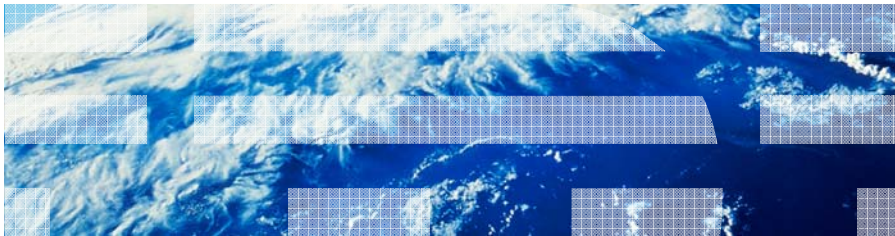


DB2 10 for LUW Upgrade Best Practices

Melanie Stopfer

IBM Software Group
mstopfer@us.ibm.com



© Copyright IBM Corporation 2012
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

6.0

Speaker Bio:

Melanie Stopfer is a Consulting Learning Specialist and Developer for IBM Software Group. As a Certified DB2 Advanced Technical Expert and Learning Facilitation Specialist, she has provided in-depth technical support to IM customers specializing recovery, performance and database upgrade and migration best practices since 1988. In 2009, Melanie was the first DB2 LUW speaker to be inducted into the IDUG Speaker Hall of Fame and was again selected Best Overall Speaker in 2011.

Objectives

- Describe changes to DB2 10 packaging and software and operating system requirements.
- Understand DB2 10 changes to DDL, utilities, tools and monitoring.
- Configure a database and instance to implement DB2 10 new features.
- Analyze DB2 10 changes that affect applications and database runtime behavior.
- Develop a DB2 10 upgrade strategy.



© Copyright IBM Corporation 2012

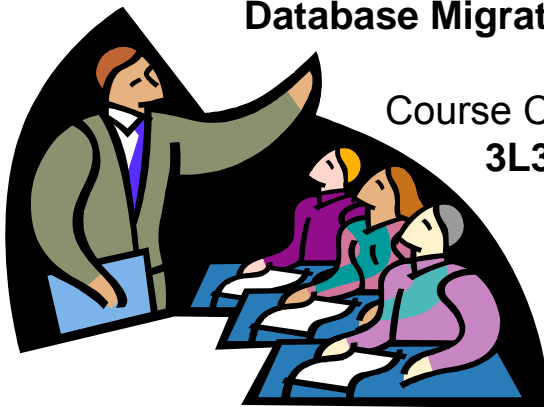
2

Abstract:

Are you responsible for upgrading your database to DB2 10 the newest version for Linux, UNIX and Windows? Attend this presentation to learn about often overlooked but very important tips and tricks to understand before upgrading. Are you looking for details about what will change when you upgrade? Do you know what will not change? What new features will be implemented when you create a new database versus upgrade your existing database. Planning your upgrade is much easier if you know these best practices.

New Course - DB2 10.1 for LUW New Features and Database Migration Considerations

- Course Title: **DB2 10.1 for LUW New Features and Database Migration Considerations**



Course Code: **CL313** (classroom)
3L313 (ILO- Instructor Led Online)

- This course can be customized to meet your needs.
- Check Out IBM Education Offerings at:
<http://www.ibm.com/software/data/education/>

© Copyright IBM Corporation 2012

3

Course Title: DB2 10.1 for LUW New Features and Database Migration Considerations
Course Code: CL313 , 3L313 (ILO)
Course Length : 2 days
Skill Level: Intermediate
Delivery Type: Classroom

Course Description :

Get the technical information about the new functions and features available in DB2 10.1 for Linux, UNIX, and Windows and plan the migration for DB2 Version 9.5, 9.7 or 9.8 database systems to DB2 LUW 10.1.

DB2 Version 10.1 for Linux, UNIX, and Windows continues to deliver database efficiency, simplicity, and reliability. Important new features and enhancements address the needs of your business. Whether your needs are improving business critical reliability and performance, simplifying consolidation and security, increasing business insight, reducing costs, or providing a resilient system for the valuable information assets of your company, DB2 Version 10.1 delivers.

The course will describe the adaptive compression feature of DB2 LUW 10.1 that improves table compression through an advanced row compression technique that uses two levels of compression dictionaries (table-level and page-level) to improve compression ratios, particularly as data changes.

DB2 LUW 10.1 support for Multi-temperature storage allows administrators to define storage groups that can be used to assign data based on performance characteristics and migrate data to a storage group with lower cost devices or lower performance when the demand for that data declines.

The lectures will describe the DB2 LUW 10.1 performance related features including improved query optimizer techniques and functionality including star schema query optimization, improved data and index prefetching, and improved use of statistical views.

We will discuss how to implement system period and application period temporal tables associated with Time Travel Query to assign time-based state information to your data, so you can query the past state of your data.

DB2 LUW 10.1 introduces row and column access control (RCAC), as an additional layer of data security. Row and column access control is sometimes referred to as fine-grained access control or FGAC. We will discuss using RCAC to control access to a table at the row level, column level, or both and how RCAC can be used to complement the table privileges model.

The database high availability feature HADR is significantly enhanced with DB2 LUW 10.1. We will discuss how with DB2 10.1 you can define up to three standby databases. The lecture will also explain how to implement one standby database with the new time-delayed replay feature, which helps to recover from application errors that cause data loss on the primary.

We will cover various migration planning and installation considerations to help move existing DB2 LUW databases to DB2 LUW 10.1 to take advantage of the new features and functions.

Planning Your Upgrade to DB2 10.1 LUW

DB2 10 LUW Upgrade from *pre-DB2 Version 10.1* copy:

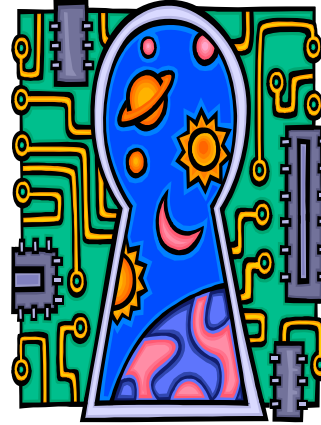
- DB2 9.7-> DB2 10.1
- DB2 9.5 -> DB2 10.1
- DB2 9.8 -> DB2 10.1
 - pureScale

- If DB2 servers are running on release before DB2 9.5:

- Migrate to DB2 9.5 latest fixpack
- Then upgrade to DB2 10.1

See notes for more details/URLs:

- Upgrading to DB2 V10.1 Roadmap
- DB2 Upgrade Portal
- DB2 V10.1 English Manuals
- DB2 V10.1 Translated Manuals



SC27-3890-00 What's New for DB2 V10
SC27-3881-00 Upgrading to DB2 V10.1 Guide

© Copyright IBM Corporation 2012

4

DB2 10 for Linux, UNIX and Windows became generally available on April 30, 2012.

Refer to the DB2 Information Center for details about the installation and upgrade process. It is highly recommended that you consult the DB2 IC while planning and executing an upgrade. You should also test the upgrade process on non-production servers first. Your first point of reference to learn how to upgrade each component of your DB2 environment and how to create a upgrade plan is the **SC27-3881-00 Upgrading to DB2 Version 10 Guide**. The content of this guide is available in the *DB2 Information Center*. You may also download the English manuals from the following website:

<http://www-01.ibm.com/support/docview.wss?uid=swg27024478>

The translated manuals can be downloaded from:

<http://www-304.ibm.com/support/docview.wss?rs=71&uid=swg27024479>

Checking Whether Upgrade is Possible

Direct upgrades are supported from Version 9.5 and higher systems. Upgrading your DB2 server requires that you install a DB2 10 copy and then upgrade all the instances and databases to be able to run them under the DB2 10 copy. You can directly upgrade existing DB2 9.8, 9.7 or DB2 9.5 instances and databases to DB2 10. On Windows operating systems, you have an option to automatically upgrade an existing pre-V10 DB2 copy. If you choose to upgrade your existing DB2 copy during installation, you only need to upgrade your databases after installation.

Review also the DB2 upgrade portal at www.ibm.com/support (formerly known as DB2 migration portal) that provides access to additional resources and up-to-date information about the upgrade process as they become available. These resources include educational material, white papers, and webcasts for upgrade. The DB2 Upgrade Portal is located at: <http://www-01.ibm.com/support/docview.wss?rs=71&uid=swg21200005>

The Upgrading to DB2 Version 10 Roadmap is located at:

<http://www-304.ibm.com/support/docview.wss?uid=swg21573228>

Product packaging enhancements

- Functionality in DB2 features and DB2 product editions details:
 - <http://pic.dhe.ibm.com/infocenter/db2luw/v10r1/index.jsp?topic=/com.ibm.db2.luw.licensing.doc/doc/r0053238.html>
- DB2 Database Partitioning Feature (DPF) no longer included or available for any DB2 database editions.
 - Included in all IBM InfoSphere Warehouse V10 product editions
- DB2 pureCluster Feature is separately priced feature.
 - refers to IBM DB2 pureScale (DB2 9.8)
- To read about products and view related licensing and marketing information, see DB2 product page:
 - <http://www.ibm.com/software/data/db2/linux-unix-windows>

© Copyright IBM Corporation 2012

5

As IBM data servers continue to evolve, the names and the packaging of the DB2 components change to respond to market needs. DB2 Database Partitioning Feature (DPF) is no longer included in or available for any DB2 database editions. It is included in all IBM InfoSphere Warehouse product editions. In DB2 10.1, occurrences of “DB2 pureCluster Feature” in documentation or messages refer to the IBM DB2 pureScale Feature.

The functionality in DB2 features and DB2 product editions is explained at:

<http://pic.dhe.ibm.com/infocenter/db2luw/v10r1/index.jsp?topic=/com.ibm.db2.luw.licensing.doc/doc/r0053238.html>

IBM has updated product packaging to provide simplicity through fewer packages, and greater value through more function and features included in the base DB2 editions. To read about these products and to view the related licensing and marketing information, see the DB2 product page at <http://www.ibm.com/software/data/db2/linux-unix-windows>.

DB2 Advanced Enterprise Server Edition

•Storage Optimization - DB2 Deep Compression technology helps clients reduce the size of their databases across many different types of data including row data, indexes, temp tables, XML and binary objects. This can further reduce your storage requirements while improving the efficiency and performance of your disk and IO. The newly introduced Adaptive Compression enriches this feature by applying advanced row compression technology that enables more granular, adaptive and dynamic updates, enhances query performance, and improves data availability. In fact, Adaptive Compression has provided 7x or greater overall space savings for more than one client, with some tables achieving 10x space savings¹.

•InfoSphere Optim Configuration Manager for DB2 offers centralized management of database and client configuration.

•InfoSphere Data Architect is a collaborative data design solution that helps you discover, model, relate, and standardize diverse and distributed data assets.

•InfoSphere Optim Performance Manager Extended Edition for DB2 provides end-to-end database monitoring to improve quality of service. It can Identify, diagnose, solve and prevent performance problems before they impact the business.

•InfoSphere Optim Query Workload Tuner for DB2 provides expert recommendations-- on query structure, access paths, and indexes-- to maximize application performance, reduce specialized skill requirements, and lower the total cost of ownership.

•InfoSphere Optim pureQuery Runtime for Linux, UNIX and Windows provides a runtime environment and application programming interface that enhances the performance of existing in-house database client applications written using Java or C without having to modify them. It also helps in rapid development of new applications.

Suggestions for a good *upgrade strategy*

1. Review new, deprecated, discontinued functionality for DB2 10.1 and for any releases between release are upgrading from and DB2 10.1.
2. Plan how to and modify database applications and routines. Ensure they run successfully in DB2 10.1.
3. Set up DB2 10.1 test server and create test databases.
4. Test database applications and routines on DB2 10.1 test database to determine whether run successfully. If application requires a client, use DB2 10.1 client.
5. Upgrade DB2 servers and clients in test environment. Determine what issues are and how to resolve. Use info to adjust upgrade plan.
6. Upgrade DB2 servers to DB2 10.1 in production environment. Ensure they operate as expected.
7. Upgrade clients to DB2 10.1 in production environment. Ensure clients operate as expected.
8. Test database applications and routines in DB2 10.1 upgraded environment to determine whether run as expected.
9. Make upgraded environment available to users.
10. Plan to remove use of deprecated functionality and incorporate new functionality from DB2 10.1



© Copyright IBM Corporation 2012

6

Your environment has several components such as DB2 servers, DB2 clients, database applications, scripts, routines and tools. Planning your upgrade requires a thorough understanding of the upgrade process of each component in your environment.

First, devise a strategy on how to approach your environment upgrade. You must determine the order in which you are going to upgrade each component. The characteristics of your environment and the information in upgrade essentials, especially the best practices and restrictions, can help you determine your strategy.

The above list is an example of a good *upgrade strategy* in which you test your database applications and routines and determine that they run successfully in DB2 10.1.

After you have a strategy that will give you the outline for your upgrade plan, you can define the upgrade plan details for each component in your environment. An *upgrade plan* should include for each component:

- Upgrade prerequisites
- Pre-upgrade tasks
- Upgrade tasks
- Post-upgrade tasks

If you have previous upgrade plans, review them and compare them with the upgrade plan for DB2 10.1. Include in your new plan any steps related to internal procedures to request access, software installation or other system services within your organization.

Finally, plan to remove the use of deprecated functionality and incorporate new functionality from DB2 10.1. Although you are only required to remove the use of discontinued functionality, you should also plan to remove the use of deprecated functionality after upgrade because they will become unsupported in a future release. Also, you should take advantage of new functionality for your database products, applications, and routines to enhance functionality and improve performance.

Upgrade restrictions for DB2 servers

- Upgrading to a DB2 10.1 non-root installation is supported from a DB2 9.5 and DB2 9.7 non-root installation. Upgrading to a DB2 10.1 non-root installation from a pre DB2 10.1 root installation is not supported.
- On Windows, upgrade action shows for existing DB2 copies that can be upgraded during installation of DB2 10.1. This action automatically installs DB2 10.1 and upgrades all instances running on DB2 copy. This action also uninstalls DB2 copy and any add-on products installed in this copy. If do not choose upgrade action, must manually upgrade instances after installation.
- On Linux/UNIX, upgrade action is not available and can only install a new copy of DB2 10.1. Then manually upgrade instances after installation.
- Instance bit size is determined by the operating system where DB2 10.1 is installed, and support for 32-bit kernels and 64-bit kernels has changed.
- Upgrading from system with multiple copies of DB2 9.5 and DB2 9.7 of all levels is supported.
- Upgrading from a partitioned database environment with multiple database partitions is supported.
- Restoring full database offline backups from pre-DB2 10.1 copies is supported. Rolling forward of logs from a previous level is not possible.
- In upgraded databases with RESTRICT_ACCESS db cfg set to YES, must grant USAGE privilege to non-DBADM users on SYSDEFAULTUSERWORKLOAD. Otherwise, users are unable to submit any work to database.

© Copyright IBM Corporation 2012

7

Before you start to upgrade your DB2 server, you must understand what the support for upgrade is and what restrictions exist.

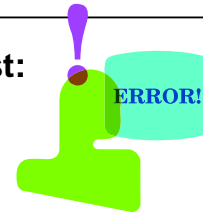
What is unsupported?

- **DB2 10.1 installation fails if following situations exist:**

- Operating system is not supported. Must upgrade to supported version operating system before you upgrade to DB2 10.1 or upgrade to new DB2 server that meets requirements.
- A 32-bit kernel is running on Linux and UNIX operating systems except for Linux on x86. A 64-bit kernel must be installed before installing DB2 10.1.
- A version of DB2 before Version 9.5 is installed on Windows operating systems.

- **db2iupgrade command fails if following situations exist:**

- Do not have authorization to upgrade instance.
- Instance trying to upgrade is active. Run db2stop command to stop instance.
- Instance already at DB2 10.1 or later. Run db2iupdt command to update to different fix pack levels or copies of DB2 10.1.
- Try to upgrade from DB2 10.1 back to DB2 9.7 or DB2 9.5.
- db2ckupgrade fails and causes db2iupgrade command to fail. db2iupgrade command calls db2ckupgrade command to verify whether cataloged local databases are ready for upgrade to DB2 10.1.



You must upgrade to supported version of operating system before you upgrade to DB2 10.1 or upgrade to new DB2 server that meets operating system requirements.

Operating system support changes

- If have DB2 9.5 copy on SUSE Linux Enterprise Server 10, first apply DB2 9.5 Fix Pack 4 or later, before upgrade operating system to SUSE Linux Enterprise Server 11.
- If upgrading pre-DB2 10.1 copy on POWER3 processor-based systems, first upgrade to POWER4 processor-based systems before upgrading to DB2 10.1. POWER3 processor-based systems are not supported in DB2 10.1.
- Review installation requirements for DB2 servers and IBM data server clients to determine whether operating system version is supported and if need to upgrade operating system before installing DB2 10.1.
 - **db2prereqcheck** - checks whether system meets installation prerequisites
- For details regarding known issues, see:
 - AIX: www.ibm.com/support/docview.wss?uid=swg21165448
 - HP-UX: www.ibm.com/support/docview.wss?uid=swg21257602
 - Linux: <http://www.ibm.com/software/data/db2/linux/validate/>
 - Solaris: www.ibm.com/support/docview.wss?uid=swg21257606



© Copyright IBM Corporation 2012

9

The supported UNIX, Linux and Windows operating systems have changed in DB2 Version 10.1. Review the installation requirements for DB2 servers and IBM data server clients to determine whether your operating system version is supported and if you need to upgrade your operating system before installing DB2 10.1. Newer versions of operating systems can also bring new hardware requirements.

Performing hardware and operating system upgrades separately from DB2 database product upgrade simplifies problem determination if you encounter upgrade difficulties. If you upgrade your software or hardware before a DB2 database product upgrade, ensure that your system is operating as expected before attempting to upgrade your DB2 database product.

If you have a DB2 9.5 copy on SUSE Linux Enterprise Server 10, first apply DB2 9.5 Fix Pack 4 or later, before you upgrade the operating system to SUSE Linux Enterprise Server 11.

If you are upgrading a pre-DB2 10.1 copy on POWER3 processor-based systems, first upgrade to POWER4 processor-based systems before upgrading to DB2 10.1. POWER3 processor-based systems are not supported in DB2 Version 10.1.

Before you install DB2 database products on AIX operating systems, ensure that the system you choose meets the necessary operating system, hardware, software, and communications requirements. The db2prereqcheck command checks whether your system meets the prerequisites for the installation.

db2prereqcheck command

- Check if system meets DB2 10.1.0.0 prerequisites:
 - **db2prereqcheck -v 10.1.0.0 -s**
- **-v** *<version>*
- **-p** Checks prerequisites for DB2 pureScale environment
- **-s** Print prerequisite validation summary on screen
- **-c** Checks the prerequisites for thin client
- **-o** *<report-file>*
- Uses resource XML file that contains prerequisites. Default path of XML file located in:
 - *<DB2 installation>/cfg/DB2prereqs.xml*

© Copyright IBM Corporation 2012

10

Checks whether your system meets the prerequisites for the installation of a specific version of DB2 Database for Linux, UNIX, and Windows. By using this command, you can determine whether your system satisfies the prerequisites before you download DB2 Database for Linux, UNIX, and Windows and start the installation process. The db2prereqcheck command uses a resource XML file that contains the prerequisites. The default path of the XML file is located in DB2 installation/cfg/DB2prereqs.xml. You must have read or write permissions on the XML file. Do not modify the contents of the XML file.

- **-c** Checks the prerequisites for thin client. The **-c**, **-u**, **-g**, and **-p** parameters are mutually exclusive.
- **-u** Checks the uDAPL requirement for the DB2 pureScale® environment (Linux and AIX only)
- The **-c**, **-u**, **-g**, and **-p** parameters are mutually exclusive. On RoCE networks, you must ensure the AIX and uDAPL software prerequisites are satisfied. The db2prereqcheck command does not automatically validate these levels.
- **-g** Checks the GPL compilation requirement (Linux only). The **-c**, **-u**, **-g**, and **-p** parameters are mutually exclusive.
- **-p** Checks the prerequisites for the DB2 pureScale environment (Linux and AIX only). The **-c**, **-u**, **-g**, and **-p** parameters are mutually exclusive.
- **-t** network configuration type. The **-t** parameter validates prerequisites requirement for a specific type of network configuration (Linux only). The network configuration type must be SINGLE_IB_PORT_CLUSTER, MULT_IB_PORT_CLUSTER, SINGLE_ROCE_PORT_CLUSTER or MULTI_ROCE_PORT_CLUSTER.
- **-s** Print prerequisite validation summary on screen.
- **-i** Checks the prerequisites for the latest DB2 version that is defined in the resource XML file
- To display the latest version number in the XML file, run **-i** with **-s**. The **-i** and **-v** parameters are mutually exclusive.
- **-v** version - Checks the prerequisites for the specific DB2 version. You can use this parameter with other db2prereqcheck command parameters. The **-i** and **-v** parameters are mutually exclusive
- **-f** xml-file - Specifies the name of the XML resource file. The default XML file (with file path : DB2 installation/cfg/DB2prereqs.xml) will be used if the **-f** parameter is not specified.
- **-o** report-file - Specifies a name for the report file. For example, db2prereqcheck.rpt.

INSTALLATION METHODS

- Use **db2setup** command or **response file** installation method
 - **`./db2setup -l /temp/db2setup.log -t /tmp/db2setup.trc`**
 - **`./db2setup -r responsefile_directory/response_file`**
 - **-c** validates contents of response file before install fix pack. Ensures contents of response file correct before installation.



- **db2_install** is deprecated and might be removed in future release
- **GC27-3884-00 Installing DB2 Servers**
- **GC27-3883-00 Installing IBM Data Server Clients**
- **GI13-2047-00 Getting Started with DB2 Installation on Linux and Windows**

© Copyright IBM Corporation 2012

11

The **db2setup** command queries the operating system to determine the existing language settings. If the language setting of your operating system is supported by **db2setup**, then that language will be used when displaying the DB2 Setup wizard. If your system uses the same code pages but different locale names than those supported by the DB2 interface, you can still see the translated **db2setup** by setting your **LANG** environment variable to the appropriate value.

