

5-9 November
Athens Hilton
Athens, Greece

Session: D12

DB2 9 Migration Tips and Tricks – Things Not Commonly Mentioned!

IDUG® 2007
Europe

Melanie Stopfer
IBM North American Lab Services

07 November 2007 • 16:15 – 17:15

Platform: DB2 LUW

INTERNATIONAL
DATA USER GROUP

GoFurther



Title:

DB2 9 Migration Tips and Tricks - Things Not Commonly Mentioned!

Abstract:

Are you responsible for migrating your database to DB2 9? Are you responsible for preparing applications, routines and scripts for migration? Are you looking for details about what has changed? If so, don't miss this session! Melanie will unveil tips and tricks about changes made to DB2 9 that are often overlooked but very important to understand before migrating. Planning your migration can be much easier if you understand things not commonly mentioned.

Outline:

- I. Planning your migration to DB2 9.
- II. Changes to development software and operating system support.
- III. Changes to DB2 packaging, installation and product components.
- IV. Changes to database creation behavior.
- V. Changes to registry variables, dbm and db cfg parameters.
- VI. Changes that affect DDL.
- VII. Changes to default database runtime behavior.
- VIII. Changes that affect applications.
- IX. Changes to utilities and tools.
- X. Changes to monitoring.

Objectives

After completing this session, you should be able to:


- Describe changes to packaging and software and operating system requirements.
- Understand changes to DDL, utilities, tools and monitoring.
- Configure database and instance to implement new features.
- Analyze changes that affect applications and database runtime behavior.
- Develop a migration strategy to DB2 9.

© Copyright IBM Corporation 2007

2

Objectives:

- Describe changes to packaging and software and operating system requirements.
- Understand changes to DDL, utilities, tools and monitoring.
- Configure database and instance to implement new features.
- Analyze changes that affect applications and database runtime behavior.
- Develop a migration strategy to DB2 9.



IBM Information Management software

- I. Planning Your Migration to DB2 9**
- II. Changes to development software and operating system support.**
- III. Changes to DB2 packaging, installation and product components.**

© Copyright IBM Corporation 2006
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

4.0.3

Let's look at planning Your Migration to DB2 9, changes to development software and operating system support, and changes to DB2 packaging, installation and product. components.

Migration Considerations

- If DB2 UDB Windows Version 8 installed, when run setup.exe command have two options:
 1. **New** - install DB2 9 as new copy and leave existing DB2 8 copy intact
 2. **Migrate** - migrates all instances, DAS, uninstalls existing DB2 V8 copy
 - Refer to the *Migrating DB2 servers (Windows)*
- Install migrate option **not** available on Linux and UNIX
 - See *Migrating DB2 servers (Linux and UNIX)*
- Can migrate from any fix pack
 - Consider applying latest DB2 UDB Version 8.2 fix pack
- For Migration to DB2 9 from DB2 UDB Version 8 FAQs see:
 - <http://www-1.ibm.com/support/docview.wss?uid=swg21239997>
- **Basic memory requirement typically increases between 0.5 and 1.5 MB per connection, depending on your operating system.**
- Restore command automatically runs MIGRATE DATABASE on full database **offline backups** created **on any level** of DB2 8 and **any instance bit size** (32 or 64).
- DB2 Version 9 migration roadmap – Great info – Check it out!
 - <http://www-1.ibm.com/support/docview.wss?uid=swg21227461>



© Copyright IBM Corporation 2007

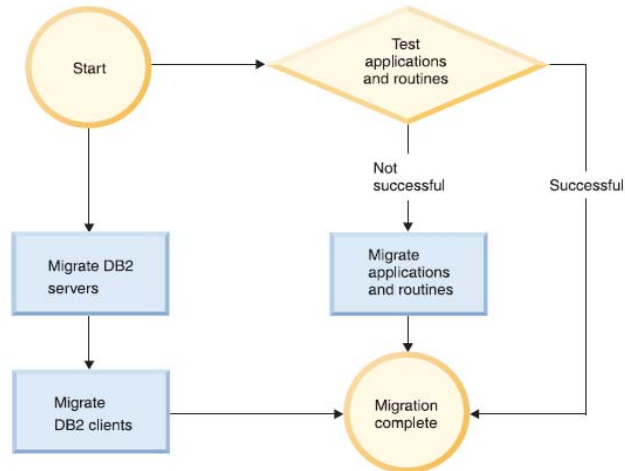
4

For Migration to DB2 9 from DB2 UDB Version 8 FAQs see:

<http://www-1.ibm.com/support/docview.wss?uid=swg21239997>

Migration from previous releases

- **db2ckmig** – checks for migration issues and logs concerns
- **db2imgr** implicitly runs **db2ckmig**
- **dasmigr** - migrates DAS
- **db2exmigr** - migrates explain tables and preserves data
- In DB2 9, support will only include migration from 1 past version
- A new section in the documentation outlines V7 to DB2 9 migration paths



- * **DB2 V8 clients can access DB2 9 servers**
- * **DB2 9 clients can access DB2 V8 servers**

© Copyright IBM Corporation 2007

5

Past DB2 versions have allowed instance migration from two previous versions:

- Support for V8 included migration from V6 and V7

In DB2 9, support will only include migration from one past version:

- Only DB2 V8 instances can be migrated to DB2 9 instances
- If db2imgr detects that the instance is not a V8 instance, an error message is displayed stating instances from versions older than V8 cannot be migrated to DB2 9
- DJ V2 customers must also migrate via DB2 V8

A new section in the Migration Guide documentation (in the Migrating environments with specific characteristics chapter) is included to outline V7 to DB2 9 migration paths which will include a description on the required two-step process (V7 to V8, then V8 to DB2 9).

The migration roadmap can be found at:

http://www-128.ibm.com/developerworks/db2/roadmaps/db2_migration_roadmap.html

Operating System Requirements

- Check operating system requirements at:
<http://www.ibm.com/software/data/db2/udb/sysreqs.html>
- Featured documents for DB2 for Linux, UNIX and Windows:
<http://www-1.ibm.com/support/docview.wss?uid=swg27006816>
- For latest information on supported Linux distributions, go to:
<http://www.ibm.com/db2/linux/validate>
- Before installing a DB2 database system, update Linux kernel parameters:
 - Default values for particular kernel parameters on Linux are not sufficient when running a DB2 database system.
 - To modify these kernel parameters, edit /etc/sysctl.conf file. If file does not exist, it should be created.
- See notes for details.

© Copyright IBM Corporation 2007

6

The following lines are examples of what should be placed into the /etc/sysctl.conf file for Linux:

```
kernel.sem=250 256000 32 1024
#Example shmmax for a 64-bit system
kernel.shmmax=1073741824
#Example shmall for 90 percent of 16 GB memory
kernel.shmall=3774873
kernel.msgmax=65535
kernel.msgmnb=65535
```

Run sysctl with -p parameter to load in sysctl settings from the default file /etc/sysctl.conf

Benefits of multiple DB2 copies

- Ability to install multiple DB2 9 copies on the same system
 - DB2 9 and V8.2 can coexist on the same machine
 - Each DB2 copy can be same or different Version or Fixpak level
 - Supported on all platforms
 - ability to install to any path, and have multiple copies
 - no need for providing multiple fixpaks
- Applications frequently require different versions of DB2 to be on the system at the same time
 - These applications may also need to run on the same machine
- Users can test on same machine before moving the production database to the latter version of the product
 - Smoother and less stressful upgrades for users
 - ISV's can now effectively hide DB2 from the rest of the system

db2swtch -l (lists installation copies on Windows)

db2swtch -d <copyname> (switches Windows copy)



© Copyright IBM Corporation 2007

7

Coexistence of multiple DB2 versions and fix packs (Linux and UNIX)

You can install multiple copies of DB2 database systems on Linux or UNIX operating systems without the need for alternate FixPaks. Key benefits of this feature include:

- **Install anywhere:** You can install DB2 database systems using any path that you choose.
- **Install any number of times:** You can install two or more copies of the same database system on one computer. The code levels can be different as well.
- **Service each copy independently:** You can update one copy without affecting any of the other copies.

You can use this feature in a number of ways:

- Database administrators can deploy one version of a DB2 database system for production, and a newer version to test new fix packs. Then, the production instance can be rolled over to the new installation path on an instance-by-instance basis, without having to apply the fix pack to the earlier version (although in-place fix packs are available). You can simply uninstall the earlier version when you no longer need it.
- Database administrators can have independent copies of DB2 database systems for different purposes. This independence allows different databases on the same computer to run at different fix pack levels. For example, one group, such as human resources, can apply fixes without affecting other groups, such as finance.
- If you are an independent software vendor, you can embed your own DB2 copy into your product.

Reference only: More notes

- Multiple DB2 Copies – See Notes

© Copyright IBM Corporation 2007

8

Multiple DB2 copies on the same computer (Windows)

With DB2 9, you can use multiple DB2 copies on the same computer. Each DB2 copy can be at the same or different code levels. The benefits of doing this include:

- The ability to run applications that require different DB2 versions on the same machine at the same time.
- The ability to run independent copies of DB2 products for different functions.
- The ability to test on the same computer before moving the production database to the latter version of the DB2 product.
- For independent software vendors, the ability to embed a DB2 server product into your product and hide the DB2 database from your users.

A DB2 copy can contain one or more different DB2 products. This refers to the group of DB2 products that are installed at the same location.

Differences when only one DB2 copy is installed:

- During installation, a unique Default DB2 copy name is generated, which you can later change.
- Applications use the Default DB2 copy in an environment similar to the DB2 Version 8 environment.

Differences when multiple DB2 copies are installed on the same computer:

- DB2 Version 8 can coexist with DB2 9, with restrictions described below.
- Optional: You can configure each DB2 copy to use a different Information Center.

Note: You can have only one copy of the DB2 Information Center installed on the same system at the same Release level. Specifically, you can have a Version 8 Information

Center and a DB2 9 Information Center, but you cannot have one Information Center at DB2 9 FixPak1 and another at DB2 9 fix pack 2 on the same machine. You can

however configure the DB2 database server to access these Information Centers remotely.

• Only the IBM DB2 .NET Data Provider from the Default copy is registered in the Global Assembly Cache. If Version 8 is installed with DB2 99, the IBM DB2 .NET 2.0 Provider from DB2 9 is also registered in the Global Assembly Cache. Version 8 does not have a 2.0 .NET provider.

• Each DB2 copy must have unique instance names. For a silent install with NO_CONFIG=YES, the default instance will not be created. However, when you create the instance after the installation, it must be unique. The name of the default instance will be the <DB2 copy Name>, if it is less than 8 characters. If it is more than 8 characters, or if an instance of the same name already exists, a unique name for the instance is generated to ensure uniqueness. This is done by replacing any characters that are not valid for the instance name with underscores and generating the last 2 characters. For performance reasons, the DB2 Control Center should only be used from one DB2 Copy at a single time on a machine.

Changes to querying the installation on Linux and UNIX

- Native UNIX/Linux OS utilities can no longer be used to query the DB2 installation (lspp, rpm, pkglist, swlist).
- Use new DB2 tool called **db2ls** to query the DB2 installation.
- Use DB2 command line help for syntax options.

```
root@localhost:~
File Edit View Terminal Tabs Help
[root@localhost ~]# db2ls
```

Install Path	Level	Fix Pack	Special	Install Number	Instal
/opt/ibm/db2/V9.0	9.0.0.0	0		0	Fri Ju

```
[root@localhost ~]#
```

© Copyright IBM Corporation 2007

9

Due to the nature of tar files, native UNIX/Linux OS utilities can no longer be used to query the DB2 installation.

Instead of using pkglist (Sun), rpm (Linux), lspp (AIX), or swlist (HP-UX), use a new DB2 tool called db2ls to query the DB2 installation.

The db2ls tool can retrieve information about the install path, DB2 version (including release and fixpack level), and installed features. In the past, different query tools and scripts were used for different platforms. With DB2 9, the same query tool and query scripts can be used across all UNIX and Linux platforms.

