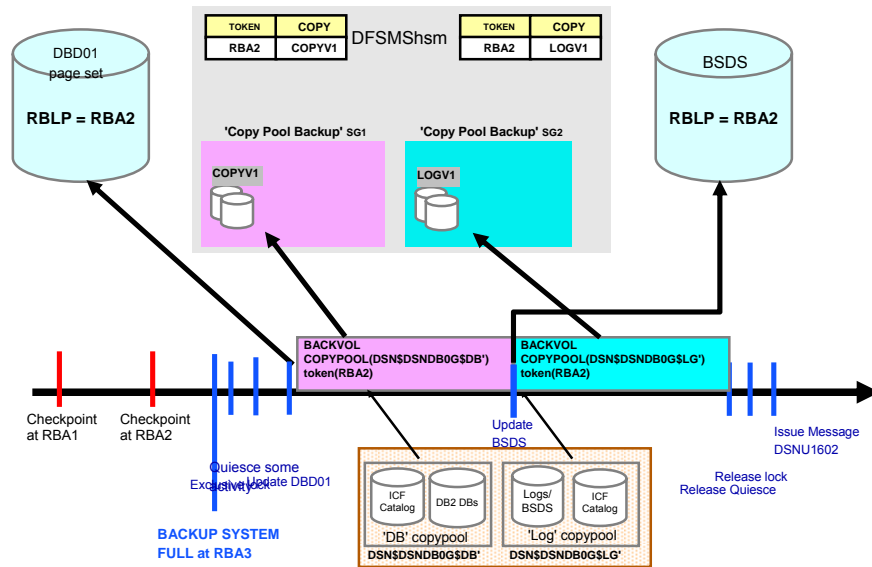
- Easier, more flexible, less disruptive, faster recovery
- Handle large numbers of table spaces & indexes
- Two new utilities introduced:
 - ▶ **BACKUP SYSTEM**: Fast volume-level backups
 - DB2 databases and logs
 - Data sharing group scope
 - z/OS V1R5, DFSMSHsm Fast Replicate, DFSMSdss, & FlashCopy required (FlashCopy Version 2 recommended)
 - ▶ **RESTORE SYSTEM**
 - To an arbitrary point on the log
 - Handles creates, drops, LOG NO events

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.

BACKUP SYSTEM operation



Backup system will do the following:

Obtain an exclusive lock (new lock type) to ensure that only one BACKUP SYSTEM process is running at a time.

Quiesce some system activities but much less disruptive than -set log suspend

Update the DBD01 header page and write the page to DASD.

Take full volume DASD-to-DASD copies of the 'DB' COPYPOOL

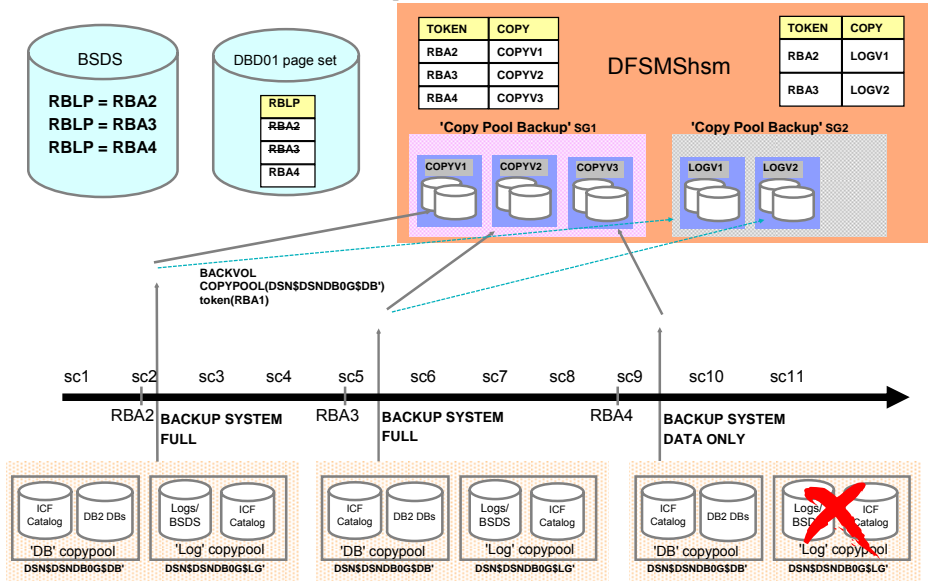
Update the BSDS with the system copy information

If "full" system backups were requested, take full volume copies of the 'LG' COPYPOOL.

Resume all system activities that were quiesced above.

Release the lock.

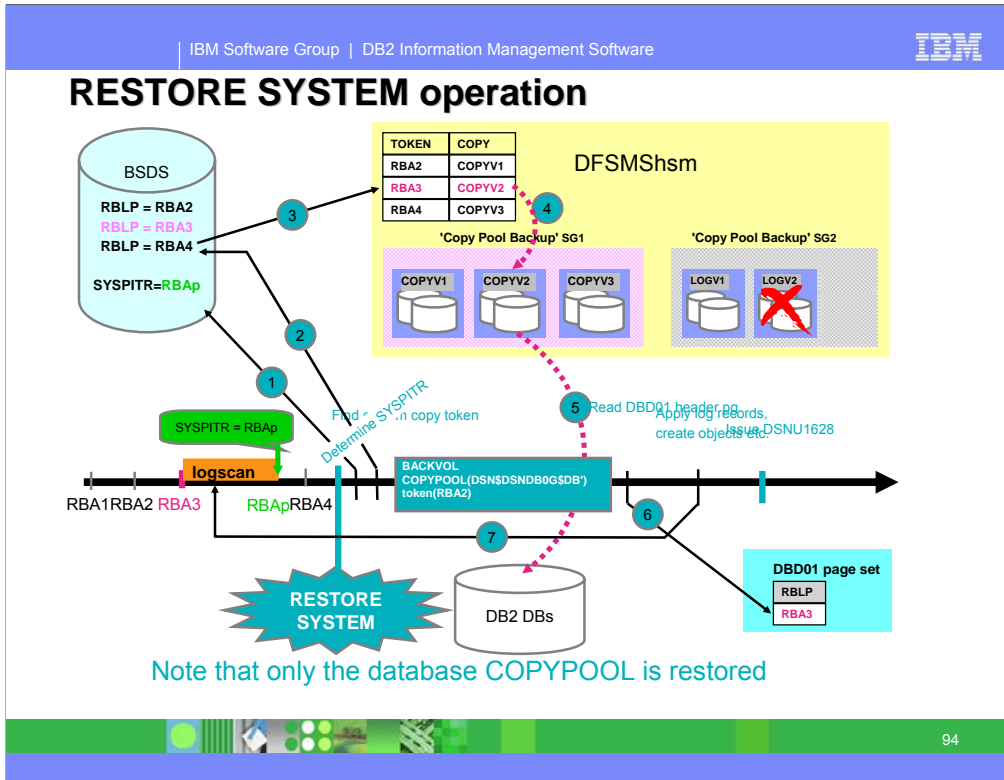
BACKUP SYSTEM operation



As mentioned before, you can use the new BACKUP SYSTEM utility to back up an entire DB2 subsystem or an entire DB2 data sharing group with a single command. The BACKUP SYSTEM utility has two options that you can specify:

FULL: In this case, the backup contains both the logs and databases. This is referred to as a full system backup. When taking a FULL backup, the copies of the log are always taken after the database copies.

DATA ONLY: This type of copy only contains the database data. Notice that a BACKUP SYSTEM DATA ONLY does NOT copy the log data. Such a copy is referred to as a data-only system backup.



You can use RESTORE SYSTEM to restore a system to an arbitrary point in time. It is sufficient to have DATA ONLY backups (provided the logs are available to roll forward from).

To be able to run the RESTORE SYSTEM utility, the system has to be in System Recover Pending mode. A DB2 system goes into System Recover Pending mode after a conditional restart is performed, using a special type of conditional restart record called a PITR conditional restart control record (CRCR).

After the conditional restart completes the system enters into System Recover Pending mode. This means that:

Only the RESTORE SYSTEM utility is allowed to execute.

The DB2 data remains unavailable until the RESTORE SYSTEM utility has completed.

DB2 has to be recycled to reset the System Recover Pending mode.

The visual above shows roughly the steps which DB2 performs if you run the RESTORE SYSTEM utility (without the LOGONLY option):

The PITR CRCR determines the log truncation point

Find the BACKUP SYSTEM information in the BSDS that immediately prior to the log truncation point.

Restore the volume copies for the 'DB' pool

The DBD01 header page information is used to determine the starting point for the log scan.

Perform a 'modified' forward log scan until the log truncation point is reached

Data sharing Enhancements

- CF batching
- Reducing XES contention
- IMMEDIATE WRITE (PH2) replaced by (PH1)
- More efficient index page split
- Restart light enhancements
- DIS GBPOOL changes



There are many changes for data sharing in this new version, improving performance, availability and usability. Being able to batch requests to the coupling facility improves performance. The new locking protocol reduces contention and improves performance. The more efficient page split helps in situations where there are large numbers of index page splits. The improvements in restart light allow indoubt units of work to be resolved.

Data Sharing Enhancements

- **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 V1R4, CFLEVEL=12
- **Reduced global contention for table space L-locks**
 - ▶ Reduced XES-level contention across members
 - ▶ Improved data sharing performance, especially for OLTP
 - ▶ RELEASE(DEALLOCATE) may not be needed
 - ▶ Change to protocol 2 requires group quiesce

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

Data Sharing Enhancements ...

- Changed pages written to GBP at Phase1 instead of Phase2
 - ▶ Transactions invoking other transactions at syncpoint for same data
 - ▶ Unusual "record not found" from another member
 - ▶ Easier to manage
 - ▶ Equivalent performance
- More efficient index split processing for data sharing

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 IMMEDIATE PH1 Zparm or Bind option if needed

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

Automatic Storage Management

- No need to specify space parameters
- Start with small disk space
 - Start with small secondary extents
 - Larger secondary extents as table grows
- Improves DBA productivity
- Avoids application outage
- Reduces need to reorganize

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 Database: DB2 for z/OS

- Add partitions
- Separate partitioning & clustering
- Index enhancements
- 4096 Partitions
- Rotate partitions
- Extend columns
- Optimization improvements
- Memory and scale increased



99

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.

Utility Improvements

- ▶ **Easier restart** (V7 PQ72337)
- ▶ **Schema Evolution**
 - **REBALANCE** partitions
- ▶ **On-line REORG Enhancements**
 - **DISCARD**
 - **Avoid BUILD2 with DPSI**
 - **REORG DB2 catalog SHRLEVEL REFERENCE**
- ▶ **LOAD & UNLOAD delimited input & output**
- ▶ **SCOPE PENDING**
- ▶ **RUNSTATS non-uniform statistics on non-index columns**
- ▶ **System-level log point backup and recovery**
- ▶ **Improved defaults for performance**
- ▶ **LOBs in cross load, LOAD, UNLOAD, online check index**



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 our 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. The DB2 catalog tables can all be reorganized in SHRLEVEL REFERENCE or read only mode.

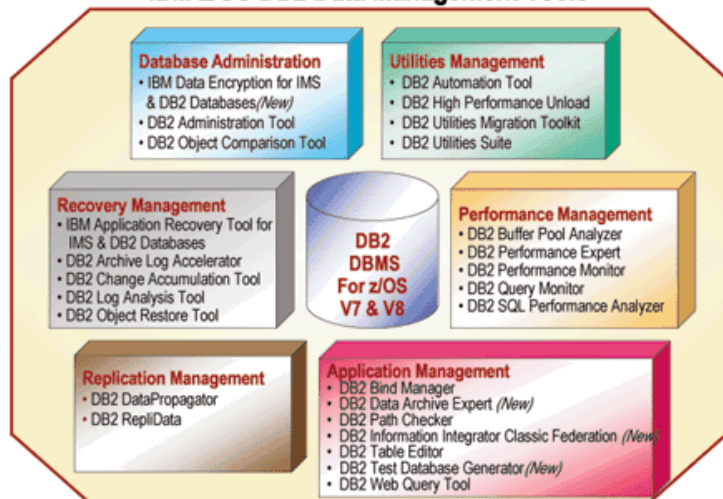
Delimited files can be used as input to LOAD or output from UNLOAD.