To start the IBM DB2 Setup Launchpad, run the **db2setup** command from the directory where the DB2 installation image resides:

- **`./db2setup -l /temp/db2setup.log -t /tmp/db2setup.trc`**
- The **-l** and **-t** recommended parameters enable DB2 logging and tracing.

To create a response file use the DB2 Setup wizard:

- Type the **db2setup** command to launch the DB2 Instance Setup Wizard.
- On the Select installation, response file creation or both panels, select the **Save my installation settings in a response file** option.

After creating a response file, you can make changes to the response file to activate or deactivate keywords.

- **`./db2setup -r responsefile_directory/response_file`**

DB2 Setup wizard: Installing Windows servers

1. Log on to system with local Administrator account defined for DB2 installation.
2. If have DB2 database product DVD, insert it into drive. If enabled, autorun feature automatically starts DB2 Setup Launchpad. If autorun does not work, use Windows Explorer to browse DB2 database product DVD and double-click setup icon to start DB2 Setup Launchpad.
3. If downloaded DB2 database product from Passport Advantage, run executable file to extract DB2 database product installation files. Use Windows Explorer to browse DB2 installation files and double-click setup icon to start DB2 Setup Launchpad.
4. From DB2 Setup Launchpad, can view installation prerequisites and release notes, or can proceed directly to installation.
5. Click Install a Product and the Install a Product window displays products available for installation.
6. If have no existing DB2 database products installed on computer, launch installation by clicking Install New. Proceed through installation following DB2 Setup wizard prompts.
7. If have at least one existing DB2 database product installed on computer, can:
 - Click Install New to create new DB2 copy.
 - Click Work with Existing to update existing DB2 copy, to add function to existing DB2 copy, upgrade existing DB2 9.5 and 9.7 copy, or to install add-on product.
8. DB2 Setup wizard determines system language, and launches setup program for that language. Online help is available to guide you through remaining steps. To invoke online help, click Help or press F1. Can click Cancel at any time to end installation.
9. Sample panels when using DB2 setup wizard lead you to installation process.
10. IBM Data Studio can be installed by running the DB2 Setup wizard.

© Copyright IBM Corporation 2012

12

Your DB2 database product is installed, by default, in the *Program_Files\IBM\sqllib* directory, where *Program_Files* represents the location of the Program Files directory.

If you are installing on a system where this directory is already being used, the DB2 database product installation path has *_xx* added to it, where *xx* are digits, starting at 01 and increasing depending on how many DB2 copies you have installed.

You can also specify your own DB2 database product installation path.

For information about errors encountered during installation, review the installation log file located in the *My Documents\DB2LOG* directory. The log file uses the following format: *DB2-ProductAbrev-DateTime.log*, for example, *DB2-ESE-Tue Apr 04 17_04_45 2012.log*.

If this is a new DB2 product installation on Vista 64-bit, and you use a 32-bit OLE DB provider, you must manually register the *IBMDADB2.DLL*. To register this DLL, run the following command:

```
c:\windows\SysWOW64\regsvr32 /s c:\Program_Files\IBM\SQLLIB\bin\ibmdadb2.dll
```

where *Program_Files* represents the location of the Program Files directory.

If you want your DB2 database product to have access to DB2 documentation either on your local computer or on another computer on your network, then you must install the *DB2 Information Center*. The *DB2 Information Center* contains documentation for the DB2 database system and DB2 related products. By default, DB2 information is accessed from the web if the *DB2 Information Center* is not locally installed.

DB2 Express® Edition and DB2 Workgroup Server Edition memory limits

If you are installing DB2 Express Edition, the maximum allowed memory for the instance is 4 GB.

If you are installing DB2 Workgroup Server Edition, the maximum allowed memory for the instance is 64 GB. The amount of memory allocated to the instance is determined by the *INSTANCE_MEMORY* database manager configuration parameter.

Important notes when upgrading from Versions 9.5 or 9.7:

The self tuning memory manager does not increase your overall instance memory limit beyond the license limits.

DB2 Setup wizard: Installing Linux/UNIX servers

- If have physical DB2 database product DVD, change to directory where DB2 database product DVD is mounted by entering: `cd /dvdrom` where */dvdrom* represents mount point of DB2 database product DVD.
- If downloaded DB2 database product image, extract product file.
 - `gzip -d product.tar.gz` where *product* name of the downloaded product.
- Untar on Linux:
 - `tar -xvf product.tar`
- Untar on AIX, HP-UX, and Solaris:
 - `guntar -xvf product.tar` where *product* name of the downloaded product.
- Change directory: `cd ./product` where *product* name of the downloaded product.
- **Note:** If downloaded National Language Package, untar into same directory which creates subdirectories (for example *./nlpack*) in same directory, and allows installer to automatically find installation images without prompting.
- Enter `./db2setup` from directory where database product image resides to start DB2 Setup wizard.
- IBM DB2 Setup Launchpad opens. From window, can view installation prerequisites and release notes, or can proceed directly to installation. Can also review installation prerequisites and release notes for late-breaking information.
- Click Install a Product and Install a Product window will display products available for installation.
- Launch installation by clicking Install New. Proceed through installation following DB2 Setup wizard's prompts.
- Sample panels when using DB2 setup wizard will lead you to installation process.
- After have initiated installation, proceed through DB2 Setup wizard installation panels and make selections. Installation help is available to guide through remaining steps. To invoke installation help, click Help or press F1. Can click Cancel at any time to end installation.

© Copyright IBM Corporation 2012

13

For non-root installations, DB2 database products are always installed in the *\$HOME*/sqlib directory, where *\$HOME* represents the non-root user's home directory. For root installations, DB2 database products are installed, by default, in one of the following directories:

- **AIX, HP-UX, and Solaris:** `/opt/IBM/db2/V10.1`
- **Linux:** `/opt/ibm/db2/V10.1`

If you are installing on a system where this directory is already being used, the DB2 database product installation path will have `_xx` added to it, where `_xx` are digits, starting at 01 and increasing depending on how many DB2 copies you have installed.

You can also specify your own DB2 database product installation path.

DB2 installation paths have the following rules:

- Can include lowercase letters (a–z), uppercase letters (A–Z), and the underscore character (_)
- Cannot exceed 128 characters
- Cannot contain spaces
- Cannot contain non-English characters

The installation log files are:

- The DB2 setup log file. This file captures all DB2 installation information including errors.
 - For root installations, the DB2 setup log file name is `db2setup.log`.
 - For non-root installations, the DB2 setup log file name is `db2setup_username.log`, where *username* is the non-root user ID under which the installation was performed.
- The DB2 error log file. This file captures any error output that is returned by Java (for example, exceptions and trap information).
 - For root installations, the DB2 error log file name is `db2setup.err`.
 - For non-root installations, the DB2 error log file name is `db2setup_username.err`, where *username* is the non-root user ID under which the installation was performed.

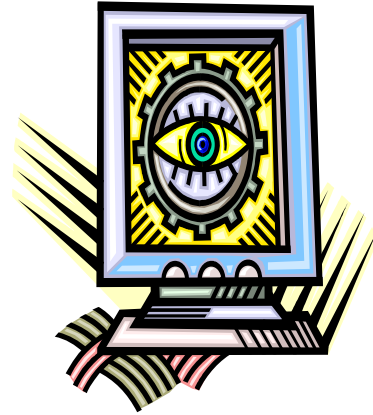
By default, these log files are located in the `/tmp` directory. You can specify the location of the log files.

There is no longer a `db2setup.his` file. Instead, the DB2 installer saves a copy of the DB2 setup log file in the `DB2_DIR/install/logs/` directory, and renames it `db2install.history`. If the name already exists, then the DB2 installer renames it `db2install.history.xxxx`, where `xxxx` is 0000-9999, depending on the number of installations you have on that machine.

Each installation copy has a separate list of history files. If an installation copy is removed, the history files under this install path will be removed as well. This copying action is done near the end of the installation and if the program is stopped or aborted before completion, then the history file will not be created.

Verify installation using CLP – SAMPLE Database

1. Create new 10.1 instance.
#DB2DIR/instance/db2icrt -u db2fenc1 db2inst1
2. Log on to system as user with SYSADM authority.
3. Start database manager
 - **db2start**
4. Create the SAMPLE database.
 - **db2sampl**
 - Can take a few minutes to process. No completion message; when command prompt returns, process is complete.
 - SAMPLE database is automatically cataloged with database alias SAMPLE when created.
 - On non-partitioned created with XML objects. On partitioned database can use **-xml** option to create XML objects.
5. Connect to SAMPLE database, retrieve a list of employees that work in department 20, and reset the database connection.
 - **db2 connect to sample**
 - **db2 "select * from staff where dept = 20"**
 - **db2 connect reset**
6. After verified installation, remove SAMPLE database to free disk space.
 - **db2 drop database sample**
 - **/opt/IBM/db2/copy1/instance/db2idrop db2inst1**



You can verify the installation by creating the SAMPLE database and running SQL commands to retrieve sample data.

The SAMPLE database component, found in the features selection, must be installed on your system and is included in a typical installation.

You require a user with SYSADM authority.

Student Notes: Reference Only

In non-partitioned database environments:

- Creates a database named SAMPLE with a Unicode (UTF-8) code set in the default database path.
- Creates relational database objects including tables, indexes, constraints, triggers, functions, procedures, multi-dimensional clustered tables and materialized query tables.
- Populates relational tables with data.
- Creates tables with XML data type columns.
- Creates indexes over XML data.
- Creates an XML schema repository that contains XML schema documents.
- All database object names are qualified with the value of the CURRENT_SCHEMA special register.

In partitioned database environments:

- Creates a database named SAMPLE with a Unicode (UTF-8) code set in the default database path.
- Creates relational database objects including tables, indexes, constraints, triggers, functions, procedures, multi-dimensional clustered tables and materialized query tables.
- Populates tables with data.
- All database object names are qualified with the value of the CURRENT_SCHEMA special register.
- The **-xml** option can be used to create tables with columns of data type XML, creates indexes on the XML columns, registers XML schemas, and populates these tables with data including XML document values. This option is only supported where XML is supported. If XML is not supported, this option is ignored.

Checklist: First steps after installation

1. [Deleting a Firefox browser profile](#)

After run DB2 Setup wizard or DB2 First Steps, may want to delete browser profile to prevent JavaScript from running automatically when visiting other Web pages with this profile.

2. [Applying DB2 licenses](#)

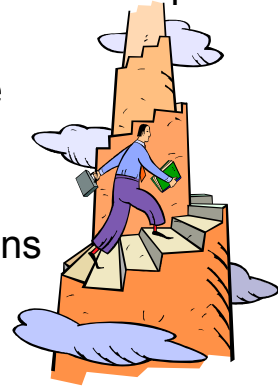
To license your product, register appropriate license key with DB2 database product.

3. [First Steps interface](#)

Launches First Steps interface, which contains links to functions that you need to begin learning about and using DB2 product.

4. [Post-installation tasks for DB2 database servers](#)

After installing DB2 database, you must complete a number of tasks.



After you install a DB2 database product, there are additional necessary and optional tasks available.

1. Deleting Firefox browser profile:

- If any Firefox browsers are open, close them.
- Open the Firefox Profile Manager.
 - On Windows
 - Open the Windows Start menu and select Run...
 - Type `firefox.exe -profilemanager` and click OK. The Firefox - Choose User Profile dialog opens.
 - On Linux and UNIX
 - Open a command prompt.
 - Switch to the Firefox directory.
 - Start Firefox with the `-profilemanager` switch. For example:
- `./firefox -profilemanager`
- Select the DB2_FIRSTSTEPS profile.
- Click Delete Profile...
- Click Exit to close the Firefox - Choose User Profile dialog.



© Copyright IBM Corporation 2012

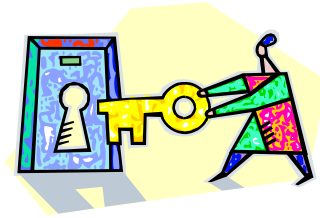
17

If you use this profile to view other Web pages that have JavaScript, the JavaScript will run automatically without warnings.

If you use Firefox as your Web browser, a browser profile is created when you run DB2 First Steps. If you agree, a browser profile named DB2_FIRSTSTEPS is created.

2. Applying DB2 licenses – db2licm

- If want license key added automatically during product installation, copy license key to /db2/license directory of installation image before launching DB2 Setup wizard.
- If do not add any license files to /db2/license directory, see *"License not registered"* message from **db2licm -l** command.
- Can apply licenses manually after installation by running **db2licm -a**.
- If register valid license key, **db2licm -l** command successfully lists all products with available license information.



© Copyright IBM Corporation 2012

18

To license your product, register the appropriate license key with your DB2 database product.

If you want the license key added automatically during product installation, copy the license key to the /db2/license directory of the installation image before launching the DB2 Setup wizard.

For example, ensure you add the db2ese_c.lic or db2ese_u.lic license files to the /db2/license directory of the installation image before installing DB2 Enterprise Server Edition. If you do not add any license files to the /db2/license directory, you will see a "License not registered" message in the output from the db2licm -l command.

You can apply the licenses manually after the installation by running the db2licm -a command. If you register a valid license key, the db2licm -l command will successfully list all the products with available license information.

To use DB2 features or to be in compliance with your current product entitlement, you might be required to register additional license keys.

3. First Steps interface - db2fs

- Manually start First Steps using **db2fs**:
 - **\$ db2fs**
- Create sample database.
- Check for product updates.
- **Linux and UNIX**
 - db2fs command located in *INSTHOME*/sql/lib/bin directory, where *INSTHOME* is instance home directory. SYSADM authority required.
 - Alternatively, on Linux operating systems can start First Steps from Main Menu by selecting IBM DB2 > First Steps.
- **Windows**
 - db2fs command located in *DB2DIR*\bin directory, where *DB2DIR* is set to location specified during DB2 installation.
 - Alternatively, can start First Steps from Start menu by selecting Programs > IBM DB2 > [DB2 Copy Name] > Set Up Tools > First Steps or from DB2 database product folder on desktop.



© Copyright IBM Corporation 2012

19

Launch the First Steps interface, which contains links to the functions that you need to begin learning about and using the DB2 product.

From First Steps you can:

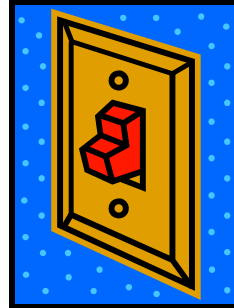
- Create the sample database or your own database.
- Check for product updates.

Product updates

Ensure that you are aware of the available product updates and enhancements for DB2 database products. With the update service you can view a list of available DB2 database product updates, and learn details about DB2 database product updates.

4. Post-installation tasks: DB2 Windows servers

- Add users to DB2ADMNS and DB2USERS user groups
- Update 32-bit DB2 instances to 64-bit instances
- Validating DB2 copy: **db2val**
- Set up notification and contact lists
- Change default DB2 and default IBM database client interface copy after installation:
db2swtch -l
db2swtch -d DB2_copy_name
- Checking for DB2 updates
- Installing SSH Server component after installing DB2 product
 - required to use Optim tools
 - <http://www.ibm.com/developerworks/data/library/techarticle/dm-0506finnie/index.html>



© Copyright IBM Corporation 2012

20

After installing DB2 database on Windows systems, you must complete a number of tasks.

•Adding your user ID to the DB2ADMNS and DB2USERS user groups (Windows)

•After successfully completing a DB2 installation, you now have to add users to the DB2ADMNS or the DB2USERS groups for users that need to run local DB2 applications and tools on the machine. The DB2 installer creates two new groups. You can either use a new name or accept the default names. The default group names are DB2ADMNS and DB2USERS. This process is also applicable to instance based clients.

•Updating your 32-bit DB2 instances to 64-bit instances (Windows)

•If you have more than one 32-bit DB2 copy on your system, you can convert them to 64-bit instances.

•Validating your DB2 copy

•The **db2val** command ensures that your DB2 copy is functioning properly.

•Setting up notification and contact lists

•For your DB2 database product to contact you or others regarding the status of your databases, set up notification and contact lists. If you did not do this during the DB2 database product installation, you can manually set up these lists.

•Changing the default DB2 and default IBM database client interface copy after installation (Windows)

•The **db2swtch** command can be run from any DB2 copy or IBM data server driver copy. If no arguments, launches the utility in graphical mode. The **-l** displays a list of DB2 copies and IBM data server driver copies on the system. To switch the default DB2 copy to the name specified: **db2swtch -db2 -d DB2_copy_name-client**. The **-d** specifies the name of DB2 copy or IBM data server driver copy. To switch the default client interface copy to the name specified: **db2swtch -client -d name of DB2 copy or IBM data server driver copy-d DB2_copy_name**. To switch both default DB2 copy and client interface copy to the name specified:
db2swtch -d DB2_copy_name.

•IBM data server client connectivity using multiple copies

•When using multiple DB2 copies of DB2 database products or multiple data server driver copies, various options are available for applications to access the DB2 databases. Existing applications will continue to function properly.

•Working with existing DB2 copies

•You can install and run multiple DB2 copies on the same computer. Each DB2 copy can either be at the same DB2 database product level or at a different DB2 database product level.

Student Notes: For Reference Only

© Copyright IBM Corporation 2012

21

- Checking for DB2 updates**

- Check for product updates to ensure that you are aware of the available product updates and enhancements for DB2 products.

- Installing the SSH Server component after installing the DB2 product (Windows)**

- The IBM Secure Shell (SSH) Server for Windows provides a secure platform to run commands from remote machines. If you chose not to install the IBM SSH Server for Windows component while installing a DB2 server, you can install this component manually. You require this component to use Optim tools. See the following URL for details:

- <http://www.ibm.com/developerworks/data/library/techarticle/dm-0506finnie/index.html>

4. Post-installation tasks: DB2 Linux/UNIX servers

- List DB2 database products installed on system: **db2ls**
 - To list info about DB2 database products or features in particular installation path **q** parameter must be specified:
 - **db2ls -q -p -b baseInstallDirectory**
 - **q** specifies querying product or feature. Parameter is mandatory.
 - **p** specifies listing displays products rather than listing features.
 - **b** specifies installation directory of product or feature. Parameter is mandatory if not running command from installation directory.
 - Verifies the basic functions of a DB2 copy by checking state of installation files, instance setup, and local database connections.
 - To validate all instances for DB2 copy: **db2val -a**
 - Main menu entries for DB2 tools (Linux)
 - After installation, can add several DB2 tools to Main menu.

© Copyright IBM Corporation 2012

22

After installing DB2 database on Linux or UNIX systems, you must complete a number of tasks.

•Listing DB2 database products installed on your system (Linux and UNIX)

On supported Linux and UNIX operating systems, the **db2ls** command lists the DB2 database products and features installed on your system, including the DB2 Version 10.1 HTML documentation. With the ability to install multiple copies of DB2 database products on your system and the flexibility to install DB2 database products and features in the path of your choice, you need a tool to help you keep track of what is installed and where it is installed. On supported Linux and UNIX operating systems, the **db2ls** command lists the DB2 products and features installed on your system, including the DB2 HTML documentation. The **db2ls** command can be found both in the installation media and in a DB2 install copy on the system. The **db2ls** command can be run from either location. The **db2ls** command can be run from the installation media for all products except IBM® Data Server Driver Package. The **db2ls** command can be used to list:

- Where DB2 database products are installed on your system and list the DB2 database product level
- All or specific DB2 database products and features in a particular installation path.

The output that the **db2ls** command lists is different depending on the ID used:

- When the **db2ls** command is run with root authority, only root DB2 installations are queried.
- When the **db2ls** command is run with a non-root ID, root DB2 installations and the non-root installation owned by matching non-root ID are queried. DB2 installations owned by other non-root IDs are not queried.

The **db2ls** command is the only method to query a DB2 database product. You *cannot* query DB2 database products using Linux or UNIX operating system native utilities, such as **pkginfo**, **rpm**, **SMIT**, or **swlist**. Any existing scripts containing a native installation utility that you use to query and interface with DB2 installations must change. You *cannot* use the **db2ls** command on Windows operating systems.

•Validating your DB2 copy

- The **db2val** command ensures that your DB2 copy is functioning properly. The **db2val** tool verifies the core function of a DB2 copy by validating installation files, instances, database creation, connections to that database, and the state of partitioned database environments. This validation can be helpful if you have manually deployed a DB2 copy on Linux and UNIX operating systems using **tar.gz** files. The **db2val** command can quickly ensure that all the configuration has been correctly done and ensure that the DB2 copy is what you expect it to be. You can specify instances and databases or you can run **db2val** against all of the instances. The **db2val** command can be found in the *DB2-install-path*\bin and *sqlib*\bin directories

•Main menu entries for DB2 tools (Linux)

After installation, you can add several DB2 tools to the Main menu:

- Check for DB2 Updates
- Command Line Processor (CLP)
- Command Line Processor Plus (CLPPlus)
- First Steps

Student Notes: For Reference Only

© Copyright IBM Corporation 2012

23

These DB2 tools can be added to main menu automatically or manually. Main menu entries are created automatically when any of the following DB2 commands are run. The db2icrt and db2iupdt commands must be run by root. The db2nrcfg and db2nrupdt are for non-root install and are run by the instance owner.

- db2icrt
- db2iupdt
- db2nrcfg
- db2nrupdt.

To see the menu entries you might need to restart the desktop. The main menu entries are automatically removed when either of the following commands are run:

- db2_deinstall (only removes the menu entries for the DB2 non-root instance when the non-root installation is removed)
- db2idrop.

In addition, you can manually create or remove main menu entries by running the following commands:

- db2addicons – to add menu entries
- db2rmicons – to remove menu entries.