Changes to 32-bit support

- DB2 9 32-bit instance support
 - Linux on x86, Windows on x86, Windows on X64 (when using DB2 for Windows on x86 Install image)
- The following are **NOT SUPPORTED** as a 32-bit instance
 - AIX, Sun, HP PA-RISC, HP IPF, Linux on x86_64, Linux on POWER, Linux on zSeries
- The -w option of **db2icrt**, **db2ilist**, and **db2iupdt** commands is ignored



Impact on applications:

- 32-bit applications can run on a 64-bit instance as long as they are linked to 32-bit libraries on the instance. Simply link your applications to the DB2 libraries in \$INSTHOME/sqlib/lib32 where INSTHOME is your instance home directory, in order for them to run successfully.
- 32-bit applications previously linked to DB2 libraries need to be re-linked if they used the /sqlib/lib library path instead of /sqlib/lib32 library path.

© Copyright IBM Corporation 2007

10

With the exception of x86-based platforms, the emphasis is on 64-bit operating system support, although both 32-bit and 64-bit application support continues.

In response to growing business needs for faster, more scalable applications, IBM is prioritizing DB2 database server support on 64-bit hardware and operating systems, and phasing out 32-bit server support on several platforms.

64-bit server support delivers increased processing power, memory, and application performance.

In DB2 9, support for 32-bit instances is no longer available on UNIX operating systems and Linux (except for Linux x86). Only 64-bit instances can be created on these platforms for both newly created DB2 9 databases and migrated databases. The 64-bit instance supports local 32-bit applications and 32-bit server side routines, except for Linux on IPF which does not support any local 32-bit libraries. Support for 32-bit instances will remain for 32-bit Windows and 32-bit Linux operating systems.

The following topics describe how the 64-bit instance will affect DB2 commands, applications, and routines.

Impact on applications:

•32-bit applications can run on a 64-bit instance as long as they are linked to 32-bit libraries on the instance. Simply link your applications to the DB2 libraries in \$INSTHOME/sqlib/lib32 where INSTHOME is your instance home directory, in order for them to run successfully.

•32-bit applications that have previously been linked to DB2 libraries will need to be re-linked if they used the /sqlib/lib library path instead of /sqlib/lib32 library path.

Impact on server routines (UDFs, stored procedures):

•32-bit unfenced stored procedures and UDFs in any supported language are not supported on a 64-bit server.

To resolve this issue, drop and recreate these routines as FENCED, or move them to 64-bit if performance is a concern.

•64-bit unfenced stored procedures and UDFs continue to be supported on 64-bit systems.

•Java fenced stored procedures and UDFs are supported, but must use a 64-bit Java Virtual Machine. Java does not support unfenced Stored Procedures.

You must rebuild 32-bit routines that use LOB locators if you migrated from 32-bit V8 instance to a DB2 9 64-bit instance.

Impact on db2icrt, db2iupdt and db2ilist:

•The bit size for the instance can no longer be specified when you create or migrate an instance. Instead, it is determined by the DB2 9 installation image that you choose to install. As a result, the -w bit-width parameter of db2icrt, db2iupdt, and db2ilist will be ignored (but will be accepted for compatibility reasons).

Component and name changes (1 of 2)

- New names for DB2 9 product features:

- DB2 UDB is no longer used
- No “V” for Version

Previous feature name	DB2 9 feature name
IBM DB2 Universal JDBC Driver	IBM DB2 Driver for JDBC and SQLJ - 2 MB jar files
DB2 CLI and ODBC Driver	IBM DB2 Driver for ODBC and CLI - 2 MB driver files
DB2 Application Development Client and DB2 Administration Client	DB2 Client - Includes DB2 V8 Administration Client plus precompilers
DB2 Run-Time Client Lite and DB2 Run-Time Client	DB2 Runtime Client - This is DB2 Run-Time Client Lite – approximately 12 MB

© Copyright IBM Corporation 2007

11

The Application Development Client and the Administration Client have been combined into a single DB2 Client, which includes the application development tools. On Windows, you can now install the DB2 Client from a server installation image.

db2iprune is a command line utility that you can use to reduce the size of your DB2 Windows product installation image.

This tool is useful for large-scale deployments of DB2, as well as for embedding DB2 within an application. It consists of an input file and a db2iprune executable. The input file, or .prn file, contains a full list of removable components and is used to indicate which features and languages you would like removed from the installation image.

The db2iprune executable (db2iprune.exe) removes the cabinet (.cab) files associated with those features and languages. The result is a new, smaller DB2 installation image that can be installed using the regular DB2 installation methods.

Additional restrictions

- Linux on IA64 will NOT support **local** 32-bit client applications or 32-bit external routines. It will still support remote connections from 32-bit client applications.
- Co-existence of the Windows for x86 Install image and Windows for X64 install image on the same computer will NOT be supported.
- There is limited support for Version 7 Clients connecting to DB2 9 Database Servers.

Component and name changes (2 of 2)

- New names for DB2 9 product features:

Previous feature name	DB2 9 feature name
Development Center	Developer Workbench - A new separately installable feature of DB2 9
DB2 Add-in for Visual Studio .NET	IBM Database Add-Ins for Visual Studio 2005 -For DB2 9, this feature is no longer included with the DB2 client and server products, and must be installed separately
DB2 Geodetic Extender	DB2 Geodetic Data Management Feature

© Copyright IBM Corporation 2007

12

The DB2 Development Center has been replaced with an Eclipse-based tool called Developer Workbench (provided free of charge). The Developer Workbench has all existing functionality of the Development Center, but with some additional new features such as routine comparison and deployment tools, new binary deploy capability, SQLJ development support, new unified debugger for SQL or Java Stored Procedures, and XML support. A migration wizard is provided for migration of existing Development Center projects into the Developer Workbench.

Discontinued in DB2 9

- Data Links Manager
- Warehouse Manager
- Type 3 JDBC support
- Text, Audio, Image, and Video (AIV) Extenders
- DB2 DataJoiner
- Extended Storage option for buffer pools
- Database logging using raw devices is deprecated

Deprecated in DB2 9

- XML Extender
- CLISHEMA CLI keyword
- DB2 JDBC Type 2 driver is deprecated
 - Use the IBM DB2 Driver for JDBC and SQLJ instead.

There is no longer an excuse not to use DB2!

DB2 Express-C 12 Months License and Subscription

do it yourself

Free to build, deploy, distribute
pureXML included
Community assistance
Download from
www.ibm.com/db2/express

... or let IBM help


24*7 support
FixPaks & Upgrade Protection
SQL Replication
High Availability - HADR
... at a price that is competitive with
open source data servers

Learn more at
www.ibm.com/db2/express

© Copyright IBM Corporation 2007


13

DB2 Express-C is IBM's no charge full function data server. Available since January 2006, it has transformed the “free database” market. Unlike other so called “free” databases, DB2 Express-C has no crippling limits on the amount of data you can store, the number of databases per server or the number of users. And, this is not a marketing gimmick ... this is real DB2. The same DB2 that you are running in your shops, the same DB2 that has been on the top of performance charts longer then any other data server in the industry. The same DB2 that has world’s first and only hybrid relational/XML storage engine. And it is the same DB2 that has most the advanced autonomic capabilities that allows DBAs to spend time doing important tasks. DB2 Express-C is available for Linux and Windows servers and is optimized for 2 processors and 4GB of memory and ... it is absolutely free. You can download it free from <http://www.ibm.com/db2/express>



IBM Information Management software

IV. Changes to database creation behavior.



© Copyright IBM Corporation 2006
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

4.0.3

Let's look at changes to database creation behavior.

Blocking Public access to DB2 Catalogs

▪ CREATE DATABASE SALES ... **RESTRICTIVE**

The new RESTRICTIVE option on the CREATE DATABASE command can be used to create a database where no privileges are automatically granted to PUBLIC

- Access then need to be granted on a *need-to-know* basis
- Traditionally, when a database is created, DB2 grants SELECT access to the catalogs to PUBLIC
- Users may indirectly gain knowledge about some data values by accessing the catalogs!



(1/4)

© Copyright IBM Corporation 2007

15

The CREATE DATABASE command has a new option that prevents DB2 from granting many privileges to PUBLIC. The DB2 Catalogs contain, among other things, statistics about data distribution in a table, which are needed by the query optimizer to determine the best way to execute a query. Users may indirectly gain knowledge about some data values by accessing the catalogs.

If the RESTRICTIVE option is included on the CREATE DATABASE command, it causes the RESTRICT_ACCESS database configuration parameter to be set to YES and no privileges are automatically granted to PUBLIC.

If the RESTRICTIVE option is not present then the RESTRICT_ACCESS database configuration parameter is set to NO and all of the following privileges are automatically granted to PUBLIC.

- CREATETAB
- BINDADD
- CONNECT
- IMPLSCHEMA
- EXECUTE with GRANT on all functions and procedures in schema SYSPROC
- BIND on all packages created in the NULLID schema
- EXECUTE on all packages created in the NULLID schema
- CREATEIN on schema SQLJ
- CREATEIN on schema NULLID
- USE on table space USERSPACE1
- SELECT access to the SYSIBM catalog tables
- SELECT access to the SYSCAT catalog views
- SELECT access to the SYSSTAT catalog views
- UPDATE access to the SYSSTAT catalog views

DB2 9 Database Storage Defaults

- When creating a default database in DB2 9, you get a database with:

- **Automatic storage**

- Single Point of Storage Management for all Table spaces
- No explicit container definitions provided
- Containers automatically created across storage paths
- Default for all databases including DPF

```
CREATE DATABASE tp1 AUTOMATIC STORAGE YES  
ON sp1, sp2, sp3 DBPATH ON dbp1
```

```
CREATE TABLESPACE tbspx
```

- Can override with

```
CREATE DATABASE tp1  
AUTOMATIC STORAGE NO  
DBPATH ON dbp1
```



© Copyright IBM Corporation 2007

16

By creating a database with AUTOMATIC STORAGE, the need for constant monitoring of space is eliminated. If maintenance needs to be performed on the storage spaces, storage paths can be added as needed.

In DB2 9, automatic storage support is added for multi-partition databases. In addition, the Control Center has been enhanced to let you create databases that use automatic storage, and to enable you to add storage paths to existing databases.

If you have the Enterprise Server Edition with the Database Partitioning Feature (DPF), you may now use multi-partition databases that support automatic storage.

Also, you can specify paths in addition to drive letters when creating a database in a Windows operating environment.

You can create an automatic storage database or add a storage path to an existing automatic storage database using the Control Center.

To add a storage path to an existing database using the Control Center:

1. Open the Add Storage window: Expand the object tree until you see the Table Spaces folder of the database to which you want to add a storage path.
2. Right-click the Database and select Manage Storage --> Add Automatic Storage from the pop-up menu. The Add Storage window opens.
3. Click Add. The Add Storage Path window opens.
4. Specify the storage path.

Auto-resize for DMS table spaces

- DB2 9 can automatically increase container file size. Default:
 - YES for AUTOMATIC STORAGE table spaces
 - NO for DATABASE MANAGED STORAGE table spaces
 - Avoids -289 SQLCODE, if no space on file system
 - Defaults per database partition:
 - INITIALSIZE=32 MB,
 - INCREASESIZE=32 MB and MAXSIZE=NONE
 - Containers must all be files - no raw devices

```
ALTER/CREATE TABLESPACE <tsName>
MANAGED BY DATABASE
USING (<listOfFileContainers>)
[INITIALSIZE integer]
[AUTORESIZE {NO|YES}]
[INCREASESIZE integer
{PERCENT|K|M|G}]
[MAXSIZE {NONE | integer {K|M|G}}]
```



© Copyright IBM Corporation 2007

17

DMS table spaces are made up of file containers or raw device containers, and their sizes are set when the containers are assigned to the table space. The table space is considered full when all of the space within the containers has been used. However, unlike SMS, you can add or extend containers using the ALTER TABLESPACE statement, allowing more storage space to be given to the table space. DMS table spaces also have a feature called auto-resize.