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>

DB2 Tools Product Portfolio

IBM z/OS DB2 Data Management Tools

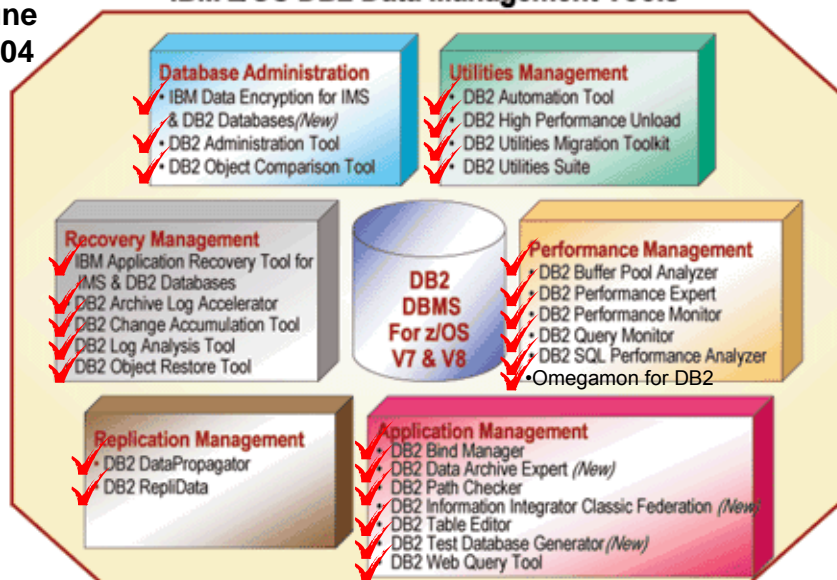


As you have seen, the depth and breadth of the DB2 Data Management Tools has grown, even as the tools have become more mature. Changing all of these tools to be ready for the changes in DB2 UDB for z/OS Version 8 was a challenge. Note that the DB2 tools portfolio has been growing rapidly since we started in 2000 with 8 products. We now have 24 products for DB2 for z/OS. We also have comparable and compatible products for DB2 on Linux, Unix and Windows. The products are constantly improving in feature and function thru continuous investment, improvement and innovation.

ibm.com/software/data/db2imstools/

✓ V8 Ready
June
2004

IBM z/OS DB2 Data Management Tools



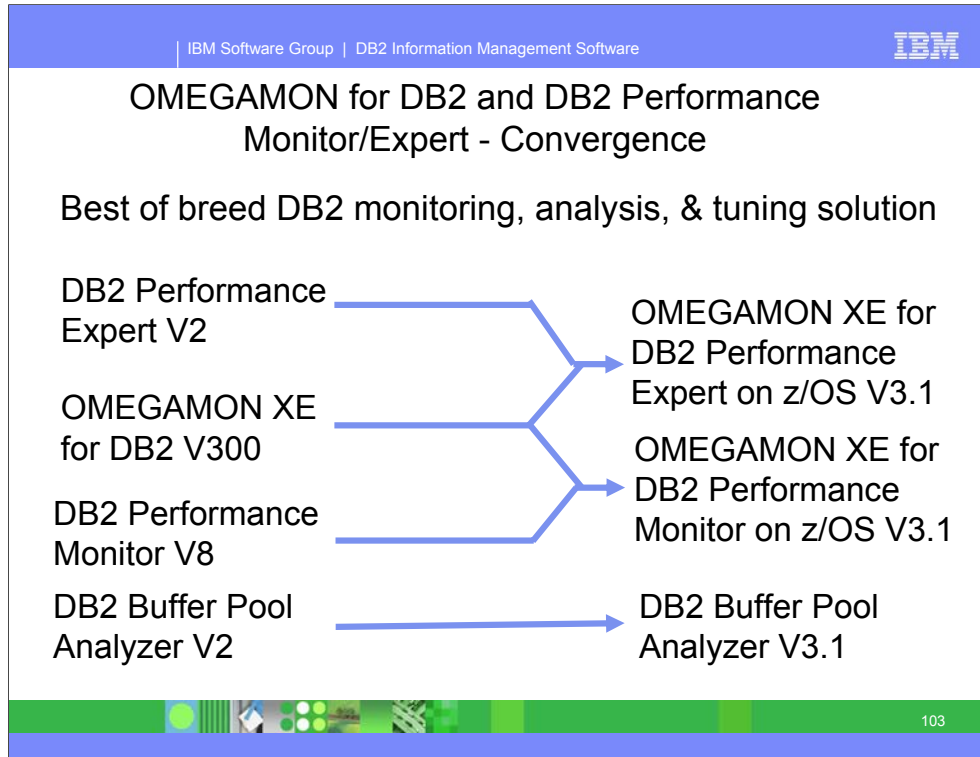
102

The 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 they are also ready for V8.



Taking the best from both products means to combine unique functions with the existing ones of the other offering, to replace identical functions with one of both solutions and to add / implement missing functions into existing components. The target, of course, is to keep as much functions as possible either coming from OMEGAMON or coming from DB2 PM/PE. We also considered the actuality (newest DB2 version support as well the effort) to be spent to provide the functions.

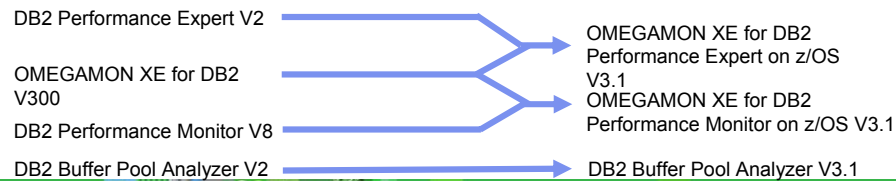
For simplification you might say, the new product is the DB2 PM/PE base services and functions (reporting, PWH, and DB2 API access) with the infrastructure, the integration capability and end user interface of OMEGAMON.

Although we prefer to sell OMEGAMON XE for DB2 PE, we still see a need to offer performance monitoring with and without specific buffer pool analysis, buffer pool expert advices, and buffer pool simulation because of market needs. Therefore we will continue to offer the OMEGAMON XE for DB2 PE and the OMEGAMON XE for DB2 PM as separate offerings. However, our intention is to add specific expert functions only to the OMEGAMON XE for DB2 PE and to support only base monitoring support to OMEGAMON XE for DB2 PM too, i.e. new DB2 functions and versions will be added to the PM and PE, but new specific expert functions will be added to PE only.

For those customers who do not have OMEGAMON XE for DB2 PM on z/OS we will continue to offer the DB2 Buffer Pool Analyzer as a separate stand-alone offering.

OMEGAMON for DB2 and DB2 Performance Monitor/Expert - Convergence Objectives

- Best of breed DB2 monitoring, analysis, and tuning solution
 - Combine best function from products to build converged product
 - OMEGAMON family - integrated cross zSeries monitoring and system management
 - Leverage OMEGAMON real time monitoring and enterprise-wide systems management and DB2 PM/PE's legendary reporting and warehouse functions
 - Merge performance data collection into single engine (server)
 - Reduce footprint and system resources



Taking the best from both products means to combine unique functions with the existing ones of the other offering, to replace identical functions with one of both solutions and to add / implement missing functions into existing components. The target, of course, is to keep as much functions as possible either coming from OMEGAMON or coming from DB2 PM/PE. We also considered the actuality (newest DB2 version support as well the effort) to be spent to provide the functions.

For simplification you might say, the new product is the DB2 PM/PE base services and functions (reporting, PWH, and DB2 API access) with the infrastructure, the integration capability and end user interface of OMEGAMON.

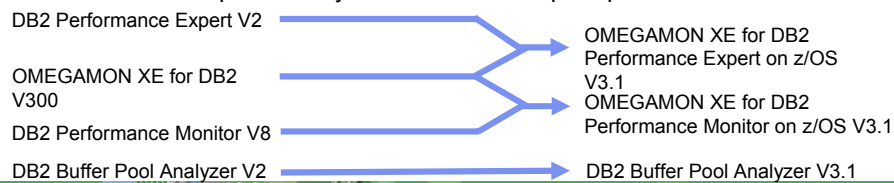
Although we prefer to sell OMEGAMON XE for DB2 PE, we still see a need to offer performance monitoring with and without specific buffer pool analysis, buffer pool expert advices, and buffer pool simulation because of market needs. Therefore we will continue to offer the OMEGAMON XE for DB2 PE and the OMEGAMON XE for DB2 PM as separate offerings. However, our intention is to add specific expert functions only to the OMEGAMON XE for DB2 PE and to support only base monitoring support to OMEGAMON XE for DB2 PM too, i.e. new DB2 functions and versions will be added to the PM and PE, but new specific expert functions will be added to PE only.

For those customers who do not have OMEGAMON XE for DB2 PM on z/OS we will continue to offer the DB2 Buffer Pool Analyzer as a separate stand-alone offering.

OMEGAMON for DB2 and DB2 Performance

Monitor/Expert - Convergence Objectives

- Offer the best of breed DB2 monitoring, analysis, and tuning solution
 - Combine the best functionality from both product offerings to build a new converged product
 - As a member of the OMEGAMON family - provide an integrated cross zSeries monitoring and system management offering
 - Satisfy the requirements of both customer sets
 - Leverage OMEGAMON's legendary realtime monitoring and enterprise-wide systems management and DB2 PM/PE's legendary reporting and warehouse functions
 - Long term support for OMEGAMON's XE and Classic VTAM interfaces
 - Merge the performance data collection logic into a single engine (server)
 - Reduce the footprint and system resources of prior products



Taking the best from both products means to combine unique functions with the existing ones of the other offering, to replace identical functions with one of both solutions and to add / implement missing functions into existing components. The target, of course, is to keep as much functions as possible either coming from OMEGAMON or coming from DB2 PM/PE. We also considered the actuality (newest DB2 version support as well the effort) to be spent to provide the functions.

For simplification you might say, the new product is the DB2 PM/PE base services and functions (reporting, PWH, and DB2 API access) with the infrastructure, the integration capability and end user interface of OMEGAMON.

Although we prefer to sell OMEGAMON XE for DB2 PE, we still see a need to offer performance monitoring with and without specific buffer pool analysis, buffer pool expert advices, and buffer pool simulation because of market needs. Therefore we will continue to offer the OMEGAMON XE for DB2 PE and the OMEGAMON XE for DB2 PM as separate offerings. However, our intention is to add specific expert functions only to the OMEGAMON XE for DB2 PE and to support only base monitoring support to OMEGAMON XE for DB2 PM too, i.e. new DB2 functions and versions will be added to the PM and PE, but new specific expert functions will be added to PE only.

For those customers who do not have OMEGAMON XE for DB2 PM on z/OS we will continue to offer the DB2 Buffer Pool Analyzer as a separate stand-alone offering.

Product Enhancements

- What OMEGAMON customers get with the converged product
 - Usage of DB2 IFI API
 - Provide more consistency and better quality of data (less dependence on control block changes)
 - World Class Batch Reporting – In-depth problem analysis
 - With the recent extensions of
 - Package Level Accounting
 - Locking suspension – new output format for direct spreadsheet usage
 - Extended SQL Activity report with input host variables
 - Additional predefined report layouts
 - Performance Warehouse
 - Expert analysis (ROT and SQL Performance queries)
 - DB2 Connect Monitoring
 - Snapshot history for online adhoc problem analysis
 - Notification of exceptional events – deadlocks, timeouts, coupling facility rebuild, data set overflow

Product Enhancements

- What DB2 PE/PM customers get with converged product
 - Integrated cross zSeries monitoring (enterprise wide)
 - Near term history
 - Object Analysis
 - WLM enclave information for stored procedures
 - Smaller footprint (fewer address spaces)
 - VTAM and Web Client
 - Monitor DB2 messages and master console
 - EDM pool (DBD,SK, CT etc section) content display
 - 18 new Customer requirements

Product Functionality

- All SPE and New Functions PTFs of the former OMEGAMON and DB2 PM/PE releases are also included
 - New & extended workspaces at XE of information available on Classic
 - Thread monitoring
 - Distributed Data Facility (DDF)
 - Enclave
 - CICS Transplex capability in OMEGAMON XE for DB2 enhancement provides correlation ID to correlate a DB2 thread with a CICS transaction
 - Correlation with CICS requires OMEGAMON DE and current release of OMEGAMON XE for CICS
 - Extended DBM1 virtual storage reporting in OLM and reports
 - Handling of DB2 V8 threads
 - SAP & PeopleSoft reporting
 - EDM pool (DBD,SK, CT etc section) content display

Product Enhancements

The screenshot displays the Thread Activity tool interface. On the left, a tree view shows the navigation path: Enterprise > OS/390 Systems > PM06 > DB2 > D761-PM06-DB2 > Thread Activity. The 'Thread Activity' and 'DB2 Connect Server' nodes are circled in red. The main area contains two bar charts and a table.