Before running the db2addicons command, the DB2 instance environment must be set for the current user. The instance environment can be setup with: *Instance_HOME*/sqllib/db2profile (for Bourne shell and Korn shell users), or *Instance_HOME*/sqllib/db2chsrc (for C shell users), where *Instance_HOME* is the instance owner's home directory.

db2val - DB2 copy validation tool

- **-o** Specifies only installation files will be validated. If specified, -i, -a, -b, and -s are ignored. Validation of instance, database, and extended security not performed
- **-i instance_name** Specifies instance to validate. For multiple instances, specify parameter multiple times.. On Windows, if not specified, current instance is default. On Linux/UNIX, parameter only be used by root users in a root installation of a DB2 copy.
- **-a** Validates all instances in DB2 copy. On Linux /UNIX, parameter only be used by root users in a root installation of a DB2 copy. Overrides parameter -i.
- **-b <db_name>** Validates database creation and connections to database specified. Only active DB2 instances will be validated. Parameter ignored for DB2 client and DB2 pureScale instances.
- **-t <trace_file>** Applies only to Linux/UNIX. Specifies full path and name of trace file.
- **-d** Deprecated and might be removed in a future release. Use the -t parameter instead.
- **-s** Starts DB2 database manager for specified instance that is part of partitioned database environment.
- **-l <log_file>** Writes log to file name specified. Default on Linux/UNIX is /tmp/db2valxx.log and on Windows My Documents\DB2LOG\db2valxx.log.
- **-? | -h** Displays usage information

© Copyright IBM Corporation 2012

24

Verifies basic functions of DB2 copy by checking state of installation files, instance setup, and local database connections.

You can use the db2val command only to validate installation files in a DB2 pureScale environment.

Instance validation requires one of the following authorities:

- On root copies, root authority is required on Linux and UNIX operating systems.
- SYSADM plus one of the following authorities:
 - Instance owner
 - Root access on Linux and UNIX operating systems, or Local Administrator authority on Windows operating systems

To validate the instance TEST1 and the database DATA1, run the following command:

db2val -i TEST1 -b DATA1

To validate all the instances for the DB2 copy, run the following command:

db2val -a

To validate only the DB2 installation files, run the following command:

db2val -o

Windows Installation Considerations

- During installation on Windows, if select **DB2 copy with upgrade action** in **Work with Existing** window and have local databases cataloged on instances, a message box warns that you must run **db2ckupgrade** from DB2 database product CD.
- Then can choose one of following actions:
 - Ignore message and continue the installation process.
 - Run **db2ckupgrade**. If command runs successfully, continue installation process. If find errors, quit installation process, fix any errors, and then rerun installation process.
 - Quit installation process.

© Copyright IBM Corporation 2012

25

Before you upgrade your DB2 server, review the upgrade essentials for DB2 servers, including recommendations, restrictions, and disk space requirements to identify the changes or restrictions that can affect your upgrade. You must be ready to address any issues before upgrade in order to have a successful upgrade.

If index root page does not have enough free space during upgrade, index will grow by 1 page. If free page cannot be found in index object, page will be requested from tablespace. If tablespace is full, entire index object marked invalid and rebuilt when the underlying table is accessed for first time after upgrade.

Because type-1 indexes are discontinued in DB2 9.7. Converting them before upgrade eliminates the overhead of index rebuild when you access tables using these indexes for the first time after upgrading to DB2 10.1.

db2ckupgrade - Check database for upgrade

db2ckupgrade <database> -l filename

- In partitioned database, db2ckupgrade will check each database partition.
- On Linux/UNIX, install new DB2 copy and run from *DB2DIR/bin* directory where *DB2DIR* is location where DB2 copy is installed
- On Windows, insert DB2 product CD and run from *db2\Windows\Utilities* directory on CD
- SYSADM authority

db2ckupgrade -e -l filename

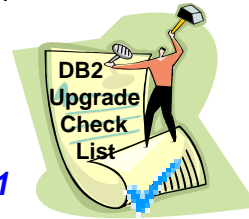
- **-e** specifies all local cataloged databases are to be scanned

db2ckupgrade <database> -l filename -not1

- **-l filename** specifies log file to keep list of errors and warnings

-not1 disables check for type-1 indexes. If not specified, checks for type-1 indexes and generates **type1_index_database-name.db2** script file, in same directory indicated for log file, containing REORG INDEXES ALL commands with the ALLOW WRITE ACCESS and CONVERT clauses for each identified type-1 index.

Type 1 indexes were not allowed on DB2 9.7 databases.



© Copyright IBM Corporation 2012

26

The **db2ckupgrade** command must be issued prior to upgrading the instance to verify that your databases are ready for upgrade. Backup all databases prior to upgrade, and prior to the installation of the current version of DB2 database product on Windows operating systems. Once the codebase has been installed you should then run the db2ckupgrade before upgrading the instance or databases. The db2ckupgrade fails to run against databases which are catalogued as remote databases.

The **db2ckupgrade** command verifies that a database can be migrated. In a partitioned database environment, the db2ckupgrade command will check each database partition.

SYSADM authority is required to execute. No database connection is required.

-not1 parameter when upgrade DB2 9.5 databases

- For Version 9.5 databases, if -not1 parameter omitted, db2ckupgrade calls db2IdentifyType1 command to identify type-1 indexes and generate script to convert type-1 indexes to type-2 indexes for specified database. db2IdentifyType1 can take a long time to complete. Running time of db2IdentifyType1 is proportional to number of tables in database and number of database partitions. Take into account following performance considerations:
 - For Version 9.5 databases with large number of tables, large number of database partitions, or both, first run db2IdentifyType1 command on specific schemas or tables by using the -s or -t parameters until you process all tables. Then run the db2ckupgrade command with the -not1 parameter.
 - For Version 9.5 partitioned database environments, run db2ckupgrade from catalog partition for faster performance, to detect all type-1 indexes.

© Copyright IBM Corporation 2012

27

Convert type-1 indexes to type-2 indexes because type-1 indexes are discontinued in DB2 9.7 and marked invalid during upgrade. Converting them before upgrade eliminates the overhead of index rebuild when you access tables using these indexes for the first time after upgrading to DB2 9.7. You should only perform this task if you know or suspect that your database has type-1 indexes.

By default, all new indexes created in pre-9.7 releases were type-2 indexes except when you created an index on a table that already had type-1 indexes, in which case the new index is also type-1. You might have type-1 indexes on databases that you created on DB2 UDB V7 or earlier and that you upgraded all the way through to DB2 9.5 or databases under an instance where the **DB2_INDEX_TYPE2** registry variable was set to OFF.

```
db2 SELECT TABSCHEMA, TABNAME FROM SYSIBMADM.ADMINTABINFO WHERE  
INDEX_TYPE = 1
```

```
db2 load query table <tablename>
```

```
db2IdentifyType1 -d database-name -o convert-t1-indexes-dbname.db2
```

```
db2 -tvf convert-t1-indexes-dbname.db2
```

db2ckupgrade verifies following conditions are true:

- Catalogued database actually exists.
- Database not in inconsistent state.
- Database not in backup pending state.
- Database not in restore pending state.
- Database not in rollforward pending state.
- Tables not in load pending state.
- Tables not in a redistribute pending state.
- For Version 9.8 or later, that all table space container paths use same mount point.
- For Version 9.8 FixPack 3 or later, that I/O write operations for database are not suspended or not being suspended.
- There are no MQT's that depend on system views.
- Table spaces are in normal state.
- Database does not contain user-defined types (UDTs) with the name ARRAY, BINARY, CURSOR, DECFLOAT, ROW, VARBINARY, or XML.
- Database does not contain built-in DATALINK data type.
- Database does not have a schema with name SYSPUBLIC.
- Database does not have orphan rows in system catalog tables that cause database upgrade to fail.
- Database enabled as an HADR primary database allows successful connections.
- An HADR database role is not standby.
- If SYSCATSPACE is DMS table space and AUTORESIZE not enabled, SYSCATSPACE has at least 50% free pages of total pages.
- A database is not enabled for XML Extender.

© Copyright IBM Corporation 2012

28

A local database must pass all of these checks to succeed at the upgrade process.

The **db2iupgrade** command implicitly calls the **db2ckupgrade** command with the **-not1** parameter, and specifies **update.log** as the log file for **db2ckupgrade**. The default log file created for **db2iupgrade** is **/tmp/db2ckupgrade.log.processID**. The **db2iupgrade** fails if the **db2ckupgrade** command finds that any of the above listed conditions are not true, and returns the DBI1205E error code. The user needs to resolve these errors before upgrading the instance.

db2chkupgrade verification (continued)

- db2chkupgrade writes warning message to log file (-l parameter) for any of following conditions:
 - Column names, routine parameter names, or variable names are called NULL.
 - Type-1 indexes exist in database.
 - Workload connection attributes contain asterisks (*).
 - Database enabled for DB2 WebSphere MQ functions.



© Copyright IBM Corporation 2012

29

For partitioned database environments, run the db2chkupgrade command before you issue the db2iupgrade command. The db2chkupgrade command will check all partitions and will return errors found in any partition. If you do not check whether all database partitions are ready for upgrade, subsequent database upgrades could fail even though the instance upgrade was successful.

Disk space requirements (1)

- **System catalog table space (SYSCATSPACE)**

- Increase to twice the total of used space.
- System catalog table space is required for both old and new database catalogs during upgrade.



- **Temporary table space (TEMPSPACE1)**

- Increasing total size to twice total size of system catalog table space.

- **db2 "SELECT SUBSTR(TBSP_NAME,1,15) NAME, TBSP_TYPE TYPE, TBSP_AUTO_RESIZE_ENABLED AUTO_RESIZE, TBSP_NUM_CONTAINERS CONTAINERS, TBSP_TOTAL_PAGES TOTAL_PGS, TBSP_USED_PAGES USED_PGS, TBSP_FREE_PAGES FREE_PGS, TBSP_MAX_SIZE MAX_SZ, TBSP_PAGE_SIZE PG_SZ FROM SYSIBMADM.TBSP_UTILIZATION WHERE TBSP_CONTENT_TYPE IN ('ANY','SYSTEMP')"**

© Copyright IBM Corporation 2012

30

You must be aware that the upgrade process requires additional disk space. Ensure that you have enough free disk space to complete this process successfully. The above disk space recommendations are applicable for upgrading to DB2 10.1.

Ensure sufficient free space on system catalog and system temporary table spaces for databases you are upgrading.

To increase the amount of free space on your System Managed Space (SMS) table spaces, free sufficient disk space on the corresponding file systems or increase the size of your file systems if you are using a volume manager.

To increase the amount of free space on your Database Managed Space (DMS) table spaces, you can increase the size of existing containers. You can also add additional containers although this might trigger data rebalancing. You can reduce the size of the containers after upgrade.

Disk space requirements (2)

- **Log file space**

- Upgrade changes to system catalog objects are single UOW.
- Set LOGSECOND to twice current value of LOGPRIMARY and LOGSECOND if file system containing log files has enough disk free space.
 - If already have large log file space, might not be necessary.
- **Only need to increase log space on catalog partition.**
- **Must update these values before upgrade the instance to DB2 10.1**
- Optional: Consider infinite logging (LOGSECOND=-1)

```
db2 GET DB CFG FOR sample | grep '(LOG[FPS])' tee logsize.txt
db2 UPDATE DB CFG FOR sample using LOGSECOND
(current_value of LOGPRIMARY + current_value of LOGSECOND)
```



- **Index space**

- Each index on every populated table requires 1 additional page per index to use following functionality:
 - Real-time statistics.
 - Deferred cleanup roll out for MDC tables.
 - Index rebuild on a populated table.
- If have type-1 indexes, will be marked invalid and recreated as type-2 indexes at restart or at first access after upgrade. Ensure have enough disk space to convert type-1 indexes.
- Type-2 indexes require one more byte per row.

- **Automatic storage files**

- If enable automatic storage on existing database by issuing ALTER DATABASE with ADD STORAGE, creates SQLSGF.1 and SQLSGF.2 files that are required for maintaining automatic storage.

© Copyright IBM Corporation 2012

31

You must update these LOGSECOND values before you upgrade the instance to DB2 10.1, because you will not be able to update these DB CFG parameters until issue the UPGRADE DATABASE command. If this command fails because there is insufficient log file space, then you can set these database configuration parameters to higher values and then re-issue the UPGRADE DATABASE command.

The new database configuration parameter settings for log space can be restored to their original value after the upgrade is complete.

If have limited amount of free disk space for indexes, can get error message SQL0289N that indicates the table space is full. Ensure that you have enough free pages in the corresponding index table space to account for one additional page per index on populated tables before:

- Populating tables in databases created in DB2 Version 9.5 or later, real-time statistics are enabled by default in these newly created databases.
- Enabling deferred cleanup roll out by setting DB2_MDC_ROLLOUT to DEFER, or when DB2_WORKLOAD is set to SAP.
- Reorganizing or recreating indexes on populated tables.

Pre-upgrade tasks for DB2 servers

1. Ensure at least 1 free page of index space per object index to eliminate overhead of potential index rebuild.
2. If use distributed transactions, ensure databases do not contain any indoubt transactions by using LIST INDOUBT TRANSACTIONS to get list of indoubt transactions and interactively resolve any indoubt transactions.
3. Convert type-1 indexes to type-2 indexes.
4. Verify databases are ready for DB2 upgrade to identify any problems before actual upgrade. Must resolve them before proceed with upgrade.
5. Optional: Stop HADR on primary and standby databases.
6. Upgrade from DB2 Query Patroller to WLM. **Query Patroller is discontinued.**
7. Back up databases to be able to upgrade them to a new upgraded system or restore them in the original pre-upgrade system.
8. Back up configuration and diagnostic information to have record of current configuration that can compare with configuration after upgrade. Use this information to create new instances or databases using same configuration that had before upgrade.
9. Archive all of DB2 log files, either for SQL or Q replication if log files needed by Capture or Q Capture, or for HADR if needed to create standby database.
10. Review disk space requirements to ensure have enough free disk space, system temporary table space and log space for the upgrade and increase table space and log file sizes if necessary.
11. Windows only: If obtained customized code page conversion tables from the DB2 support service, need to backup all files in DB2OLD\conv directory where DB2OLD is the location of your existing pre-DB2 10.1 copy.
12. Linux only: Change raw devices to block devices.
13. Optional: Upgrade DB2 server in test environment to identify upgrade issues and verify applications, scripts, tools and routines work as expected before upgrading DB2 server in production environment.
14. If diaglevel is 2 or less, set diagnostic error capture level to 3 or higher.
15. Take DB2 server offline for upgrade.
16. Refresh data in existing MQTs. All MQTs that depend on system views are dropped during database upgrade. After upgrade, must refresh data in existing MQTs by using REFRESH TABLE.

© Copyright IBM Corporation 2012

32

If you use distributed transactions involving DB2 databases, ensure that the databases to be upgraded do not contain any indoubt transactions by using the LIST INDOUBT TRANSACTIONS command to get a list of indoubt transactions and to interactively resolve any indoubt transactions.

In DB2 10, all significant upgrade events are logged in the **db2diag** log files when the **diaglevel** database manager configuration parameter is set to 3 (default value) or higher. If this parameter is set to 2 or less, set this parameter to 3 or higher before upgrade.

Back up databases before AND after upgrade

After upgrade instances to DB2 10.1, cannot backup databases until upgrade them!

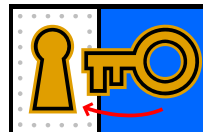
- Perform a full offline back up for each of your local databases:

1. To disconnect all applications and users:

db2 FORCE APPLICATION ALL

2. Backup database using BACKUP DATABASE.

db2 BACKUP DATABASE database_alias TO backup-dir



- In **partitioned database, back up all database partitions.**
- If activated and configured DB2 Advanced Copy Services (ACS) on DB2 9.5 or later, use the USE SNAPSHOT parameter to perform snapshot backup.
 - Can only restore snapshot backup to instance of same version.
 - Cannot use snapshot backup to upgrade to new version.
- If performed full online or offline database backup recently and cannot perform another one before upgrading, can perform incremental offline database backup instead if **TRACKMOD=ON**.

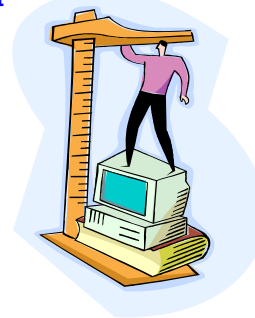


Before and after the upgrade process to DB2 10.1, it is strongly recommended that you perform a full *offline* database backup. If an error occurs during the upgrade process, you need full database backups to recover and upgrade your databases.

After you upgrade your instances to DB2 10.1, you cannot backup databases until you upgrade them.

Backing up DB2 server configuration and diagnostic info

1. Collect information from DB2 servers by running db2support command for all databases going to upgrade or convert to DB2 pureScale, in all your instances. Collect information about database system catalog, DB CFG and DBM CFG settings, DB2 registry variables settings, explain table data, and diagnostic information required by DB2 support in case of problems.
db2support output-directory -d database-name -cl 0
2. Back up information about all packages for applications associated with each database.
db2 LIST PACKAGES FOR ALL SHOW DETAIL > /upgrade/sample_pckg.txt
3. If enabled audit facility, back up audit cfg for each instances:
db2audit describe > audit_instance-name.cfg
4. Back up all external routines.
cp -R \$INSTHOME/sqllib/function \$INSTHOME/routine_backup
Optional Steps: db2support HTML report includes following info.
5. Use GET DBM CFG to back up settings for each instance:
db2 GET DBM CFG > dbm_instname.cfg
6. Use GET DB CFG to back up settings for each database.
If not same settings, back up DB CFG for each database partition.
db2 CONNECT TO database_alias
db2 GET DB CFG FOR database_alias SHOW DETAIL > db_database_alias.cfg
7. Use db2look to save additional information.
db2look -d sample -e -o sample_tbs.db2 -l -x
8. Use db2set command to back up DB2 profile registry variables settings:
db2set -all > reg_instname.txt
9. If set DB2 environment variables, use appropriate system command to list environment variables values.
set | grep DB2 > env_instname.txt (AIX example)



© Copyright IBM Corporation 2012

34

Backing up your settings for database and database manager configuration parameters before DB2 server upgrade, or conversion to DB2 pureScale, allows you to verify DB2 server behavior after upgrade, or converting to DB2 pureScale, and to re-create instances and databases.

In addition, you can collect information from your DB2 servers about the database system catalogs, DB2 registry variables settings, explain table data, and diagnostic information that can help in problem determination if you encounter any post-upgrade differences in the database manager behavior or performance.

The db2support -cl 0 parameter collects the database system catalog, database and database manager configuration parameters settings, DB2 registry variables settings. The information collected is stored in the db2support.zip compressed zip file under the output directory. A summary report in HTML format is included. In the db2supp_opt.zip file that is also included, you should check the optimizer.log file to verify that the collection of information was performed successfully.

Keep this zip file for several months after you complete the upgrade, or conversion to DB2 pureScale. The information in the zip file can help in quickly resolving any performance issues with the new release.


The LIST PACKAGES ... FOR SCHEMA clause allows you to list all packages for a specific schema, if your application has several schemas you need to repeat this command for each schema name or use the FOR ALL clause.

When backing up external routines, INSTHOME is set to home directory of instance owner. If you have specified a full path that is not under the default routines path when you created your external routines in the database, you must ensure the existing libraries remain on their original location.

The SHOW DETAIL clause on GET DB CFG displays the values calculated by the database manager when configuration parameters are set to AUTOMATIC.

When possible, use the output from the set command and run the db2set command to set these environment variables as registry variables in the DB2 profile registry.

Instance upgrade - db2iupgrade

- Call instance upgrade explicitly with **db2iupgrade** command
- Call implicitly when install DB2 10.1 on Windows and select **Work with Existing option** and choose a pre-10.1 copy with upgrade action
- db2iupgrade does following things:
 - Upgrades existing instance to new instance under DB2 10.1 copy.
 - Upgrades instance profile registry variables.
Global profile registry variables are not upgraded. 
 - Upgrades dbm cfg file.
 - Sets jdk_path dbm cfg parameter.
 - Upgrades db2audit.cfg audit cfg file when audit facility enabled.
 - Uses SSLconfig.ini SSL configuration file to set new dbm cfg parameters to corresponding SSL parameter value in this file and upgrades instance profile registry setting DB2COMM=SSL.
 - For successful instance upgrade, all files must exist for all instances and all files must have write access granted.

© Copyright IBM Corporation 2012

35

Only DB2 Enterprise Server Edition instances (instance type *ese*) can be upgraded using the db2iupgrade command.

The db2iupgrade command calls the db2ckupgrade command with the -not1 parameter, and specifies update.log as the log file for db2ckupgrade. The default log file created for db2iupgrade is /tmp/db2ckupgrade.log.processID. Verify that local databases are ready for upgrade before upgrading the instance. The -not1 parameter disables the check for type-1 indexes. The log file is created in the instance home directory for Linux and UNIX operating systems or in the current directory for Windows operating systems. The instance upgrade will not continue if the db2ckupgrade command returns any errors.

For Linux and UNIX operating systems

If you use the db2iupgrade command to upgrade a DB2 instance from a previous version to the current version of a DB2 database system, the DB2 Global Profile Variables defined in an old DB2 database installation path will not be upgraded over to the new installation location. The DB2 Instance Profile Variables specific to the instance to be upgraded will be carried over after the instance is upgraded.

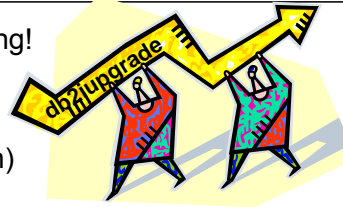
If you are using the su command instead of the login command to become the root user, you must issue the su command with the - option to indicate that the process environment is to be set as if you had logged in to the system using the login command.

You must not source the DB2 instance environment for the root user. Running the db2iupgrade command when you sourced the DB2 instance environment is not supported.

On AIX 6.1 (or higher), when running this command from a shared DB2 copy in a system workload partition (WPAR) global environment, this command must be run as the root user. WPAR is not supported in a DB2 pureScale environment.