As space is consumed in a DMS table space that can be automatically resized, the DB2 database system might extend one or more file containers. SMS table spaces have similar capabilities for growing automatically but the term auto-resize is used exclusively for DMS.

Changes to default table spaces

- DMS table space default type changed to LARGE
 - 6 byte RID (4 bytes for data page + 2 bytes for slot)
 - Table size limits increased to 1.1 trillion rows
 - 4K pg=2 TB, 8K pg=4 TB, 16K pg=8 TB, 32K pg=16 TB
- Userspace1 default to:
 - DMS, Large RID, AUTORESIZE ON
- SYSCATSPACE default to:
 - DMS, Small RID (3 bytes for page + 1 byte for slot), AUTORESIZE ON
- TEMPSPACE1 default to:
 - SMS, Large RIDs
 - Removes limit of 255 rows per temporary data page
- Default page size larger than 4 KB may be set for default buffer pool and table spaces when creating new database
CREATE DATABASE ... PAGESIZE 8K

© Copyright IBM Corporation 2007

18

In DB2 9, temporary tables now support larger row identifiers (RIDs) by default, and the previous limit of 255 rows per temporary data page no longer exists. In previous DB2 versions we recommended a small page size for temporary tables (to support efficient consumption of the data page space), but due to new large RID support, this is no longer the case for DB2 9.

The maximum size of a large table space is 16 TB which is enabled by larger record identifiers (RID).

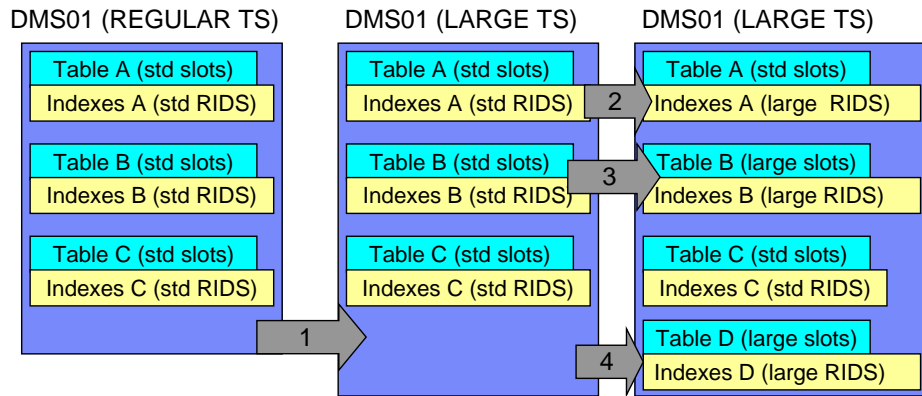
The system table space (SYSCATSPACE) is Automatic Storage DMS. The user table space (USERSPACE1) is Automatic Storage DMS, and Temp table space (TEMPSPACE1) is Automatic Storage SMS.

For some applications with large average row sizes, the default page size of 4 KB for the default buffer pool and table spaces might not be sufficient for all tables. In DB2 9 database you can increase the page size at database creation to 8 KB or 16 KB to allow wider rows to be stored on a page and allow the application table spaces to share the default buffer pool with the default system table spaces. To increase the default page size, set the PAGESIZE option of CREATE DATABASE to the new value.

For example: CREATE DATABASE mydb PAGESIZE 16 K

The default page size represents the default page size for all future CREATE BUFFERPOOL and CREATE TABLESPACE statements. By having a uniform page size across the database, administration is simplified because you only have to maintain a single bufferpool, and even if there are multiple bufferpools, they could be a consistent page size if all table spaces were the same page size. DB2 9 continues to support four different page sizes: 4 KB, 8 KB, 16 KB, and 32 KB.

Enabling Large RIDs after Alter Tablespace ... CONVERT



1. ALTER TABLESPACE DMS01 CONVERT TO LARGE
2. REORG ALL INDEXES FOR TABLE A ALLOW READ ACCESS
3. REORG TABLE B ALLOW READ ACCESS
4. CREATE TABLE D.....CREATE INDEXON D.....

© Copyright IBM Corporation 2006

19

This graphic shows a regular DMS table space containing three tables. All of the tables begin with standard 4-byte RIDs for the data and index components. **Reorganizing the table automatically rebuilds all indexes, converting any Type-1 indexes to Type-2 indexes and all indexes would be large RID indexes, thus allowing both large slots and large page numbers.**

An ALTER TABLESPACE is used to convert the table space to a large table space, capable of supporting large RIDs. The statement would be:

ALTER TABLESPACE DMS01 CONVERT TO LARGE

This would allow the table space to be extended beyond the size limits for regular DMS table spaces. The existing tables and indexes, still have 4-byte RIDs.

Next a REORG is used to reorganize the indexes for one of the tables as follows:

REORG ALL INDEXES FOR TABLE A ALLOW READ ACCESS

This would rebuild all of the indexes on table A with large RIDs that could support a larger number of pages for the table. Since this was an index reorganization, the data component was not rebuilt and the data pages would continue to have 1-byte slots with the standard limits for rows per page.

Next an offline table reorganization is used to rebuild table B and all of its indexes as follows:

REORG TABLE B ALLOW READ ACCESS

Table B would now be enabled for large slots, which would allow more data rows per page and the indexes would allow for a larger number of total data pages. Any new tables created after the table space is converted to a large table space would automatically be enabled for large RIDs and large slots.

Using either the IMPORT utility or the LOAD utility with the REPLACE option will empty the table and the new pages will be enabled for large slots.

IBM Information Management software

V. Changes to registry variables, dbm and db cfg parameters

© Copyright IBM Corporation 2006
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

4.0.3

Let's look at changes to registry variables, dbm and db cfg parameters.

Automatic Configuration for Database options

- **AVG_APPLS = AUTOMATIC**
 - Set to 3 if DB2_WORKLOAD=SAP and MAXAPPLS > 3
 - Otherwise, set to 1 (DB2 8 default)
- **NUM_IOSERVERS = AUTOMATIC**
 - Maximum level of parallelism for any single table space
 - For SMS, based on number of containers
 - For DMS, based on maximum containers in any stripe set
 - Multiplied by setting for DB2_PARALLEL_IO
 - **V8.2 default:** `db2set DB2_PARALLEL_IO=*,6`

Number of prefetchers = max(max over all table spaces(parallelism setting * [SMS: # containers; DMS: max # containers in stripe set]), 3)
- **NUM_IOCLEANERS = AUTOMATIC**
 - Based on the number of CPUs configured – ensures that number of cleaners will be less than the number of CPUs
 - Adjusted for Number of Logical Database Partitions (DPF)

Number of page cleaners = max(ceil(# CPUs / # local logical database partitions) - 1, 1)

© Copyright IBM Corporation 2007

21

The database configuration options AVG_APPLS, NUM_IOSERVERS, and NUM_IOCLEANERS can be set to AUTOMATIC to allow DB2 set values based on the database and system configuration at the time the database is started.

avg_appls - Average number of active applications

Configuration Type Database

Parameter Type Configurable Online

Propagation Class Statement boundary

Default [Range] Automatic [1 – maxappls]

Unit of Measure Counter This parameter is used by the query optimizer to help estimate how much buffer pool will be available at run-time for the access plan chosen.

By default, this parameter is set to Automatic for databases created in IBM DB2 9.1. When *avg_appls* is set to AUTOMATIC, the optimizer uses a value of 3 if the registry variable DB2_WORKLOAD=SAP and if the value of *max_appls* is greater than 3. Otherwise, when *avg_appls* is set to AUTOMATIC, the optimizer uses a value of 1.

num_ioservers - Number of I/O servers

Configuration Type: Database

Parameter Type: Configurable

Default [Range] Automatic [1 – 255]

Unit of Measure Counter

When Allocated When an application connects to a database

When Freed When an application disconnects from a database

I/O servers, also called prefetchers, are used on behalf of the database agents to perform prefetch I/O and asynchronous I/O by utilities such as backup and restore. This parameter specifies the number of I/O servers for a database. No more than this number of I/Os for prefetching and utilities can be in progress for a database at any time. An I/O server waits while an I/O operation that it initiated is in progress. Non-prefetch I/Os are scheduled directly from the database agents and as a result are not constrained by *num_ioservers*.

If this parameter is set to AUTOMATIC, the number of prefetchers started will be based on the parallelism settings of the table spaces in the current database partition. (Parallelism settings are controlled by the DB2_PARALLEL_IO environment variable.) For each DMS table space, the value of this parallelism setting will be multiplied by the maximum number of containers in the table space stripe set. For each SMS table space, the value of this parallelism setting will be multiplied by the number of containers in the table space. The largest result over all table spaces in the current database partition will be used as the number of prefetchers to start. There will always be at least three prefetchers started when this parameter is set to AUTOMATIC.

Reference Notes, Continued

- Automatic Configuration for Database options – See Notes

© Copyright IBM Corporation 2007

22

When this parameter is set to AUTOMATIC, the number of prefetchers to start will be calculated at database activation time based on the following formula:

number of prefetchers = max(max over all table spaces (parallelism setting * [SMS: # containers; DMS: max # containers in stripe set]), 3)

num_jocleaners - Number of asynchronous page cleaners

Configuration Type Database
Parameter Type Configurable
Default [Range] Automatic[0 – 255]
Unit of Measure Counter

This parameter allows you to specify the number of asynchronous page cleaners for a database. These page cleaners write changed pages from the buffer pool to disk before the space in the buffer pool is required by a database agent. As a result, database agents should not have to wait for changed pages to be written out so that they might use the space in the buffer pool. This improves overall performance of the database applications. If you set the parameter to zero (0), no page cleaners are started and as a result, the database agents will perform all of the page writes from the buffer pool to disk. This parameter can have a significant performance impact on a database stored across many physical storage devices, since in this case there is a greater chance that one of the devices will be idle. If no page cleaners are configured, your applications might encounter periodic log full conditions.

If this parameter is set to AUTOMATIC, the number of page cleaners started will be based on the number of CPUs configured on the current machine, as well as the number of local logical database partitions in a partitioned database environment. There will always be at least one page cleaner started when this parameter is set to AUTOMATIC.

The number of page cleaners to start when this parameter is set to AUTOMATIC will be calculated using the following formula:

number of page cleaners = max(ceil(# CPUs / # local logical database partitions) – 1, 1)

This formula ensures that the number of page cleaners is distributed almost evenly across your logical database partitions, and that there are no more page cleaners than there are CPUs.