Top Ten In-DB2-CPU Time Threads

Elapsed Time	Plan	Package Name	CPU Rate	Thread Status	In DB2 CPU Time	In DB2 Time	Get Page	Updates	Commits	Authorization	Job Name	Interval Start
01:41:20.000	ADB	ADBMAN	0.0	SWAPPED-OUT	00:00:00.194	01:41:20.000	10824	0	13	MON	MON	07/27/05 15:22:05
01:48:20.000	DISTSERV		0.0	WAIT-REMREQ	00:00:00.134	01:48:20.000	960	0	0	MON	db2bp	07/27/05 15:22:05
01:47:20.000	DISTSERV		0.0	WAIT-REMREQ	00:00:00.078	01:47:20.000	480	0	0	MON	db2bp	07/27/05 15:22:05
03:57:53.200	ADB	ADBMAN	0.0	SWAPPED-OUT	00:00:00.037	03:57:54.680	71	4	4	HELM	HELM	07/27/05 15:22:05
00:23:09.200	KOZPLAN		0.0	NOT-IN-DB2	00:00:00.019	00:23:09.317	0	0	0	DB2PM	D761DM10	07/27/05 15:22:05
00:00:00.000	KOZPLAN		0.0	NOT-IN-DB2	00:00:00.003	00:00:00.020	0	0	0	DB2PM	D761DM10	07/27/05 15:22:05
00:00:00.000	DISTSERV	SYSSH200	0.0	WAIT-REMREQ	00:00:00.002	00:00:00.007	2	0	1	DEA	java.exe	07/27/05 15:22:05

DB2 to CICS Transplex

File Edit View Help

Physical Page: 1 of 2 Page: 1 of 2

Thread Activity

- System Status
- Detailed Thread Exception
- Lock Conflicts
- Subsystem Management
- Log Manager
- Utility Jobs
- EDM Pool
- Buffer Pool Management
- ...
- Thread Detail
- Thread Activity by Package
- Distributed Allied Thread Summary
- Distributed Database Access Thread Summary
- All Threads Connected to DB2
- Enclave Thread Summary
- CICS Transplex**
- Link Wizard...
- Link Anchor...

Top Ten In-DB2-CPU Time Threads

This bar chart shows the top ten threads by CPU time. The x-axis represents CPU time from 0.0 to 0.5. The y-axis lists threads: KO2540IF, KO2520HP, and 2540IF.

Top Ten In-DB2 Time Threads

This bar chart shows the top ten threads by DB2 time. The x-axis represents time from 0 to 60000. The y-axis lists threads: KO2540HP, KO2540IF, and TRANSPLX.

Top Ten In-DB2-CPU Time Threads

Thread Status	In DB2 CPU Time	In DB2 Time	Get Page	Updates	Commits	Authorization	Job Name	Interval Start				
00:24:2500	TRANSPLX	CICSUBZ1	0.0	NOT-IN-DB2	00:00:28.780	00:24:41.564	775	192	0	MIHANS	CICSR24L	02/18/05 14:20:4
12:08:45:300	KO2540IF	0.0	NOT-IN-DB2	00:00:20.180	12:08:44.570	0	0	0	0	D2DB020	D2DB020	02/18/05 14:20:4
12:08:35:900	KO2540HP	0.0	NOT-IN-DB2	00:00:06.443	12:08:35.146	0	0	0	0	D2DB020	D2DB020	02/18/05 14:20:4
1 Day	KO2540IF	0.0	NOT-IN-DB2	00:00:03.346	1 Day	0	0	0	0	D2DB058	D2DB058	02/18/05 14:20:4
11:29:28:700	KO2540IF	0.0	NOT-IN-DB2	00:00:00.486	11:29:27.907	0	0	0	0	D2DB020	D2DB020	02/18/05 14:20:4
11:41:20:000	KO2540IF	0.0	NOT-IN-DB2	00:00:00.344	11:41:20.000	0	0	0	0	D2DB011	D2DB011	02/18/05 14:20:4

Ready Hub Time: Wed, 02/16/2005 02:22 PM Server Available. Thread Activity by Plan - localhost - WONGSU

Correlate DB2 Thread to CICS transaction

File Edit View Help

Transaction Processor Utilization

Task Number: 00156

CPU Seconds: 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26

Transaction Analysis

DB2 Correlation Identifier	System ID	CICS Region Name	CICS SYSIDNT	Transaction ID	User ID	Terminal ID	Task Number	Wait Type	Elapsed Time	CPU Time	Program ID	Exceeds MAXR Threshold	P	N
ENTRDB210001	SYS_	CICSR24L	R24L	DB21	CICSR24L	0009	00156	Interval	00:23:40.10	00:00:24.03	CICSDB21	No	n/s	

Ready | Hub Time: Wed, 02/16/2005 02:19 PM | Server Available. | Transaction Analysis - localhost - WONGSU

DB2 Connect monitoring in XE

The screenshot displays the DB2 Connect Server monitoring interface. On the left is a tree view of the server hierarchy. The main area is divided into two sections: a bar chart titled 'Server Key Events' and a table titled 'DB2 Connect Server'.

Server Key Events Bar Chart:

Event Type	BHJINTEL	B555Z5ZP
AGENTS WAITING FOR TOKEN	~10	~10
STOLEN AGENTS	~170	~10
MAX AGENT OVERFLOWS	~40	~10

DB2 Connect Server Table:

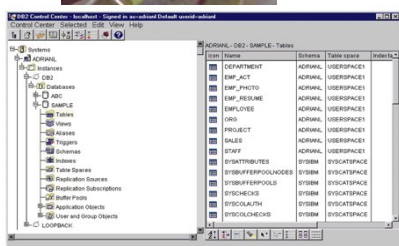
Name	Ip Address	Status	Node Name	Node Number	Server Product Version ID	Server Instance Name	Server Version	Time Zone Displacement	Gateway Snapshot Time
BHJINTEL	9.152.196.89	ACTIVE	NIP	0	SQL08015	DB2	5	2	1041209101220123
B555Z5ZP	9.152.196.89	ACTIVE	NIP	0	SQL08015	DB2	5	2	1041209101220123

DB2 Family



Other DB2 Family Improvements

- Control Center
- Development Center
- Warehouse Manager
- Replication & Replication Center
- DB2 Connect
- Common Client
- Java Universal Driver
- LOAD delimited input



A wide range of other improvements are being made in components that work across the DB2 family. These graphical interfaces, wizards and clients improve consistency across the family and usability.

ODBC Improvements for USS

SQLConnect user and password support

If provided in V7, just checked if syntactically correct

ODBC Unicode support

CURRENTAPPENSCH in .INI file can be set to Unicode

New so called wide APIs available

Non-wide functions changed to accept UTF-8 string

Cursor extensions allows ODBC application to

Specify whether the server should release read locks when a query is closed

Specify whether the server should close a query implicitly when there are no more rows for a non-scrollable cursor

The values for the userid and password arguments on input to SQLConnect and SQLDriverConnect APIs will be propagated to the target server via the new USER/USING clause for use in authenticating the end user. User authentication will be performed only when a userid and password are both provided on the API call which will result in the following CONNECT statement being executed:


```
CONNECT TO :host_variable USER :host_variable  
USING :host_variable
```

The DB2 UDB for z/OS Version 8 ODBC driver enables ODBC applications to store and manipulate UNICODE data to a DB2 UDB for OS/390 Version 7 or later server.

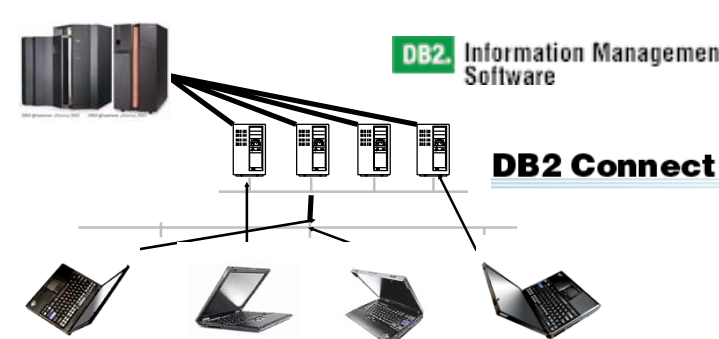
IBM Software Group | DB2 Information Management Software IBM

DB2 Connect Version 8.2

- ✓ Performance & Scalability
- ✓ Improved Application Availability
- ✓ Innovative manageability
- ✓ Programmer productivity



DB2 UDB
V8.2
FP 10, 7A



DB2 Information Management Software

DB2 Connect

115

DB2 Connect improvements are noted in announcements and in the What's New in DB2 UDB Version 8.1? book.

ibm.com/software/data/db2/udb/pdfs/db2q0.pdf

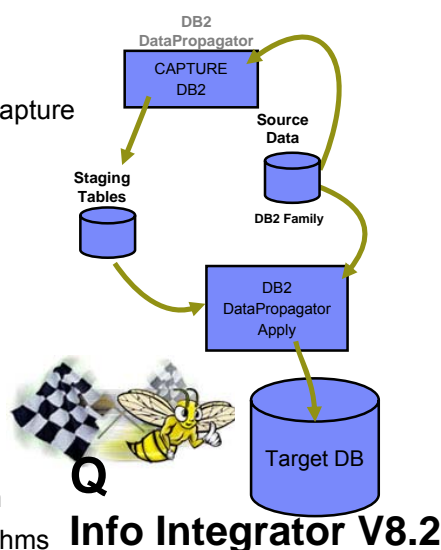
Plan to use V8.1.5 (fixpak 5) or later for DB2 Connect. If you want the DRDA encryption support, then you need V8.2 (fixpak 7A) or later. Fixpak 10 is out and is the recommended level. Version 7.2 end of service was September 2004 and then extended for customers moving to V8. While DB2 V8 works for some customers with DB2 Connect V7.2 (fixpak 10, 11 or 12), most of the improvements only deliver with V8.1.4 or fixpak 4 or later. I recommend adding the fixes from fixpak 5 at a minimum. Also see the web:

ibm.com/software/data/db2/db2connect/

<http://www.ibm.com/software/data/db2/udb/support/downloadv8.html>

DB2 Replication Version 8

- Performance
 - ▶ Multithreaded transaction based Capture
 - ▶ Multiple Capture Schemas
 - ▶ Reduced need for joins
- Usability
 - ▶ New Replication Admin features
 - ▶ Monitoring
 - ▶ More adaptive, Less rules
 - ▶ Long name support
- Reliability
 - ▶ Emphasis on continuous operation
 - ▶ Improved restart and failure algorithms

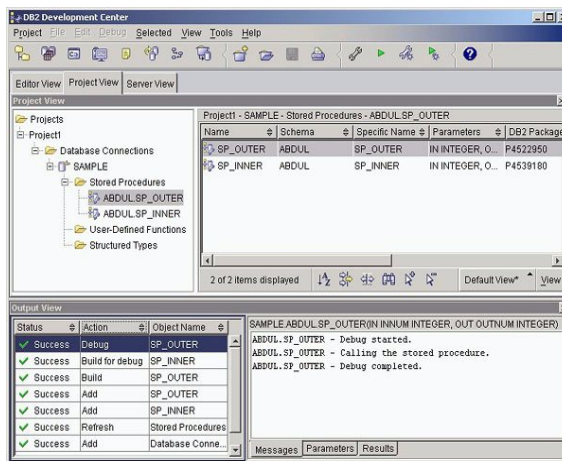


Replication is rearchitected in Version 8. The architecture for data capture has changed to improve performance and availability. See the new Redbook, The Practical Guide to DB2 UDB Data Replication V8, SG24-6828-00. V8 improves performance, throughput, has lower latency, real time alert monitor, and much improved usability and reliability. See the replication improvement and Stinger previews for the outlook on replication using queues and other improvements for more robust, higher volume, lower latency replication. With these enhancements, replication can be used for operational replication and disaster recovery situations.