Upgrading to DB2 10 Instance – db2iupgrade

- **BACKUP DATABASE** before upgrading!
- db2stop force
- db2 terminate
- **cd /opt/ibm/db2/V10** (code install path)
- **db2iupgrade –u fencedid inst1**
 - Root access on Linux/UNIX or Local Administrator on Windows
 - Calls **db2ckupgrade** with **–e, –not1** and **–l db2ckupgrade.log**
 - Fails if db2ckupgrade fails



```
>>db2iupgrade -d -k -a--AuthType- -u--FencedID--InstName-><
```

- d turns debug mode on. Use when instructed by DB2 Support.
- k keeps pre-upgrade instance type. If not specified instance is upgraded if supported.
- a specifies the AUTHENTICATION type. Default is SERVER.
- u specifies user id which fenced UDF and stored procedures run.
- j configures DB2 Text Search server's service name and port number. See student notes for details.

For Windows, use / instead of – before options: **db2iupgrade /d inst1**

© Copyright IBM Corporation 2012

36

Before starting the instance migration, the instance must be stopped. You must be logged in as root to perform the majority of tasks. It is important that the DB2 runtime instance environment, normally established using the db2profile script, has not been set. If it is, the instance migration commands will not work. The db2iupgrade command upgrades an instance to a DB2 copy of the current release from a DB2 copy of a previous release. The DB2 copy from where you are running the db2iupgrade command must support instance upgrade from the DB2 copy that you want to upgrade.

On Linux and UNIX, this command is located in the DB2DIR/instance directory, where DB2DIR represents the installation location where the new release of the DB2 database system is installed. This command does not support instance upgrade for a non-root installation. On Windows, this command is located in the DB2PATH\bin directory, where DB2PATH is the location where the DB2 copy is installed. To move your instance profile from its current location to another location, use the /p option and specify the instance profile path. Otherwise, the instance profile will stay in its original location after the upgrade.

The db2iupgrade command calls the db2ckupgrade command with the -not1 parameter and specifying db2ckupgrade.log as the log file. Verify that local databases are ready for upgrade before upgrading the instance. The -not1 parameter disables the check for type-1 indexes.

A log file giving details of what took place is written to the instance home directory (UNIX-like systems) or to the current directory path (Windows). If any errors are reported, check this log file for further details.

-j "TEXT_SEARCH"

Configures DB2 Text Search server using generated default values for service name and TCP/IP port number. Cannot be used if the instance type is client.

-j "TEXT_SEARCH, servicename"

Configures DB2 Text Search server using provided service name and automatically generated port number. If service name has port number assigned in services file, it uses assigned port number.

-j "TEXT_SEARCH, servicename, portnumber"

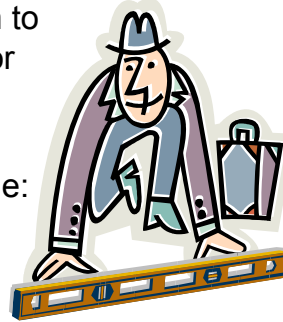
Configures DB2 Text Search server using provided service name and port number.

-j "TEXT_SEARCH, portnumber"

Configures DB2 Text Search server using default service name and provided port number. Valid port numbers must be within 1024 - 65535 range.

Validating Instance Upgrade

- Review **db2iupgrade** log file written to UNIX/Linux instance home directory or Windows current path
- Switch to the instance owner and issue:
 - db2start**
 - ✓ **db2level**
- **DB21085I Instance "inst1" uses "64" bits and DB2 code release "SQL10010" with level identifier "08010107". Informational tokens are "DB2 v10.1.0.0", "s090521", "LINUXIA3297", and Fix Pack "0". Product is installed at "/opt/ibm/db2/V10.1".**
- On Windows, db2level shows DB2 copy name.



© Copyright IBM Corporation 2012

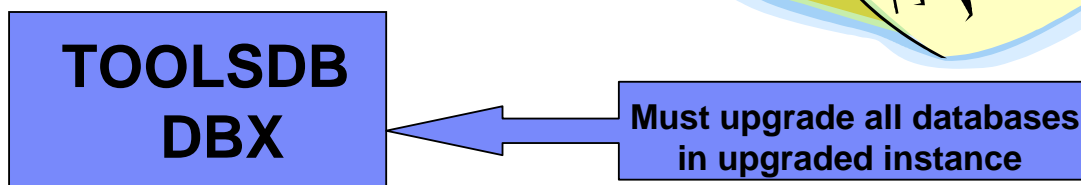
37

The db2iupgrade log file is created in the instance home directory for Linux and UNIX operating systems or in the current directory for Windows operating systems. The instance upgrade will not continue if the db2ckupgrade command returns any errors.

If the db2iupgrade command completes successfully, switch to the instance owner and issue the **db2start** to restart the instance. To confirm you are now running DB2 10.1 issue the **db2level** command.

Database directory upgrade

- During first time access, database directory is implicitly upgraded if necessary.
- Database directory is accessed when issue:
 - **LIST DATABASE DIRECTORY**
 - **UPGRADE DATABASE**
- Obtain list of all local databases that must be upgraded:
db2 list db directory > db-dir-before.txt



© Copyright IBM Corporation 2012

38

The database directory is implicitly upgraded if necessary during first time access.

After the instance upgrade has been completed, all databases need to be upgraded to the latest version. Obtain a list of all local databases using the list database directory command.

Database upgrade

- Call explicitly using **UPGRADE DATABASE** command
- Following database entities converted during database upgrade:
 - Database configuration file
 - Log file header
 - Table root page for all tables
 - Index root page for all tables
 - Catalog tables
 - Buffer pool files
 - History file
- For recoverable databases, UPGRADE DATABASE renames all log files in active log path with the extension .MIG. After upgrade databases successfully, can delete all S*.MIG files.
- UPGRADE DATABASE upgrades files SQLSPCS.1, SQLSPCS.2 , SQLSGF.1, and SQLSGF.2 to support new functionality on automatic storage table spaces such as removing storage paths from database and rebalancing automatic storage table spaces after add or drop storage paths from database.

For details about log migration, see [Post-upgrade tasks for DB2 servers](#) discussed later in this topic.

Upgrade database - runstats on system tables

- UPGRADE DATABASE automatically collects statistics for all system catalog tables during database upgrade.
- *Which RUNSTATS is called for automatic statistics collection?*

auto_runstats	User profile	RUNSTATS command
Enabled	Exists	RUNSTATS command with SET PROFILE parameter using information in STATISTICS_PROFILE column in SYSCAT.TABLES.
Enabled	Does not exist	RUNSTATS command with default parameters
Disabled	N/A	RUNSTATS command from most recent call to RUNSTATS command. ¹

¹ If statistics previously collected for table, RUNSTATS is issued as indicated in table. If there are no statistics collected for table, RUNSTATS is not issued.

© Copyright IBM Corporation 2012

40

The automatic collection of statistics for all system catalog tables ignores any exclusion policies defined in the health monitor. Also, if you have manually modified your system catalog table statistics via updates to SYSSTATS views, manually reissue these updates to the SYSSTATS views.

Upgrading Local Databases to DB2 10

- **db2 UPGRADE DATABASE *database_alias* -REBINDALL**

- Instance owner authority required
- **REBINDALL** rebinds all packages during upgrade and ensures step not missed and that applications not started before completed

db2 UPGRADE DATABASE TOOLSDB -REBINDALL

db2 UPGRADE DATABASE DBX -REBINDALL



```
db2 CONNECT to <dbname>
db2 GET DB CFG FOR <dbname> SHOW DETAIL
> <dbname>-dbcfg-after.txt
db2 LIST TABLESPACES SHOW DETAIL
> <dbname>-tsp-after.txt
db2 LIST PACKAGES FOR ALL SHOW DETAIL
> <dbname>-pkg-after.txt
db2 CONNECT RESET
...
Compare before and after txt files.
```

© Copyright IBM Corporation 2012

41

The **db2ckupgrade** command must be issued prior to upgrading the instance to verify that your databases are ready for upgrade. The **db2iupgrade** command implicitly calls the **db2ckupgrade**. Backup all databases prior to upgrade, and prior to the installation of the current version of DB2 database product on Windows operating systems.

The **upgrade database** command will only upgrade a database to a newer version, and cannot be used to convert an upgraded database to its previous version. The database must be cataloged before upgrade.

If an error occurs during upgrade, it might be necessary to issue the **TERMINATE** command before attempting the suggested user response. For example, if a log full error occurs during upgrade (SQL1704: Database upgrade failed. Reason code "3"), it will be necessary to issue the **TERMINATE** command before increasing the values of the database configuration parameters **LOGPRIMARY** and **LOGFILSIZ**. The CLP must refresh its database directory cache if the upgrade failure occurs after the database has already been relocated (which is likely to be the case when a "log full" error returns).

After each database has been upgraded repeat the statistics gathering as above and compare the two sets of information. With the exception of the system packages, the two sets of information should be identical.

The new **REBINDALL** option specifies that a **REBIND** of all packages is performed during upgrade. Performing the **REBINDs** automatically during the upgrade ensures that the step is not missed and will help ensure that other applications are not started before the **REBINDs** are complete.

Upgrading Tools Catalog and Runstats

- **db2tdbmgr -d TOOLSDB -s SYSTOOLS**
 - Upgrades Tools Catalog
 - Instance owner authority required
- Complete TOOLSDB upgrade **before** upgrading Tools Catalog
- **Runstats** on objects so optimizer uses new functionality.
 - Use REORGCHK to update statistics on multiple objects.



```
db2 REORGCHK UPDATE STATISTICS
    ON TABLE [ALL |
    SCHEMA <schema-name> |
    SYSTEM]
```

© Copyright IBM Corporation 2012

42

The db2tdbmgr command migrates specific tools catalog database objects after running the UPGRADE DATABASE command on the tools catalog database.

It is critical that the upgrade of TOOLSDB is completed before any attempt is made to upgrade the tools catalog to the latest version. Access to the database is restricted until the upgrade is completed, and so any attempt to upgrade the tools catalog would fail.

Rebinding packages in upgraded databases

- During database upgrade, all packages for user applications and routines are marked as invalid.
- Packages implicitly rebound first time used after upgrade database.
 - To eliminate overhead, explicitly rebind invalid packages.
 - Must explicitly rebind inoperative packages.
 - **Alternatively, specify REBINDALL on UPGRADE DATABASE**
 - Applies only to C, C++, COBOL, FORTRAN, and REXX embedded SQL applications.
- Complete Runstats first before rebind static embedded SQL packages!
- Then rebind all valid and invalid packages in each database:
 - **db2rbind database-name -l logfile all**
 - From IBM Data Studio, open task assistant to rebind packages.
- Test your applications and tools to ensure that the server is working as expected.
- After rebound all packages, can automatically take advantage of optimizer improvements.

© Copyright IBM Corporation 2012

43

The new **REBINDALL** option specifies that a REBIND of all packages is performed during upgrade. Performing the REBINDs automatically during the upgrade ensures that the step is not missed and will help ensure that other applications are not started before the REBINDs are complete.

Upgrading explain tables - db2exmig

- If must maintain explain table info gathered from previous releases, upgrade explain tables to DB2 10.1.
- Can manually upgrade explain tables after upgrade database, or can re-create explain tables and gather new information.

db2exmig -d dbname -e explain_schema

- renames original explain tables, creates new set of tables by using EXPLAIN.DDL file, and copies contents of original explain tables to new tables. Finally, the tool drops original explain tables. db2exmig preserves any user-added columns in explain tables.

© Copyright IBM Corporation 2012

44

Run a number of performance tests before upgrading your DB2 server. The **db2batch** benchmark tool helps you to collect elapsed and CPU times for running queries. You can use this tool to develop performance tests. Record the exact environment conditions where you run your tests.

Also, keep a record of the **db2expln/db2exfmt** command output for each test query. Compare the results before and after upgrade. This practice can help to identify and correct any performance degradation that might occur.

Upgrade event monitor output tables

- As event monitors enhanced, tables require new columns.
- **EVMON_UPGRADE_TABLES** stored procedure
 - upgrades definitions of existing tables along with all data
 - automatically deactivate event monitor before upgrade
 - automatically reactivates after upgrade completed

call EVMON_UPGRADE_TABLES (*evmon_name*,
evmon_type, *options*, *num_evmons_evaluated*,
num_evmons_to_upgrade, *num_evmons_upgraded*)

or

call evmon_upgrade_tables (null, null, null, ?, ?, ?)

- Can also use ALTER EVENT MONITOR statement to add new logical groups to event monitor as alternative

© Copyright IBM Corporation 2012

45

Beginning in DB2 10.1, you can upgrade event monitor output tables after you upgrade the DB2 product. This capability lets you retain any data that might exist in event monitor tables that you had before you upgraded.

As event monitors are enhanced in the DB2 product, the tables they produce might change. For example, new columns might be added to a table for reporting new monitor elements. Before 10.1, if you had existing event monitors that wrote to tables that contained data that you wanted to retain, and you wanted to collect the data in the newly-added columns, you were required to manually alter them after upgrading to the new release. This alteration involved adding any of the new columns that you might want to use. If you did not add the new columns, the event monitor would work as it had in the previous release, capturing only the data supported by that the event monitor in that release.

Unformatted event tables that had changed could not be upgraded at all; you were required to drop them and then re-create them.

The EVMON_UPGRADE_TABLES stored procedure upgrades the definitions of existing event monitor tables to match those produced by the current level of the DB2 product. This feature lets you keep any existing tables that you might have, along with all the data they contain, eliminating the need to manually alter, or to drop and re-create tables.

Note: Starting in 10.1, you can also use the ALTER EVENT MONITOR statement to add new logical groups to an event monitor. You can use this approach as an alternative to EVMON_UPGRADE_TABLES to add logical data groups added in a new release. However, you cannot use ALTER EVENT MONITOR to modify logical groups that are already associated with the event monitor; if a logical data group already associated with the event monitor has changed, the only way to modify the event monitor is using the EVMON_UPGRADE_TABLES procedure.

The EVMON_UPGRADE_TABLES procedure works with both regular and UE tables. For regular tables, the procedure adds any new columns needed, drops old columns that are no longer required, and alters any columns as needed. For UE tables, the procedure adds new columns and modifies existing columns as needed to allow the UE table to be processed by the db2evmonfmt tool, or the EVMON_FORMAT_UE_TO_TABLES or EVMON_FORMAT_UE_TO_XML routines.

Important: Any active event monitors must be deactivated for the upgrade process to work properly. The EVMON_UPGRADE_TABLES procedure automatically deactivates any active event monitors before it begins upgrading tables. Do not reactivate any event monitors with tables being processed by EVMON_UPGRADE_TABLES, or the upgrade process will fail. Any event monitors that were active before the upgrade are activated again after the upgrade has completed.

Implications of not upgrading event monitor tables

As in past releases, you can choose to not upgrade your event monitor tables. However, any new columns that have been added to the event monitor in the new release will not be populated with data, and will not be available for queries. Also, the values for any monitor elements that previously existed in the old release and that increased in size in the new release might be truncated. For example, if a monitor element increases in size from VARCHAR(20) to VARCHAR(128) in the new release, and you do not upgrade the previously-existing tables, the column that contains the monitor element values will continue to store only 20 characters of data, even though the system may be sending 128-bytes of data for that monitor element to the event monitor.

Student Notes: For Reference Only

Upgrading tables produced by `EVMON_FORMAT_UE_TO_TABLES`

When used with UE tables, the `EVMON_UPGRADE_TABLES` procedure upgrades the UE table itself; it has no effect on any regular tables that you might have created using the `EVMON_FORMAT_UE_TO_TABLES` procedure. After you use `EVMON_UPGRADE_TABLES` to upgrade a UE table, you can also upgrade the output tables produced by `EVMON_FORMAT_UE_TO_TABLES`. As of DB2 Version 10.1 `EVMON_FORMAT_UE_TO_TABLES` procedure supports a new option: `UPGRADE_TABLES`. When you run the `EVMON_FORMAT_UE_TO_TABLES` procedure with this option, any existing tables produced by the procedure are altered so that the table columns match the output produced by the new version of `EVMON_FORMAT_UE_TO_TABLES` procedure.

UPGRADE DATABASE fails if following exist:

- Do not have authorization to upgrade the database.
- Cataloged database does not exist.
- Database upgrade encounters any of problems described in reason codes of error message SQL1704N.
- User-defined distinct types (UDTs) encountered with names ARRAY, BINARY, CURSOR, DECFLOAT, ROW, VARBINARY, or XML. Must drop these UDTs and re-create with different names before database upgrade.
- Database objects were created using restricted schema names described in the error message SQL0553N. **List of restricted schema names now includes SYSPUBLIC.**
- Database is enabled as a high availability disaster recovery (HADR) standby database.

© Copyright IBM Corporation 2012

47

Changes in existing functionality introduced in DB2 10.1 can potentially impact your applications, scripts, maintenance processes, and any other aspects related your DB2 server upgrade process.

Changes in existing functionality introduced in pre-DB2 10.1 releases can also have an impact.

Upgrading in a test environment allows you to learn about possible issues, evaluate the impact on your environment and find a resolution.

Preparation Steps: Reversing DB2 server upgrade

- Ensure have SYSADM, as well as root on Linux/UNIX or Local Administrator authority on Windows.
- Perform following steps before upgrading your DB2 server:
 - Review disk space requirements.
 - Take offline full backup of all databases that upgrading.
 - Back up all dbm cfg values for each instance and all db cfg values for each database.
- Keep existing pre-DB2 10.1 copy during upgrade of your DB2 server. To do this:
 - Select Install New option to create new copy when installing DB2 10.1.
 - Do not select Work with existing option
 - Then choose a pre-DB2 10.1 copy with upgrade action that is available on Windows.
- Keep all S*.MIG files in active log path in case want to rollforward through these log files after reversing upgrade. For recoverable databases, UPGRADE DATABASE renames log files in active log path with extension .MIG.
- In partitioned databases, must perform this procedure on all database partitions.



Reversing DB2 server upgrade involves creating a plan using the steps in this procedure to fall back to the DB2 release from which you upgraded your DB2 server. There is no utility to fall back to a previous release of DB2 database after upgrading your DB2 server.

Performing an upgrade in a test environment will help you identify any issues with the process and avoid having to reverse the upgrade.

Reversing DB2 server upgrade

1. Log on to DB2 server as user with SYSADM authority.
2. Drop all databases in DB2 10.1 by running DROP DATABASE.
3. Log on to DB2 server as root on Linux/UNIX or user with Local Administrator authority on Windows.
4. Drop DB2 10.1 instances by running db2idrop. Does not remove database files. Need to drop databases before dropping instances.
5. If upgraded pre-DB2 10.1 instances to DB2 10.1, re-create instances in pre-DB2 10.1 by running db2icrt. Then restore DBM CFG values for each instance using UPDATE DBM CFG.
6. For each pre-DB2 10.1 instance, log on to DB2 server as instance owner and restore upgraded databases from pre-DB2 10.1 offline full backup by running RESTORE DATABASE. Cannot upgrade databases from DB2 10.1 to pre-DB2 10.1 release.
7. If recreated instances using same instance owner had before upgrade and did not upgrade database to DB2 10.1 instance, database is still pre-DB2 10.1 release and can access it by just re-cataloging it.
8. If have recoverable databases and want to rollforward through log files had before upgrade, rename all S*.MIG files in active log path using .LOG extension and issue ROLLFORWARD DATABASE.

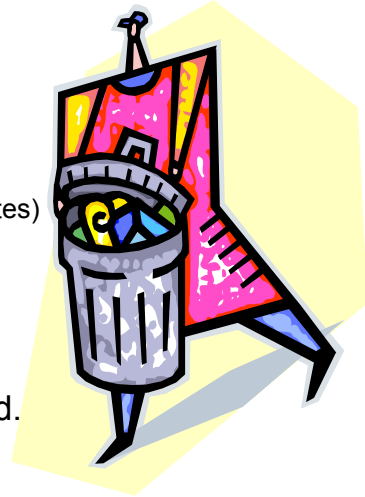
© Copyright IBM Corporation 2012

49

To reverse a DB2 server upgrade, you need to perform the above steps.

Deprecated or discontinued functionality that affects DB2 server upgrades

- DB2 JDBC Type 2 driver is discontinued.
- Control Center tools is discontinued.
- Deprecated or discontinued commands
 - Commands related to Control Center and DAS (see notes)
 - Additional commands – see upcoming graphics
- Raw logs depreciated in DB2 9.1.
- Net Search Extender (NSE) has been deprecated.
- Query Patroller is discontinued.
- If upgrading from DB2 9.5, review changes made in DB2 9.7 that might impact environment after upgrading to DB2 10.1.



© Copyright IBM Corporation 2012

50

You should be aware of functionality that is deprecated or discontinued in DB2 10.1 that can affect the upgrade of your DB2 server. Also, you should be aware of the DB2 products that are no longer supported because upgrade from these products to DB2 10.1 is unsupported. To deal with these functionality changes, you must perform additional tasks before or after upgrade. The above list describes changes that are not included in the pre-upgrade and post-upgrade tasks for DB2 servers.

The DB2 JDBC Type 2 driver for Linux, UNIX, and Windows has been discontinued and it is no longer available. You should use the IBM® Data Server Driver for JDBC and SQLJ instead.

The Control Center tools have been discontinued in DB2 10.1. Start using IBM Data Studio and IBM Optim tools.

The use of raw devices for database logging has been deprecated since DB2 9.1 and will be removed in a future release. You should use a file system instead of a raw device. Using a file system with non-buffered I/O capabilities enabled, such as Concurrent I/O (CIO) or Direct I/O (DIO), can give you performance comparable to that of using raw devices.

Net Search Extender (NSE) has been deprecated and might be discontinued in a future release. For NSE environments with comparative small workloads, if you are not using these deprecated features or commands, consider migrating to DB2 Text Search to take advantage of functionality not supported in NSE such as linguistic processing in all supported languages and search improvements on XML documents.

