Considerations for
DB2 V8 Transition

Sponsored by
SoftBase
http://www.softbase.com

Craig S. Mullins
Mullins Consulting, Inc.
http://www.CraigSMullins.com
The Top Ten Dos and Don’ts

1. Do Your Homework
2. Do Know the Modes
3. Don’t Forget to Bone up on CCSID and Unicode Issues
4. Don’t Proceed Without Reviewing Good, Old COBOL
5. Don’t Expect Performance to Stay Exactly the Same
6. Do Prepare for Storage Changes
7. Do Review Your DSNZPARMs
8. Do Prepare for DFSORT
9. Do Study Up on New Features
10. Don’t Expect all 3rd Party Tools to Immediately Exploit V8
1. Do Your Homework

Know the pre-requisites
Know the history
Understand the “big”-ness of DB2 V8
Prepare adequate time to learn before migrating
DB2 UDB for z/OS Version 8

- Became GA on March 26, 2004
- Large and complex release
- Lots of new functions and features
- New migration process
Ten years of DB2 Release history

- DB2 V4 – GA in ‘95 – End of Service Dec ‘01
- DB2 V5 – GA June ’97 – EOS Dec ‘02
- DB2 V6 – GA June ’99 – EOS June ’05
- DB2 V7 – GA March ’01 – EOS TBD
- DB2 V8 – GA March ’04 – EOS TBD

Why the sudden jump to a 3 year release cycle?

- Y2K created unique situation
  - Many customers in a long-term ‘freeze’ for system changes
  - Required IBM to support migration to V7 from V6 OR V5
  - IBM recognized that many customers would take several years to migrate from V5 to V6 and/or to V7
  - It took a longer time than for past releases for customers to migrate to V7
  - (Estimate that only by YE 2003 were more than ½ of DB2 customers on V7)
DB2 Version Support Details

Installation and Migration

Environment requirements

- DB2 V7
- V8 must run on zSeries in 64-bit mode
  - Z800, z900 (or bigger) box
- z/OS R3 or higher (see next slide)

Installation and Migration paths

- Install brand new V8 Subsystem
- Migrate V7 to V8
  - COMPAT, ENFM, then NFM
  - V5-V6-V7-V8
  - V5-V7-V8
z/OS Requirements

z/OS level required

- R3 is minimum
  - WLM Goal Mode is enforced with z/OS R3 or later
  - z/OS R3 is out of service as of March 2005
- With R4
  - Improved data sharing performance
  - Improved Unicode conversion performance
- With R5
  - Multilevel Security including row level security
  - System Level PIT Recovery utilities
    - **BACKUP SYSTEM**
    - **RESTORE SYSTEM** without LOGONLY option
  - 31-bit DSAB support
- With R6
  - zSeries Application Assist Processor (Java offload)
  - WLM enhancement for stored procedures
Fallback SPE

- UQ81009 must be applied first
- Information is stored in BSDS /SCA
  - Checked at start-up time by DB2

MUST START DB2 V7 AT LEAST ONCE AFTER THE SPE HAS BEEN APPLIED
Health Check

DSNTIJPM

- Checks code page configuration
  - Evidence of multiple CCSIDs in the same encoding scheme
- Type 1 Indexes
- Data Capture on Catalog tables
- Partitioned tables that use selective partition locking
- Partitioned tables that have truncated limit key (PQ48024 in V6/V7)
- Stored procedures that use LANGUAGE COMPJAVA
- Stored procedures that use DB2 SPAS
- Use of DSNWZPR module by DSNWZP

Also shipped with DB2 V7 as DSNTIPJ8 via PQ84421
Before Migrating...

Ensure a stable and useful V7 environment:

- Reasonably up-to-date on maintenance
- Save accounting reports
- Save critical access paths for important applications
  - Be sure you have PLAN_TABLE access paths
  - Also DSN_STATEMENT_TABLE and DSN_FUNCTION_TABLE
  - Alternately, Visual Explain reports…
    - Visual Explain V8 XML files for the V7 access.
2. Do – Understand the Modes

- 3 modes of DB2 execution in V8
  1. Compatibility Mode (CM)
  2. Enabling New Function Mode (ENFM)
  3. New Function Mode (NM)
Compatibility Mode (CM)

- Can migrate from V7, not earlier releases
- Catalog is EBCDIC, no long names
- Fallback to V7 is possible
- No new function is available
  - Although there will be VSCR and some optimizer enhancements
Compatibility Mode is transitional phase