AUTOCONFIGURE during CREATE DATABASE

- Configuration Advisor enabled on newly created DB2 9 database by [default](#)
- database configuration parameter settings will likely be different from the default configuration parameter settings from earlier DB2 versions
 - existing applications/scripts that have dependencies on the old, hard-coded db cfg parameters at database creation may be affected

To disable:

```
db2set DB2_ENABLE_AUTOCONFIG_DEFAULT=NO
```



© Copyright IBM Corporation 2007

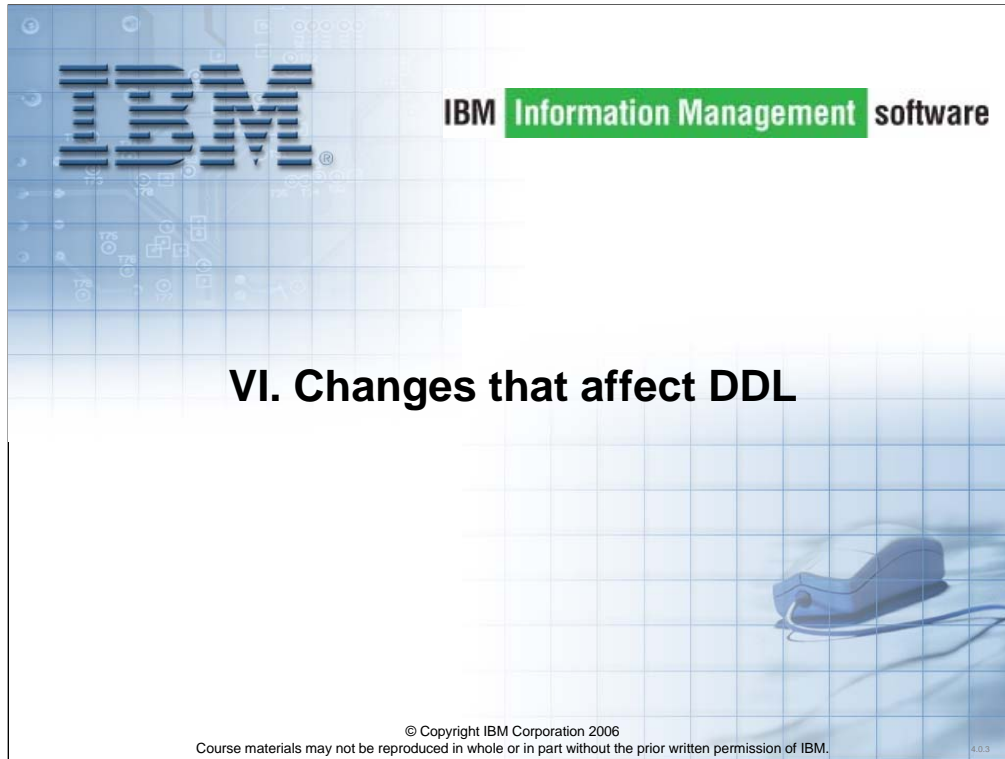
23

When the CREATE DATABASE command is issued, the Configuration Advisor also runs automatically. This means that the database configuration parameters are automatically tuned for you according to your system resources. To disable the Configuration Advisor from running at database creation, please refer to the `db2_enable_autoconfig_default` registry variable.

Autoconfigure calculates and displays initial values for the buffer pool size, database configuration, and database manager configuration parameters, with the option of applying these recommended values. Running the AUTOCONFIGURE command on a database will recommend enablement of the Self Tuning Memory Manager. However, if you run the AUTOCONFIGURE command on a database in an instance where SHEAPTHRES is not zero, sort memory tuning (SORTHEAP) will not be enabled automatically. To enable sort memory tuning (SORTHEAP), you must set SHEAPTHRES equal to zero using the UPDATE DATABASE MANAGER CONFIGURATION command. Note that changing the value of SHEAPTHRES may affect the sort memory usage in your previously existing databases.

If the Configuration Advisor is enabled on a newly created DB2 9 database, the database configuration parameter settings will likely be different from the default configuration parameter settings from earlier DB2 versions. Consequently, any existing applications or scripts that have dependencies on the old, hard-coded database configuration parameters at database creation time may be affected by this change.

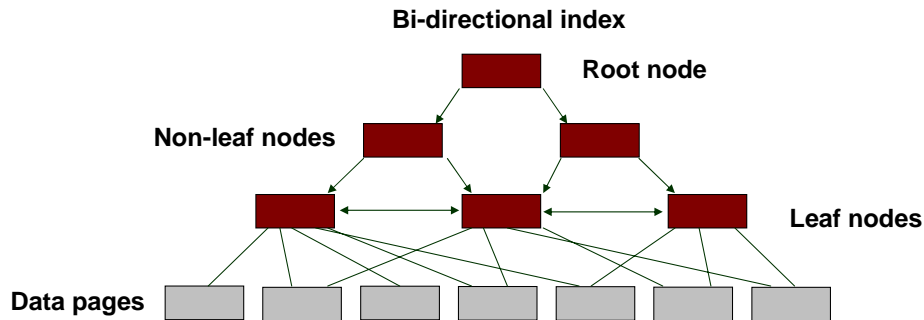
When the CREATE DATABASE command is issued, the Configuration Advisor also runs automatically. This means that the database configuration parameters are automatically tuned for you according to your system resources. To disable the Configuration Advisor from running at database creation, please refer to the `db2_enable_autoconfig_default` registry variable. The table in the graphic shows the options for the autoconfigure advisor and the defaults that will be used during database creation if no options are included with the CREATE DATABASE command.



Let's look at changes that affect DDL.

Reverse scans enabled by default

- New primary keys, unique keys and indexes will **ALLOW REVERSE SCANS** by default
 - When one forward scan index and one reverse scan index is created, the two indexes can be replaced with a single one that is enabled for reverse scans.
 - The second index will not be created and DB2 will issue a duplicate index warning message.



25

In DB2 9, all new primary keys, unique keys and indexes (except extended index) will ALLOW REVERSE SCANS by default. Consequently, the access plan may change and query execution times may improve because the optimizer may be able to use the reverse index scan in some SQL statements. This feature also leads to improved index maintenance.

Attention:

If you create two indexes on the same table, one specifying ASC and the other DESC, and if you do not specify the DISALLOW REVERSE SCANS option in the CREATE INDEX statement, the two indexes will default to ALLOW REVERSE SCANS. As a result, the latter index will not be created and DB2 will issue a duplicate index warning message.

Extended index (user-defined extended index types)

To support user-defined index types, the DB2 database manager allows you to create and apply your own logic for the primary components that make up how an index works. Those components that can be substituted are:

- Index maintenance. This allows the ability to map index column content to an index key. Such a mapping is done through a user-defined mapping function. Exactly one structured type column can participate in an extended index. Unlike an ordinary index, an extended index might have more than one index entry per row. Multiple index entries per row could enable a text document to be stored as an object with a separate index entry for each keyword in the document.

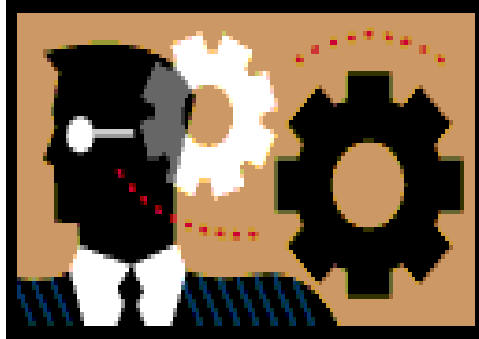
- Index exploitation. This enables the application designer to associate filtering conditions (range predicates) with a user-defined function (UDF) that would otherwise be opaque to the optimizer. This enables DB2 to avoid making a separate UDF call for each row, and thereby avoids context switching between client and server, greatly improving performance.

Increased identifier length limits

Identifier length limits increased from 18 bytes to 128 bytes.

▪ Objects that allow up to 128-byte identifiers include:

- Indexes
- Index specifications
- Methods
- Routine-specific names
- User-defined functions (UDFs)




© Copyright IBM Corporation 2007

26

The ability to increase identifier limits from DB2 V8 18 byte limit to 128-byte DB2 9 enhancement enables you to specify more descriptive names for database objects.

Alter Table command

- In DB2 9, the ALTER command now allows:
 - drop column, alter column type, alter column nullability
 - After 3 ALTERs on same table, requires REORG 
 - Table in READ-ONLY status until REORG

```

>>-ALTER TABLE--table-name----->
.-----
v .-----
>+--+ALTER-----+--+COLUMN-.-----| column-definition |----->
|      |-----+--+COLUMN-.-----+--+CASCADE--.-----|
|      |-----+--+column-name-----+--+RESTRICT-'-----|
|      |-----+--+DATA TYPE--altered-data-type+----->
|      |-----+--+NOT NULL-----+-----|
|      |-----+--+DROP-----NOT NULL-----+-----|

```

© Copyright IBM Corporation 2007

27

You can now use the ALTER TABLE statement to:

- Drop a column, using the new DROP COLUMN clause
- Change a column type, using the ALTER COLUMN SET DATA TYPE clause
- Change the nullability attribute of a column, using the SET NOT NULL or the DROP NOT NULL clause

When changing these table attributes using SQL, it is no longer necessary to drop the table and then recreate it, a time consuming process that can be complex when object dependencies exist.

A table alteration that affects the row format of the data is known as a REORG-recommended operation, and requires table reorganization before most subsequent operations on the table are allowed.

When an alter of any of these types is performed, the table is updated to the new design, but only SELECT scans will be allowed (no inserts or updates) until a REORG is completed.

- No more than three separate ALTERs will be allowed before a REORG will be required.

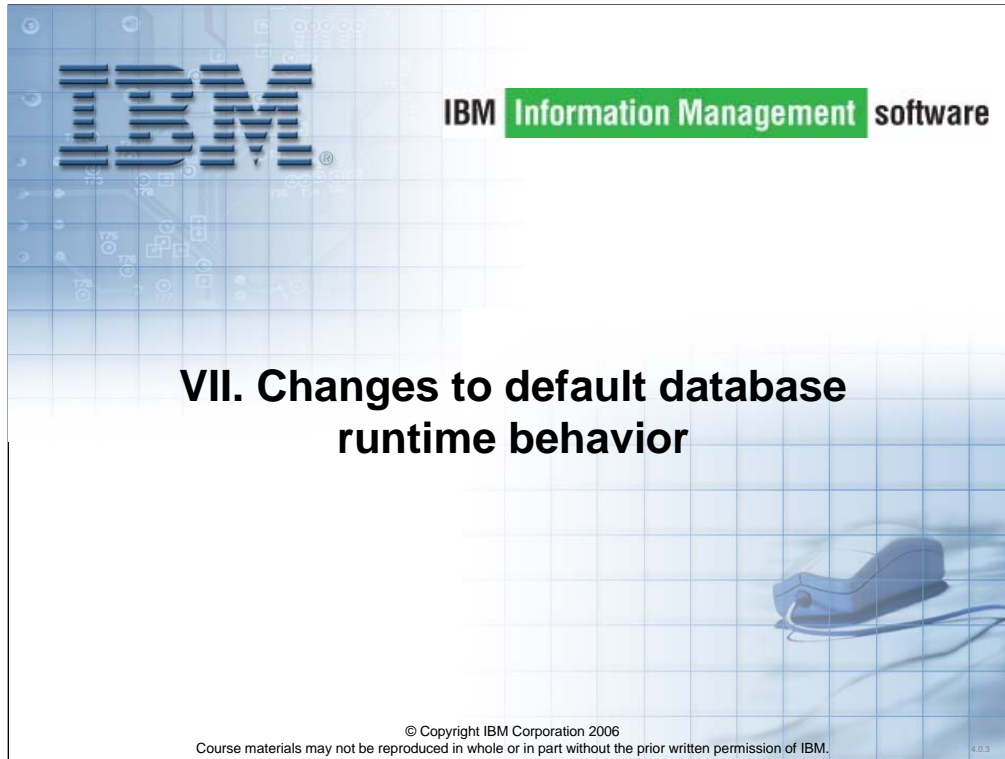
A trigger is dependent on the column if it is referenced in the UPDATE OF column list, or anywhere in the triggered action.

DROP CASCADE is the default behavior. It specifies that any views, indexes, triggers, SQL functions, or constraints that are dependent on the column being dropped are also dropped.

If an SQL function is a dependency of another database object, it may be unable to be dropped via the Cascade option (drop function cascade is not supported at present).

Note that a trigger is dependent on the column if it is referenced in the UPDATE OF column list or anywhere in the triggered action. The first dependent object detected is logged to the administrator log.

DROP RESTRICT specifies that the column cannot be dropped if any views, indexes, triggers, or constraints are dependent on the column (SQLSTATE 42893).



Let's look at changes to default database runtime behavior.

The Self Tuning Memory Manager in DB2 9.1 can manage the Database shared memory areas that directly impact application performance. Multiple database buffer pools can be difficult to optimize, so STMM can be used to increase or decrease the size of buffer pools based on the current workload.