Discontinued Control Center tools and features

- Control Center and associated wizards and advisors
 - Add Partitions launchpad
 - Alter Database Partition Group wizard
 - Backup wizard
 - Configuration advisor
 - Configure Database Logging wizard
 - Configure Multisite Update wizard
 - Create Cache Table wizard
 - Create Database wizard
 - Create Table Space wizard
 - Create Table wizard
 - Design advisor
 - Drop Partition launchpad
 - Health Alert Notification
 - Health Indicator Configuration launchpad
 - Load wizard
 - Recommendation advisor
 - Redistribute Data wizard
 - Restore wizard
 - Set Up Activity Monitor wizard
 - Set Up HADR Databases wizard
 - Create Federated Objects wizard
 - Storage Management Setup launchpad
 - Troubleshooting wizard
- Activity Monitor
- Command Editor
- Configuration Assistant
- Control Center plug-in extensions
- Event Analyzer
- Health Center
- Indoubt Transaction Monitor
- Journal
- License Center
- Memory Visualizer
- Query Patroller Center
- Satellite Administration Center
- Task Center
- User interface to access Spatial Extender functionality
- User interface to Visual Explain

© Copyright IBM Corporation 2012

51

As a result, the following associated DB2 commands are also discontinued:

- dasauto (Autostart DB2 administration server command)
- dasCRT (Create a DB2 administration server command)
- dasdrop (Remove a DB2 administration server command)
- dasmigr (Migrate the DB2 administration server command)
- dasupdt (Update DAS command)
- daslist (Display DAS name command)
- db2admin (DB2 administration server command)
- db2am (Start Activity monitor center command)
- db2ca (Start the Configuration Assistant command)
- db2cc (Start Control Center command)
- db2ce (Start Command Editor command)
- db2daslevel (Show DAS level command)
- db2eva (Event analyzer command)
- db2hc (Start Health Center command)
- db2indbt (Start Indoubt Transaction Monitor Center command)
- db2journal (Start Journal command)
- db2lc (Start License Center command)
- db2memvis (Start Memory Visualizer center command)
- db2tc (Start Task Center command)

The Replication Center has not been deprecated. It is still available as part of the Replication tools.

Resolution to Control Center tools discontinued

- Use **AUTOCONFIGURE** command
 - Recommendations from Configuration Advisor
 - Wizard for Configuration advisor is discontinued; Configuration Advisor is not.
- Use **db2advis** command
 - Wizard for Design advisor is discontinued; Design Advisor is not.
- Use **IBM Data Studio** and **IBM InfoSphere Optim** tools
 - Perform similar tasks previously perform with Control Center tools.
 - Provide enhanced features for those tasks.
 - Control Center tools and related features are discontinued
- **Data Studio 3.1.1 supports following DB2 10.1 features:**
 - Adaptive compression for table rows
 - Special registers for temporal tables in server profiles
 - Time-based data management with temporal tables
 - Data management using multi-temperature storage
 - Data security with row and column access control (RCAC)

© Copyright IBM Corporation 2012

52

Start using the AUTOCONFIGURE command to get recommendations from the configuration advisor. The wizard interface for the configuration advisor is discontinued.

Start using the db2advis command to get recommendations from the design advisor. The wizard interface for the design advisor is discontinued. The design advisor is not deprecated.

The Control Center tools and related features are discontinued in DB2 10.1, but are still supported in DB2 9.7. Only DB2 9.7 DB2 servers or earlier releases are supported. Become familiar with the IBM Data Studio and IBM InfoSphere Optim tools and start using them in place of the Control Center tools.

IBM tool by IBM product edition

IBM tool by IBM product edition.	
IBM tool	IBM product editions that include the tool
IBM Data Studio	All DB2 database products, excluding DB2 Express-C, and all IBM InfoSphere Warehouse product editions
IBM InfoSphere Optim Performance Manager Enterprise Edition	DB2 Advanced Enterprise Server Edition, DB2 Database Enterprise Developer Edition, IBM InfoSphere Warehouse Departmental Edition, and IBM InfoSphere Warehouse Advanced Departmental Edition
IBM InfoSphere Optim Performance Manager Extended Edition	IBM InfoSphere Warehouse Enterprise Edition and IBM InfoSphere Warehouse Advanced Enterprise Edition
IBM InfoSphere Optim Query Tuner and IBM InfoSphere Optim Query Workload Tuner	DB2 Performance Optimization Feature for Enterprise Server Edition, IBM InfoSphere Warehouse Enterprise Edition, IBM InfoSphere Warehouse Advanced Enterprise Edition

© Copyright IBM Corporation 2012

53

IBM Data Studio and IBM InfoSphere Optim tools are available to perform similar tasks that you used to perform with the Control Center tools. These recommended tools provide enhanced features for those tasks.

The above table provides details about IBM tools included in DB2 database and IBM InfoSphere Warehouse product editions.

Recommended tools versus Control Center tools (1)

Control Center tool	IBM tool	Additional information
Activity Monitor Event Analyzer	IBM InfoSphere Optim Performance Manager	Relevant information: <ul style="list-style-type: none"> • Monitoring with Optim Performance Manager Added benefits: <ul style="list-style-type: none"> • Monitoring profiles and system template descriptions • Reporting with Optim Performance Manager • End-to-end database monitoring for applications (Extended Insight Analysis dashboard)¹
Command Editor	IBM Data Studio ²	Relevant information: <ul style="list-style-type: none"> • Modifying SQL and XQuery statements in the SQL and XQuery editor • DB2 command support • Diagramming access plans with Visual Explain Added benefits: <ul style="list-style-type: none"> • Using templates for SQL and XQuery • Validating SQL and XQuery statements
Configuration assistant	IBM InfoSphere Optim Configuration Manager	Relevant information: <ul style="list-style-type: none"> • Exploring configuration information about data clients and data servers in your environment Added benefits: <ul style="list-style-type: none"> • Exploring configuration changes • Creating a client rule to control and optimize the configurations of the managed clients
<p>© Copyright IBM Corporation 2012</p>		

54

The above table shows the recommended tools to use in place of the Control Center tools.

IBM InfoSphere Optim Performance Manager Extended Insight is a separately priced feature for IBM InfoSphere Optim Performance Manager. It is part of IBM InfoSphere Optim Performance Manager Extended Edition.

IBM Data Studio consist of three components. You can install any combination of these components and DB2 database products into a common shared environment.

The Index Advisor and Query Advisor require an active license for IBM InfoSphere Optim Query Tuner or IBM InfoSphere Optim Query Workload Tuner.

Recommended tools versus Control Center tools (2)

Control Center	IBM Data Studio	<p>Relevant information:</p> <ul style="list-style-type: none"> •Connecting to databases •Creating and altering data objects with the Data Object editor •Managing data •Analyzing impact and dependency for data objects •Modifying privileges on data objects •Generating DDL scripts •Administering databases with task assistants •Diagramming access plans with Visual Explain •Updating and viewing database statistics •Modifying SQL and XQuery statements in the SQL and XQuery editor •Connecting to a federated database •Changing federated objects <p>Added benefits:</p> <ul style="list-style-type: none"> •Making basic database object changes with a change plan •Change management by using forward engineering from a model or the compare and migrate wizard •Copying database objects •Developing a Java applications that use pureQuery annotated methods •Deploying routines and SQL scripts to multiple systems.
Control Center associated wizards and advisors	IBM Data Studio	<p>The Task assistants provide similar functionality to the Backup, Configure database logging, Create database, Load, Restore, and Set up HADR wizards in the Control Center tools.</p> <p>Added benefits:</p> <ul style="list-style-type: none"> •Using links to find detailed information about the message number or SQL code in the Messages section •Running commands on multiple objects •Management of cluster members in DB2 pureScale environments
© Copyright IBM Corporation 2012		
55		

The above table shows the recommended tools to use in place of the Control Center tools.

Recommended tools versus Control Center tools (3)

Health Center IBM Data Studio web console	IBM Data Studio web console IBM InfoSphere Optim Performance Manager	<p>Relevant information:</p> <ul style="list-style-type: none"> • Monitoring database health and availability • Monitoring the status of utilities that operate on your databases • Monitoring with Optim Performance Manager <p>Added benefits:</p> <ul style="list-style-type: none"> • Opening the Data Studio web console from the Data Studio full client • Accessing more advanced performance monitoring functionality in IBM InfoSphere Optim Performance Manager 4.1.0.1 or later.
Memory Visualizer	IBM InfoSphere Optim Performance Manager	<p>Relevant information:</p> <ul style="list-style-type: none"> • Monitoring with Optim Performance Manager (Memory dashboard) <p>Added benefits in InfoSphere Optim tools:</p> <ul style="list-style-type: none"> • Finding buffer pools with a low hit ratio and high activity
Query Patroller Center	IBM InfoSphere Optim Performance Manager	<p>Relevant information:</p> <ul style="list-style-type: none"> • Configuring and monitoring a workload management solution • Migrating from Query Patroller to DB2 workload manager <p>Added benefits:</p> <ul style="list-style-type: none"> • Reporting with Optim Performance Manager • Configuring autonomic performance objectives for workloads (IBM InfoSphere Optim Performance Manager Extended Edition) • Monitor and control active work with DB2 workload manager.

The above table shows the recommended tools to use in place of the Control Center tools.

Recommended tools versus Control Center tools (4)

Task Center	IBM Data Studio	<p>Relevant information:</p> <ul style="list-style-type: none"> •Creating and managing jobs •Scheduling command scripts <p>Added benefits:</p> <ul style="list-style-type: none"> •Scheduling Executable/shell Scripts and configuring email notifications to report on job completion.
User Interface to Spatial Extender	IBM Data Studio	<p>Access to Spatial Extender functionality from the Control Center is deprecated. Alternatively, you can use the InfoSphere Optim tools or DB2 CLP commands to perform similar tasks.</p> <p>Added benefits:</p> <ul style="list-style-type: none"> •See Added benefits for the Control Center and Control Center associated wizards and advisors in these tables.
Visual Explain	IBM Data Studio	<p>Use the SQL and XQuery editor, which provides access to Visual Explain and enhanced query tuning capabilities.</p> <p>Added benefits:</p> <ul style="list-style-type: none"> •Creating and deploying optimization profiles for SQL statements •Use advisors to generate and act on recommendations for: <ul style="list-style-type: none"> ◦Collecting statistics for database objects using the Statistics Advisor ◦Reviewing recommendations for improving SQL statement access paths using the Access Path Advisor ◦Creating or modifying indexes that improve the performance of single SQL statements using the Index Advisor³ ◦Rewriting queries to resolve the problems that are identified by the Query Advisor³
<p>© Copyright IBM Corporation 2012</p>		

57

The above table shows the recommended tools to use in place of the Control Center tools.

Upgrade impact from DB2 command changes (1)

- **db2cat**
 - Table space properties such as TRANSFERRATE, OVERHEAD, WRITETRANSFERRATE, WRITEOVERHEAD, PREFETCHSIZE, EXTENTSIZE and PAGESIZE no longer reported in db2cat output
- **db2icrt, db2iupdt, db2cluster_prepare**
 - -m and -cf parameters are deprecated. Instead use **-mnet** and **-cfnet** to indicate network name.
- **db2look**
 - -l parameter modified to generate DDL statements for user-defined storage groups
- **CREATE DATABASE**
 - AUTOMATIC STORAGE parameter is deprecated
 - MANAGED BY SYSTEM clause in USER TABLESPACE parameter is deprecated
 - Some values for COLLATE USING parameter are deprecated. Collations based on Unicode Collation Algorithm of Unicode Standard version 4.0.0 is deprecated. For new Unicode databases, use COLLATE USING clause with language-aware-collation or locale-sensitive-collation instead of UCA400_NO, UCA400_LSK, or UCA400_LTH.
- **EXPORT, IMPORT**
 - Worksheet Format (WSF) is discontinued.

© Copyright IBM Corporation 2012

58

The changes in DB2 10.1 to CLP and system commands can impact your existing applications and scripts after you upgrade to DB2 10.1. Modify any existing scripts and applications to discontinue using depreciated or discontinued options and parameters.

The changes to commands include new parameters, modifications to existing parameters, deprecated or discontinued parameters, and modifications to command output. The above table lists the changes that impact applications and scripts.

Upgrade impact from DB2 command changes (2)

- **db2pd**

- **-catalogcache** output reports details about new table space cache.
- **-hadr** output has completely changed. Existing fields renamed to match output from new **MON_GET_HADR** table function.
 - Fields added to support monitoring for new HADR multiple standby.
 - Formatting of output changed from multiple field names on one line and field values on subsequent line to having one field per line.
 - Serviceclasses parameter output now includes UOW throughput, activity throughput, activity lifetime average, CPU Velocity, and CPU Utilization.
- **-storagepaths** output includes storage group configuration details.
- **-tablespaces** output includes storage group identifier that table space is associated with and table space storage statistics.
- **-tcbstats** output for column and data for IndexObjSize is deprecated. If perform a reorg to reclaim extents, IndexObjSize output does not accurately reflect number of pages in index object because value still includes pages that were released during reorg.
 - Use INDEX_OBJECT_P_SIZE or INDEX_OBJECT_L_SIZE columns of **ADMIN_GET_INDEX_INFO** table function to obtain accurate values.
- **-workclasssets** output is no longer in table format; it now lists basic work class info followed by work class attribute info.

© Copyright IBM Corporation 2012

59

The changes in DB2 10.1 to CLP and system commands can impact your existing applications and scripts after you upgrade to DB2 10.1. Modify any existing scripts and applications to discontinue using deprecated or discontinued options and parameters.

The changes to commands include new parameters, modifications to existing parameters, deprecated or discontinued parameters, and modifications to command output. The above table lists the changes that impact applications and scripts.

Upgrade impact - **RUNSTATS** – new syntax

DETAILED

- Calculates extended index statistics: CLUSTERFACTOR and PAGE_FETCH_PAIRS statistics gathered for relatively large indexes.
- Not all index entries are examined, CPU sampling technique employed instead to improve performance.
- Option cannot be used for views.
 - **SAMPLED**
 - Specifying does not change default functionality from DETAILED. Left in for compatibility . Cannot be used for views.
 - **UNSAMPLED**
- When used with DETAILED, forces RUNSTATS to examine every entry in index to compute extended index statistics. Cannot be used for views and cannot be used with INDEXSAMPLE. Increases RUNSTATS resource consumption, while rarely providing significant improvement over DETAILED or SAMPLED DETAILED (equivalents).

INDEXSAMPLE BERNOULLI

- Use to collect index statistics on sample of rows in index. *If numeric-literal were 10, representing a 10 percent sample, each row would be included with probability 0.1 and be excluded with probability 0.9.* Not supported on statistical views.

INDEXSAMPLE SYSTEM

- Use to collect statistics on a sample of the index pages. *System sampling* considers each page individually. Only a percentage of index pages as specified through numeric-literal parameter is retrieved and used for the statistics collection. Option is not supported on statistical views.

The above options have been added to the runstats command.

New Registry Variables

- Global profile registry variables set with **db2set -g** are not upgraded. **Redefine them after upgrade.**
- **DB2_INDEX_PCTFREE_DEFAULT**
 - Specifies default percentage of each index page to leave as free space when building index.
 - Default is 10.
- **DB2_XSLT_ALLOWED_PATH**
 - Controls whether DB2 instance refers to external entities defined inside of XSLT style sheet.
 - By default, not set so no access allowed to external entities.
 - If using document function of XSLT, may set to directories where XML files can be downloaded..

© Copyright IBM Corporation 2012

61

Changes to DB2 registry variables, configuration parameters, database physical design characteristics, and database authorities and privileges can result in DB2 server behavior changes that might impact your upgrade.

As a general rule, instance profile variables that you set in your DB2 profile registry or your system environment retain their values after an instance upgrade. Some global profile registry variables, such as DB2SYSTEM and DB2PATH, are set by the DB2 installation procedure or instance upgrade. However, the global profile registry variables that you set by running the db2set command with the -g option are not upgraded. Therefore, you must define them after upgrade.

Upgrade impact on existing registry variables

- **DB2_EXTENDED_OPTIMIZATION**

- ENHANCED_MULTIPLE_DISTINCT setting is deprecated although preserved during instance upgrade.
- If use multiple distinct queries, remove setting to start using new enhancements in DB2 10.1.

- **DB2BPVARS**

- Numprefetchqueues and prefetchqueuesize are discontinued as optimization improvements render these options obsolete. DB2PBVARS is still deprecated.

- **DB2_NO_FORK_CHECK**

- Registry variable is no longer deprecated.
- Continue use to have DB2 runtime client minimize checks to determine if current process is result of fork call.

- **DB2NTNOCACHE**

- Registry variable is no longer deprecated. Continue use to override undocumented 192 MB limit for cache.

- **DB2_PMODEL_SETTINGS**

- Use new SRVLST_EQUAL_WEIGHT option to override default behavior in which member weights are computed based on load, and have non-zero member weights in server list always identical.

© Copyright IBM Corporation 2012

62

The above list describes the upgrade impact of changes to existing registry variables.

You should remove the use of registry variables that are deprecated because the functionality associated with the variable is obsolete or has been replaced by new functionality. If you are upgrading from DB2 9.5 or earlier, consider removing deprecated registry variables in pre-DB2 10.1 releases because the functionality associated with the variable is obsolete or has been replaced by new functionality. Also, remove the use of discontinued registry variables in pre-DB2 10.1 releases as they do not have the intended effect.

New database manager configuration parameters

- **wlm_dispatcher**
 - Enables (YES) or disables (NO) DB2 WLM dispatcher.
 - By default, enabled WLM dispatcher controls only CPU limits.
- **wlm_disp_concur**
 - Specifies how DB2 WLM dispatcher sets thread concurrency level.
 - Can also manually set thread concurrency level to fixed value.
- **wlm_disp_cpu_shares**
 - Enables (YES) or disables (NO) control of CPU shares by DB2 WLM dispatcher.
 - By default, enabled WLM dispatcher controls only CPU limits.
- **wlm_disp_min_util**
 - Specifies minimum amount of CPU utilization necessary for service class to be included in DB2 WLM-managed sharing of CPU resources.

© Copyright IBM Corporation 2012

63

Existing database and database manager configuration parameters also, as a general rule, retain their values after upgrade. However, the default values assigned to new parameters or the new default values assigned to existing parameters could impact the behavior or performance of your applications.

The above list describes the upgrade impact of the default values of new database manager configuration parameters.

Upgrade impact of changes to DBM CFG parameters

- **alt_diagpath**
 - For DB2 pureScale, alternate diagnostic data directory path set from Null to INSTHOME/sqllib/db2adump/ \$m.
 - If V10.1 Server Edition or V10.1 DB2 Database Partitioning Feature instance then remains NULL.
- **cf_diagpath**
 - Diagnostic data directory path for cluster caching facility (CF) set from Null to INSTHOME/sqllib/db2dump/ \$m in DB2 pureScale.
- **diagpath**
 - Diagnostic data directory path new default. New value \$m resolves to DIAG<number> to specify unique diagnostic log path for all database partitions, CFs, or members.
 - V10.1 Server Edition (Linux/UNIX): INSTHOME/sqllib/db2dump/
 - V10.1 (Windows): ProgramData\IBM\DB2\db2build\DINSTESE\DIAG0000
 - V10.1 DB2 Database Partitioning Feature (Linux/UNIX): INSTHOME/sqllib/db2dump/ \$m
 - V10.1 DB2 pureScale (Linux/UNIX): INSTHOME/sqllib/db2dump/ \$m
- **No DBM CFG parameters deprecated or discontinued in DB2 10.1.**

© Copyright IBM Corporation 2012

64

The above list describes the upgrade impact of changes to database manager configuration parameters.

No database manager configuration parameters have been deprecated or discontinued in DB2 10.1. However, if you are upgrading from DB2 9.5 or earlier, consider removing deprecated database manager configuration parameters in pre-DB2 10.1 releases because the functionality associated with the parameters is obsolete or has been replaced by new functionality. Also, remove the use of discontinued database manager configuration parameters in pre-DB2 10.1 releases as they do not have the intended effect.

New database configuration parameters

- **dft_schemas_dcc**
 - Controls default setting for DATA CAPTURE CHANGES on newly created schemas for replication purposes.
- **hadr_replay_delay**
 - Specifies time in seconds that passed from when data is changed on primary before changes reflected on standby database.
- **hadr_spool_limit**
 - Allows log replay on HADR standby database to be behind HADR primary database. If there is spike in transaction volume or slow replay caused by specific operations and log receive buffer fills up, log data is written (or spooled) to disk and read later.
- **hadr_target_list**
 - Enables multiple high availability disaster recovery (HADR) standbys, specifies list of up to three target host:port pairs that act as HADR standby databases.
- **log_appl_info**
 - Specifies that application information log record is written at start of each update transaction.
- **log_ddl_stmts**
 - Specifies that extra information regarding DDL statements be written to log.
- **mon_uow_execlist**
 - Enables (ON) or disables (OFF) collection of execution list info by UOW event monitor. By default, OFF. It is child parameter of mon_uow_data DB CFG parameter.
- **mon_uow_pkglist**
 - Enables (ON) or disables (OFF) collection of package list info by UOW event monitor. By default, OFF. It is child parameter of mon_uow_data DB CFG parameter.
- **system_time_period_adj**
 - Specifies how to handle situation of history row for system-period temporal table potentially being generated with end timestamp < begin timestamp.

The above list describes the upgrade impact of the default values of new database configuration parameters.