64-bit architecture storage above the bar is available

- Buffer pools which can be page fixed for performance
- RID pool and SORT pool
- Compression dictionaries
- EDM pool for DBD cache and dynamic statement cache

Page fixing for performance

- PGFIX(YES) by bufferpool
- Frequent page read or writes when there is adequate real storage
- Avoids processing time for fixing and freeing pages every time there is an I/O
CM Migration Issues

Consider migrating all subsystems to V8 CM before moving forward to ENFM phase

Validate existing applications against V8 base code in CM

◆ Go through all major business cycles before moving on to ENFM
  ● monthly, quarterly, bi-annual processing

Ensure you have good backup and restore procedures in place before beginning the ENFM process
**Fallback & Coexistence**

**Fallback**
- Fallback to V7 only permitted from V8 CM
- The fallback SPE must be on.
- The V8 catalog structure will remain in V7 after fallback
- Returning from ENFM or NFM to CM mode is ONLY possible using restore processing

**Release coexistence**
- All data sharing members must be in CM before moving to ENFM
- All data sharing members must be at least V7
Enabling New Function Mode (ENFM)

- Enabling New Function Mode (ENFM)
  - Converts catalog to Unicode and long names, one tablespace at a time, via ALTER and online reorg
  - 18 catalog table spaces converted, with indexes – in the order shown below:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>SYSVIEWS</td>
<td>10.</td>
</tr>
<tr>
<td>2.</td>
<td>SYSDBASE</td>
<td>11.</td>
</tr>
<tr>
<td>3.</td>
<td>SYSDBAUT</td>
<td>12.</td>
</tr>
<tr>
<td>4.</td>
<td>SYSDDF</td>
<td>13.</td>
</tr>
<tr>
<td>5.</td>
<td>SYSGPAUT</td>
<td>14.</td>
</tr>
<tr>
<td>6.</td>
<td>SYSGROUP</td>
<td>15.</td>
</tr>
<tr>
<td>7.</td>
<td>SYSGRTNS</td>
<td>16.</td>
</tr>
<tr>
<td>8.</td>
<td>SYSHIST</td>
<td>17.</td>
</tr>
<tr>
<td>9.</td>
<td>SYSJAVA</td>
<td>18.</td>
</tr>
</tbody>
</table>

- The process (one job) is interruptible and restartable
  - **No fallback to CM or V7**
  - **Still no new function available**
ENFM Issues and Concerns

- Do not move on to ENFM until you are absolutely certain that you will not need to return to Version 7
- Cannot begin ENFM processing if any active DB2 V7 subsystems still exist in data sharing group
  - You will not be able to start a DB2 V7 member in the group once ENFM process begins
- Restoring entire system from backup copy is ONLY option
- Clean up catalog prior to ENFM, removing unwanted objects
- Take image copy of DB2 catalog and directory after job DSNTIJNE has completed processing or DSNTIJNE paused
ENFM Recommendations

Do NOT stay in ENFM phase any longer than necessary

- Duration of ENFM will depend on the size of the catalog:
  - The larger the catalog the longer ENFM takes *Most catalogs under one hour*

Verify that BP8K0, BP16K0 and BP32K are created

- BP8K0, BP16K0 must be created prior to ENFM. Will fail with resource unavailable if they do not exist
  - For data sharing must also define GBP8K0, GBP16K0 and GBP32K buffer pools

Increase size of the catalog table space and index space
VSAM data sets prior to ENFM

- 2 byte length to existing CHAR columns
Consider backing up the entire subsystem prior to ENFM

- Quiesce V8 CM system before running DFSMS to create copy of entire V8 CM subsystem
  - including DB2 catalog and directory, logs, BSDS and application data

**Keep an eye on SYSVIEWS size**

- Parse trees stored in SYSVIEWS will take up more space than in previous releases, so it is recommended that you increase the allocation size.

**Change Data Capture:**

- During ENFM, Change Data Capture (CDC) is turned off automatically for any catalog table that has it enabled. Once in NFM, CDC can be turned back on, but it is manual
User indexes on the catalog:

- Will be automatically rebuilt as a result of REORG of table space. But that can result in a much larger index if columns were increased in size and marked as VARCHAR.

- Therefore, consider dropping user-created catalog indexes on VARCHAR columns that expand in V8. These indexes will need more space and it is best to drop them before running DSNTIJNE, and then to recreate them as NOT PADDED indexes in NFM.