publib-b.boulder.ibm.com/Redbooks.nsf/RedpieceAbstracts/sq246828.html

Development Center

- ✓ Build, test & deploy Java & SQL Stored Procedures
- ✓ Build SQL Table functions
- ✓ Extract data from XML documents



117

See Development Center articles on DB2 Developer Domain.

www7b.boulder.ibm.com/dmdd/

DB2 Development Center -- The Next-Generation AD Tooling for DB2

www7b.boulder.ibm.com/dmdd/library/techarticle/0207alazzawe/0207alazzawe.html

DB2 Development Add-Ins for Visual Studio 6.0 - The Next-Generation AD Tooling

www7b.boulder.ibm.com/dmdd/library/techarticle/0208alazzawe/0208alazzawe.html

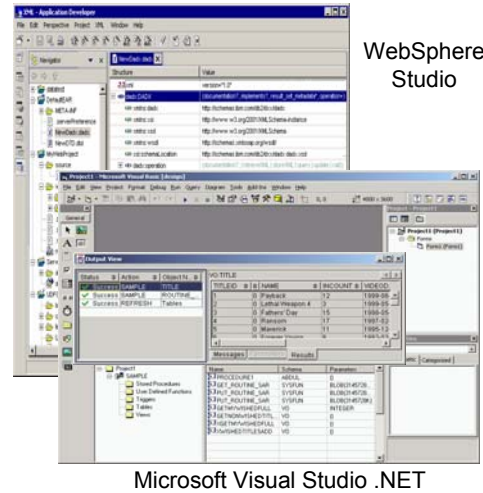
Developing Enterprise Java Applications Using DB2 Version 8

www7b.boulder.ibm.com/dmdd/library/techarticle/0209hutchison/0209hutchison.html

Productivity for Application Development

Developer Productivity for Both Worlds: Java & .NET

- **Flexible Access Style**
 - SQL
 - JDBC, Java Beans, SQLJ
 - ADO, ODBC, OLE
 - Web Services
- **Seamless Integration with Integrated Development Environments**
 - Java...WebSphere Studio
 - .NET...Visual Studio .NET
 - DB2...Development Center
 - Eclipse framework



WebSphere Studio

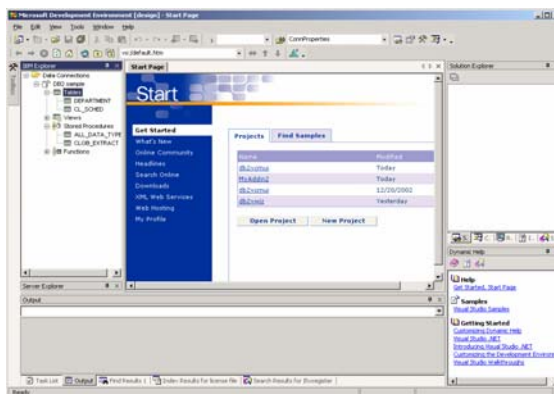
Microsoft Visual Studio .NET

DB2 works with application development to meet a wide variety of application needs, using varied languages, varied vendors, and varied application development styles. Whether you choose Java or Microsoft .NET, regardless of the access style, DB2 database access must be seamless. Programmers will use their integrated development environment, language, access style, and still interact easily with DB2. DB2 provides plug-ins to the IDEs.

Visual Studio.NET Add-ins

What are they?

- IBM Explorer
- IBM Projects (DB2 Database Project)
- DB2 SQL Editor
- DB2 .Net Data Provider tools
- Dynamic Help for DB2 Add-Ins
- DB2 Tools menu Add-in



Technology Previews and Beta programs for .NET features are ongoing. For more details, please refer to the following links:

Technology Previews and Betas

<http://www7b.boulder.ibm.com/dmdd/downloads/dotnetbeta/>

Visual Studio .NET zone on Developer Domain

<http://www7b.boulder.ibm.com/dmdd/zones/vstudio/>

DB2 QMF Version 8

DB2 V8 support

- long names, multirow fetch

QMF for Windows

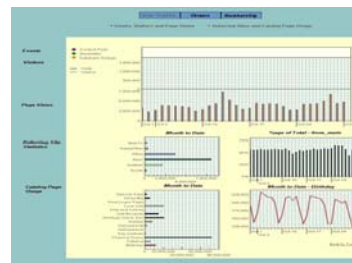
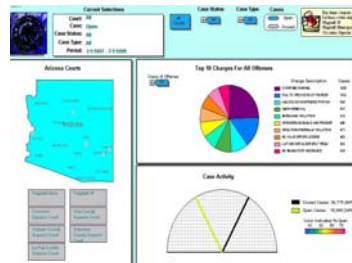
- Visionary Studio

QMF for WebSphere

- Browser access
- Java class & Web services



WebSphere software



What's in QMF Version 8? Easier, faster, and more global on-demand access to enterprise data and analysis through support for DB2 Version 8 plus:

QMF for Windows: - new drag-and-drop data visualization with Visionary Studio (in addition to existing summary reports, charts, and spatial data maps): across / pivot / top formatting, conditional formatting, rich HTML reports, multi-dimensional analysis (OLAP), visual query building interface, new visual database explorer, support for DB2 V8.1 features

QMF for WebSphere: new, greatly enhanced user interface for Web-based data access through an ordinary browser, visual display of customized report libraries, rapid, robust query development: Expression Builder, Java class API & Web services API for custom Web-based applications.

<http://www.ibm.com/software/data/qmf/>


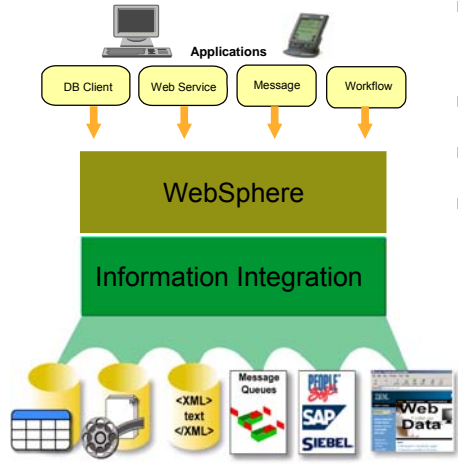
IBM Software Group | DB2 Information Management Software

IBM

DB2 Information Integrator Version 8

Integrating diverse business information across and beyond the enterprise

It's here now
The next generation of data integration software

- Data federation
 - ▶ SQL Content XML
- Data replication
- Data transformation
- Part of a complete business integration solution
 - ▶ Best XML data store = DB2
 - ▶ WebSphere business integration
 - ▶ Open standards

121

Our vision for Information Integration builds upon products we have been developing for a long time, and extends the work into new areas.

Data federation includes read/write access across diverse data and content sources: Database flavor (SQL), Content flavor (CM API) & XML flavor (XML API).

Data placement: Caching and Replication over heterogeneous information

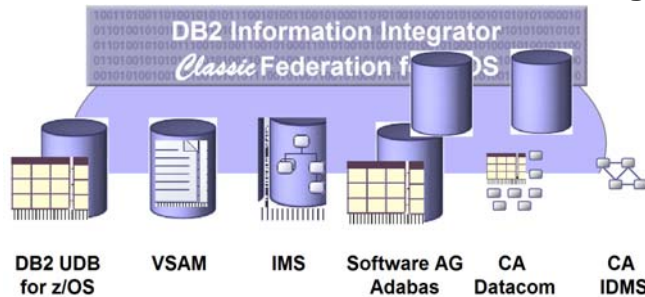
Data transformation: SQL, XML, Advanced search and mining & Metadata management

Part of a complete business integration solution: with the best data store for XML data = DB2, WebSphere business integration, provided by an open platform based on industry standards.

<http://www.ibm.com/software/data/integration/>

DB2 Information Integrator Classic Federation for z/OS

- **Standards-based access**
 - ODBC, JDBC, or Call Level Interface
- **No mainframe programming required**
- **Read & write mainframe data sources using SQL**



122

IICF provides standards-based access via ODBC, JDBC, or Call Level Interface. The code is multi-threaded with native drivers for scalable performance. Being Metadata-driven means No mainframe programming required; Fast installation & configuration; and Ease of maintenance.

Classic Federation works with existing and new:

Mainframe infrastructure

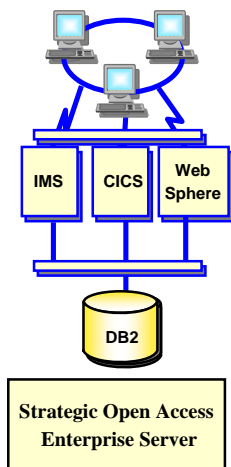
Application infrastructure

Toolsets

You can read from and write to mainframe data sources using SQL. This product can help in situations needing to federate the data or in conversions to DB2. We also have a product for VSAM transparency.

<http://www.ibm.com/software/data/integration/iicf/>

Network Computing Improvements



- **Application Connectivity for JAVA**
- **Improve data sharing member routing**
- **LOCATION aliases at requester & server**
- **Extended function, standards**
- **Reduced processing**
- **Stored Procedure Enhancements**
- **Java Universal Driver**

DB2 V8 also provides many improvements for network computing: security, availability, usability and performance. The improved data sharing member routine is more robust. Having the ability to use multiple names for a server adds flexibility. Increased levels of standards are implemented. Performance improvements will reduce cpu costs substantially. Improved granularity for stored procedures help with managing your work load, using the WLM.

The Java Universal Driver is used across the DB2 family.

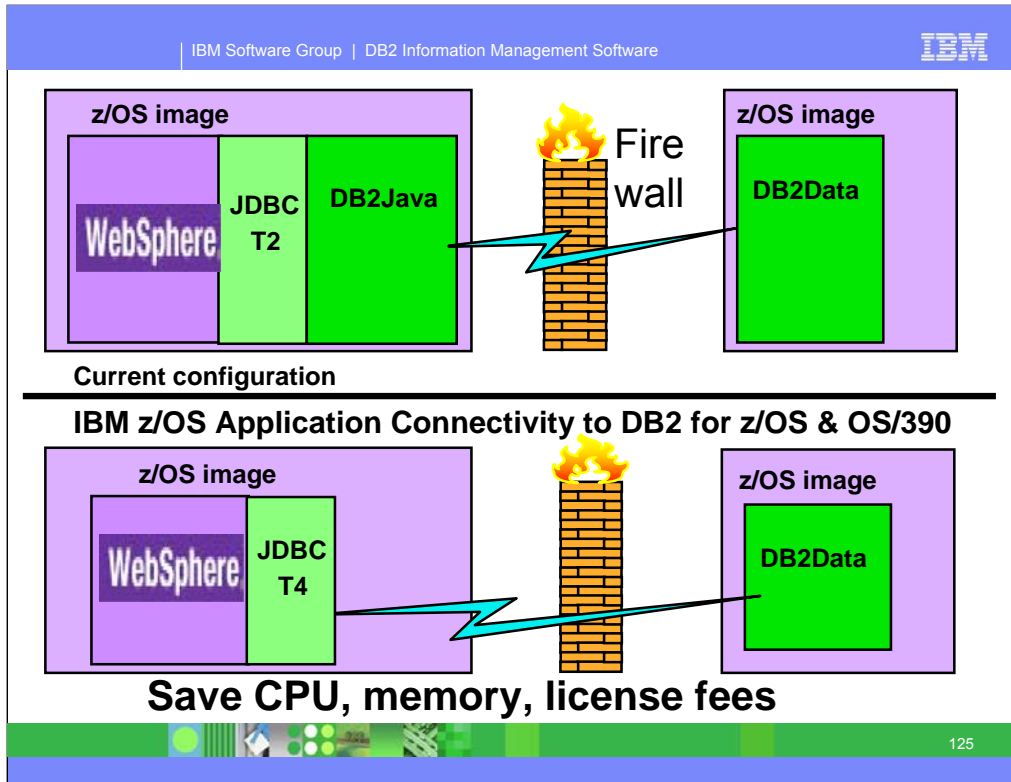
z/OS Application Connectivity to DB2 for z/OS & OS/390

- Pure Java type 4 JDBC driver
- From Java application on z/OS
- To DB2 for z/OS server or Linux, Unix & Windows
- JDBC 2.0 and 3.0, JDK V1.4
- Connect to DB2 for z/OS and WebSphere Application Server for z/OS



WebSphere software

A new IBM z/OS Application Connectivity to DB2 for z/OS and OS/390 delivers performance, scalability and flexibility to JAVA applications, as announced December 16, 2003. The announcement is for DB2 UDB for z/OS and OS/390 Version 7 and for DB2 UDB for z/OS Version 8. Please see the next foil and announcement for more information.