Changes to existing DB CFG parameters

- **auto_reorg**
 - Supports reorganizing indexes for volatile tables.
- **auto_stats_views**
 - Enables and disables automatic statistic collection on statistical views.
- **hadr_local_host, hadr_local_svc, hadr_peer_window, hadr_remote_host, hadr_remote_inst, hadr_remote_svc, hadr_syncmode, hadr_timeout**
 - Updates take effect on HADR primary without deactivating database. Issue STOP HADR on primary, followed by START HADR AS PRIMARY. Can make updates to HADR primary without having impact on applications.
 - New HADR cfg parameters have same behavior: **hadr_replay_delay, hadr_spool_limit, hadr_target_list**
- **mon_uow_data**
 - Default value still NONE. It is parent parameter to mon_uow_execlist and mon_uow_pkglist.
- **mon_req_metrics**
 - Default value changed from BASE to NONE.
- **mon_act_metrics**
 - Default value changed from BASE to NONE.
- **mon_obj_metrics**
 - Default value changed from BASE to NONE.
- **mon_lw_thresh**
 - Default value changed from 5000000 to 4294967295.

© Copyright IBM Corporation 2012

66

The above list describes the upgrade impact of changes to existing database configuration parameters.

You should remove the use of database configuration parameters that are deprecated because the functionality associated with the variable is obsolete or has been replaced by new functionality. See Some database configuration parameters have been changed to determine the upgrade impact of deprecated database configuration parameters.

If you are upgrading from DB2 9.5 or earlier, consider removing deprecated database configuration parameters in pre-DB2 10.1 releases because the functionality associated with the parameter is obsolete or has been replaced by new functionality. Also, remove the use of discontinued database configuration parameters in pre-DB2 10.1 releases as they do not have the intended effect

Changes to database physical design characteristics

- CHAR or VARCHAR type
 - Casting XML data to CHAR or VARCHAR type that is too small causes data to be truncated to fit specified data type and no error is returned.
- DECIMAL type
 - Casting XML data to DECIMAL type that has insufficient space for digits to right of decimal separator, trailing digits are truncated to fit specified data type and no error is returned.
- XML data of incompatible types
 - Comparing XML data of incompatible types, comparison returns FALSE.

Upgrade impact from SQL statement changes

- **ALTER DATABASE**
 - Statement is deprecated. CREATE STOGROUP or ALTER STOGROUP provide same functionality.
- **ALTER TABLE, CREATE TABLE**
 - COMPRESS clause for default compression method changed; STATIC or ADAPTIVE indicates table compression method.
 - IMPLICITLY HIDDEN clause can be used in any type of column.
- **CREATE INDEX**
 - DETAILED keyword now specifies sampling technique used when processing index entries to collect extended index statistics with SAMPLED DETAILED or UNSAMPLED DETAILED keywords.
- **DECLARE GLOBAL TEMPORARY TABLE**
 - Implicitly hidden columns in declared temporary tables are not supported. Occurs when using LIKE clause to create from a base table that contains implicitly hidden columns.
- **ALTER WORKLOAD, CREATE WORKLOAD**
 - Modify scripts and applications that use COLLECT UNIT OF WORK DATA PACKAGE LIST clause. Use new syntax to collect package list information:
**ALTER WORKLOAD REPORTS COLLECT UNIT OF WORK DATA BASE
INCLUDE PACKAGE LIST**


© Copyright IBM Corporation 2012

68

The changes to SQL statements in DB2 10.1 can impact your existing applications and scripts after you upgrade to DB2 10.1.

The changes to SQL statements include new default behaviors and modifications to statement output. In addition, some statements are changed, deprecated or discontinued. The above list shows the changes that impact applications and scripts.

Supported Clients

- **CLIENTS that can access remote DB2 10.1 Server:**
 - **DB2 9.1, DB2 9.5, DB2 9.7**
 - **If use pre-10.1 clients to access DB2 10.1 servers, applications are restricted to functionality available for that client.**
- If different versions of client and DB2 server are located on same system, local client-to-server connections using IPC are not supported. Establish connection as remote connection (loopback connection) by using TCP/IP.
- When later-level client accesses earlier-level server, functionality of the later level of client is not available to server.
- Upgrading client involves installing DB2 10.1 client copy and then upgrading client instance.
- Current level of client installed determines way to proceed with upgrade to DB2 10.1.
 - Can directly upgrade to DB2 10.1 clients from DB2 9.5, or DB2 9.7.
 -  **- If have DB2 9.1 or earlier clients, migrate to any DB2 9.5 client first.**
- 32-bit applications on HP-UX is discontinued.
 - Upgrade to 64-bit before upgrading to DB2 10.1, so applications can run on native 64-bit HP-UX Itanium-based environments.

© Copyright IBM Corporation 2012


69

A client instance allows connecting application to database and keeps information about client configuration, cataloged nodes, and cataloged databases.

The upgrade process consists of all the tasks that you must perform to have your DB2 environment running successfully after upgrading. These components are DB2 servers, DB2 clients, database applications, and routines. Upgrading DB2 servers to DB2 10.1 requires an understanding of upgrade concepts, upgrade restrictions, upgrade recommendations, and your DB2 server. When you have a complete understanding of what upgrading your DB2 server involves, you can create your own upgrade plan. The upgrade of each of the components in your DB2 environment requires that you perform different tasks:

- *Upgrading DB2 servers* involves upgrading your existing instances and databases so that they can run in the new release.
- *Upgrading clients* involves upgrading your client instances to keep the configuration of your existing clients.
- *Upgrading database applications and routines* involves testing them in the new release and modifying them only when you must support changes in this new release.

Upgrading to a new DB2 10.1 Server (1)

- Perform a full **offline** database backup of your existing databases
 - If database backups performed recently, can perform incremental *offline* database backup instead.
 - Log on to new DB2 server as root on Linux and UNIX or user with Local Administrator authority on Windows.
 - Install DB2 10.1 on new DB2 server.
 - Create DB2 10.1 instances on new DB2 server by running **db2icrt**.
 - If similar resources, restore DBM CFG values for each instance.
 - Transfer pre-DB2 10.1 backup files for all databases want to upgrade to new server.
 - Upgrade database using RESTORE DATABASE:
db2 RESTORE DATABASE *sample* FROM /db2/backups
 - If performed incremental *offline* database backup before upgrade, must have access to most recent full *offline* database backup and incremental *offline* database backup and use automatic incremental restore to upgrade database.
 - manual incremental restore will fail because each RESTORE DATABASE  tries to upgrade database before database is completely recovered.
- db2 RESTORE DATABASE *sample* INCREMENTAL AUTOMATIC
TAKEN AT *timestamp* WITHOUT PROMPTING**
- In partitioned database, must execute RESTORE DATABASE on all database partitions starting with catalog partition first.

© Copyright IBM Corporation 2012

70

If you want to upgrade to a new DB2 10.1 server, re-create your instances and then upgrade your databases by restoring a pre-DB2 10.1 database backup. After restoring the database backup, the RESTORE DATABASE command automatically runs the UPGRADE DATABASE command.

Upgrading to a new DB2 10.1 Server (2)

- If log file size is still not large enough, the following error is returned:
 - SQL1704N Database upgrade failed. Reason code "3".
You must increase the log file size and attempt to upgrade the database again.
 - After database upgrade completed, reset size of log files to pre-upgrade values.
- Optional: Configure new server to use new resources available to calculate buffer pool sizes, and DBM and DB CFG values. Run command to display only recommended values:
**db2 CONNECT TO sampledb2 AUTOCONFIGURE
USING MEM PERCENT 80 WORKLOAD TYPE complex
NUM_STMTS 1 TPM 73 ADMIN_PRIORITY performance IS_POPULATED YES
NUM_REMOTE_APPS 15 ISOLATION CS APPLY NONE;**
- If choose not to run and apply, manually configure DB2 server to use new resources.
- Restore any external routines that backed up in pre-upgrade tasks.
- Verify database upgrade successful. Connect to upgraded databases and issue query:
cd samplefile-dir-clp
db2 connect to sample
db2 -tvf testdata.db2
where samplefile-dir-clp is DB2DIR/samples/clp on Linux and UNIX and
DB2DIR\samples\clp on Windows, DB2DIR represents the location specified
during DB210.1 installation, and sample is database name.
- After upgrading DB2 server, perform recommended post-upgrade tasks such as resetting diagnostic error level, adjusting log space size, and rebinding packages. In addition, verify that upgrade of DB2 server was successful.

© Copyright IBM Corporation 2012

71

If you want to upgrade to a new DB2 10.1 server, re-create your instances and then upgrade your databases by restoring a pre-DB2 10.1 database backup. After restoring the database backup, the RESTORE DATABASE command automatically runs the UPGRADE DATABASE command.

Upgrading server using online backups from previous release

- Perform this task only under the following conditions:
 - If cannot upgrade existing instances and databases.
 - If did not perform full *offline* database backups recently or incremental *offline* database backups as indicated in Pre-upgrade tasks for DB2 servers.
- Transfer pre-DB2 10.1 online database backup files for all databases that want to upgrade to DB2 server.
- If have DB2 copy with same version as online database backups, install DB2 copy of same version. if performed online database backups from DB2 9.5 copy, must have DB2 9.5 copy installed on DB2 server.
- If do not have instance running on DB2 copy with same version as online backups, create instance under DB2 copy.
- As SYSADM, rebuild databases by:
**RESTORE DB *db-name* REBUILD WITH ALL TABLESPACES IN DATABASE
TAKEN AT *timestamp-backup*;
ROLLFORWARD DB *db-name* TO END OF LOGS AND STOP;**
- Can choose to rebuild database with just subset of table spaces, but must drop all table spaces in restore pending state after issue ROLLFORWARD DATABASE.
- Verify databases that rebuild are in consistent state by issuing GET DB CFG:
GET DB CFG FOR sample | FIND "consistent"

© Copyright IBM Corporation 2012

72

You can rebuild your database on a previous release by using online database backups from the same release and then upgrade to DB2 10.1.

Before upgrading your DB2 server, ensure that you have root access on Linux and UNIX operating systems or Local Administrator authority on Windows. All necessary full or incremental online pre-DB2 10.1 database backups of your databases so that you can rebuild your databases by using these online backups.

Perform this task only under the following conditions:

- If you cannot upgrade the existing instances and databases.
- If you did not perform full *offline* database backups recently or incremental *offline* database backups as indicated in Pre-upgrade tasks for DB2 servers.

Upgrading partitioned database environments

- Install DB2 10.1 on all database partition servers, upgrade the instances and then upgrade the databases.
- Use only Install New option in Install a Product panel to install DB2 10.1.
 - If choose to upgrade action when select Work with Existing option on Install a Product panel, installation fails.
- Perform a full offline backup for all database partitions. Verify databases are ready for upgrade, and perform any other pre-upgrade tasks that apply.
- Upgrade each instance on database partition server that owns the instance. Run UPGRADE DATABASE on catalog partition.
- Catalog partition must be available when issue UPGRADE DATABASE regardless on what database partition issue command from.
- If any database partitions are not available, these database partitions are not upgraded.

© Copyright IBM Corporation 2012

73

Upgrading partitioned database environments requires that you install DB2 10.1 as a new copy in all database partition servers, upgrade the instances and then upgrade the databases.

Ensure that you have root access on Linux and UNIX operating systems or Local Administrator authority on Windows.

Ensure that you have SYSADM authority.

Post-upgrade tasks for DB2 servers (1)

- If reset diaglevel to 3 or higher as recommended in pre-upgrade tasks, reset to value before upgrade.
- Existing pre-DB2 10.1 tables with row compression enabled have classic row compression enabled. If want to use adaptive compression, alter tables.
- If changed log space setting as recommended in pre-upgrade tasks, reset logfilsiz, logprimary, and logsecond to pre-upgrade values.
- Ensure existing libraries for external routines remain on original location before upgrade.
- Activate database to start database and all necessary database services.
- Automatic storage table spaces inherit media attribute values, including overhead, device read rate and data tag attributes, from the storage group it is using by default. After upgrading to DB2 10.1, existing table spaces retain their settings and OVERHEAD and DEVICE READ RATE attributes for the storage group are set to undefined. Can set media attributes with the ALTER STOGROUP statement. [...](#)
- Consider setting new registry variables and configuration parameters that can impact behavior of DB2.
- If automatic collection of statistics failed on certain system catalog tables during database upgrade, update statistics on these tables.
- If did not use REBINDALL on UPGRADE DATABASE then rebind packages to validate packages and use updated statistics or new index information.

© Copyright IBM Corporation 2012

74

After upgrading DB2 servers, you should perform several post-upgrade tasks to ensure that your DB2 servers perform as expected and at their optimum level.

Perform the above post-upgrade tasks that apply to your DB2 server.

Post-upgrade tasks for DB2 servers (2)

- Refresh data in existing MQTs by using REFRESH TABLE. MQTs on unicode databases using language aware collation and involves LIKE predicate or substring function involved in a basic predicate, need to be refreshed.
- Migrate Explain Tables to retain explain information previously gathered.
- If tables with XML columns created in DB2 9.5, convert XML storage object to DB2 10.1 format by recreating to use XML compression and statistics collection to estimate inline length.
- If obtained customized code page conversion tables from DB2 support service, copy all files for those tables from the *DB295or97/conv* to *DB210/conv*. Do not need to copy standard code page conversion tables.
- If upgraded existing DB2 9.5 or 9.7 on Windows, can restore customized code page conversion tables that backed up as part of pre-upgrade tasks.
- Upgrade existing target tables for event monitors that write to tables and to unformatted event (UE) tables by using new EVMON_UPGRADE_TABLES procedure.
- Test applications and tools to ensure DB2 server is working as expected.
- Back up your databases after DB2 server upgrade is complete.
- If recoverable databases, UPGRADE DATABASE renamed all log files in active log path using .MIG extension. After verifying database upgrade successful and backing up databases, can delete S*.MIG files.
- Must migrate SQL Replication in order to support new LSN formats.

© Copyright IBM Corporation 2012

75

After upgrading DB2 servers, you should perform several post-upgrade tasks to ensure that your DB2 servers perform as expected and at their optimum level.

Perform the above post-upgrade tasks that apply to your DB2 server.

If you obtained customized code page conversion tables from the DB2 support service, you need to backup all of the files in the DB2OLD\conv directory where DB2OLD is the location of your existing pre-10 DB2 copy. You do not need to backup standard code page conversion tables. Upgrading your pre-10 DB2 copy removes these tables because standard code page tables are contained in a DB2 10 library.

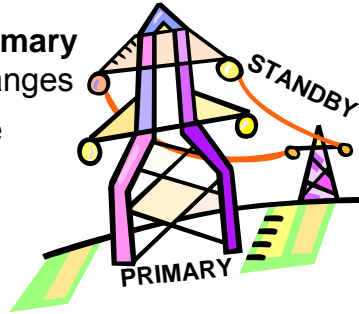
Post upgrade - What to do next?

- If upgraded existing DB2 9.5, or 9.7 copy, database log directories are changed. Review db2diag.log file which has entries detailing new log directories. If user defined log directory is used, after upgrade the location of log files will be /usr/logpath/NODE0000/LOGSTREAM0000. Old log directory will only contain renamed log files. If default database directory is used, /home/db2user/db2inst/NODE0000/SQL00001/SQLOGDIR, after upgrade location of log files will be /home/db2user/db2inst/NODE0000/SQL00001/LOGSTREAM0000. Old log directory will only contain renamed log files.
- If upgrade DB2 server running HADR, initialize HADR replication. During upgrade to DB2 10.1, database role is changed from primary to standard. Upgrade of standby databases is not supported.
- When DB2 server performance is stable, take advantage of optimizer improvements and collect statistics for new functionality by updating statistics for your upgraded databases. During database upgrade to DB2 10.1, statistics collected from existing database tables retain their values. Statistics for new characteristics on tables and indexes have a value of -1 to indicate there is no information gathered. Only need these statistics if using new functionality.
- After updating statistics for upgraded databases, determine if index or table reorganization is necessary by running REORGCHK. Table and index reorganization can help improve performance.
- At this point, resume all of maintenance activities such as backing up databases and updating statistics. Remove any DB2 9.5, 9.7 or 9.8 copies that no longer need.

Perform the above post-upgrade tasks that apply to your DB2 database products or add-on features.

HADR (High Availability Disaster Recovery)

- Consider **logbufsz**, **logfilsiz**, **logprimary**, **logsecond**, **locklist** & **maxlock** changes
- Backup upgraded Primary Database
- Restore on Standby
- Reinitialize HADR



Primary **db2 stop hadr on db dbx**

Standby **db2 deactivate database dbx**
db2 stop hadr on db dbx

Primary **UPGRADE INSTANCE & DATABASE**
db2 backup db dbx online to ...

Standby **db2iupgrade -u <fencedid> <standbyinstance>**
db2 restore db dbx from ...
db2 start hadr on db dbx as standby

Primary **db2 start hadr on db dbx as primary**

© Copyright IBM Corporation 2012

77

Upgrading a primary database to DB2 10.1 changes the database role from primary to standard. Upgrading standby databases to DB2 10.1 is not supported because these databases are in roll forward pending state. Because of these restrictions, upgrading an HADR environment to DB2 10.1 requires that you stop HADR, upgrade your DB2 server where the primary database resides, and then re-initialize HADR. The primary database is taken out of HADR (its status is changed from “primary” to “standard”).

Adjusting log space size in upgraded databases

- Set appropriate size for log files since is one of important factors in tuning server.
- If increased log files sizes as pre-upgrade task, can restore additional free space to server.
- On partitioned database, must adjust log space size on catalog database partition.
- Restore log file size settings to values before upgrade:
db2 UPDATE DB CFG FOR sample using LOGSECOND *previous-value*
- If you enabled infinite active logging, disable it by running the following commands:
db2 UPDATE DB CFG FOR sample using LOGARCHMETH1 *previous-value*
db2 UPDATE DB CFG FOR sample using LOGSECOND *previous-value*
- To support larger log record headers, increase log space setting, by approximately 10% - 15% over what used for DB2 9.7.
- To support larger log record headers, increase logbufsz and softmax by 10% - 15% over what used for DB2 9.7.
- logfilsiz changes take effect only when database is reactivated.



Consider: Adopting adaptive compression

- **CREATE/ALTER TABLE COMPRESS NO | COMPRESS YES ADAPTIVE | COMPRESS YES STATIC**
 - adaptive is default when create new BUT static when upgrade if compression enabled
 - SYSCAT.TABLES.ROWCOMPMODE= A or S or blank
- Estimate storage space savings: **ADMIN_GET_TAB_COMPRESS_INFO.**
 - estimation of savings potential if compress yes static versus adaptive
 - replaces ADMIN_GET_TAB_COMPRESS_INFO_V97 with ESTIMATE mode option
- **ADMIN_GET_TAB_DICTIONARY_INFO**
 - replaces ADMIN_GET_TAB_COMPRESS_INFO_V97 with REPORT mode option
- Once ADAPTIVE COMPRESSION is enabled:
 - Existing table rows are not immediately subject to adaptive compression.
 - Subsequent modification of existing rows or input of new rows lead to application of adaptive compression.
- If want to compress all existing rows, perform classic table REORG with RESETDICTIONARY.
 - Subsequent reorg for purposes of better compressing data rows may no longer be required.
 - Can use ADMIN_MOVE_TABLE procedure instead of performing classic table reorg.

© Copyright IBM Corporation 2012

79

Existing tables that have row compression enabled from a pre-DB2 Version 10.1 database will be upgraded to have classic row compression enabled. If you want to use adaptive compression you must enable it after the upgrade is performed.

You can enable adaptive compression by using the COMPRESS YES ADAPTIVE clause in the CREATE TABLE and ALTER TABLE statements. The ADAPTIVE keyword is now the default option for the COMPRESS YES clause.

For a table enabled to use adaptive compression, rows are compressed during the following data change operations: Inserts, Updates, Imports, Loads, Redistributes, Reorgs, and Online table moves.

After adaptive compression is enabled for a table with existing data, rows that are subsequently inserted might trigger the creation of additional page-level dictionaries as new data is added.

Adaptive Row Compression – Why?

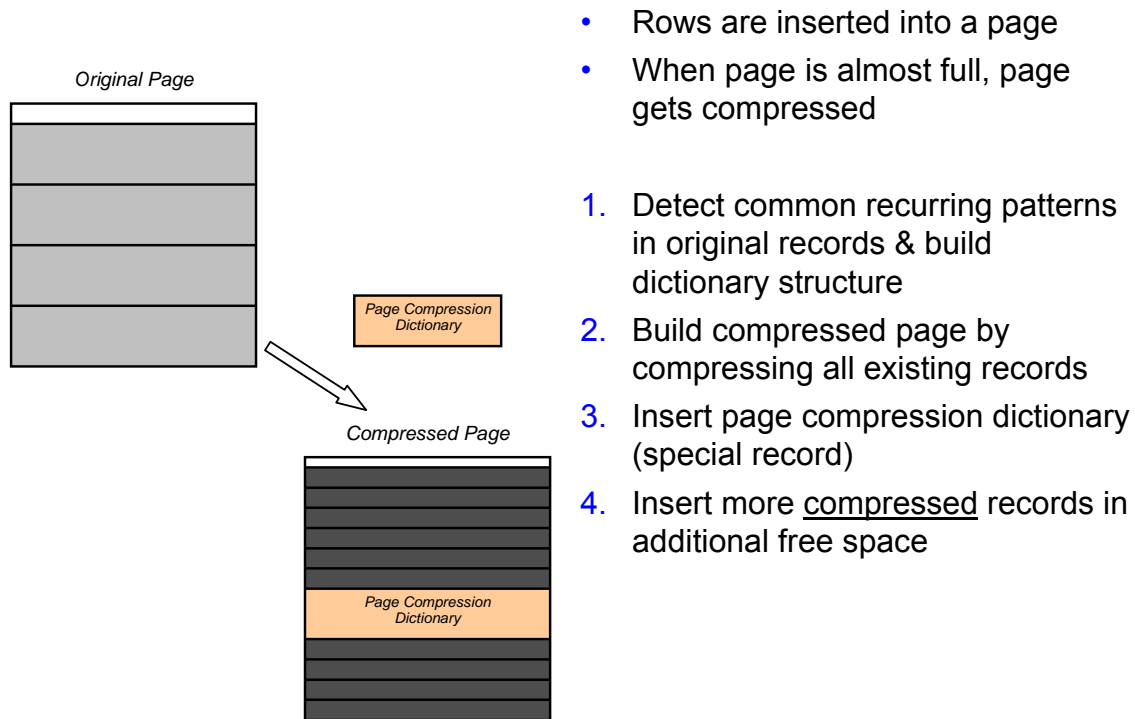
- Classic row compression works very well in many cases
 - Very fast and robust
 - Not sensitive to data clustering and ordering
- Classic row compression has limitations
 - Dictionary requires classic table reorganization to refresh
 - Not sensitive to data clustering and ordering
 - Dictionary capacity limits compression ratios for some large tables
 - Theoretical maximum: 10x compression (90% savings)
- Better compression
 - Classic row compression: Typically saves ~40%-75%
 - Adaptive row compression: Typically saves ~75%-85%
 - Adaptive typically saves 30% over classic row compression
- More automatic
 - Automatic compression in DB2 9 can produce up to 2x the data size of best compression (after REORG)
 - In DB2 10, storage requirements for automatic and best compression typically differ by less than 20%

© Copyright IBM Corporation 2012

80

DB2 compression has different algorithms and characteristics than competitor compression algorithms. It has a reduced TCO and is an industry-leading compression technology for row stores.