  - If you don’t drop them, when you get to NFM you can ALTER the index to NOT PADDED (if key column is VARCHAR) and rebuild the index(es).
ENFM Steps

Run DSNTINST (install CLIST)

- Enter ENFM in DSNTIPA1 panel

Calculates preliminary space allocations for shadow data sets

Lists image copy dataset allocation information

Generates installation jobs

- DSNTIJNE – ENFM processing
- DSNTIJNH – halt DSNTIJNE
  - Halts the table space conversion process after last successful table space
- DSNTIJNF – turn NFM on
- DSNTIJNG – update DECP for NFM
- DSNTIJEN – return to ENFM status
- DSNTIJNR – convert RLST for long name support
DSNTIJNE Processing ***

DSNTIJNE is the key job in the ENFM processing

- DSNTIJNE can be stopped after the REORG of any table space (with DSNTIJNH)
- Changes types and lengths of existing catalog columns
- Converts catalog data to Unicode
- Changes buffer pools for several catalog table spaces
- Changes catalog index with varying length columns to NOT PADDED
- Changes page size of several catalog table spaces
- Catalog changes
  - SYSDUMMY1 is moved to SYSEBCDC
  - No columns added to existing catalog tables
  - Most catalog tables and directory have columns that change type
  - CHAR => VARCHAR(128 or greater)
  - VARCHAR => VARCHAR(128 or greater)
  - Index padding
    - Many catalog indexes are changed to NOT PADDED
    - User indexes on the catalog will NOT be changed to NOT PADDED
Unicode conversion step

- Table spaces converted to Unicode using online REORG with SHRLEVEL REFERENCE
- FOR BIT DATA columns will not be converted to Unicode
- Table spaces are marked ‘UNICODE’ during online REORG
- If process fails, the old catalog data is maintained
- Once table space is Unicode there is no going back
  - other than via restoring entire DB2 subsystem
If space problems in DSNTIJNE the succeeding steps will be skipped

- Last step performs –TERM UTIL
- The old catalog data is still maintained
- The tablespace switch does not take place

Use only DSNTIJNH to halt DSNTIJNE processing

- No other means of halting conversion processing should be used.
- Restarting DSNTIJNE will skip those table spaces that have already been converted and continue for those table spaces that have failed (or that remain to be processed).
Fallback & Coexistence

**Fallback**

- No supported return to V8 CM and no fallback to V7
- Recover by doing a PIT recovery of entire DB2 subsystem or group logs, BSDS, catalog, directory and data and indexes
  - All CF structures must be purged

**Coexistence**

- ENFM & NFM are group scope
  - Once in ENFM all members are considered to be in a status of ENFM.
  - A V7 subsystem cannot be started if group is in ENFM
  - A new V8 member could be started, it joins the group in ENFM status
New Function Mode (NFM)

- **New Function Mode (NFM)**
  - Catalog is Unicode, with long names, new page sizes
  - DBRMs are Unicode (by default)
    - All new function is enabled
    - NEWFUN(YES/NO) is precompiler parameter
    - NEWFUN(NO) creates V7 compatible DBRM

- **Other Notes**
  - -DIS GROUP command shows mode information
  - Newly installed V8 subsystems are NFM
  - Plans and/or packages that access the catalog may be invalid
  - Queries accessing catalog may need to be rewritten
NFM Catalog Considerations

- **Storage Requirements?**
  - Possible **increase** in space required for system catalog due to CHAR -> VARCHAR conversion issues
    - Length bytes are required for the new VARCHAR columns
  - Possible **decrease** in space required for system catalog due to index padding
    - DB2 defined indexes on the system catalog are converted to not padded (for indexes with variable keys)
  - **Estimate?** Some are seeing about a 10% increase in disk storage required for the catalog and directory
    - But if you’ve never REORGed an older catalog you could see a decrease
Fallback & Coexistence

Fallback

- DSNTIJEN – allows the system to return to ENFM
  - Fallback to V8 CM or V7 is not supported
- PIT recovery using system restore to V7

Coexistence

- V8 NFM is group-wide
Timeframe Estimation

- Apply Fallback SPE
- Compatibility Mode: Approx 2 – 3 months?
- Enabling New Function Mode: < 1 day?
- New Function Mode: Until Next Version…
3. Don’t Forget to Bone up on CCSID and Unicode Issues

1. Review your CCSID settings and prepare

- DB2 Version 8 requires you to provide a DSNHDECP module that specifies valid, non-zero CCSIDs for single-byte character sets (SBCS) for both EBCDIC and ASCII.
  - For Far East languages like Chinese, and Korean, you must also specify valid, non-zero CCSIDs for mixed-byte (MBCS) and double-byte (DBCS) character sets for both EBCDIC and ASCII.