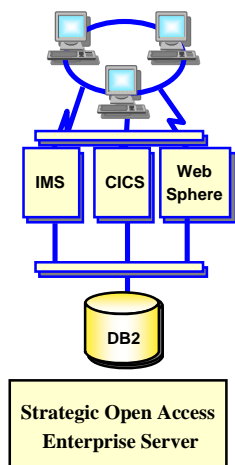
z/OS[®] Application Connectivity to DB2[®] for z/OS and OS/390[®] is a no-charge, optional feature of DB2 Universal Database[®] Server for z/OS V7 and V8. This feature consists of a component known as the DB2 Universal Database Driver for z/OS, Java[™] Edition, a pure Java, type 4 JDBC driver designed to deliver high performance and scalable remote connectivity for Java-based enterprise applications on z/OS to a remote DB2 for z/OS database server. The driver:

- Supports JDBC 2.0 and 3.0 specification and JDK V1.4 to deliver the maximum flexibility and performance required for enterprise applications
- Delivers robust connectivity to the latest DB2 for z/OS and WebSphere[®] Application Server for z/OS
- Provides support for distributed transaction support
- Allows custom Java applications that don't require an application server to run in a remote partition and connect to DB2 z/OS

See the December 16, 2003 announcement or the web for more:

<http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/tips0356.html?Open>

Network Computing Improvements



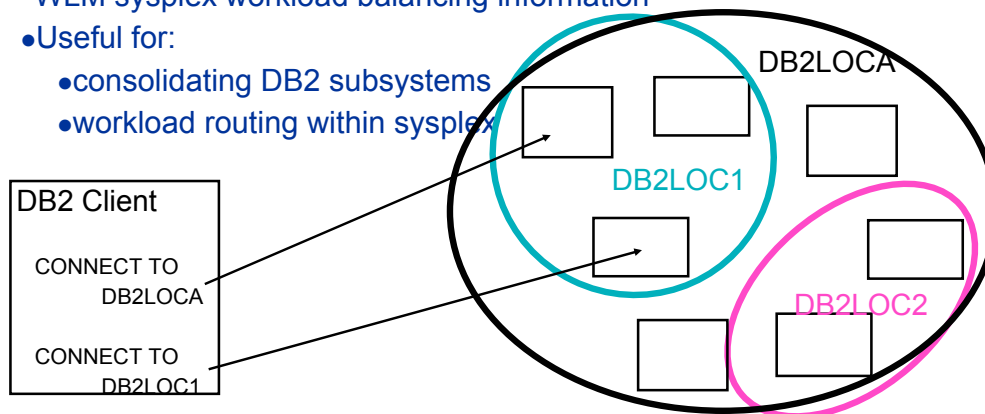
- Encrypt data flowing on the wire
- CDB improvements for data sharing member routing
- LOCATION aliases at requester and server (e.g. CONNECT)
- Extended DESCRIBE for CLI & Java
- LOB streaming
- SQL Cancel
- Auto-close for WITH HOLD cursors
- Reduced block FETCH CPU cost
- Reduced TCP/IP CPU cost
- Stored Procedure Enhancements
- Java Universal Driver

We are also working on many improvements for network computing: security, availability, usability and performance. Performance improvements will reduce cpu costs substantially.



Server LOCATION Aliases

- BSDS for each DB2 member can include multiple LOCATION names for the DB2 server
- Each LOCATION name is registered with WLM separately for WLM sysplex workload balancing information
- Useful for:
 - consolidating DB2 subsystems
 - workload routing within sysplex



127

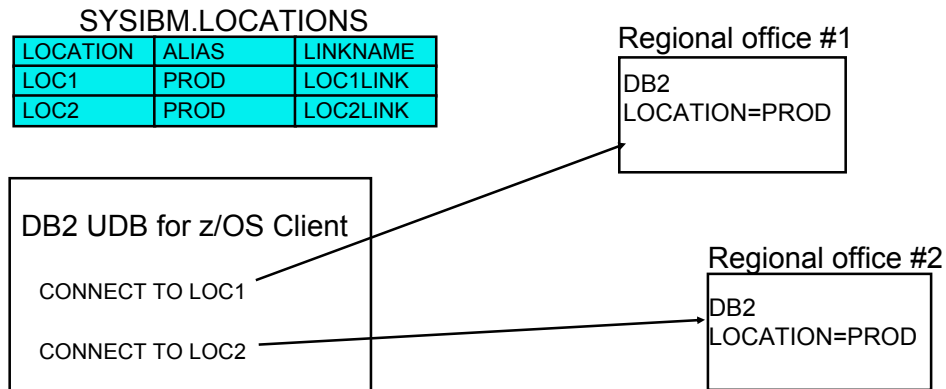
The new server location alias capability allows you to assign multiple LOCATION names to a give DB2 data sharing group, and have those LOCATION names managed separately by WLM for workload routing purposes.

As an example, you might want to have one LOCATION name for handling SQL requests from CICS, and another LOCATION for ad-hoc query users. This gives you the ability to dedicate a subset of the DB2 data sharing members to a specific set of clients (e.g. DB2LOC1 consists of 2 members processing CICS requests, DB2LOC2 consists of 2 members for ad-hoc query, etc.).

This capability is also useful when combining multiple DB2 subsystems into a single data sharing group, since it allows applications to continue to connect to DB2 using whatever DB2 LOCATION name was used previously.

Client LOCATION Aliases

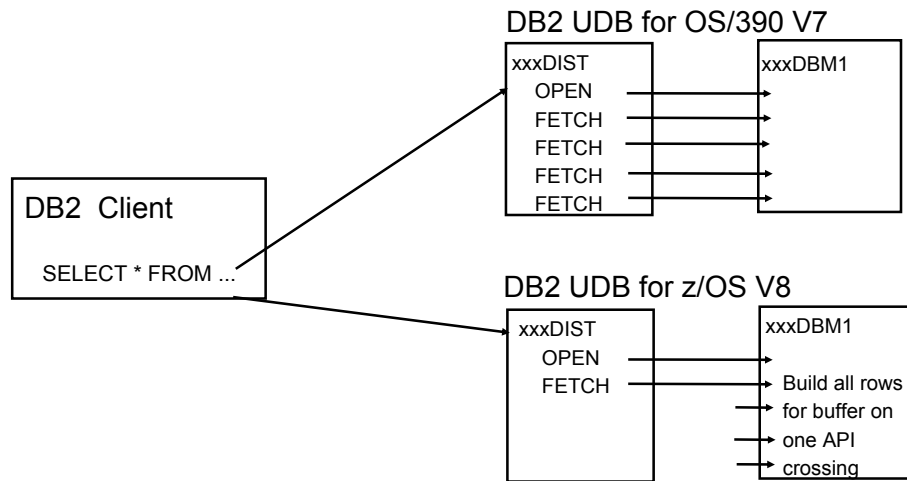
- CDB is enhanced so client LOCATION name does not have to match remote server's LOCATION name



Client LOCATION alias support allows you to refer a remote DB2 server by more than one LOCATION name. This can be particularly useful when accessing regional DB2 systems that are all configured with the same LOCATION name.

DDF Block FETCH

- DDF automatically uses multi-row FETCH for read-only queries
- Reduces CPU cost of read-only queries significantly



The DDF block fetch support is enhanced to eliminate the address space switches that occurred previously on each FETCH operation. With this new enhancement, DB2 will perform the address space switch only once per DRDA query block, which will have significant CPU cost reduction for all DRDA read-only queries. In ODBC, use the array interface to make the next improvement with this technique.

Stored Procedures & RRS

- Stored Procedures improvements
 - WLM management of TCBs
 - TCBs are added/removed based on WLM recommendations
 - MAX FAILURES on CREATE PROCEDURE
 - Integrated debugger for SQL Procedure Language
 - COMPJAVA (HPJ) is replaced by interpretive Java
- RRS support for CAF migration
 - CAF-style "implicit" connection support

z/OS Workload Manager function can determine appropriate resource utilization and provide a method of changing the number of tasks within a stored procedures address space.

Customers can set a limit on how many times each stored procedure or user defined function can fail before it is stopped. We already allow the customer to specify a max abend value for all stored procedures and user defined functions on a single DB2 image.

DB2 for z/OS Version 8 will no longer support LANGUAGE COMPJAVA stored procedures.

Applications that have been written to use CAF implicit connect can be easily converted to RRSAF.

Key Java, XML and WebSphere Improvements

- Improved function & consistency
 - ◆ JDBC / SQLJ 3.0 standard
 - ◆ Java Universal Driver used across family
- ◆ Unicode improvements
- Complemented by more consistent SQL
- Increased integration with WebSphere
- XML Publishing



Java support will be more consistent across platforms as we use a single code base across the DB2 family. The improved consistency also adds new function to DB2 and improves integration with WebSphere and Java.

The Java Universal Driver is updated to support the JDBC/SQLJ 3.0 standard, including improvements like savepoints, connection pooling improvements, the ability to reuse PreparedStatements, multiple open ResultSets for a single stored procedure, WITH HOLD cursors, and improved BLOB and CLOB support. Very substantial improvements in Unicode, allowing join of Unicode tables with EBCDIC and Ascii and converting DB2 catalog to Unicode.

<ftp://ftp.software.ibm.com/software/db2storedprocedure/db2zos390/techdocs/F01.pdf>

Key Java Improvements

■ Java Universal Driver




- ▶ Updated to support JDBC/SQLJ 3.0 standard
 - savepoint support
 - connection pooling improvements
 - reusing PreparedStatements
 - standard connection pool properties
 - reset of server connection at getConnection
 - new metadata for PreparedStatements
 - return autogenerated keys
 - multiple open ResultSets for a single stored procedure
 - WITH HOLD cursors
 - improved BLOB/CLOB support



Java support will be more consistent across platforms as we use a single code base across the DB2 family. The improved consistency also adds new function to DB2 for z/OS and improves integration with WebSphere and Java.

We will be supporting both Type 2 and Type 4 Java drivers which will be updated to support the JDBC/SQLJ 3.0 standard, including improvements like savepoints, connection pooling improvements, the ability to reuse PreparedStatements, multiple open ResultSets for a single stored procedure, WITH HOLD cursors, and improved BLOB and CLOB support.

Enterprise Applications & DB2 UDB for z/OS

 4.6 certified	<ul style="list-style-type: none"> -64 bit virtual storage -Unicode -Schema evolution -System-level backup and recovery 	<ul style="list-style-type: none"> -Longer Table Names & Column Names -Additional statistics -Convert Column Type -Altering CLUSTER Option
	 8.45 certified	<ul style="list-style-type: none"> -Multi-row fetch & insert -Multiple DISTINCT Clauses -Lock Contention on volatile Tables -Fast Retrieval of Most Recent Value
 7.8 certified		

Most of the key items in Version 8 help our key enterprise application partners: SAP, PeopleSoft and Siebel, and also help many other applications and customers. The 2 GB address space limit is a major problem for customers who have large dynamic statement cache, many concurrently open data sets and long running units of work. Unicode helps vendors support multinational companies and improve character conversions. These applications often insert and fetch many rows. Using a technique called array fetch & insert or multirow fetch & insert can be as much as 50% less cpu time locally, but improves remote access more. SQL flexibility improvements allow DB2 to be efficient in performance and in productivity for our partners. There are roughly 50 items, with some differences, for each partner. See the book DB2 for z/OS V8: Through the Looking Glass & What SAP Found, SG24-7088 for more detail. SAP R/3 4.6 and PeopleSoft PeopleTools 8.45 were certified for V8 less than four months after general availability.