Example: Creation of Page-Level Dictionaries



© Copyright IBM Corporation 2012

81

Compression is applied for each row individually.

Combination of two compression algorithms. Both algorithms detect different kinds of patterns: Globally recurring byte sequences and locally recurring byte sequences.

Compression via table-level dictionary is applied first. When records are placed in a page, compression via page-level dictionary is applied.

Rows are compressed during DML operations and admin tasks such as: INSERT, UPDATE, IMPORT, LOAD, REORG, and REDISTRIBUTE.

Page Dictionary Management

- Management of page-level dictionaries is fully automatic
 - Creation happens when page is almost full
 - Recreation may be triggered when page becomes full again
 - Recreation is only triggered if actual compression ratio is significantly less than projected compression ratio (based on last dictionary build)
 - Deletion happens if record is placed in a committed empty page
- Database manager can decide to skip (re-)building page-level dictionaries
 - Decision is based on runtime stats (generated savings)
 - For dictionary recreation, age of existing dictionary is also considered
- Goal: No noticeable CPU overhead if adaptive compression generates little savings

Consider: 10 Best Practices for DB2 10 (1)

1. Use standard row format
 - Row format for compressed tables has little impact on the storage consumption
2. Use estimation capabilities to gauge compression savings
 - Focus on largest tables first
 - Use new ADMIN_GET_TAB_COMPRESS_INFO
3. During first adoption, choose REORG or ADMIN_MOVE_TABLE to compress
4. When you REORG:
 - Start with the smallest tables, work your way up
 - Use a temporary table space for the REORG, and specify the RESETDICTIONARY option
5. When you use ADMIN_MOVE_TABLE:
 - Put your tables into new-style DMS table spaces to facilitate file system space reclamation
 - Use ADMIN_MOVE_TABLE to compress and move your tables
6. Reclaim disk space via ALTER TABLESPACE REDUCE MAX

© Copyright IBM Corporation 2012

83

10 best practices for DB2 10 storage are listed above.

Consider: 10 Best Practices for DB2 10 (2)

7. Use large table spaces over regular ones
8. Use separate table spaces for tables with different compression settings
 - Perform table space backups
 - Apply backup compression for table spaces that contain uncompressed tables
9. Group correlated columns
 - Adaptive compression can very effectively compress long sequences of similar column values
 - Compression ratio improves
10. Cluster your data
 - Sort before load, so that row order respects similarity
 - Compression ratio improves when similar rows are stored in the same page

10 best practices for DB2 10 storage are listed above.

Consider: Offline Log Archive Compression

- db2 update db cfg for tp1 using **LOGARCHCOMPR1 ON**
- db2 update db cfg for tp1 using **LOGARCHCOMPR2 ON**
- Dynamic parameters
- Use same compression algorithm as backup compression
- Save offline log archive space
- Consider time to compress when copying from active log path to offline archive log path



© Copyright IBM Corporation 2012

85

DB2 10.1 provides an opportunity to compress archived logs. It uses the same compression algorithm as the backup compression option. Just as there is a timing cost to compress backups, there is a timing cost to back up logs, but since the log files are smaller it is not as great of a timing cost.

Default Storage Group - IBMSTOGROUP

- **IBMSTOGROUP** is default storage group when CREATE DATABASE ... AUTOMATIC STORAGE YES (default).
- If create database with AUTOMATIC STORAGE NO, first storage group created with CREATE STOGROUP is default.
 - **CREATE DATABASE ...AUTOMATIC STORAGE clause is deprecated**
- Designate default storage group:
CREATE/ALTER STOGROUP ... SET AS DEFAULT
 - no impact to existing table spaces using old default storage group
- **SYSCAT.STOGROUPS. DEFAULTSG='YES'**
 - can not drop current default storage group

© Copyright IBM Corporation 2012

86

If a database has storage groups, the default storage group is used when an automatic storage managed table space is created without explicitly specifying the storage group.

When you create a database, a default storage group named IBMSTOGROUP is automatically created. However, a database created with the AUTOMATIC STORAGE NO clause, does not have a default storage group. The first storage group created with the CREATE STOGROUP statement becomes the designated default storage group. There can only be one storage group designated as the default storage group. **Note:** Although, you can create a database specifying the AUTOMATIC STORAGE NO clause, the AUTOMATIC STORAGE clause is deprecated and might be removed from a future release.

You can designate a default storage group by using either the CREATE STOGROUP or ALTER STOGROUP statements. When you designate a different storage group as the default storage group, there is no impact to the existing table spaces using the old default storage group. To alter the storage group associated with a table space, use the ALTER TABLESPACE statement.

You can determine which storage group is the default storage group by using the SYSCAT.STOGROUPS catalog view.

You cannot drop the current default storage group. You can drop the IBMSTOGROUP storage group if it is not designated as the default storage group at that time. If you drop the IBMSTOGROUP storage group, you can create another storage group with that name.

Consider: Multi-Temperature Storage

```
CREATE STOGROUP fastsg ON '/p1', '/p2', '/p3'
```

```
CREATE TABLESPACE history_sales
```

- default goes to automatic storage group IBMSTOGROUP

```
CREATE TABLESPACE current_sales
```

```
USING STOGROUP fastsg
```

```
ALTER TABLESPACE current_sales
```

```
USING STOGROUP fastsg
```

- When Alter Tablespace, it rebalances online.

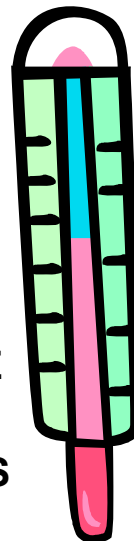
```
ALTER TABLESPACE current_sales
```

```
REBALANCE SUSPEND | RESUME
```

```
SELECT * FROM TABLE
```

```
(MON_GET_REBALANCE_STATUS
```

```
('CURRENT_SALES', -1))
```



© Copyright IBM Corporation 2012

87

Multi-temperature data storage can provide fast access to data. You can manage your IT budget more efficiently by configuring your database so that only frequently accessed data (hot data) is stored on expensive fast storage, such as solid-state drives (SSD), and infrequently accessed data (cold data) is stored on slower, less-expensive storage, such as low-rpm hard disk drives.

As hot data cools down and is accessed less frequently, you can dynamically move it to the slower storage, thereby extending the life of your less expensive storage assets that are used for storing warm and cold data.

In database systems, there is a strong tendency for a relatively small proportion of data to be hot data and the majority of the data to be cold data. Current data is often considered to be hot data, but it typically becomes cold as it ages. These sets of multi-temperature data pose considerable challenges to DBAs who want to optimize the use of fast storage by trying not to store cold data there. As a data warehouse consumes more storage, optimizing the use of fast storage becomes increasingly important for managing storage costs. With your hot data stored on your fastest storage assets, multi-temperature data storage can help to reduce the time it takes to retrieve your most frequently accessed data, while reducing the cost of storing infrequently accessed warm and cold data.

A major advance in DB2 Version 10.1 is the ability to create storage groups, which are groups of storage paths. A storage group contains storage paths with similar characteristics. Some critical attributes of the underlying storage to consider when creating or altering a storage group are available storage capacity, latency, data transfer rates, and the degree of RAID protection.

These storage groups can be used to create different classes of storage (multi-temperature storage classes) where frequently accessed (or hot) data is stored in storage paths residing on fast storage, while infrequently accessed (or cold) data is stored in storage paths residing on slower, less expensive storage. After you create storage groups that map to the different classes of storage in your database management system, you can assign automatic storage table spaces to those storage groups, based on which table spaces have hot or cold data. You can

use storage groups to physically partition table spaces managed by automatic storage. You can dynamically reassign a table space to a different storage group by using the ALTER TABLESPACE statement with the USING STOGROUP option.

A database-managed table space can be converted to an automatic storage table space by executing an ALTER TABLESPACE statement and specifying the MANAGED BY AUTOMATIC STORAGE clause on the table space. Note that after this is done, it is necessary to perform a rebalance operation on the table space by executing an ALTER TABLESPACE statement and specifying the REBALANCE clause on the table space. In 10.1, the rebalance operation is enhanced to manually SUSPEND and RESUME a rebalance operation during performance sensitive periods.

You can take further advantage of organizing your data into storage groups by configuring DB2 workload manager (WLM) to prioritize activities based on the priority of the data being accessed. The ADMIN_GET_STORAGE_PATHS table function can be used to get the list of automatic storage paths for each database storage group, including file system information for each storage path. Other table functions that have been added or modified to support the monitoring of storage groups include: MON_GET_REBALANCE_STATUS, MON_GET_TABLESPACE and MON_GET_CONTAINER.

Consider: Reclaiming unused space in indexes

- Deleting substantial data results in unused space in tables and associated indexes
- Space can not be used by any other object until reorg
- Use new online index reorg functionality to reclaim unused index space in DMS table spaces:
- **REORG INDEX FOR TABLE ... RECLAIM EXTENTS ...**
- **REORG INDEXES ALL FOR TABLE .. RECLAIM EXTENTS ..**
- Call db2Reorg API with **DB2REORG_INDEX_RECLAIM_EXTENTS** for reorgFlags in db2ReorgStruct data structure
- Set automatic index reorg and specify **reclaimExtentsSizeForIndexObjects** attribute in ReorgOptions element in input XML file

© Copyright IBM Corporation 2012

88

A new mechanism to reclaim unused space in indexes has been introduced to provide a more efficient way to free space for indexes that reside in DMS table spaces. Deleting a substantial amount of data from tables on a regular basis results in unused space in tables and associated indexes. This space cannot be used by any other object in the same table space until reorganization takes place.

In Version 10.1, you can use the new online index reorganization functionality to reclaim unused index space on tables that reside in DMS table spaces. This functionality is available through the following options:

- Issuing the REORG INDEX FOR TABLE or REORG INDEXES ALL FOR TABLE command with the new RECLAIM EXTENTS clause.
- Calling the db2Reorg API and specifying the new DB2REORG_INDEX_RECLAIM_EXTENTS value for the reorgFlags parameter in the db2ReorgStruct data structure.
- Setting automatic index reorganization and specifying the reclaimExtentsSizeForIndexObjects attribute in the ReorgOptions element in the input XML file.

INDEX index-name RECLAIM EXTENTS

Specifies the index to reorganize and reclaim extents that are not being used. This action moves index pages around within the index object to create empty extents, and then free these empty extents from exclusive use by the index object and makes the space available for use by other database objects within the table space. Extents are reclaimed from the index object back to the table space. ALLOW READ ACCESS is the default, however all access modes are supported.

TABLE table-name RECLAIM EXTENTS

Specifies the table to reorganize and reclaim extents that are not being used. The table-name variable must specify a multidimensional clustering (MDC) or insert time clustering (ITC) table. The name or alias in the form: schema.table-name can be used. The schema is the user name under which the table was created. If you omit the schema name, the default schema is assumed.

For **REORG TABLE RECLAIM EXTENTS** when the **ON DATA PARTITION** clause is specified, the access clause only applies to the named partition. Users can read from and write to the rest of the table while the extents on the specified partition are being reclaimed. This situation also applies to the default access levels.

Consider ITC: Insert Time Clustering Tables

- ITC and MDC tables both use block based allocation and block indexes
- MDC tables are clustered based on dimensions specified by `CREATE TABLE ... ORGANIZE BY`



- ITC tables cluster data by using virtual column which clusters rows that are inserted at similar time, together:

CREATE TABLE ... ORGANIZE BY INSERT TIME ...

- Can convert existing tables with `ADMIN_MOVE_TABLE` or export/import. Can not alter to become ITC table.

© Copyright IBM Corporation 2012

89

Insert time clustering (ITC) tables provide an effective way of maintaining data clustering and easier management of space utilization.

ITC tables have similar characteristics to MDC tables. For example, these table types use block based allocation and block indexes. ITC and MDC tables differ in how data is clustered. ITC tables cluster data by using a virtual column which clusters rows, that are inserted at a similar time, together. Clustering dimensions on MDC tables are specified by the creator.

ITC tables are created with the `CREATE TABLE` command by specifying the `ORGANIZE BY INSERT TIME` clause.

A convenient, online way to convert existing tables to ITC tables is the `ADMIN_MOVE_TABLE` procedure. Another method to convert existing tables to ITC tables is export/import or a load from table. Existing tables cannot be altered to become ITC tables.

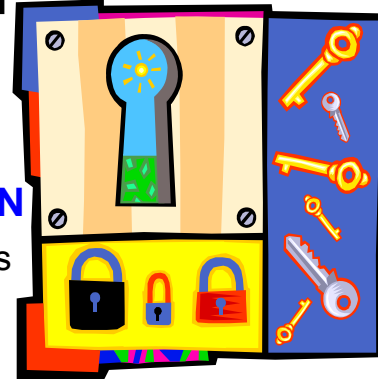
Consider: Partitioned tables remain accessible

- When add or attach data partition,
**partitioned table remains accessible to queries
when data partition added or attached**

- **ALTER TABLE ... ADD PARTITION**
or
ALTER TABLE ATTACH PARTITION

- Table remains accessible to dynamic queries running under RS, CS, or UR isolation level.
- DB2 9.7 Fixpack 1, enabled table to remain accessible when detach data partition

- Requires all partitioned indexes (local indexes)



© Copyright IBM Corporation 2012

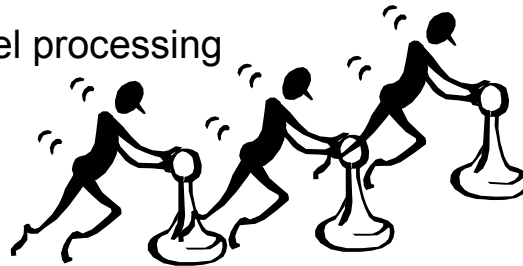
90

Partitioned tables remain accessible to queries when a data partition is added or attached

The process of adding or attaching a data partition to a partitioned table by using the ALTER TABLE statement with the ADD PARTITION or ATTACH PARTITION clause has been enhanced. The partitioned table now remains accessible to dynamic queries running under the RS, CS, or UR isolation level.

Consider: More Parallelism with db2move

- **db2move** now supports parallel processing
 - have load operations for tables in schema(s) spread across multiple threads
- If **COPY** action specified when copying schema, can now load tables in schema in parallel by using **PARALLEL *number-of-threads***
- If PARALLEL is specified without number of threads, db2move utility chooses appropriate value
 - Range for *number-of-threads* is 0-16.
 - If 0 or 1, load performed serially



© Copyright IBM Corporation 2012

91

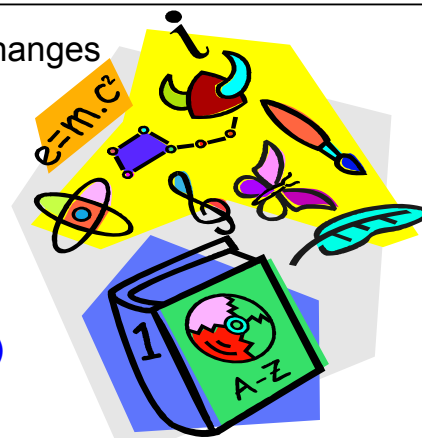
Specify **PARALLEL *number-of-threads*** option to have the load operations for the tables in the schema(s) spread across a number of threads. The value range for *number-of-threads* is 0-16.

If PARALLEL is not specified, no threads are used and the load operations are performed serially. If PARALLEL is specified without a number of threads, the db2move utility will choose an appropriate value. If PARALLEL is specified and *number-of-threads* is provided, the specified number of threads is used. If *number-of-threads* is 0 or 1, the load operation is performed serially.

The maximum value that can be specified for *number-of-threads* is 16. This is an optional option available to the COPY action.

Consider: Tracking Change History

- New change history event monitor tracks changes to system that might affect performance
 - Including automatic changes
 - STMM or automatic reorg/runstats
- **CREATE EVENT MONITOR *BKP_REST* FOR CHANGE HISTORY WHERE EVENT IN (BACKUP, RESTORE) WRITE TO TABLE**
- Control Options: ALL, BACKUP, CFGALL, DBCFG, DBCFGVALUES, DBMCFG, DBMCFGVALUES, DDLALL,DDLDATA, DDLFEDERATED, DDLMONITOR, DDLSECURITY, DDLSQL, DDLSTORAGE, DDLWLM, LOAD, MOVETABLE, REDISTRIBUTE, REGVAR, REGVARVALUES, REORG, RESTORE, ROLLFORWARD, RUNSTATS, UTILALL
- See **CREATE EVENT MONITOR (change history) statement in Information Center**



© Copyright IBM Corporation 2012

92

New event monitor tracks configuration and registry changes and DDL and utility executions

The change history event monitor captures changes to database and database manager configuration and registry settings, execution of DDL statements, and execution of utilities. You can use this data to determine if the appearance of any problems with your database coincides with any of these events.

Some examples of changes to the system that might affect the performance or behavior of work on the system include:

- v Creating or dropping an index unexpectedly
- v The failure of scheduled maintenance to run
- v Changes to a database configuration parameter or DB2 registry setting

Unintentional or unanticipated changes affecting database operations might be made by a user; for example, a DBA might execute DDL to drop an index. Or, changes might occur automatically, without any user interaction; for example, the self-tuning memory manager (STMM) might change a configuration parameter, or automatic table reorganization might reorganize a table. Either way, you can use the change history event monitor to track several different types of

The change history event monitor can capture change-related events for a number of actions and operations, including:

- Database and database manager configuration parameter changes
- Registry variable changes
- Execution of DDL statements
- Change history event monitor startup
- Execution of the following DB2 utilities and commands:
 - LOAD
 - ADMIN_MOVE_TABLE procedure invocations
 - BACKUP DATABASE (ONLINE option only)
 - RESTORE DATABASE (ONLINE option only)
 - ROLLFORWARD DATABASE (ONLINE option only)
 - REDISTRIBUTE DATABASE PARTITION GROUP
 - REORG
 - RUNSTATS

Student Notes: Reference Only

Manually tracking changes to the database server can be a difficult task. Historically, the information for different types of changes has been captured through different interfaces. Configuration updates are written to the diagnostic log files (db2diag log files), while utility progress is captured in the database history file. The change history event monitor provides you with a single interface that captures the events that change the behavior and performance characteristics of your database system. Using the event monitor tables, you can investigate any change events that are of interest.

The change history event monitor can record certain types of changes made even when the database is offline. Generally, information related to events that occur while the change history event monitor is inactive or the database is offline are not captured. However, the change history event monitor can be configured to capture the registry variable values that are in effect when the an event monitor is activated. Similarly, database and database manager configuration parameter values can be captured when a change history event monitor is activated. When capturing configuration parameter values, the event monitor can detect if any configuration parameters were changed while the event monitor was inactive, and so the event monitor captures configuration parameter values only if changes occur.

Consider: Usage list database object

- Monitor all unique sections (DML statements) that reference a table or index during execution
- **CREATE USAGE LIST** *usage-list-name*
FOR TABLE | INDEX *object-name*
LIST SIZE 100 | *integer-value*
WHEN FULL WRAP | DEACTIVATE
INACTIVE | ACTIVATE ON START DATABASE
- **SET USAGE LIST ACTIVE | INACTIVE | RELEASED**
- **MON_GET_USAGE_LIST_STATUS** (schema, name, member)
 - **STATE, LAST_STATE_CHANGE, LIST_SIZE, USED_ENTRIES, LIST_MEM_SIZE, WRAPPED**



© Copyright IBM Corporation 2012

94

The CREATE USAGE LIST statement defines a usage list. A usage list is a database object for monitoring all unique sections (DML statements) that have referenced a particular table or index during their execution. This statement can be embedded in an application program or issued interactively. It is an executable statement that can be dynamically prepared only if DYNAMICRULES run behavior is in effect for the package.

The privileges held by the authorization ID of the statement must include either DBADM or SQLADM authority.

CREATE USAGE LIST *usage-list-name* **FOR TABLE | INDEX** *object-name* **LIST SIZE** *integer-value* **WHEN FULL WRAP | DEACTIVATE INACTIVE | ACTIVATE ON START DATABASE**

The table name must not specify an alias, catalog table, created temporary table, hierarchy table, detached table, nickname, typed table, or view.

Indexes defined on tables other than untyped tables or materialized query tables are not supported. The name must specify a physical index; Block Indexes (BLOK), Clustering indexes (CLUS), Dimension block indexes (DIM), Regular indexes (REG), and Physical indexes over XML column (XVIP).

LIST SIZE specifies that the size of this list is integer-value entries. The minimum size that can be specified is 10 and the maximum is 5000. The default size is 100 entries. The default is to wrap when the list becomes full.

If the usage list is not activated for monitoring whenever the database is activated, collection must be explicitly started using the SET USAGE LIST statement. This clause is the default.