- See Appendix A of the *DB2 Installation Guide*, GC18-7418 for information on choosing valid CCSIDs for ASCII and EBCDIC.

2. Understand the impact of Unicode on V8
CCSID stands for Coded Character Set Identifier, a unique value that identifies a code page.

For example:
- CCSID 37 equates to U.S. EBCDIC
- CCSID 437 equates to U.S. ASCII PC
- CCSID 819 equates to U.S. AIX ASCII
- CCSID 1208 equates to Unicode.

DB2 also uses a System CCSID (SCCSID), specified on the DSNTIPF Installation Panel, and stored in DSNHDECP. This is the default CCSID for the DB2 system.
CCSID Issues

There is the potential for incorrect character interpretation (data corruption) if multiple CCSIDs within one encoding scheme

- Terminal emulator using incompatible code page
- Incorrect CCSID specification
- Change of CCSID specification

Must be fixed prior to moving to V8

- V7 APAR PQ56697 introduces checks for non-zero valid CCSIDs and issues warning when trouble found
- V7 APAR PQ89018 will provide CCSID detection if DECP value does not match the emulator value
- APAR PQ97373 allows the user to specify the names of the SPUFI plans on the defaults panel. DB2 system programmers can then bind multiple SPUFI plan pairs for various EBCDIC CCSIDs, according to the CCSIDs of end-user TSO terminal sessions.
Example ***

This problem document shows the impact of an incompatible CCSID

Problem
In Version 8, I am receiving a syntax error on the following lines of C code in SYSIN:
#include <stdio.h>
EXEC SQL INCLUDE SQLCA;
struct dsnMsg {short len; char txt[7][132];} dsnMsg = {7 * 132};
extern short dsnh2w(char *valca); struct dsnMsg *int *);

Error:
DSNH0071 E DSNHTOER LINE 5 COL 35 ILLEGAL CHARACTER IGNORED:
'I' ("AD'X")

The same program compiles in V7.1 without error.

Cause
The precompile step is being run with a DBCP with CCSID(500), but the C program contains square brackets in CCSID 1047 codepoints ("AD'X", "BD'X").

OPTIONS SPECIFIED: HOST(CPP),SOURCE,XREF,MARGINS(1,90),STDSQL(0)
OPTIONS USED - SPECIFIED OR DEFAULTED

APPOST
APPOSTSQL
ATTACH(TSO)
CCSID(500) <---- defaulted from DBCP
NOPADNTSTR
CONNECT(Z)
FLAG(I)
HOST(CPP)
JDBC(590)
LINKCOUNT(60)
MARGINS(1,80)
NOW_MOUNT дир
OKPASS
OPTIONS
PERIOD
SOURCE
STDSQL(0)
SQL(DB2)
XREF

struct dsnMsg {short len; char txt[7..132];}
AA9A3D3AD4CA899A489548894AAAAF8A67FB80 <---- square
brackets are "BD'X", "BD'X" (1047 codepoints)
23942304254270286939355023819373755132580

The message DSNH0071 E DSNHTOER LINE 5 COL 35 ILLEGAL CHARACTER
IGNORED: 'I' ("AD'X") explains that the character that can't be recognized
in CCSID 500 is the left square bracket in CCSID 1047.

Solution
Use the CCSID(1047) precompile option to precompile this program.
CCSID Issues

3270 CCSID 37

UPDATE EMP
SET NAME = 'MÜLLER'
WHERE NAME = 'SMITH'

Appl. Enc. Scheme: EBCDIC

MÜLLER
= x'D4FCD3D3C5D9'

3270 CCSID 273

SELECT TEL
FROM EMP
WHERE NAME = 'MÜLLER'

MÜLLER
= x'D45AD3D3C5D9'

DB2 for z/OS  
SCCSID = 37

EBCDIC Table EMP

<table>
<thead>
<tr>
<th>NAME</th>
<th>FIRSTNME</th>
<th>TEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMITH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MÜLLER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Miller</td>
<td>TOM</td>
<td>222</td>
</tr>
<tr>
<td>Croft</td>
<td>Anne</td>
<td>333</td>
</tr>
</tbody>
</table>

Result: 0 Rows !!
When an SQL statement references table objects with multiple CCSID sets...