Shortages in locking memory can cause lock escalations which can impact concurrency and application performance, so STMM can be used to increase or decrease the memory available for locks. The database configuration parameters locklist and maxlocks can be set by STMM and can be adjusted when the demand for lock memory changes.

Another type of memory that can have a big impact on application performance is sort memory, which is used to support Hash Joins, Merge Joins, Index Anding as well as normal internal sort operations. The configuration parameters sheapthres_shr and sortheap can be set by STMM and adjusted when the demand for sort memory changes.

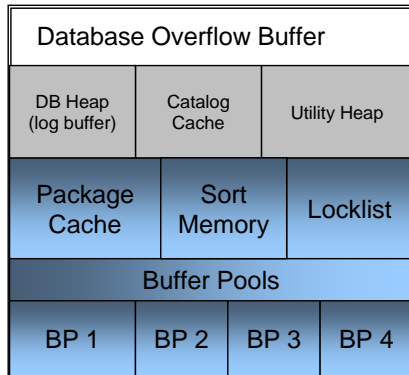
STMM can also manage the database package cache size, which can vary in size depending on the usage of static and dynamic SQL by the applications currently using a DB2 database.

Some areas of database shared memory are not supported for management by STMM. The database utility heap, defined by util_heap_sz, is critical to the performance of DB2 utilities like Backup, Restore and the LOAD utility. Since these utilities do not run continuously it would be difficult for STMM to predict the correct amount of memory needed for the utilities. The LOAD utility was designed to 'self-optimize' its performance by acquiring a percentage of the currently available memory in the utility heap when it start processing. A similar automatic buffer calculation was also added for Backup and Restore utilities. Since STMM can not know when these utilities are scheduled to run, it will be necessary for a DBA to configure the size of the utility heap, which can be changed using DB2 commands. A DBA might decide to take memory from a large buffer pool and assign it to the utility heap just before the scheduled LOAD processing starts.

The database heap size and catalog cache are also not currently supported for automatic management by STMM.

Database shared memory in DB2 9

Total Database Shared Memory defined by DATABASE_MEMORY



Automatic Management:

- Pckcachesz=AUTOMATIC
- Sheapthres_shr=AUTOMATIC
- Sortheap=AUTOMATIC
- Sheapthres=0
- Locklist=AUTOMATIC
- Maxlocks=AUTOMATIC
- Create/Alter Bufferpool ...
size AUTOMATIC
- **Self_Tuning_Mem=ON**

© Copyright IBM Corporation 2007

29

Is on by default for newly created DB2 9.1 databases

New configuration parameter SELF_TUNING_MEM must be set to ON

```
update db cfg for database <db_name> using self_tuning_mem on
```

Set each parameter that you wish to tune to AUTOMATIC

```
update db cfg for database <db_name> using locklist automatic
```

In the buffer pool case, an alter bufferpool command is necessary

```
alter bufferpool ibmdefaultbp size automatic
```

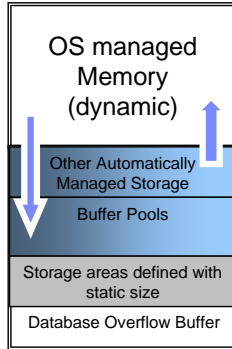
Or for new buffer pools

```
create bufferpool <bp_name> size automatic
```

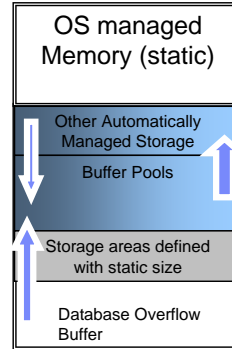
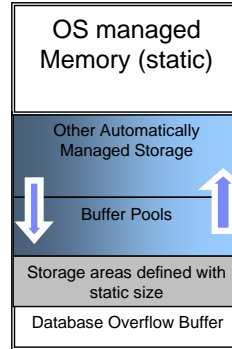
Feature can be turned on dynamically

DATABASE_MEMORY = Automatic/Computed/value

DATABASE_MEMORY=AUTOMATIC
(Default AIX, Windows DB2 9 databases)



DATABASE_MEMORY=value



DATABASE_MEMORY=Computed
(Linux, HP-UX, Solaris DB2 9 default.
Migrate value if DB2 V8=Automatic)

(1/2)

© Copyright IBM Corporation 2007

30


This shows examples of the three different ways to set the database configuration parameter DATABASE_MEMORY.

When DATABASE_MEMORY is set to AUTOMATIC on the supported platforms, the overall size of database shared memory can grow or shrink based on activity in the database and also the demands of other databases and applications running on the system. So additional operating system memory might be added to buffer pools to improve performance. Later, some memory might be released from the sort memory and made available to the operating system.

When DATABASE_MEMORY is set to COMPUTED, DB2 would calculate the size of database shared memory and reserve 20% for the overflow buffer. The size database shared memory would not grow or shrink. Memory could be exchanged between heaps that are set for automatic management. So an active buffer pool might take some memory from sort memory and later an inactive buffer pool might give some its memory to the locklist.

Setting DATABASE_MEMORY to a value is similar to setting it to COMPUTED, in that STMM would not be exchanging memory with the operating system and overall database memory size would be static. This mode of operation would be useful when the current settings for the memory heaps are relatively small but a target memory size for the database has been decided on. This could be used when several databases are sharing a system and the DBA wants to assign some fixed portion of the system memory based on the relative priority of the application. For example, two production databases are sharing a large system with several test databases. If all of these databases had DATABASE_MEMORY set to AUTOMATIC, a test database might acquire a much larger portion of the system's memory resources than one of the production databases. In this case, the DBA could estimate a maximum allocation for each database and set DATABASE-MEMORY to that size.

STMM and Sort Memory

- STMM can tune Sort Memory if SHEAPTHRES is set to 0
 - All sorts will use database shared memory defined by SHEAPTHRES_SHR
 - If SHEAPTHRES_SHR is AUTOMATIC then overall sort memory managed by STMM and SORTHEAP must also be set to AUTOMATIC
 - If SHEAPTHRES_SHR is not AUTOMATIC then overall sort memory is fixed but SORTHEAP may be set to AUTOMATIC
 - Changes in SORTHEAP can now effect active sorts
- No STMM tuning for sorts if SHEAPTHRES is NOT 0 
 - If no STMM, DB2 only uses database shared sort memory if SMP parallelism or Connection Concentrator are enabled
 - Private and Shared Sort Memory managed using the V8.2 model

© Copyright IBM Corporation 2007

31

Sort memory is allocated for several important types of processing, including Hash Joins, Merge Joins, Multiple Index Anding as well as sort operations for running SQL functions like ORDER BY and GROUP BY and CREATE INDEX. STMM can be used to manage the total amount of memory available for these functions as well as the setting of SORTHEAP, which regulates the memory available for each sort-memory based function.

For STMM to automatically manage the memory used for sorting the DBM configuration parameter, SHEAPTHRES needs to be set to 0. This implies that all sorts running in the instance will use the database shared memory defined by SHEAPTHRES_SHR.

If SHEAPTHRES_SHR is set to AUTOMATIC, then STMM will manage the shared sort memory for the database and the total memory available for sort operations may grow or shrink depending on the application workload. If SHEAPTHRES_SHR is set to AUTOMATIC, then SORTHEAP, which defines the maximum memory size for a single sort operation, must also be set to AUTOMATIC. This helps to avoid the following conditions:

1. The available sort memory is underutilized because SORTHEAP is small and there are not many concurrent sorts.
2. The available sort memory is over-utilized, either because SORTHEAP is too large or there are a large number of concurrent sorts.

If SHEAPTHRES_SHR is NOT set to AUTOMATIC, STMM can still be used to dynamically set an optimal size for SORTHEAP, if SORTHEAP is set to AUTOMATIC.

If the DBM configuration parameter SHEAPTHRES is set to a value other than 0, then sort memory will be handled as it was before DB2 9.1 as follows:

- The setting of SHEAPTHRES will be used as a soft limit for all private sort memory for the instance.
- If SMP Parallelism or Connection Concentrator are in use then some sorts may use Database Shared memory instead of private agent memory. The database configuration parameter SHEAPTHRES_SHR can set a hard limit for the total shared database memory available for sorts.

Automatic statistics collection (Auto RUNSTATS)

Automatically runs the RUNSTATS utility in the background

- Ensures the correct statistics are collected and maintained
- If create DB2 9 database, **automatic RUNSTATS enabled**.
 - RUNSTATS utility automatically scans tables in background on a periodic basis to ensure statistics are kept up-to-date for optimizer
- Databases migrated from a previous DB2 version are unaffected.
- To turn off auto RUNSTATS, issue:
db2 update db cfg for sample using auto_runstats off

- **db2set DB2_AUTOMATION_DEFAULT**
 - =**STMM_OFF**, disables STMM
 - =**AUTO_RUNSTATS_OFF**, disables AUTORUNSTATS
 - =**V82**, no STMM, no AUTORUNSTATS, and no AUTOCONFIGURE



© Copyright IBM Corporation 2007

32

To turn off auto RUNSTATS, issue the database update command:

```
db2 update db cfg for sample using auto_runstats off
```

Or change the AUTO_RUNSTATS configuration parameter using the Control Center.

Automatic statistics profiling advises when and how to collect table statistics.

Can be enabled or disabled by using the **auto_stats_prof**, **auto_tbl_maint** and **auto_maint** database configuration parameters.

Fast Communication Manager (FCM) Enhancements

- Separate sender and receiver communications daemons **improve** communication speed
- NUMA partitioning of FCM resources by logical partition now supported on any platform that offers per-partition affinity
- Number of FCM buffers and FCM channels can be dynamically changed

Configuration parameters set to AUTOMATIC by default

- **fcm_num_buffers** – tunes number of FCM buffers
- **fcm_num_channels** – tunes number of FCM channels

New FCM monitor elements

- **ch_free** monitor element indicates the number of inter-node communication channels that are currently free
- **ch_free_bottom** monitor element indicates the lowest number of free inter-node communication channels reached during processing
- **db2pd –fcm** command has new high water mark (**hwm** option)

© Copyright IBM Corporation 2007

33

The re-architecture of the DB2 communications subsystem has resulted in several enhancements involving Database Partitioning Feature (DPF) servers.

Separate sender and receiver communications daemons are now used to improve communication speed.

- NUMA partitioning of FCM resources by logical partition is now supported on any platform that offers per-partition affinity.
- The number of FCM buffers and FCM channels can be dynamically changed during execution, eliminating the need for tuning. There are several other dynamic configuration improvements that automatically monitor resource usage without your involvement.

These enhancements affect several configuration parameters and monitor elements.

- The `fcm_num_buffers` parameter has a new AUTOMATIC setting that allows the DB2 database system to attempt to tune the configured parameter value if the database is not making full use of resources. This setting is turned on by default.
- The new `fcm_num_channels` configuration parameter specifies the number of FCM channels. It replaces the deprecated `fcm_num_rqb`, `fcm_num_anchors`, and `fcm_num_connect` parameters. This parameter is set to AUTOMATIC by default.

Two new monitor elements replace a number of deprecated ones.

- The new `ch_free` monitor element indicates the number of inter-node communication channels that are currently free. It replaces the deprecated `ma_free`, `ce_free`, and `rb_free` monitor elements.
- The new `ch_free_bottom` monitor element indicates the lowest number of free inter-node communication channels reached during processing. It replaces the deprecated `ma_free_bottom`, `ce_free_bottom`, and `rb_free_bottom` monitor elements.

The deprecated monitor elements no longer appear in the dynamic snapshot streams in FCM snapshots.



Let's look at changes that affect applications.

C++ Runtime Requirements for DB2 9

- To ensure that C, C++, or COBOL routines that you created prior to DB2 9 function successfully, you must migrate them.
- Migration only applies to external routines programmed in C/C++, and COBOL (procedures only).
- Use one of following:
 - **xIC.rte 8.0.0.0, xIC.50.rte 8.0.0.0**
or
 - **xIC.rte 8.0.0.4 and xIC.aix50.rte 8.0.0.4**
- See notes for details

© Copyright IBM Corporation 2007

35

To ensure that C, C++, or COBOL routines that you created prior to DB2 9 function successfully, you must migrate them. This procedure only applies to external routines programmed in C/C++, and COBOL (procedures only).