Unicode Enhancements in Version 8

- Able to join Unicode with EBCDIC or ASCII
- SQL in Unicode, statements, literals, object names, mixed CCSIDs
- Most DB2 character catalog columns in Unicode
- Collating sequence change
- Lengths and maximum lengths can change



WebSphere software



www.unicode.org

The key improvements for Unicode allow much more flexibility, with the ability to join a Unicode table to one that is ASCII or EBCDIC. SQL statements and literals can be Unicode or EBCDIC. Many of the DB2 catalog character columns will be converted to Unicode, so Unicode is for everyone. SQL is converted to Unicode before parsing to allow code-page dependent parsing. The collating sequence for Unicode is similar to ASCII, with numbers sorting lower than letters. The lengths and maximum lengths of strings can change as they are converted to Unicode, so treat lengths as variable.

Unicode enhancements in Version 8

- Able to join Unicode with EBCDIC or ASCII
- SQL in Unicode, statements, literals, object names, mixed CCSIDs / Utility statements can be in Unicode
- Most DB2 character catalog columns in Unicode (Unicode catalog)



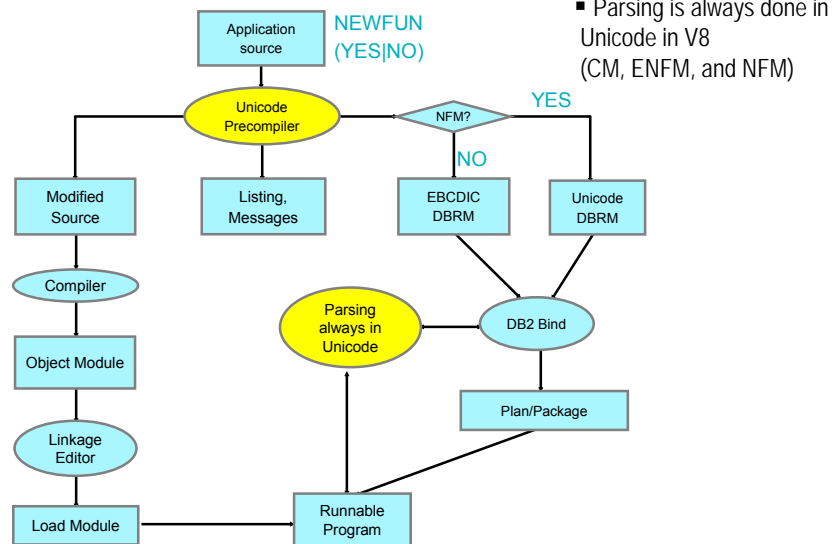
www.unicode.org

The key improvements for Unicode allow much more flexibility, with the ability to join a Unicode table to one that is ASCII or EBCDIC. SQL statements and literals can be Unicode or EBCDIC. Many of the DB2 catalog tables' character columns will be converted to Unicode, so Unicode is for everyone. SQL is converted to Unicode before parsing to allow code-page independent parsing.

The collating sequence for Unicode is similar to ASCII, with numbers sorting lower than letters. The lengths and maximum lengths of strings can change as they are converted to Unicode, so treat lengths as variable.

See z/OS Conversion Services for CCSID, Unicode definitions.

Unicode parsing/precompiler



In V8, all SQL statements are parsed and precompiled in Unicode.

The precompiler helps controlling the type of SQL statements that can be processed by DB2. (NEWFUN keyword))

Only NFM allows the use of new SQL functions. Since the precompiler executes outside DB2, it cannot ascertain the current mode of DB2. In addition to that, you might want to be able to run a partial test (precompilation and compilation without binding or executing) of an application that uses new syntax. Therefore, a new precompiler option (NEWFUN) has been added which tells the precompiler whether or not to allow new syntax, as well as to tell whether or not to produce a DBRM in EBCDIC or Unicode.

If you specify a value of **NEWFUN(NO)**, the precompiler rejects any source SQL statements that contain new V8 syntax. A successful precompilation produces an EBCDIC DBRM, which is compatible with DB2 V7 and earlier releases. The DBRM can be bound on DB2 V7 or V8.

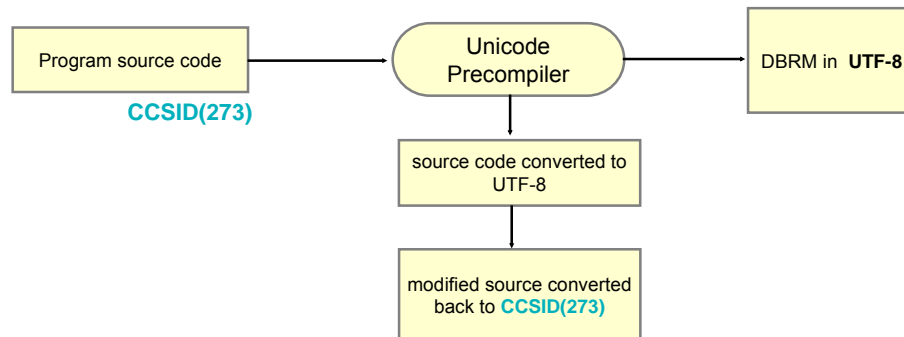
If you specify a value of **NEWFUN(YES)**, the precompiler accepts source SQL statements that contain new V8 SQL syntax. A successful precompilation produces a DBRM that is marked as V8-dependent and therefore not compatible with V7. This happens regardless of whether the program contains new syntax or not. As a consequence it can not be bound on a V7. The DBRM which is produced as a result of the precompilation with NEWFUN(YES) is in UNICODE.

NEWFUN in DSNHDECP

The default value for the NEWFUN precompiler parameter is set to NO during compatibility and EFN mode.

For a new V8 subsystem or for a subsystem which has successfully been converted to NFM, the default changes to YES. The advantage of changing the default is that there is no need to change all the precompile jobs once you get to NFM.

Unicode precompiler



- Precompilation is always done after converting the source to Unicode
- If NEWFUN=YES, you can specify the CCSID of the source program on the precompile
- Otherwise precompiler assumes that source is CCSID 37

When migrating your DB2 UDB for OS/390 and z/OS from V7 to V8, you must go through 3 different modes. Independent of the question in which mode you are currently running your DB2 subsystem, any mode of DB2 V8 now uses a UNICODE precompiler (or precompiler services). The Unicode precompiler converts the program source code to UNICODE UTF-8, performs the precompilation and then converts all statements, including the generated and converts the statements back.

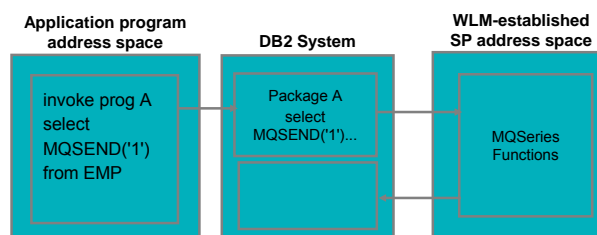
Apart from the modified source, the UNICODE precompiler also generates the corresponding DBRM. If you do not specify any additional precompiler options, the DBRM is generated in EBCDIC as long you do not specify NEWFUN(YES) on the precompile.

NEWFUN(YES) is normally the default when your subsystem is running in New Function Mode (NFM) because as part of the switch to NFM, you normally change the default value for NEWFUN to YES in DSNHDECP.

The CCSID option can only be use in NFM, and tells the precompiler what CCSID the source is coded in.

MQSeries UDFs

- Integrate MQSeries messaging operations within SQL statements
- Are already available in V7 - come in 3 flavours
 - ▶ Basic MQ scalar and table UDFs (eg. MQRECEIVE(), MQSEND() - PQ59549)
 - ▶ MQ publish/subscribe UDFs (eg. MQPUBLISH(), MQSUBSCRIBE() - PQ73340)
 - ▶ MQXML UDFs (eg. MQReadAllXML() - PQ74445)
- Set of UDFs in MQSeries Application Messaging Interface:
 - ▶ UDFs with schema name DB2MQ1C provide only single-phase commit
 - ▶ UDFs with schema name DB2MQ2C support two-phase commit
 - ▶ Separate WLM environments for each flavor of UDFs

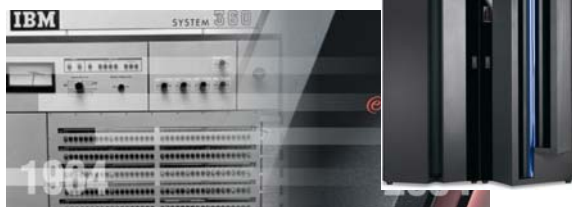


IBM System z9, zSeries & DB2 UDB for z/OS V8

Faster Processors
 Up to 54 Processors
 More memory, better value; 64 bit virtual storage
 ✓ New backup and restore
 ✓ Multilevel Security

✓ Parallel Sysplex
 ✓ Unicode conversion
 ✓ Compression
 ✓ Cryptography
 ✓ zSeries Application Assist Processor
 ✓ z/Architecture new instructions
 ✓ WLM enhanced

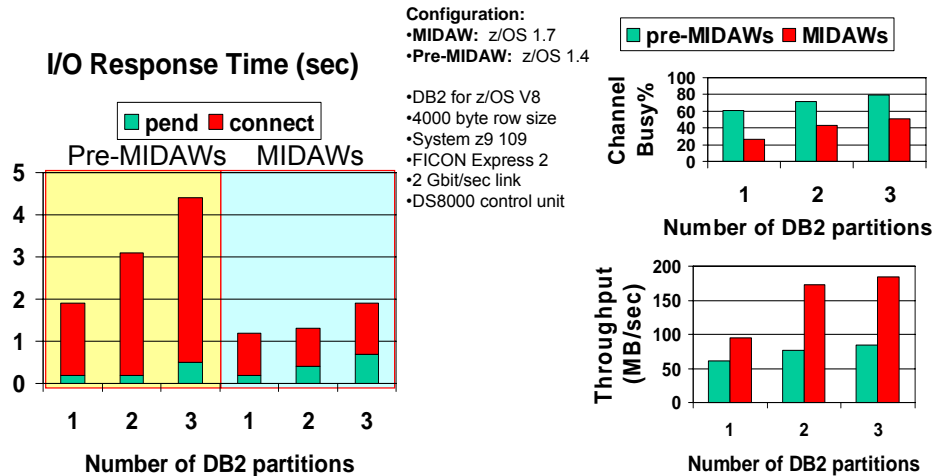
...



These are the key functions of the latest IBM System z9, zSeries z990 and z890 that DB2 UDB for z/OS uses, almost everything to deliver System z9, 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 DS8000 and Enterprise Storage Server™. 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), as we show on the next foil.

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

Parallel DB2 Table Scan, EF 4K (single channel)



140

These are some preliminary measurements with new processors, new software, new channel configurations and new disks. Note the sustained scan rates of 100 megabytes per second.

Configuration: MIDAW: z/OS 1.7 Pre-MIDAW: z/OS 1.4

DB2 for z/OS V8 4000 byte row size

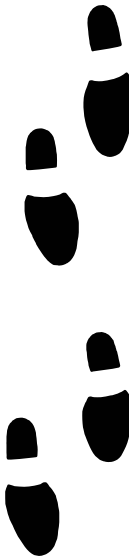
System z9 109

FICON Express 2 2 Gbit/sec link DS8000 control unit

This document contains performance information. 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 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 throughput or performance improvements equivalent to the numbers stated here.

Four step migration process

1. Test with New Install & Migrate
2. Migrate to new release without new function (**COMPAT Mode**)
 - a. Apply SPE to all members
 - b. Start all members
 - c. Start in V8 COMPAT Mode
3. **Enable New Function Mode**
4. Then in **New Function Mode**



The process of migration will take four steps.

Use a new install on a separate subsystem and practice migration on the separate subsystem. Test the function you want to use. This will ensure that you are ready to move forward.