- If an expression or comparison involves two strings which contain columns with different CCSID sets, drive to Unicode if necessary.
- If an expression or comparison involves two strings with different CCSID sets where only one of them contains a column, drive to the column's CCSID set.
- If an expression or comparison involves two strings with different CCSID sets and neither contains a column, drive to Unicode.
- String constants and special registers in a context by themselves use the application encoding scheme.
Starting with V7, DB2 supports 3 encoding schemes

- ASCII
- EBCDIC
- UNICODE

DB2 uses the same CCSIDs for Unicode regardless of language. DB2's Unicode CCSID are:

- 367 for SBCS data
- 1208 for MBCS data
- 1200 for DBCS data.
Unicode is a single character set that encodes all of the world’s scripts (sort of)

The Unicode standard provides a cross platform, cross vendor method of encoding data that enables loss-less representation and manipulation

Before Unicode

- Many Standards (ANSI, JIS, TISI)
- Provided by various vendors [IBM (ASCII and EBCDIC), HP, Microsoft…]

Foundation for globalization of data

- Customers want to use DB2 to manage data from around the world
- No pre-Unicode code page handles all characters
- Handles most current and historical languages
- Handles scientific and mathematical symbols
- Handles other symbols: wingdings, dingbats, …
- Required by many modern standards: XML, Java, LDAP, CORBA
Most of DB2 V8 operates in Unicode (see next slide)

- Make sure you have all necessary conversions defined before you attempt to bring up a V8 subsystem.
- Details are provided in:
  - DB2 Installation Guide (GC18-7418) -- Appendix A
  - DB2 V8 Technical Preview Redbook (SG24-6871) -- Appendix A.

Conversion definitions need to be set up for your environment

- Must configure Unicode Conversion Services for z/OS
- Mappings must be included for every possible combination of code page translation
Most internal DB2 processing is done in Unicode

- Parsing – Literals in parse tree are in Unicode
- Utility parsing – Option to parse in Unicode
- Precompiler – Source converted to Unicode, parsed in Unicode
- DBRM – Unicode in NFM
- Bind – Statement text converted to Unicode if necessary
- Optimization – Catalog look up in Unicode (even in COMPAT Mode)
- Authorization – Unicode internal, EBCDIC external (e.g. RACF)
- Internal names – PLAN Name, Package Name
- Special Registers – Stored in Unicode
- Tracing – Option to output trace data in Unicode (certain fields)
Unicode and the Catalog

**DB2 V7 Catalog is encoded in the SYSTEM EBCDIC CCSID**
- Static statements are stored in the catalog in EBCDIC
- View definitions are stored in the catalog in EBCDIC

**DB2 V8 Catalog is encoded in Unicode UTF-8**
- Static statements are stored in the catalog in Unicode *(by default)*
- View definitions are stored in the catalog in Unicode

**Any name which must be passed to z/OS must be convertible to EBCDIC. For example:**
- Database Name(dataset qualifier)
- Table space/Index space Name(dataset qualifier)
- DBRM Name(PDS member name)
- (UDF, SP, Exits, Fieldproc...) (PDS member name)

**Identifiers may not be able to be generated from all clients so they should really be limited to common subset that can be represented on all clients.**
SQL in the Catalog

SQL statement text stored in SYSSTMT and SYSPACKSTMT:

- EBCDIC for
  - Applications Precompiled Prior To DB2 V8
  - Applications Precompiled in NEWFUN(NO) mode
- Unicode for
  - Applications Precompiled in NEWFUN(YES) mode
Program Preparation – pre-V8

Application source

Precompiler

Modified Source

Listing, Messages

DBRM

Compiler

Object Module

Parser

DB2 Bind

Linkage Editor

Load Module

Plan/Package

Dynamic SQL
Accessing the Unicode Catalog

SQL which returns, compares or orders by a DB2 catalog column

- Evaluated now in Unicode instead of EBCDIC
- Column data will be returned in Unicode
- Comparison of column data will be evaluated in Unicode
- Ordering will cause output to be sorted in Unicode
- Result set and result set sequence may differ from Compatibility Mode (EBCDIC)