To migrate a C, C++, or COBOL routine to DB2 Version 9, do the following:

If you identify changes in DB2 Version 9 that impact your routines, edit your routine code and modify:

SQL statement syntax

SQL statements using SQL Administrative views and routines, and catalog views

If you migrated to a DB2 Version 9 64-bit instance, change your routine libraries or routine definitions according to the following.

Unfenced 32-bit Routine

Define the routine as both fenced and not threadsafe using the ALTER PROCEDURE or ALTER FUNCTION statement with the FENCED clause and the NOT THREADSAFE clause. You can not use this option if LOB locators are referenced in the routine.

Rebuild the routine source code into a 64-bit library using the DB2 Version 9 bldrtn script and redeploy the library to the DB2 server. If LOB locators are referenced in the routine, you must use this option. You can determine which routines reference lob locators by executing the following query:

```
SELECT DISTINCT a.routineschema, a.routinename, a.specifcname FROM  
syscat.routines a, syscat.routineparms b WHERE a.specifcname = b.specifcname  
AND b.locator='Y' AND a.fenced='N'
```

An advantage of this approach is that using a 64-bit library results in better routine runtime performance than using a 32-bit library.

Reference Notes Continued

- C++ Runtime Requirements for DB2 9, see notes

© Copyright IBM Corporation 2007

36

Fenced 32-bit Routine

Define the routine as not threadsafe using the ALTER PROCEDURE or ALTER FUNCTION statement with the NOT THREADSAFE clause.

Rebuild the routine source code into a 64-bit library using the DB2 Version 9 bldrtn scripts and redeploy the library to the DB2 server.

Migrated from a Version 8 32-bit instance (AIX® and Windows®)

You should specify a default entry point for your routine library using the ALTER PROCEDURE or ALTER FUNCTION statement. For example, to explicitly specify the entry point for an existing procedure use the following statement:

```
ALTER SPECIFIC PROCEDURE schema-name.specific-name EXTERNAL NAME 'library-name!function-name'
```

where library-name is the library to be loaded and function-name is the explicit entry point for the function associated with the routine.

If none of the previously mentioned situations apply, you do not need to change your routine libraries or routine definitions.

3. For routines that you did not rebuild but that you modified, rebind the routine packages to the target DB2 database. During database migration, the existing packages for your routines are invalidated.

4. Test your routines to verify your changes and to ensure that the routines run as expected using DB2 9.

MQT evaluation and usage diagnostics

- New Section in **db2exfmt explain tool** output for extended diagnostic information or use **EXPLAIN_GET_MSGS()** table function
- **db2exmigr** - migrates explain tables and keeps existing explain table data
 - **EXPLAIN_DIAGNOSTICS** and **EXPLAIN_DIAGNOSTICS_DATA**

```
Extended Diagnostic Information:
-----
Diagnostic Identifier:      2
Diagnostic Details:        EXP0022W  Index has no statistics.  The index
"CMCCAIN"."MQTIX1" has not had runstats run on it.  This can lead to
poor cardinality and predicate filtering estimates.

Diagnostic Identifier:      1
Diagnostic Details:        EXP0020W  Table has no statistics.  The table
"CMCCAIN"."MQT1" has not had runstats run on it.  This may result in a
sub-optimal access plan and poor performance.

Diagnostic Identifier:      3
Diagnostic Details:        EXP0148W  The following MQT or statistical view
was considered in query matching: "CMCCAIN"."MQT1".

Diagnostic Identifier:      4
Diagnostic Details:        EXP0149W  The following MQT was used (from those
considered) in query matching: "CMCCAIN"."MQT1".
```

© Copyright IBM Corporation 2007

37

The new diagnostic messages for Materialized Query Tables are included with the output from the db2exfmt explain tool.

The Visual Explain tool also shows the diagnostic messages when a new query is explained.

The example shows a group of messages generated by the DB2 Optimizer during the evaluation of a query.


Extended Diagnostic Information:

```
Diagnostic Identifier:      2
Diagnostic Details:        EXP0022W  Index has no statistics.  The index "CMCCAIN
"."MQTIX1" has not had runstats run on it.  This can lead to poor cardinality and
predicate filtering estimates.


Diagnostic Identifier:      1
Diagnostic Details:        EXP0020W  Table has no statistics.  The table "CMCCAIN
"."MQT1" has not had runstats run on it.  This may result in a sub-optimal access
plan and poor performance.

Diagnostic Identifier:      3
Diagnostic Details:        EXP0148W  The following MQT or statistical view was
considered in query matching: "CMCCAIN"."MQT1".

Diagnostic Identifier:      4
Diagnostic Details:        EXP0149W  The following MQT was used (from those
considered) in query matching: "CMCCAIN"."MQT1".
```



IX. Changes to utilities and tools



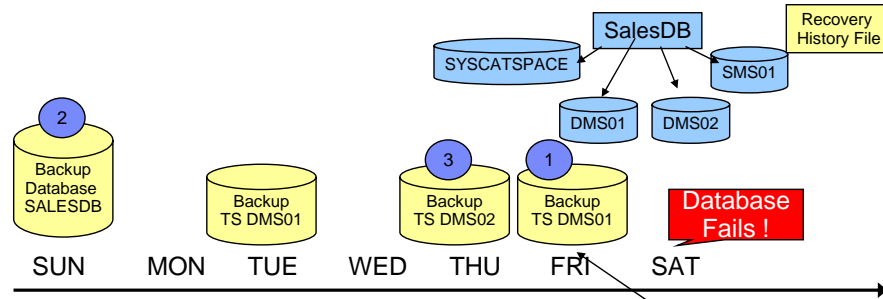
© Copyright IBM Corporation 2006
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

4.0.3

Let's look at changes to utilities and tools.

Database Disaster Recovery using Rebuild

- Necessary Table spaces will be restored automatically based on Recovery History data



- RESTORE DB SALESDB REBUILD WITH ALL TABLESPACES IN DATABASE TAKEN AT Friday (DB2 Restores Friday, Sunday and Thursday)**
- ROLLFORWARD DB SALESDB TO END OF LOGS AND STOP (Only One ROLLFORWARD needed in DB2 9)**

© Copyright IBM Corporation 2007

39

This is the same disaster recovery situation that was explained earlier. At the time of the database failure these backup images were available:

- A Database level backup on Sunday. This could be online or offline.
- A table space level backup of DMS01 on Tuesday.
- A table space level backup of DMS02 on Thursday.
- A more recent table space level backup of DMS01 on Friday.

The database needs to be restored on Friday evening. This time a Restore command with the rebuild option will be used. The steps would be:

- Issue a single Restore command with the REBUILD option pointing to the latest backup image available, the table space backup of DMS01 from Friday.

RESTORE DB SALESDB REBUILD WITH ALL TABLESPACES IN DATABASE TAKEN AT Friday

The Restore utility would use the complete list of table spaces in the database at the time of the Friday table space backup of DMS01 to establish the list table spaces

needed to rebuild the database. Using the target backup image the data for the table space DMS01 would be restored. The recovery history file would be scanned to locate the most recent backup image for the other table spaces. The table spaces SYSCATSPACE and SMS01 would be automatically restored using the database backup image taken on Sunday. Next the table space DMS02 would be restored automatically using the table space backup from Thursday. Now that all of the table spaces have been restored from some backup, the Restore utility is considered complete.

- Now a single database rollforward can be used to apply the logged changes from Sunday to Friday for ALL table spaces. This is the result of a change to the Rollforward command processing in DB2 9. The logs will be processed from the oldest backup image used forward to the end of logs, but the starting point for each table space would depend on the image used for that table space. For example, the rollforward would only need to apply changes to DMS02 from Thursday and Friday.

ROLLFORWARD DB SALESDB TO END OF LOGS AND STOP

When the command completes, all table spaces will be available for application access.

This is a much simpler sequence of commands than those needed to perform a similar recovery using DB2 8.2. In this case, only one backup image was specifically selected, the others were automatically chosen. This is similar to the processing of an automatic incremental restore.

Monitoring Database Rebuild

db2 list utilities show detail

```
ID = 3
Type = RESTORE
Database Name = TP1
Partition Number = 0
Description = automatic db rebuild
Start Time = 08/03/2006 14:52:25.830154
State = Executing
Invocation Type = User
Progress Monitoring:
  Phase Number = 1
    Total Work = 19681280 bytes
    Completed Work = 19681280 bytes
    Start Time = 08/03/2006 14:52:25.830162
  Phase Number [Current] = 2
    Description = 20060803140132
    Completed Work = 4096 bytes
    Start Time = 08/03/2006 14:52:34.761567
  Phase Number = 3
    Description = 20060803143453
    Completed Work = 0 bytes
    Start Time = Not Started
```

© Copyright IBM Corporation 2007

40

A RESTORE with the REBUILD options could cause many backup images to be restored automatically. The LIST UTILITIES SHOW DETAILS can be used to show the current processing phase. It would also list the backup image timestamps that would be part of the rebuild. The following is an example of the output from a LIST UTILITIES command issued during a database rebuild.

Database Rebuild option for RESTORE DATABASE command:

1. Recoverable Databases:

1. Rebuild Database from one or more table space backups
2. Rebuild a partial database with selected table spaces

2. Non-Recoverable Databases:

1. Rebuild a partial database with selected table spaces from single Database Backup

Syntax:

RESTORE DATABASE *source-database-alias*

REBUILD WITH - ALL TABLESPACES IN DATABASE

- ALL TABLESPACES IN IMAGE [EXCEPT]

- TABLESPACE(*tablespace-name*)

LIST HISTORY for RESTORE with REBUILD

- One record for each table space restore performed
- One record for each database restore with REBUILD

```
Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log Backup ID
-----
R D 20060803145225 R S0000171.LOG S0000184.LOG 20060803143742
-----
Contains 6 tablespace(s):
00001 TP1ACCTI
00002 TP1ACCTD
00003 TP1HIST
00004 TP1SMS
00005 USERSPACE1
00006 SYSCATSPACE
-----
Comment: REBUILD TP1 WITH RF
Start Time: 20060803145225
End Time: 20060803145253
```

Since a RESTORE command with the REBUILD option may actually process a large number of database and table space backup images during its processing a group of records will be written to the recovery history file for the database.

There will be one record written for each backup image that is processed to restore one or more table spaces.

Backup and Recovery Enhancements

- Continue RECOVER that ended during rollforward phase
 - Restart from where it stopped, rather than from the beginning
 - Use RECOVER DATABASE command with RESTART
- db2ckbkp
 - s option displays automatic storage paths in the backup image
 - t option displays table space details including container information
- Redirected restore operation can automatically generate script
 - Script is then edited, for example using different paths
 - The script is executed to perform the redirected restore



```
RESTORE DB TP1 FROM /backups
REDIRECT
GENERATE SCRIPT TP1_GEN_SCRIPT.TXT
```

© Copyright IBM Corporation 2007

42

If you issue the RECOVER DATABASE command following an incomplete recover operation that ended during the rollforward phase, the recover utility will now attempt to continue the previous recover operation, without redoing the restore phase. If you want to force the recover utility to redo the restore phase, you need to issue the RECOVER DATABASE command with the RESTART option to force the recover utility to ignore any prior recover operation that failed to complete.

A RECOVER will not complete if, for example, it was interrupted by the user (if Ctrl+C is pressed or the force application command is used) or if DB2 was unable to retrieve log files.

You can now perform a redirected restore operation by generating a redirected restore script from an existing backup image, modifying any of the paths or container sizes in the script, then running the script to recreate the database with the new set of containers.