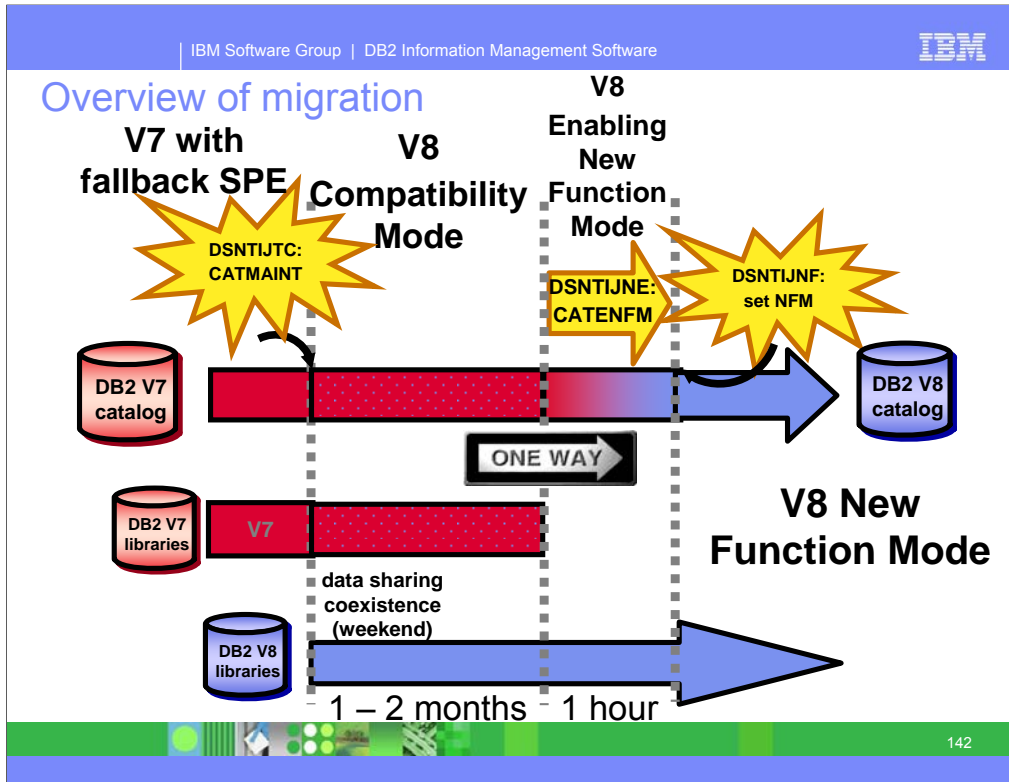
Apply the fallback SPE PQ48486 on every member of each data sharing group and on all subsystems to migrate. Applying the SPE was required only for data sharing in prior versions, but is now required for all subsystems.

Migrate to the new release in Compatibility Mode (CM). You are using the new code but almost no new function.

The third step is Enabling New Function Mode (ENFM).

Finally, after all of the members of a data sharing group are running the new version, you can switch to New Function Mode (NFM).

<ftp://ftp.software.ibm.com/software/db2storedprocedure/db2zos390/techdocs/Z41.pdf>



This is an overview of the migration process. There are two steps which modify the catalog, CATMAINT to change into Compatibility Mode (CM), and also CATENFM for Enabling New Function Mode (ENFM), which contains the major catalog changes to long names and Unicode, using a technique of an online (read only or shrlevel reference) reorg.

Keep the time for data sharing coexistence fairly short, such as over a weekend or one week. Compatibility mode will often be one or two months. Once you move to ENFM, you don't go back to CM or to V7. ENFM will generally be on the order of an hour. Then you turn the switch and are taking advantage of the new function.

Getting ready for V8

Planning and Education

- ✓ Get needed education, books
 - ➔ Classes, conferences, web
 - ➔ Understand dates, V8 changes, Unicode, migration process, ...
- ✓ Resolve incompatible changes
- ✓ Determine new functions to be used
- ✓ Plan & Install prerequisite hardware & software
- ✓ Perform health check



This is the outline of planning for a DB2 UDB for z/OS Version 8 migration. The most important step is being sure that you have all of the needed information. There are many good classes, ranging from a full transition class to conferences to specialized education for others at your site. There are many books, with more coming. Many of the presentations and books are on the web.

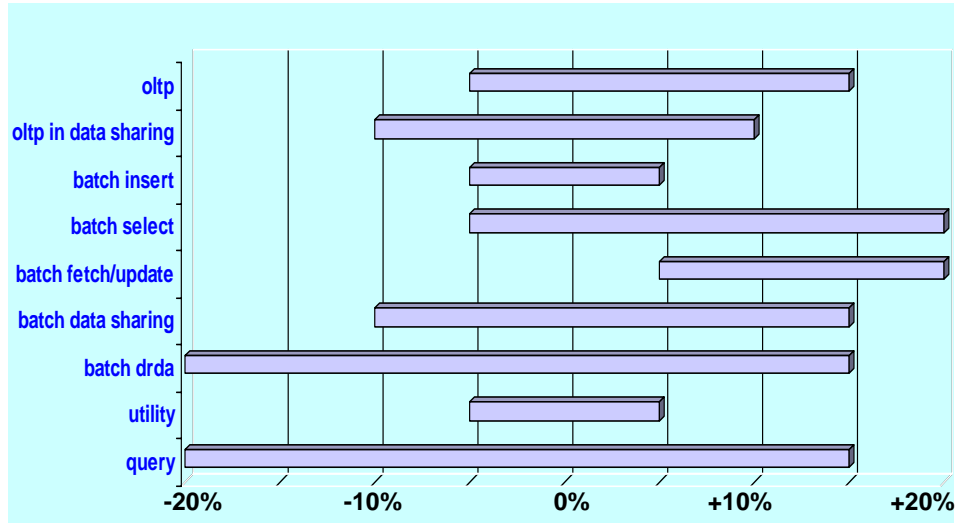
There are a substantial number of changes that can require changes in applications and databases. Get the information to the people who need it.

Reseaching the prerequisite software for the function you will be using, then installing it can take time, especially if you are not current today.

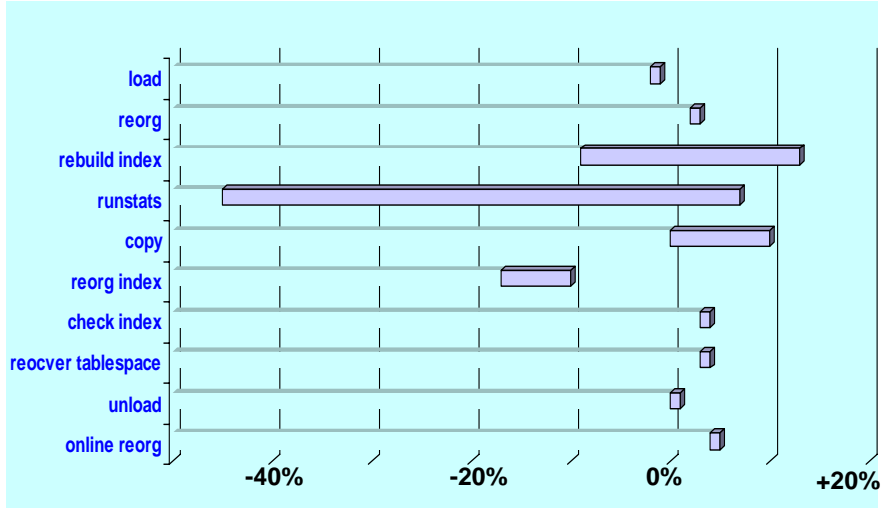
Being ready to test and to manage the inevitable surprises is very important. This is an excellent time for a health check on availability, performance, storage, commit & copy frequency, BIND currency, ...

<ftp://ftp.software.ibm.com/software/data/db2zos/S1322rm2.pdf>

CPU change - continued



Utility CPU - continued



Performance Regression

Expect some cpu performance regression

- ✓ 64 bit DBM1 virtual storage constraint relief
- ✓ Long names, long index keys
- ✓ Longer and more complex SQL statements
- ✓ Unicode catalog & SQL

No change in I/O time

→ **Plan to take advantage of V8 improvements**

Page fix, REBIND, multirow fetch & insert, ...

For applications not taking any advantage of V8 performance enhancements, some CPU time increase is unavoidable to support a dramatic improvement in user productivity, availability, scalability, portability, family consistency.

Taking advantage of the V8 performance improvements is very important to avoid regressions and to offset them. Some are relatively simple, and some have been implemented already in products like QMF and multirow fetch.

See the details in chapters 2 and 4 of SG24-6465, **DB2 UDB for z/OS Version 8 Performance Topics**.

CPU Performance Regression: Subject to change YMMV

- Performance objective: under 10% average
- Typical customer 5 to 10% higher average
- Work load variations
 - ▶ 0 to +15% online transaction
 - ▶ -5 to +10% transaction in data sharing
 - ▶ -5 to +20% batch
 - -5 to +5% insert
 - +5 to +20% fetch, select, update
 - ▶ -10 to +15% batch data sharing
 - ▶ -20 to +15% batch DRDA
 - ▶ -5 to +10% utility
 - ▶ -20 to +15% query

Performance objective is less than 10% average regression.

Typical customer workload regression is expected to be 5 to 10% higher on average, differing by work load: Here are some examples:

0 to +15% online transaction

-5 to +10% transaction in data sharing

-5 to +20% batch

-5 to +5% insert

+5 to +20% fetch, select, update

-10 to +15% batch data sharing

-20 to +15% batch DRDA

-5 to +10% utility

-20 to +15% query

■ Options with significant potential to offset an expected increase include multirow fetch, multirow insert, long term page fix and rebind.

CPU Performance Regression

Some workloads tend to have less regression:

- ✓Virtual storage constrained
- ✓Data space buffer pool or Hiperpool
- ✓SMF89 active
- ✓IRLM PC=YES
- ✓CURRENTDATA NO
- ✓BIND RELEASE COMMIT Data Sharing
- ✓Update intensive batch processing Data Sharing
- ✓I/O intensive workload

Measure all of CPU time

- ✓Not just Accounting Class 2 CPU time
- ✓Must factor in CPU time reduction from Statistics Trace

The following types of workloads tend to have less regression in V8:

- Virtual storage constrained (MINSTOR=YES, CONTSTOR=YES)
- Data space buffer pool or Hiperpool in V7
- SMF89 active in V7
- IRLM PC=YES in V7
- CURRENTDATA NO in V7
- BIND RELEASE COMMIT option in V7 Data Sharing
- Update intensive batch processing in V7 Data Sharing
- I/O intensive workload

Measure all of the CPU time. Do not just take Accounting Class 2 CPU time to compare V7 to V8. You must factor in CPU time reduction from Statistics Trace (DB2 address spaces).

Getting ready for V8



- Migrate to DB2 for z/OS V7
- Migrate to z/OS 1.4 or later (1.3 end service)
 - WLM goal mode required
 - DB2 requires z/OS Unicode services
 - Some functions need z/OS V1R5, R6, R7
- zSeries, z/Architecture 64 bit mode
- Migrate to IBM COBOL V3.3 PL/I V3.4
- IMS V9, V8, V7, CICS TS V3, V2.3
- Details in Program Directory

Migration is only from DB2 V7, so migrate to DB2 V7. When you migrate to z/OS 1.3 or later, WLM compatibility mode is not supported. Some of the function will require later versions, so moving to 1.4 will be better for many customers. End of service for 1.3 is March 2005, and you need to be current, so migrate to 1.4 or later, depending upon the functions you need.

This is a 64-bit virtual application, so the new bi-modal accommodation that allows z/OS on zSeries to run in 31 bit mode will not be able to run DB2 for z/OS V8.

The old COBOL compilers (OS/VS COBOL & VS COBOL II) will not be able to be used with the precompiler. Migrate to current IBM Enterprise COBOL compilers. Old COBOL load modules continue to run (under LE for support). COBOL V2 is now out of support, as is Enterprise COBOL V3R1. End of support for Enterprise COBOL and PL/I V3R2 is October 2005.

Get current software for other products, including IMS and CICS. I recommend CICS V3.

DB2 UDB for z/OS V8 Test Quality Improvements

- **Weekly Regression Tests**
 - Began tests earlier in development cycle
 - Broader more rigorous workload compared to V7
- **Stress Test Environment Improvements**
 - 16 processor z990 machine
 - Multiple native configurations
- **Stress Test Workload Improvements**
 - Heavy use of SQLJ and JDBC under WebSphere
 - Added Linux for zSeries client stress test
 - Stress testing multiple concurrent mixed workloads
 - 21 month stress test duration (12 months in V7)
- **Expanded Quality Partnership Program**
 - Using new functions plus regression tests
 - Duration expanded – 14 months (6 months in V7)

Since this version of DB2 has so much new function, customers often express concern about the quality. We have made many quality enhancements in our process, and have discovered problems earlier as a result. We have devoted more time and more resources to the process. As a result, we expect the quality of V8 to exceed the results of V7, which were much better than V6. The real test of a product comes from our customers.