<table>
<thead>
<tr>
<th>EBCDIC</th>
<th>hex value</th>
<th>Unicode (ASCII)</th>
<th>hex value</th>
</tr>
</thead>
<tbody>
<tr>
<td>space</td>
<td>'40'x</td>
<td>space</td>
<td>'20'x</td>
</tr>
<tr>
<td>lower case</td>
<td>'81-89'x</td>
<td>'91-99'x</td>
<td>numerals</td>
</tr>
<tr>
<td></td>
<td>'91-99'x</td>
<td>'A1-A9'x</td>
<td></td>
</tr>
<tr>
<td>upper case</td>
<td>'C1-C9'x</td>
<td>'D1-D9'x</td>
<td>upper case</td>
</tr>
<tr>
<td></td>
<td>'D1-D9'x</td>
<td>'E1-E9'x</td>
<td></td>
</tr>
<tr>
<td>numerals</td>
<td>'F0-F9'x</td>
<td>lower case</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
You do NOT have to convert your application data to Unicode

Unicode may increase your storage, but it won’t double it:

- In UTF-8, the first 127 code points are the same as that of 7-bit ASCII, with one byte being used for characters such as A-Z, a-z, and 0-9.
- Other characters are stored as one to four bytes, with accented characters often taking two bytes, and Far Eastern characters taking three to four bytes.
Some Unicode References

Unicode Web Site  http://www.unicode.org
Unicode Book (download)  http://www.unicode.org/unibook

Redbook: SG24-6079
DB2 V8: Everything You Ever Wanted to Know… and More
♦ Chapter 6 – Unicode in DB2 for z/OS
♦ Appendix B – Unicode Setup

Redbook: SG24-6871
DB2 V8 Technical Preview
♦ Appendix A – Unicode Definitions

Manual: GC18-7418
DB2 Version 8 Installation Guide
♦ Appendix A – Character Conversion

DB2 V7/V8 Unicode Support
Older COBOL compilers are no longer supported:

- **OS/VS COBOL** (out of support since 1994)
- **VS COBOL II** (out of support since 2001)
  - An existing COBOL II load module running in LE will still work though
  - But no more compiles for OS/VS COBOL or VS COBOL II

- **DB2 V8 supports**:
  - IBM COBOL V2.2 *(already end of service though!)*
  - Enterprise COBOL V3.2 or later
    - End of service coming up soon – October 2005
More on COBOL

See also the following Techdocs, available at:


In case you are not on any of the supported releases, you will need to migrate. In that case, be sure to review the following:

- *Enterprise COBOL for z/OS Compiler and Runtime Migration Guide (GC27-1409).*
5. Don’t Expect Performance to Remain the Same!

DB2 is having to perform additional work, due to factors such as:
- increased code size
- 64-bit address translation, and
- increased module linkage costs.

Performance objective is less than 10% average CPU regression — when comparing V7 and V8 subsystem with the same size buffer pools, hardware, workload, and so on.

Real storage requirements can also increase by at least ten percent as compared to V7... more on storage coming up in #6.
According to Akira Shibamiya (IBM):

Typical customer workload regression is expected to be 5 to 10% higher on average, assuming no application rewrite and no aggressive configuration change to take advantage of V8 significant performance features

- 0 to +15% Online Transaction
- -5 to +10% Online Transaction in Data Sharing
- 0 to +20% Batch
- -5 to 5% INSERT
- +5 to +20% SELECT, FETCH, UPDATE
- -10 to +15% in Batch Data Sharing
- -20 to +15% Batch DRDA
- -5 to +10% Utility
- -20 to +15% Query
Feedback from some V8 sites indicate increased CPU for the subsystem, but that can be offset by performance gains in applications.

Actual CPU increase? I’ve heard figures that range from the expected 5%-10% range… all the way up to 40%… but everyone will be different

If you are running around 90% CPU under V7 then you should think VERY carefully before moving to V8…
Improved Performance By...

- More Stage 1 predicates (compare unlike types)
- Materialized Query Tables
  - Data warehousing, aggregates
  - Automatic Query Rewrite
- Multi-row INSERT and FETCH
- CI Size = Page Size (Page Size > 4K)
- Volatile Tables
Table UDF cardinality and materialized FETCH
- Estimate the expected number of rows returned by table UDFs
- All rows are returned during the first invocation of the UDF
  - Can save (#rows_in_table_UDF_result – 1) context switches, at the cost of a one-time workfile creation, the cost to insert the rows into the workfile and the cost of deallocating the workfile.