Example of Generated Redirect Restore Script

```
• UPDATE COMMAND OPTIONS USING S ON Z ON TP1_NODE0000.out V ON;
• SET CLIENT ATTACH_DBPARTITIONNUM 0;
• SET CLIENT CONNECT_DBPARTITIONNUM 0;
• .....
• -- ** automatically created redirect restore script
• .....
• RESTORE DATABASE TP1
• -- USER <username>
• -- USING <password>
• FROM 'C:\cf49\backups\testing\lab11'
• TAKEN AT 20060523142208
• -- DBPATH ON <target-directory>
• INTO TP1
• -- NEWLOGPATH 'd:\cf49\logs\NODE0000\
• -- WITH <num-buff> BUFFERS
• -- BUFFER <buffer-size>
• -- REPLACE HISTORY FILE
• -- REPLACE EXISTING
• REDIRECT
• -- PARALLELISM <n>
• -- COMPROLIB <lib-name>
• -- COMPROPTS <options-string>
• WITHOUT ROLLING FORWARD
• WITHOUT PROMPTING
• ;
• .....
• -- ** table space definition
• .....
• -- ** Tablespace name = SYSCATSPACE
• -- ** Tablespace ID = 0
• -- ** Tablespace Type = System managed space
• -- ** Tablespace Content Type = All permanent data. Regular table space.
• -- ** Tablespace Page size (bytes) = 4096
• -- ** Tablespace Extent size (pages) = 32
• -- ** Using automatic storage = No
• -- ** Total number of pages = 4512
• .....
• SET TABLESPACE CONTAINERS FOR 0
• -- IGNORE ROLLFORWARD CONTAINER OPERATIONS
• USING (
• PATH 'SQLT0000.0'
• );
• ;
• .....
```

© Copyright IBM Corporation 2007

43

Example of generated script is shown above.

More Recovery Enhancements

- **ROLLFORWARD DATABASE:**
 - Format of output timestamp same as format specified in input file
 - Table space-level point-in-time rollforward can be reissued with earlier point-in time
- Tivoli Storage Manager support added (Windows x64)
- TSM management class no longer used for filtering when restoring or retrieving logs
- Backup image naming convention on the Windows platform has changed to match the naming convention used for all other operating systems:

`DB_alias.Type.Inst_name.NODEnnnn.CATNnnnn.timestamp.Seq_num`

© Copyright IBM Corporation 2007

44

The ROLLFORWARD DATABASE command could accept both Coordinated Universal Time (UTC) and local time as valid input formats since DB2 Version 8. In DB2 9, the format of the output timestamp is the same as the format specified in the input.

Tivoli Storage Manager (TSM) provides storage management and data access services to protect your organization's data from failures and other errors. TSM supports various communication methods, provides administrative facilities to manage the backup and storage of files, and provides facilities for scheduling backups.

TSM support has been added for all Windows x64 systems. The minimum required level of TSM client API is Version 5.3.2.

As of DB2 9, the naming convention for backup images stored on Windows operating systems has changed to match the naming convention used for all other operating systems. Backup images from earlier versions of the product that use the previous naming structure can still be restored on DB2 9 database systems.

db2batch command changes

- no longer executes embedded dynamic SQL (no **-cli** option)
- **-iso** parameter to specify isolation level (default now RR)
- XQuery statements issued by prefixing with **XQUERY** keyword
- **msw <switch> <setting>** sets state of each specified monitor switch.
<switch> uow, statement, table, bufferpool, lock, sort, timestamp, **all**
<setting> **hold** – unchanged state (default), **on**, **off**
- **-mss <snapshot>**
Specifies monitoring snapshots taken after each statement or block is executed. More than one snapshot can be taken at a time, with information from all snapshots combined into one large table before printing.
Possible snapshots: applinfo_all, dbase_applinfo, dcs_applinfo_all, db2, dbase, dbase_all, dcs_dbase, dcs_dbase_all, dbase_remote, dbase_remote_all, agent_id, dbase_appls, appl_all, dcs_appl_all, dcs_appl_handle, dcs_dbase_appls, dbase_appls_remote, appl_remote_all, dbase_tables, appl_locks_agent_id, dbase_locks, dbase_tablespaces, bufferpools_all, dbase_bufferpools, dynamic_sql, **all**. Any snapshots involving appl ID are not supported in favor of their agent ID (application handle) equivalents. By default, no snapshots taken.

© Copyright IBM Corporation 2007

45

db2batch now supports the following data types: CLOB, GRAPHIC, VARGRAPHIC, LONGVARGRAPHIC, DBCLOB, BLOB, and XML.

This XQuery example retrieves the city where the country is “Canada”.

```
XQUERY db2-fn:xmlcolumn ('XMLCUSTOMER.INFO')//addr[@country="Canada"]/city
```

-msw *switch*

Sets the state of each specified monitor switch. You can specify any of the following: uow, statement, table, bufferpool, lock, sort, and timestamp. The special switch 'all' sets all of the above switches. For each switch that you specify you must choose one of the following settings: **hold** The state of the switch is unchanged. This is the default. **on** The switch is turned on. **off** The switch is turned off.

-mss *snapshot*

Specifies the monitoring snapshots that should be taken after each statement or block is executed, depending on the -g option. More than one snapshot can be taken at a time, with the information from all snapshots combined into one large table before printing. The possible snapshots are: applinfo_all, dbase_applinfo, dcs_applinfo_all, db2, dbase, dbase_all, dcs_dbase, dcs_dbase_all, dbase_remote, dbase_remote_all, agent_id, dbase_appls, appl_all, dcs_appl_all, dcs_appl_handle, dcs_dbase_appls, dbase_appls_remote, appl_remote_all, dbase_tables, appl_locks_agent_id, dbase_locks, dbase_tablespaces, bufferpools_all, dbase_bufferpools, and dynamic_sql. The special snapshot 'all' takes all of the above snapshots. Any snapshots involving an appl ID are not supported in favour of their agent ID (application handle) equivalents. By default, no monitoring snapshots are taken.

Load Enhancements

▪ Load using Named Pipe

- Previously supported multiple input files but only processes one at a time.
- In DB2 9 “USEREXIT” allows load to process multiple input files in parallel.

```
LOAD .... SOURCE USEREXIT <executable>  
REDIRECT INPUT FROM BUFFER <input buffer>  
FILE <input file> OUTPUT TO FILE <output file> PARALLELIZE
```

▪ Load from Cursor with Remote Fetch

- No longer have to use a federated name in declare cursor to access remote database source. New DB2 9 syntax:

```
DECLARE <cursor name> CURSOR  
DATABASE <dbalias> USER <userid> USING <password>  
FOR SELECT ...
```

- See: <http://svlwas2.svl.ibm.com/dd/imdd/asis/500046/500046.html>

© Copyright IBM Corporation 2007

46

See Rex Oliva's *Introduction to DB2 9 Load Enhancements – (1) User Exits (2) Cursor Load w/Remote Fetch (Linux, Unix, Windows)* at:

<http://svlwas2.svl.ibm.com/dd/imdd/asis/500046/500046.html>

New administrative commands

- CALL ADMIN_CMD(..) enhanced to support new commands:
 - ADD CONTACT | ADD CONTACTGROUP
 - DROP CONTACT | DROP CONTACTGROUP
 - UPDATE CONTACT | UPDATE CONTACTGROUP
 - AUTOCONFIGURE
 - INITIALIZE TAPE
 - RESET ALERT CFG
 - RESET DB CFG
 - RESET DBM CFG
 - REWIND TAPE
 - SET TAPE POSITION
 - UPDATE ALERT CFG
 - UPDATE NOTIFICATION LIST

Example:

```
CALL SYSPROC.ADMIN_CMD  
( 'force application ALL' )
```

(3/4)

© Copyright IBM Corporation 2007

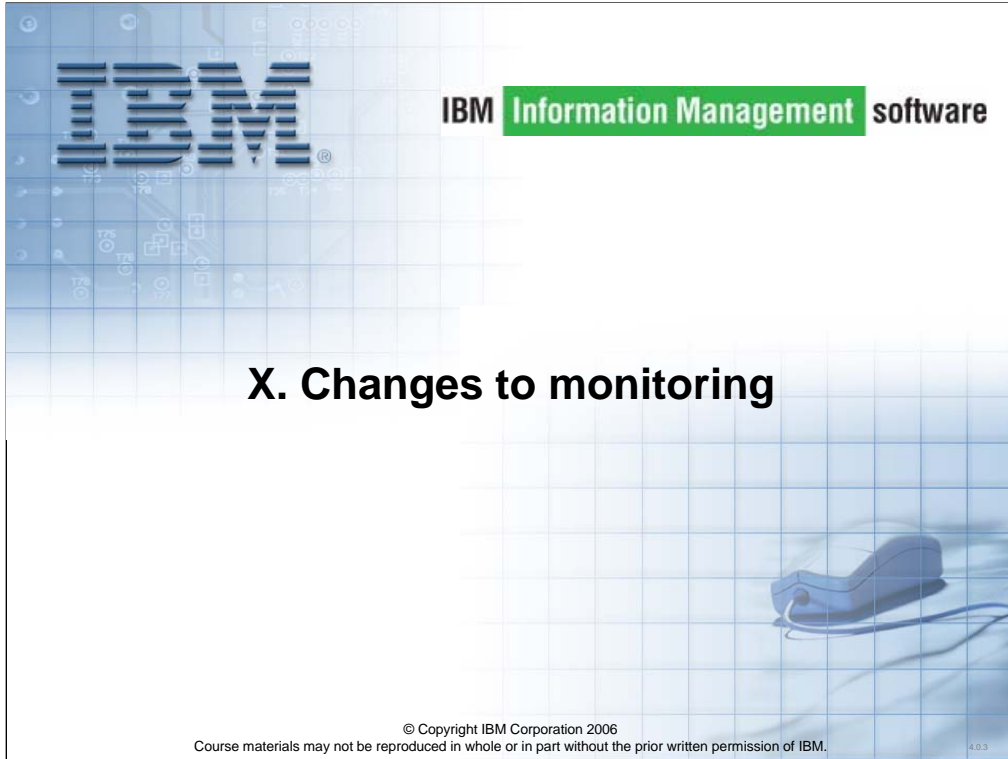
47

The ADMIN_CMD procedure is used by applications to run administrative commands using the SQL CALL statement.

CALL ADMIN_CMD(..) function was enhanced in DB2 8 Fixpak 9 with access to the following DB2 administration commands through SQL.

- IMPORT
- UPDATE DBM CFG
- UPDATE HISTORY
- LOAD
- QUIESCE / UNQUIESCE DATABASE
- QUIESCE TABLESPACES FOR TABLE
- FORCE APPLICATIONS
- DESCRIBE TABLE/INDEX/DATA PARTITION
- EXPORT (XML support)
- REDISTRIBUTE

DB2 9 includes the above listed new administrative commands.



Let's look at changes to monitoring.

Automatic storage identification

Use GET SNAPSHOT FOR TABLESPACES ON ...

- Do not use the LIST TABLESPACES
 - **db2 list tablespaces/db2 list tablespace containers** deprecated
- Example of GET SNAPSHOT FOR TABLESPACES ON <database> command:

```
Tablespace name           = SYSCATSPACE
Tablespace ID             = 0
...
Automatic Prefetch size enabled = Yes
Buffer pool ID currently in use = 1
Buffer pool ID next startup   = 1
Using automatic storage      = Yes
Auto-resize enabled          = Yes
...
Increase size (bytes)       = AUTOMATIC
```

- **Note that *Using automatic storage = Yes* and *Increase size (bytes) = AUTOMATIC* are shown in the snapshot.**

© Copyright IBM Corporation 2007

49

In DB2 9, the following changes to the db2pd -fcm command are introduced:

- You can use the new hwm option to see historical information about applications that consume large amounts of fast communication manager (FCM) resources. FCM usage statistics for applications are collected and retained so that even if the applications are not connected to the database, the details will be reported when you run the db2pd command with the -fcm hwm option.
- The output will now contain FCM channel usage statistics, including the high and low water mark values with respect to the number of channels used.