What are customers saying about DB2 V8?



What can you learn?

Gain their experiences.

At conferences, user groups, on the web

Univar

<http://www.ibm.com/software/os/zseries/index.html#>

<ftp://ftp.software.ibm.com/software/data/db2zos/db2v8dbv.pdf>

<http://www.ibm.com/support/docview.wss?rs=64&context=SSEPEK&uid=swg27005465>

GAD

<http://www.ibm.com/software/data/db2imstools/news.html>

<http://www.ibm.com/software/os/zseries/webcast/mar1/>

Deere, system backup and restore and migration to V8

<ftp://ftp.software.ibm.com//software/data/db2zos/PITR18.pdf>

ftp://ftp.software.ibm.com/software/data/db2zos/Keemle_C3.pdf

Hewitt Associates

<ftp://ftp.software.ibm.com/software/db2storedprocedure/db2zos390/techdocs/OF2.pdf>

Summary of early customer experiences

<ftp://ftp.software.ibm.com/software/data/db2zos/A01jc.pdf>

Health checking

<ftp://ftp.software.ibm.com/software/data/db2zos/G11jc.pdf>

<ftp://ftp.software.ibm.com/software/data/db2zos/G12jc.pdf>



Major objectives:

- Prove stability & reliability of DB2 for z/OS Version 8
- Test DB2 family integration, Port existing UNIX DB2 environment to DB2 for z/OS
- Test Online Schema Evolution availability

IMS, MQS with UDF & triggers, Stored Procedures, WebSphere on z/OS, DRDA, DB2 Utilities & batch data sharing environment

BIG BATCH - 320GB data, Over 4300 jobs

1.5 days compatibility & new function

SQL Coverage (BIG BATCH / all GAD) Over 95%

100% Successful

At the start of the DB2 V8 ESP program, GAD's major objectives were to test and prove the stability and reliability of the DB2 for z/OS Version 8 product; To test DB2 family integration by porting an existing UNIX DB2 environment to DB2 for z/OS; To test the introductory features of the Online Schema Evolution availability feature.

During the ESP, GAD integrated IMS, MQS with UDF & triggers, Stored Procedures, WebSphere on z/OS, DRDA, DB2 Utilities and a huge DB2 batch environment, representing a very heavy regression test workload in a data sharing environment. GAD extended this regression test suite with a BIG BATCH environment (320GB of GAD DB2 production user data), driving over 4300 jobs for 1.5 days in both V8 compatibility and new function modes. The entire regression test environment represents over 95% of the SQL functionality of our production environment for this ESP-business application and was by far the major milestone in our project. The successful running of this heavy batch workload resulted in inspired confidence in the reliability and stability of the DB2 Version 8 product to GAD.



Successful test DB2 family integration
Port existing UNIX DB2 environment to DB2 for z/OS
Without problems
Except some indexes not created automatically

Joint performance study with SVL JAVA type 4 driver

Online Schema Evolution
Praise from the DBA group
Await further enhancements in future releases

GAD also invested time in successfully testing the DB2 family integration support in Version 8, by porting an existing UNIX DB2 environment to the DB2 for z/OS platform without any problems, other than the creation of some indexes which were not created automatically. GAD have also participated in a joint performance study with SVL, evaluating the new Universal Driver component and specifically testing the JAVA type 4 driver support.

Availability enhancements in every new release of DB2 for z/OS are always important to GAD and the latest introduction, Online Schema Evolution has met with praise from the DBA group who await further enhancements in future releases of the product.

See more on the web

<http://www.ibm.com/software/data/db2imstools/news.html>

<http://www.ibm.com/software/os/zseries/webcast/mar1/>



Initial decisions separate DB2 subsystems
Migrate DB2 to V8 to test stability & reliability
DB2 to test new functions

Compatibility Mode batch benchmark for regression testing, migration and fallback procedures

New functions tested:
64 bit addressability
DPSI (Data Partitioning Secondary Indexes)
Online schema changes

When we started this adventure, we decided that we should have a DB2 on which testing all the activities that we will be called to do when we will migrate a significant DB2 to V8 to prove its stability and reliability, and a DB2 on which perform new functions testing.

On the CM DB2 we performed several batch activity to provide significant data to be a valuable benchmark for regression testing, execute migration and fallback procedure couple of time (more or less every time new ptf's batch were applied)

The new functions we decided to test were:

64 bit addressability
DPSI (Data Partitioning Secondary Indexes)
Online schema changes



More data to assure reliability and stability

Migrate huge development subsystem

CICS, IMS, DRDA transactions, WebSphere procedure (DSNALI), QMF, QMF for Windows, UNIX and Win* DRDA transaction & batch

Very good benchmark, stressed subsystem, CPU usage, reliability and stability

Assumed everything that works in V7 runs in V8 CM
Correct (with some little exception)

CCSID usage required work CCSID 500 mistake

During the second half of 2003 we realized that we didn't have enough data to assure reliability and stability so we decided to migrate to CM a huge development subsystem on which are present : CICS, IMS, DRDA transactions, Web-sphere procedure (DSNALI), QMF, QMF for Windows, UNIX and Win* DRDA transaction and batch.

This has been a very good benchmark because there is nothing like a stressed subsystem that can provide information on CPU usage, reliability and stability, that can provide data to be compared with other similar db2s in V7, that can be a thermometer of how everything run.

We started with the assumption that everything that works in V7 MUST run in V8 CM, and we realized that it's correct (with some little exception).

Partners and Vendors

- **Specifications in hand since November 2001**
- **Initial code drops in October 2002**
- **Bi-weekly PTF update deliveries**
- **More than 40 partner/vendor participants**
- **Key vendors will use features such as:**
 - ▶ **Unicode, SQL enhancements**
 - ▶ **64-bit virtual storage**
 - ▶ **System level Recovery**
 - ▶ **...**
 - ▶ **Key vendor offerings support V8 at GA**
 - ▶ **Others planned for rollout through 2004, 2005**

This is an outline of our work with many vendors for applications and tools. With a longer time to work on the product, many vendors were able to provide support much sooner than on prior versions.

DB2 UDB for z/OS Version 8 is

- ✓ **SQL OLTP Leadership:** name lengths, statements, scrolling, expressions, predicates, diagnostics, ...
- ✓ **Break through limitations:** storage, partitions, log
- ✓ **Performance enhancements:** index, materialized query tables, more efficient IO, multi-row
- ✓ **Database changes without an outage:** add partition, rotate partition, backup / restore
- ✓ **Integration**

zSeries, z/OS & ESS platform
 Middleware stack



Reengineered for e-business on demand

157

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.

DB2 UDB for z/OS Version 8 News

- **New function**
 - now {
 - Cross loader with LOBs
 - Built in functions ASCII, TIMESTAMPDIFF
 - DSN1COPY with 1000 OBIDs
 - QMF with multirow fetch
 - Online Check Index
 - z/OS 1.7 up to 7257 extents
 - LOAD, UNLOAD with LOBs
- **New and updated books**
 - Library refresh April 2005, Messages, Codes August
 - Redbooks: Business Value, Performance Topics, WebSphere, MLS, Disaster Recovery, others updated ...
 - Customer information on the web

- ✓DB2 for z/OS and WebSphere: The Perfect Couple, SG24-6319 published January 2005
<http://www.redbooks.ibm.com/redbooks/pdfs/sg246319.pdf>
- ✓Achieving the Highest Parallel Sysplex Availability DB2, REDP-3960, updated January 2005
<http://www.redbooks.ibm.com/redpapers/pdfs/redp3960.pdf>
- ✓**Planning for Multilevel Security & Common Criteria (GA22-7509)**
<http://publibz.boulder.ibm.com/epubs/pdf/e0z2e111.pdf>
- ✓**Multilevel Security & DB2 Row-Level Security Revealed, SG24-6480**
<http://www.redbooks.ibm.com/redpieces/pdfs/sg246480.pdf>
- Ready for Java <http://www.redbooks.ibm.com/redbooks/pdfs/sg246435.pdf>
- Large Objects <http://www.redbooks.ibm.com/redbooks/pdfs/sg246571.pdf>
- Stored Procedures <http://www.redbooks.ibm.com/redbooks/pdfs/sg247083.pdf>
- Webcast <http://www.ibm.com/software/os/zseries/webcast/mar1/>
- ✓ **Cross-Platform SQL Reference V2 published Sept 2004**
<http://ibm.com/developerworks/db2/library/techarticle/0206sqlref/0206sqlref.html>
- ✓ **Disaster Recovery with DB2 UDB for z/OS , SG24-6370 published November 2004**
<http://www.redbooks.ibm.com/redbooks/pdfs/sg246370.pdf>
- ✓ **DB2 for z/OS V8 publications update on web new in April 2005, one August 2005**
<http://www.ibm.com/software/data/db2/zos/v8books.html>
- ✓ **DB2 for z/OS Performance Topics SG24-6465**
 - Cross loader with LOBs PQ90263
 - Built in functions ASCII, TIMESTAMPDIFF PQ95795
 - DSN1COPY with 1000 OBIDs PK05758
 - QMF with multirow fetch PQ99482
 - Online Check Index PQ92749 & PQ96956 open
 - z/OS 1.7 up to 7257 extents PK07590 PK10594 open
 - LOAD, UNLOAD with LOBs PK10278 open

DB2 UDB for z/OS Vnext

- ❑ Integration ➤ Data Definition On Demand
- ❑ Availability ➤ XML, Unicode, LOBs
- ❑ Scalability
- ❑ Productivity ➤ SQL for DB2 family
- ↪ Total cost of ownership
- Utility enhancements



These are primary areas for our next version, carrying on some of the key deliveries from the prior work. Migration to Vnext will be from DB2 for z/OS Version 8 and will prereq z/OS 1.7

Data definition on demand extends the theme of online schema revolution from V8. Additional Unicode enhancements continue the work from V7 and V8. XML work across the DB2 family is a much larger step than in V7 or V8. SQL Procedures become more consistent across the family. While V7 and V8 removed many differences from DB2 for Linux, Unix & Windows, Vnext takes the next big step to improved consistency. Utility enhancements help with new function, more LOB and XML support, better performance and improved availability, removing the BUILD2 step from online reorg.

Release Vision: The vision for DB2 UDB for z/OS Vnext is to enhance DB2's ability to handle new and enterprise applications. Vnext improves the ability to handle new applications with XML, large objects, and many SQL and security improvements. Vnext builds upon and extends DB2 traditional strengths and the ground-breaking Version 8 in many areas: online schema evolution, Unicode, XML, DB2 family SQL, utilities, security and 64-bit virtual storage.

DB2 UDB for z/OS VNext: On-demand Database

- Enable high-volume transaction processing for next wave of Web applications
- Extend the lead in transaction processing availability, scalability and performance
- Reduce cost of ownership and zSeries-specific skill needs
- Improve data warehousing and reporting

Themes for VNext: There are four primary themes for the VNext enhancements. The first is enabling high volume transaction processing for the next wave of web applications. This work extends DB2's ability to handle the next generation of applications by adding XML support and extending SQL and security. The SQL improvements also improve the ability to port applications to DB2 UDB for z/OS and DB2 family compatibility.

The second theme is extending the lead in transaction processing availability, scalability and performance. The key changes in availability extend DB2's ability to make changes to the data definitions without disrupting the online processing. New structures are provided to improve scalability and performance. Bottlenecks for very high performance are removed.


The third theme is reducing the cost of ownership and the need for skills that are specific to the zSeries platform. This theme includes productivity improvements for administrators through consistency with the DB2 family and the relational database management industry. It also includes cost reductions through processing reductions and index compression.

The fourth theme is improving data warehousing and reporting. Today's complex applications include both transactions and reporting, so performing both well is imperative. The key improvements for reporting are optimization enhancements to improve query and reporting performance and ease of use. More queries can be expressed in SQL with new SQL enhancements.

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, 



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.