TRIGGER work file creation avoided
- When the WHEN clause (conditional trigger) evaluates false (and the trigger is not invoked)
- When old/new transition variables/tables can fit in a memory buffer
Index Enhancements

- Indexing Enhancements
  - Varying-length index keys (NOT PADDED)
    - NOT PADDED indexes can provide a true index-only access path
  - Backward index scan
  - Sparse Index for Star Join
- Long index keys
Distribution Stats

- RUNSTATS updated to collect additional stats
  - frequency value distributions for non-indexed columns or groups of columns
  - Cardinality values for groups of non-indexed columns
  - LEAST frequently occurring values (indexed and non-indexed)
  - MOST frequently occurring values (indexed and non-indexed)
  - Sort subtasks required (defined in utility syntax)

No need for DSTATS (downloadable tool for V7)
New RUNSTATS Keywords

**SORTDEVT** - device-type

**SORTNUM** - integer

**colgroup-spec**:

**COLGROUP** - (column name)

**FREQVAL** - COUNT - integer

**correlation-stats-spec**:

**FREQVAL** - NUMCOLS - integer

**COUNT** - integer

**KEYCARD**

**FREQVAL** - NUMCOLS - 1 - COUNT - 10 - MOST

**MOST**

**BOTH**

**LEAST**
64-bit Architecture and VSCR

An exabyte (EB) is a large unit of computer data storage, two to the sixtieth power bytes. The prefix exa means one billion billion, or one quintillion, which is a decimal term. Two to the sixtieth power is actually 1,152,921,504,606,846,976 bytes in decimal, or somewhat over a quintillion (or ten to the eighteenth power) bytes. It is common to say that an exabyte is approximately one quintillion bytes. In decimal terms, an exabyte is a billion gigabytes.
On migration to V8 in 64-bit real and virtual

- Buffer pools above the 2GB bar *(more on next slide)*
- EDM pool / RID pool / SORT pool / compression stored above the 2Gb bar
- EDM POOL DATASPACE SIZE becomes EDMPOOL DBD CACHE SIZE
- EDM POOL DATASPACE MAX to EDMPOOL STATEMENT CACHE SIZE
- IRLM above 2GB bar and PC=YES
  - PC=NO is no longer supported.
  - All locks are now in IRLM private storage above the bar.
Buffer Pool Issues

Create New Buffer Pools & Change Defaults
- ADD BP8K0 default 1000
- ADD BP16K0 default 500
- BP0 min size 2000 default 20000
- BP32K default 250

Buffer Pool sizing
- **New limit for virtual buffer pools is 1 TB (was 1.6 GB)**
- For Virtual Pools & data spaces
  - V8 BP size is VPSIZE & data space size
- For Virtual Pools & hiperpool
  - V8 BP size is (VPSIZE + HPSIZE)
- Maintains original virtual pool and hiperpool sizes for fallback
Some customers are planning to migrate to CM relatively quickly for the VSCR and optimization enhancements

- With no immediate plans to then migrate to ENFM and NFM
- Allows immediate storage and (some) optimizer benefits
  - And gives time to digest all of the new features and functionality to best determine how and when to use them
Important subsystems such as DB2 should not be paging to auxiliary storage

According to John Campbell (IBM) at IDUG 2005:

- DB2 V8 will increase real storage usage
- Estimated typical range of 1% to 10% more
7. Do Review DSNZPARMs

Defaults change for many, such as:

<table>
<thead>
<tr>
<th>ZPARM</th>
<th>Old Default</th>
<th>New Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTHREAD</td>
<td>70</td>
<td>200</td>
</tr>
<tr>
<td>MAXDBAT</td>
<td>64</td>
<td>200</td>
</tr>
<tr>
<td>CONDBAT</td>
<td>64</td>
<td>10,000</td>
</tr>
<tr>
<td>LOGAPSTG</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>DSMAX</td>
<td>3,000</td>
<td>10,000</td>
</tr>
<tr>
<td>CHKFREQ</td>
<td>50,000</td>
<td>500,000</td>
</tr>
<tr>
<td>IDFORE</td>
<td>40</td>
<td>50</td>
</tr>
<tr>
<td>IDBACK</td>
<td>20</td>
<td>50</td>
</tr>
</tbody>
</table>
More on ZPARMs

Check out Willie Favero’s blog for more on V8 system parameter changes

⚠️ You can find it here:

The IBM utilities for DB2 V8 require DFSORT

- No additional license (or cost) is required for the DB2 utilities to use DFSORT.
- You already have DFSORT installed.
  - You may not be using it, but it is there.
  - DFSORT is always shipped and installed with z/OS.
- The only action needed is:
  - Add the DFSORT load libraries to the link list after the OEM sort libraries, or;
  - Add a steplib/joblib to the DB2 utilities batch JCL.

http://www.ibm.com/support/docview.wss?rs=0&uid=isg1Il14047
What if You Don’t Prepare for or Have DFSORT?