Write-to-table event monitor changes

New Snapshot Table Functions & Views in DB2 9

GET SNAPSHOT FOR DATABASE ON PRODDB

↓

(DB2 8.1) `SELECT * FROM TABLE(SNAPSHOT_DATABASE('PRODDB',-1)) AS SNAPDB`

↓

Contains these statistics for Database Logging

SEC_LOG_USED_TOP	TOT_LOG_USED_TOP
TOTAL_LOG_USED	TOTAL_LOG_AVAILABLE
LOG_READS	LOG_WRITES

(DB2 9 Function) `SELECT * FROM TABLE(SNAP_GET_DB('PRODDB',-1)) AS SNAPDB`

OR (DB2 9 View) `SELECT * FROM SYSIBMADM.SNAPDB`

Contains these statistics for Database Logging

TOTAL_LOG_AVAILABLE	TOTAL_LOG_USED
SEC_LOG_USED_TOP	TOT_LOG_USED_TOP
SEC_LOGS_ALLOCATED	LOG_READS
LOG_READ_TIME_S	LOG_READ_TIME_NS
LOG_WRITES	LOG_WRITE_TIME_S
LOG_WRITE_TIME_NS	NUM_LOG_WRITE_IO
NUM_LOG_READ_IO	NUM_LOG_PART_PAGE_IO
NUM_LOG_BUFFER_FULL	NUM_LOG_DATA_FOUND_IN_BUFFER
APPL_ID_OLDEST_XACT	LOG_TO_REDO_FOR_RECOVERY
LOG_HELD_BY_DIRTY_PAGES	

© Copyright IBM Corporation 2007

50

DB2 9 introduces administrative views that provide an easy-to-use application programming interface to DB2 administrative functions through SQL. The administrative views fall into three categories:

- Views based on catalog views.
- Views based on table functions with no input parameters.
- Views based on table functions with one or more input parameters.

The administrative views are the preferred and only documented interfaces for the views based on catalog views and the views based on table functions with no input parameters because the table functions do not provide any additional information or performance benefits.

The example shows the portion of the GET SNAPSHOT FOR DATABASE report that was returned using the SNAPSHOT_DATABASE function for system log statistics in DB2 V8.1. A number of new detailed statistics were added since V8.1. These are now available using the DB2 9 function SNAP_GET_DB and also in the DB2 Administrative View, SYSIBMADM.SNAPDB.

Snapshot monitor SQL administrative views (1 of 7)

The following table lists some of the snapshot monitor SQL administrative views. Each returning monitor data about a specific area of the database system.

- Database manager monitoring:

Monitor level	SQL administrative view	Information returned
Database manager	SYSIBMADM.SNAPDBM	Database manager level information.
Database manager	SYSIBMADM.SNAPFCM	Database manager level information regarding the fast communication manager (FCM).
Database manager	SYSIBMADM.SNAPFCM_PART	Database manager level information for a partition regarding the fast communication manager (FCM).
Database manager	SYSIBMADM.SNAPSWITCHES	Database manager monitor switch settings.
Database manager	SYSIBMADM.SNAPDBM_MEMORY_POOL	Database manager level information about memory usage.

© Copyright IBM Corporation 2007

51

There are a number of different snapshot monitor SQL administrative views available, each returning monitor data about a specific area of the database system. For example, the SYSIBMADM.SNAPBP SQL administrative view captures a snapshot of buffer pool information.

Snapshot monitor SQL administrative views (2 of 7)

- Database monitoring:

Monitor level	SQL administrative view	Information returned
Database	SYSIBMADM.SNAPDB	Database level information and counters for a database. Information is returned only if there is at least one application connected to the database.
Database	SYSIBMADM.SNAPDB_MEMORY_POOL	Database level information about memory usage.
Database	SYSIBMADM.SNAPHADR	Database level information about high availability disaster recovery.
Database	SYSIBMADM.SNAPUTIL	Information about utilities.
Database	SYSIBMADM.SNAPUTIL_PROGRESS	Information about the progress of utilities.
Database	SYSIBMADM.SNAPDETAILLOG	Database level information about log files.
Database	SYSIBMADM.SNAPSTORAGE_PATHS	Returns list of automatic storage paths for the database including file system information for each storage path.

© Copyright IBM Corporation 2007

52

Before capturing a snapshot, consider if you need information from monitor elements that are under monitor switch control. If a particular monitor switch is off, the monitor elements under its control will not be collected. See the individual monitor elements to determine if an element you need is under switch control.

Snapshot monitor SQL administrative views (3 of 7)

- Application monitoring:

Monitor level	SQL administrative view	Information returned
Application	SYSIBMADM. SNAPAPPL	General application level information for each application that is connected to the database. This includes cumulative counters, status information, and most recent SQL statement executed (if statement switch is set).
Application	SYSIBMADM. SNAPAPPL_INFO	General application level identification information for each application that is connected to the database.
Application	SYSIBMADM. SNAPLOCKWAIT	Application level information regarding lock waits for the applications connected to the database.

© Copyright IBM Corporation 2007

53

All snapshot monitoring administrative views and associated table functions use a separate instance connection, which is different from the connection the current session uses. Therefore, only default database manager monitor switches are effective. Ineffective monitor switches include any that are turned on or off dynamically from the current session or application.

Snapshot monitor SQL administrative views (4 of 7)

- More application monitoring:

Monitor level	SQL administrative view	Information returned
Application	SYSIBMADM.SNAPSTMT	Application level information regarding statements for the applications connected to the database. This includes the most recent SQL statement executed (if the statement switch is set).
Application	SYSIBMADM.SNAPAGENT	Application level information regarding the agents associated with applications connected to the database.
Application	SYSIBMADM.SNAPSUBSECTION	Application level information regarding the subsections of access plans for the applications connected to the database.
Application	SYSIBMADM.SNAPAGENT_MEMORY_POOL	Information about memory usage at the agent level.

© Copyright IBM Corporation 2007

54

Reference Only: More SQL Administrative Views.

Snapshot monitor SQL administrative views (5 of 7)

- Table monitoring:

Monitor level	SQL administrative view	Information returned
Table	SYSIBMADM.SNAPTAB	Table activity information at the database and application level for each application connected to the database. Table activity information at the table level for each table that was accessed by an application connected to the database. Requires the table switch.
Table	SYSIBMADM.SNAPTAB _REORG	Table reorganization information at the table level for each table in the database undergoing reorganization.
Lock	SYSIBMADM.SNAPLOCK	Lock information at the database level, and application level for each application connected to the database. Requires the lock switch.
Dynamic SQL	SYSIBMADM.SNAPDYN _SQL	Point-in-time statement information from the SQL statement cache for database.

© Copyright IBM Corporation 2007

55

Reference Only: More SQL Administrative Views.

Snapshot monitor SQL administrative views (6 of 7)

- Buffer pool monitoring:

Monitor level	SQL administrative view	Information returned
Buffer pool	SYSIBMADM.SNAPBP	Buffer pool activity counters for the specified database. Requires the buffer pool switch.
Buffer pool	SYSIBMADM.SNAPBP_PART	Information on buffer size and usage, calculated per partition.

Reference Only: More SQL Administrative Views.

Snapshot monitor SQL administrative views (7 of 7)

- Table space monitoring:

Monitor level	SQL administrative view	Information returned
Table space	SYSIBMADM.SNAPTBSP	Information about table space activity at database & application level for each connected application, and table space level for each table space accessed by application connected to database. Requires buffer pool switch.
Table space	SYSIBMADM.SNAPTBSP_PART	Table space configuration info.
Table space	SYSIBMADM.SNAPTBSP_QUIESCER	Information about quiescers at the table space level.
Table space	SYSIBMADM.SNAPCONTAINER	Table space container configuration info at the table space level.
Table space	SYSIBMADM.SNAPTBSP_RANGE	Table space map range information.

© Copyright IBM Corporation 2007

57

Reference Only: More SQL Administrative Views.

Monitoring Database Activity using Administrative Convenience Views

Administrative Convenience Views

- Extend standard snapshot statistics with calculations (like Buffer Pool Hit Ratios)
- Can Join multiple snapshot reports for additional details

```
SELECT TOTAL_HIT_RATIO_PERCENT,  
       DATA_HIT_RATIO_PERCENT,  
       INDEX_HIT_RATIO_PERCENT, BP_NAME  
FROM SYSIBMADM.BP_HITRATIO
```

TOTAL_HIT_RATIO_PERCENT	DATA_HIT_RATIO_PERCENT	INDEX_HIT_RATIO_PERCENT	BP_NAME
87.16	87.66	80.37	IBMDEFAULTBP
60.98	61.83	0.00	CLPBUFFL
12.73	12.73	13.33	CLPBUFFS
-	-	-	IBMSYSTEMBP4K
-	-	-	IBMSYSTEMBP8K
-	-	-	IBMSYSTEMBP16K
-	-	-	IBMSYSTEMBP32K

7 record(s) selected.

© Copyright IBM Corporation 2007

58

DB2 9.1 also provides you with a set of administrative views that do not only return values of individual monitor elements, but also return computed values that are commonly required in monitoring tasks. For example, the SYSIBMADM.BP_HITRATIO administrative view returns calculated values for buffer pool hit ratios, which combine a number of individual monitor elements.

The example shows a query that uses the view SYSIBM.BP_HITRATIO to return the buffer pool hit ratios for each database buffer pool.

Snapshot Monitor SQL administrative convenience views (1 of 2)

SQL Administrative Convenience Views	Information returned
SYSIBMADM.APPLICATIONS	Information about connected database applications.
SYSIBMADM.APPL_PERFORMANCE	Information about the rate of rows selected versus the number of rows read by an application.
SYSIBMADM.BP_HITRATIO	Buffer pool hit ratios, including total, data, and index, in the database.
SYSIBMADM.BP_READ_IO	Information about buffer pool read performance.
SYSIBMADM.BP_WRITE_IO	Information about buffer pool write performance.
SYSIBMADM.CONTAINER_UTILIZATION	Information about table space containers and utilization rates.
SYSIBMADM.LOCKS_HELD	Information on current locks held.

© Copyright IBM Corporation 2007

59

These views are similar to the SQL administrative views, but provide more general information on the health of the database system.

Snapshot Monitor SQL administrative convenience views (2 of 2)

SQL Administrative Convenience Views	Information returned
ISYSIBMADM.LOCKWAITS	Information about DB2 agents working on behalf of applications that are waiting to obtain locks.
SYSIBMADM.LOG_UTILIZATION	Information about log utilization for the currently connected database.
SYSIBMADM.LONG_RUNNING_SQL	Information about the longest running SQL in the currently connected database.
SYSIBMADM.QUERY_PREP_COST	Information about the time required to prepare different SQL statements.
SYSIBMADM.TBSP_UTILIZATION	Table space configuration and utilization information.
SYSIBMADM.TOP_DYNAMIC_SQL	The top dynamic SQL statements sortable by number of executions, average execution time, number of sorts, or sorts per statement.

© Copyright IBM Corporation 2007

60

For Reference Only – More SQL Administrative Convenience Views.

Summary

After completing this session, you should be able to:

- Describe changes to packaging and software and operating system requirements.
- Understand changes to DDL, utilities, tools and monitoring.
- Configure database and instance to implement new features.
- Analyze changes that affect applications and database runtime behavior.
- Develop a migration strategy to DB2 9.
- For further information see:
 - [CG090 DB2 9 Transition](#)
 - www.ibm.com/services/learning/us

In Summary, DB2 9 has a lot of new features. The purpose of this session was to unveil changes made to DB2 9 that are often overlooked but very important to understand before migrating. Planning your migration can be much easier if you understand things not commonly mentioned.

Melanie Stopfer

IBM North America Lab Services

mstopfer@us.ibm.com