- DB2 will issue message DSNU1640I if the utility is unable to locate DFSORT

Limited Functionality

- DB2 Utilities use only the SORT and MERGE functions of DFSORT. If you want to use DFSORT for any other uses outside of DB2, you will need to separately order and license DFSORT from IBM.

Support and Maintenance for DFSORT

- Support is shipped with z/OS. If a customer has problems with DB2's use of DFSORT, they would open an ETR with DB2 Level 2 Support no different that they would do today for any other DB2 related issue.
There are a LOT of new features and functionality in DB2 V8

Take the time to review and understand the new features and:

- Determine which features your shop will most likely implement first.
- Determine which features are not on your short-term “needs” list
- Factor this into your migration planning
  - For evaluation of third party vendors and tools
  - For staff education requirements
My Top Ten New DB2 V8 Features

1. 2M SQL Limit
2. Partitioning changes
   up to 4096 Partitions, table-based partitioning, clustering separation
3. Stage 1 for Unlike Data Types
4. Data Partitioned Secondary Indexes
5. Sequences
6. Materialized Query Tables
7. Multi-Row FETCH and INSERT
8. Distribution Statistics
9. Recursive SQL
10. Online Schema Change
### NPSI versus DPSI

#### Table:

<table>
<thead>
<tr>
<th>EXP_DATE</th>
<th>CUSTNO</th>
<th>CUSTNAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>07-11-2004</td>
<td>500</td>
<td>ACME RENTAL</td>
</tr>
<tr>
<td>07-19-2004</td>
<td>100</td>
<td>MULLINS AND ASSOC.</td>
</tr>
<tr>
<td>07-25-2004</td>
<td>600</td>
<td>BMC SOFTWARE</td>
</tr>
<tr>
<td>07-31-2004</td>
<td>175</td>
<td>SUPER BANK</td>
</tr>
<tr>
<td>08-01-2004</td>
<td>400</td>
<td>SPUMCO</td>
</tr>
<tr>
<td>08-16-2004</td>
<td>333</td>
<td>SIGH BASE CORP.</td>
</tr>
<tr>
<td>08-27-2004</td>
<td>200</td>
<td>BETH-ANN INC.</td>
</tr>
<tr>
<td>08-28-2004</td>
<td>900</td>
<td>DIAL-A-DBA</td>
</tr>
<tr>
<td>09-01-2004</td>
<td>800</td>
<td>VAN THE MAN</td>
</tr>
<tr>
<td>09-02-2004</td>
<td>715</td>
<td>DBAZINE.com</td>
</tr>
<tr>
<td>09-04-2004</td>
<td>300</td>
<td>RENT-IT, INC.</td>
</tr>
<tr>
<td>09-10-2004</td>
<td>950</td>
<td>VANDALAY INDUST.</td>
</tr>
</tbody>
</table>

#### Diagram:

- **NPSI on CUSTNO**
- **DPSI on CUSTNO**
- **Cannot be unique**
10. Don’t Expect All Third Party Tools to Immediately *Exploit* V8

Toleration versus Exploitation

- **Tolerate** – will run without abending
- **Exploit** – will allow you to use the new features
  - Most vendors are implementing “phased” exploitation
  - Make sure the vendors are *very clear* to you here
- **Use your list of important features to rate your tools vendor(s)**
- **Other questions to ask**
  - Explain what exploitation means?
    - If phased, explain the phases – what is supported when
    - It may be different product by product, so watch out!
  - What about the modes?
Areas to Watch Out for!

Unicode

- This should be in phase 1 for EVERY vendor, if not you should worry – the catalog is Unicode, after all!

Features that may be low on the exploit list:

- 4,096 partitions
  - Some vendors will phase in support for up to 1000 and support more later
- Multi-level security
- Sequence Objects
- MQTs
References

- DB2 UDB for z/OS Version 8: Everything You Ever Wanted to Know, ... and More (SG24-6079)
- DB2 UDB for z/OS Version 8 Technical Preview (SG24-6871)
- DB2 UDB for z/OS V8: Through the Looking Glass & What SAP Found There (SG24-7088)
- DB2 UDB for z/OS V8 Installation Guide (GC18-7418)
- DB2 UDB for z/OS V8 Administration Guide (SC18-7413)
- DB2 UDB for z/OS Version 8 Release Planning Guide (SC18-7425)
- DB2 for z/OS V8 Information Roadmap
- Links to articles, etc. about DB2 versions and migration issues
  http://www.db2portal.com/version.shtml