Tracking sections with unique keys: A usage list keeps tracks of all unique sections (DML statements only) that have referenced a particular object. References are aggregated within the list with the unique key of executable ID, representing the section doing the reference, and the monitor interval ID at the time of the reference. Each list entry keeps a count of section executions related to that entry and a set of statistics outlining the affect that the section had on the object across those executions.

Memory allocation: Memory is allocated the first time that the object for which the usage list is defined is referenced by a section.

Memory allocation in a partitioned database environment or DB2pureScale environment: If the state of a usage list for a partitioned table or index is set to active, memory is allocated for each data partition when the data partition is first referenced by the section. Similarly, in a partitioned database environment or DB2 pureScale environment, memory is allocated at each active member. If a member is unavailable at the time of activation, then the memory is allocated when the member is next activated (if the state of the usage list is still set to active). This also applies when a member is added to the cluster.

Student Notes: Reference Only

Implicit reactivation of an active usage list: If the state of an INACTIVE ON START DATABASE usage list is set to active in a partitioned database environment or DB2 pureScale environment, then its behavior is similar to the ACTIVE ON START DATABASE clause until the state of the usage list is explicitly set to inactive or the instance is recycled. That is, if the state of a usage list is active when a database member is deactivated or offline, and that database member is subsequently reactivated, the usage list for this member is also implicitly reactivated.

Inactive usage lists remain inactive upon database member reactivation: If the state of an ACTIVE ON START DATABASE usage list is set to inactive in a partitioned database environment or DB2 pureScale environment, then its behavior is similar to the INACTIVE ON START DATABASE clause until the state of the usage list is explicitly set to active or the instance is recycled. That is, if the state of a usage list is inactive when a database member is deactivated or offline, and that database member is subsequently reactivated, the state of the usage list for this member will remain inactive.

Multiple usage lists: Multiple usage lists can be created for the same table or index, however, it is recommended that only one of them be activated. Activating all of them affects database performance and memory usage.

Usage list size considerations: When the state of a usage list is set to active, the memory for the usage list is allocated from the monitor heap. At the maximum list size setting, the usage list is approximately 2MB. For partitioned tables or indexes, memory is allocated for each data partition. For example, if a partitioned table has three data partitions defined, approximately 6MB of memory is allocated. Therefore, activating multiple usage lists imposes more memory requirements on the monitor heap. It is therefore suggested that a reasonable list size is selected, or that you set the `mon_heap_sz` configuration parameter to AUTOMATIC so that the database manager manages the monitor heap size.

Performance considerations: To maintain high performance, create usage lists such that they are limited to the amount required to gather the information you need. Each usage list requires system memory; system performance can degrade as additional usage lists are activated.

The `MON_GET_USAGE_LIST_STATUS` table function returns current status of a usage list. The `MON_GET_INDEX_USAGE_LIST` table function returns information from a usage list defined for an index.

Consider: Usage list database object (cont'd)

- **MON_GET_TABLE_USAGE_LIST** (schema, name, member)
 - LAST_UPDATED, NUM_REFERENCES, NUM_REF_WITH_METRICS, ROWS_INSERTED, ROWS_DELETED, ROWS_UPDATED, ROWS_READ, OVERFLOW_CREATES, OVERFLOW_ACCESSES, LOCK_WAIT_TIME, LOCK_WAIT_TIME_GLOBAL, LOCK_WAITS, LOCK_WAITS_GLOBAL, LOCK_ESCALS, LOCK_ESCALS_GLOBAL, DIRECT_WRITES, DIRECT_WRITE_REQS, DIRECT_READS, DIRECT_READ_REQS, OBJECT_DATA_L_READS, OBJECT_DATA_P_READS, OBJECT_DATA_GBP_L_READS, OBJECT_DATA_GBP_P_READS BIGINT, OBJECT_DATA_GBP_INVALID_PAGES, OBJECT_DATA_LBP_PAGES_FOUND, OBJECT_DATA_GBP_INDEP_PAGES_FOUND_IN_LBP, OBJECT_XDA_L_READS, OBJECT_XDA_P_READS, OBJECT_XDA_GBP_L_READS, OBJECT_XDA_GBP_P_READS, OBJECT_XDA_GBP_INVALID_PAGES, OBJECT_XDA_LBP_PAGES_FOUND, OBJECT_XDA_GBP_INDEP_PAGES_FOUND_IN_LBP
- **MON_GET_INDEX_USAGE_LIST** (schema, name, member)
 - USAGELISTSHEMA, USAGELISTNAME, INDSHEMA, INDNAME, MEMBER, DATA_PARTITION_ID, EXECUTABLE_ID, MON_INTERVAL_ID, LAST_UPDATED, NUM_REFERENCES, NUM_REF_WITH_METRICS, OBJECT_INDEX_L_READS BIGINT, OBJECT_INDEX_P_READS BIGINT, OBJECT_INDEX_GBP_L_READS BIGINT, OBJECT_INDEX_GBP_P_READS BIGINT, OBJECT_INDEX_GBP_INVALID_PAGES, OBJECT_INDEX_LBP_PAGES_FOUND, OBJECT_INDEX_GBP_INDEP_PAGES_FOUND_IN_LBP

© Copyright IBM Corporation 2012

96

The MON_GET_USAGE_LIST_STATUS table function returns current status of a usage list.

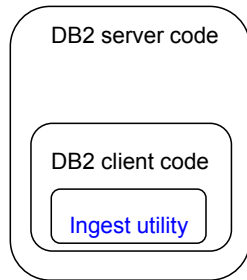
The MON_GET_INDEX_USAGE_LIST table function returns information from a usage list defined for an index.

The MON_GET_TABLE_USAGE_LIST table function returns information from a usage list defined for a table.

Consider INGEST Utility

Deployment options

Client-side utility

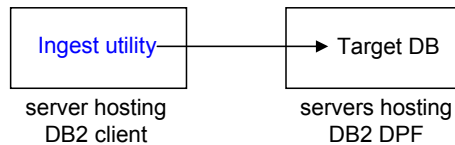


Supported environments

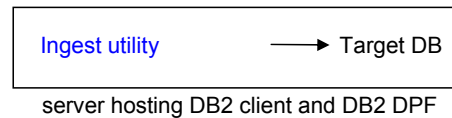
- serial
- DPF
- pureScale

Topology options

1. Remote client



2. Local client



© Copyright IBM Corporation 2012

97

This slide shows deployment options for the utility. Note that the utility is a client-side utility, but could run on the same machine as the server or its own dedicated machine.

DB2 LUW ingest utility is a high-speed client-side utility that ingests data from files or pipes into DB2 LUW tables, using SQL-like commands. Utility features include flexible input formats allowing fields in various orders and formats, support for INSERT, UPDATE, DELETE, and MERGE, support for SQL expressions and functions to construct column values, error recovery, restart, and monitoring.

Most basic Ingest Utility command

```
INGEST FROM FILE my_file.del
FORMAT DELIMITED
INSERT INTO my_table;
```

file
containing
input records

Fields are
separated by
commas and
correspond to
table columns.

Insert records
into the target
table

© Copyright IBM Corporation 2012

98

This slide shows the simplest most basic form of the INGEST command. You can use this when the input fields are separated by commas and correspond to the table columns.

You may specify:

	Parameter		Range	
	Default	Explanation		
•COMMIT_COUNT	0 to max 32-bit integer		0	Number of rows each flusher
	writes in a single transaction before issuing a commit.			
•COMMIT_PERIOD	0 to 2,678,400 (31 days)	1 second		Number of seconds between committed transactions.

Ingest input types and formats (1)

- Input types: file or named pipe (can also specify multiple files or multiple pipes)
- Input formats
 - DELIMITED
 - Equivalent to "OF DEL" on the IMPORT and LOAD commands.
 - Fields are separated by a one-byte character (default is comma).
 - Records are always varying length (delimited by CRLF or LF).
 - Field definition list is optional and defaults to specified or implied table columns.
 - POSITIONAL
 - Equivalent to "OF ASC" on the IMPORT and LOAD commands.
 - Fields are at fixed offsets in each record and have a fixed length.
 - Records can be varying length (delimited by CRLF or LF) or fixed length.
 - Field definition list is required.
 - Allows numbers to be specified in binary

This slide contains more details about the input types and formats that the utility supports.

Ingest Input types and formats (2)

- Other options
 - INPUT CODEPAGE parameter allows specifying codepage of input data.
 - IMPLICITLYHIDDENMISSING or IMPLICITLYHIDDENINCLUDE specify whether implicitly hidden columns are to be included when the column list is omitted on the INSERT or REPLACE statement (or the INSERT clause of the MERGE statement).

PARAMETER	RANGE	DEFAULT	EXPLANATION
• COMMIT_COUNT	0 to max 32-bit integer	0	Number of rows each flusher writes in a single transaction before issuing a commit.
• COMMIT_PERIOD	0 to 2,678,400 (31 days)	1 second	Number of seconds between committed transactions.

This slide contains more details about other formatting options that the utility supports.

Ingest with SQL statements - UPDATE

- The INGEST command can specify a subset of the INSERT, UPDATE, DELETE, or MERGE statements.

```
-- Update records in the table.
INGEST FROM FILE my_file.del FORMAT DELIMITED
(
    $key_fld1 INTEGER EXTERNAL,
    $key_fld2 INTEGER EXTERNAL,
    $data_fld1 CHAR(8),
    $data_fld2 CHAR(8),
    $data_fld3 CHAR(8)
)
UPDATE my_table
SET (data_col1, data_col2, data_col3) =
    ($data_fld1, $data_fld2, $data_fld3)
WHERE (key_col1 = $key_fld1) AND
      (key_col2 = $key_fld2);
```

© Copyright IBM Corporation 2012

101

This slide shows an example of an UPDATE statement. Here we update the non-key columns in the table with the values from the input file.

Ingest with SQL statements - MERGE

```
-- Merge input records into the table.
INGEST FROM FILE my_file.del FORMAT DELIMITED
(
  ... same field definition list as before ...
)
MERGE INTO my_table
ON (key_col1 = $key_fld1) AND (key_col2 = $key_fld2)
WHEN NOT MATCHED THEN
  INSERT VALUES($key_fld1, $key_fld1,
    $data_fld1, $data_fld2, $data_fld3)
WHEN MATCHED THEN
  UPDATE SET (data_col1, data_col2, data_col3) =
    ($data_fld1, $data_fld2, $data_fld3);
```

This slide shows an example of a MERGE statement. For each input record, if the key matches an existing row in the table, we update the non-key columns with new values. Otherwise, we add a new row.

Ingest with SQL statements - DELETE

```
-- Delete records from the table that correspond to
-- input records.
INGEST FROM FILE my_file.del FORMAT DELIMITED
(
  ... same field definition list as before ...
)
DELETE FROM my_table
WHERE (key_col1 = $key_fld1) AND
      (key_col2 = $key_fld2);
```

This slide shows an example of a DELETE statement. For each input record, if the key matches an existing row in the table, we delete the row.

Monitoring – Example of INGEST LIST

```
=> INGEST LIST
Ingest job ID      =
  DB2100000:20101116.123456.234567:34567:45678
Ingest temp job ID = 4
Database Name      = MYDB
Input type         = FILE
Target table       = MY_SCHEMA.MY_TABLE
Start Time         = 01/10/2012 11:54:45.773215
Running Time       = 00:02:03
Number of records processed = 30,000
```

This slide shows an example of a type of monitoring

Monitoring – Example of INGEST GET STATS

=> INGEST GET STATS FOR 4 EVERY 3 SECONDS

Ingest job ID = DB2100000:20101116.123456.234567:34567:45678
Database = MYDB
Target table = MY_SCHEMA.MY_TABLE

Overall ingest rate (records/second)	Overall write rate (writes/second)	Current ingest rate (records/second)	Current write rate (writes/second)	Total records
3333	65432	76543	87654	108765
3334	75432	86543	97654	118765
3335	85432	96543	107654	128765

etc (new row every 3 seconds until INGEST command ends)

Temp job ID from
INGEST LIST
command

This slide shows an example of another type of monitoring

Comparison to IMPORT and LOAD – Table types

Table type	_Import	Load	Ingest
Created global temporary table	no	no	no
Declared global temporary table	no	no	no
Detached table	no	no	no
Multidimensional clustering (MDC) table		yes	yes
Materialized query table (MQT) that is maintained by user	yes	yes	yes
Nickname	relational except ODBC	no (SQL02305N)	yes
Range-clustered table (RCT)	yes	no	yes
Range-partitioned table	yes	yes	yes
Summary table	no	yes	yes
Typed table	yes		no
Typed view	yes		no
Untyped (regular) table	yes	yes	yes
Updatable view	yes	no (SQL02305N)	yes

This slide compares the 3 utilities by the types of tables they support.

Comparison to IMPORT and LOAD – Column types

Column data type	Import	Load	Ingest
Numeric: SMALLINT, INTEGER, BIGINT, DECIMAL, REAL, DOUBLE, DECFLOAT	yes	yes	yes
Character: CHAR, VARCHAR, NCHAR, NVARCHAR, plus corresponding FOR BIT DATA types	yes	yes	yes
Graphic: GRAPHIC, VARGRAPHIC	yes	yes	yes
Long types: LONG VARCHAR, LONG VARGRAPHIC	yes	yes	yes
Date/time: DATE, TIME, TIMESTAMP(p)	yes	yes	yes
DB2SECURITYLABEL	yes	yes	yes
LOBs from files: BLOB, CLOB, DBCLOB, NCLOB	yes	yes	no
inline LOBs	yes	yes	no
XML from files	yes	yes	no
inline XML	no	no	no
distinct type (note 1)	yes	yes	yes
structured type	no	no	no
reference type	yes	yes	yes

Notes: (1) Supported if based on a supported built-in type.

This slide compares the 3 utility by the column types they support.

Comparison to IMPORT and LOAD – Input types and formats

Input type	Import	Load	INGEST
cursor	no	yes	no
device	no	yes	no
file	yes	yes	yes
pipe	no	yes	yes
multiple input files, multiple pipes, etc	no	yes	yes

Input format	Import	Load	INGEST
ASC (including binary)	yes, except binary	yes	yes
DEL	yes	yes	yes
IXF	yes	yes	no
WSF (worksheet format)	yes, but will be discontinued in V10.1	no	no

© Copyright IBM Corporation 2012

108

This slide compares the 3 utility by input type and input format.

Consider: Data Prefetching Monitoring Elements

- pool_failed_async_data_reqs
- pool_failed_async_index_reqs
- pool_failed_async_other_reqs
- pool_failed_async_temp_data_reqs
- pool_failed_async_temp_index_reqs
- pool_failed_async_temp_xda_reqs
- pool_failed_async_xda_reqs
- pool_queued_async_data_pages
- pool_queued_async_data_reqs
- pool_queued_async_index_pages
- pool_queued_async_index_reqs
- pool_queued_async_other_reqs
- pool_queued_async_temp_data_pages
- pool_queued_async_temp_data_reqs
- pool_queued_async_temp_index_pages
- pool_queued_async_temp_index_reqs
- pool_queued_async_temp_xda_pages
- pool_queued_async_temp_xda_reqs
- pool_queued_async_xda_pages
- pool_queued_async_xda_reqs
- pool_sync_data_gbp_reads
- pool_sync_data_reads
- pool_sync_index_gbp_reads
- pool_sync_index_reads
- pool_sync_xda_gbp_reads
- pool_sync_xda_reads
- prefetch_waits
- skipped_prefetch_data_p_reads
- skipped_prefetch_index_p_reads
- skipped_prefetch_temp_data_p_reads
- skipped_prefetch_temp_index_p_reads
- skipped_prefetch_temp_xda_p_reads
- skipped_prefetch_uow_data_p_reads
- skipped_prefetch_uow_index_p_reads
- skipped_prefetch_uow_temp_data_p_reads
- skipped_prefetch_uow_temp_index_p_reads
- skipped_prefetch_uow_temp_xda_p_reads
- skipped_prefetch_uow_xda_p_reads
- skipped_prefetch_xda_p_reads

Many new monitor elements have been added in Version 10.1.

New monitor elements provide added insight into the operation of your DB2 server.

Consider: New Monitor Elements

- **Connection request and authentication processing times**

- total_connect_authentication_proc_time
- total_connect_authentication_time
- total_connect_request_proc_time
- total_connect_request_time
- total_connect_requests

- **Package cache**

- max_coord_stmt_exec_time_args
- db2_max_coord_stmt_exec_time_args
- db2_max_coord_stmt_exec_time_arg
- db2_max_coord_stmt_exec_time_arg_type
- max_coord_stmt_exec_time
- max_coord_stmt_exec_timestamp
- max_coord_stmt_exec_time
- routine_id

© Copyright IBM Corporation 2012

110

There are new monitor elements for connection request and authentication processing times

There are new monitor elements that report the following items:

- The maximum statement execution time
- The input variables that were associated with that execution of the statement
- The time that the statement started executing
- The routine that was associated with that statement

Consider: New miscellaneous monitor elements

- **disabled_peds** - The number of times that partial early distinct operations were disabled because insufficient sort heap was available.
- **edu_ID** - The ID of the EDU with which a memory pool is associated.
- **evmon_waits_total** - The number of times that an agent waited for an event monitor record to become available.
- **index_jump_scans** - The number of jump scans. A jump scan is an index scan where there is a gap in the index start and stop keys and sections of the index that will not yield results are skipped.
- **index_name** - The name of an index.
- **index_schema** - The name of an index schema.
- **mon_interval_id** - The value of the MON_INTERVAL_ID global variable at the time that a transaction was completed.
- **num_page_dict_built** - The number of page-level compression dictionaries that were created or re-created for a table.
- **post_threshold_peds** - The number of times that partial early distinct operations received less memory than requested because the sort heap threshold was exceeded.
- **total_peas** - The total number of times that partial early aggregation operations were executed.
- **total_peds** - The total number of times that partial early distinct operations were executed.
- **total_stats_fabrications** - The total number of statistics fabrications that real-time statistics gathering performed.
- **total_stats_fabrications_time** - in milliseconds
- **total_sync_runstats** - The total number of synchronous RUNSTATS command activities that real-time statistics gathering triggered.
- **total_sync_runstats_time** - in milliseconds

© Copyright IBM Corporation 2012

111

The following lists new miscellaneous monitor elements.

- **disabled_peds** - The number of times that partial early distinct operations were disabled because insufficient sort heap was available.
- **edu_ID** - The ID of the EDU with which a memory pool is associated.
- **evmon_waits_total** - The number of times that an agent waited for an event monitor record to become available.
- **index_jump_scans** - The number of jump scans. A jump scan is an index scan where there is a gap in the index start and stop keys and sections of the index that will not yield results are skipped.
- **index_name** - The name of an index.
- **index_schema** - The name of an index schema.
- **mon_interval_id** - The value of the MON_INTERVAL_ID global variable at the time that a transaction was completed.
- **num_page_dict_built** - The number of page-level compression dictionaries that were created or re-created for a table.
- **post_threshold_peds** - The number of times that partial early distinct operations received less memory than requested because the sort heap threshold was exceeded.
- **total_peas** - The total number of times that partial early aggregation operations were executed.
- **total_peds** - The total number of times that partial early distinct operations were executed.
- **total_stats_fabrications** - The total number of statistics fabrications that real-time statistics gathering performed.
- **total_stats_fabrications_time** - in milliseconds
- **total_sync_runstats** - The total number of synchronous RUNSTATS command activities that real-time statistics gathering triggered.
- **total_sync_runstats_time** - in milliseconds

Upgrade impact from system catalog changes (1)

- System catalog objects modified to support new functionality which can impact existing applications and scripts after upgrade to 10.1.
 - Modifications consist of new columns, changed column data types, or longer column lengths.
- New columns added in V10.1 that impact applications and scripts:
 - [SYSCAT.AUDITUSE](#) - AUDITEXCEPTIONENABLED
 - [SYSSTAT.COLUMNS](#) - ROWBEGIN, ROWEND, QUALIFIER, FUNC_PATH
 - [SYSCAT.CONTEXTS](#) - AUDITEXCEPTIONENABLED
 - [SYSCAT.DATATYPES](#) - NULLS, FUNC_PATH, CONSTRAINT_TEXT
 - [SYSCAT.DBAUTH](#) - CREATESECUREAUTH
 - [SYSCAT.EVENTMONITORS](#) - VERSIONNUMBER, MEMBER
 - [SYSCAT.EVENTTABLES](#) - TABOPTIONS
 - [SYSCAT.INDEXCOLUSE](#) - VIRTUAL, TEXT
 - [SYSCAT.INDEXES](#) - PERIODNAME, PERIODPOLICY, MADE_WITHOUTOVERLAPS, NULLKEYS, FUNC_PATH

Upgrade impact from system catalog changes (2)

- [SYSCAT.INDEXXMLPATTERNS](#) - SCALE
- [SYSCAT.PACKAGES](#) - BUSTIMESENSITIVE, SYSTIMESENSITIVE, KEEPDYNAMIC, STATICSDYNAMIC, MEMBER
- [SYSCAT.ROLES](#) - AUDITEXCEPTIONENABLED
- [SYSCAT.ROUTINES](#) - SECURE
- [SYSCAT.ROWFIELDS](#) - NULLS, QUALIFIER, FUNC_PATH, DEFAULT
- [SYSCAT.SCHEMATA](#) - AUDITPOLICYID, AUDITPOLICYNAME, AUDITEXCEPTIONENABLED, DATACAPTURE
- [SYSCAT.SERVICECLASSES](#) - MAXDEGREE, CPUSHARES, CPUSHARETYPE, CPULIMIT, SORTMEMORYPRIORITY, SECTIONACTUALOPTIONS, COLLECTAGGUOWDATA
- [SYSCAT.TABCONST](#) - TRUSTED, PERIODNAME, PERIODPOLICY
- [SYSCAT.TABLES](#) - ROWCOMPMODE, AUDITEXCEPTIONENABLED, CONTROL, TEMPORALTYPE
- [SYSCAT.TABLESPACES](#) - SGNAME, EFFECTIVEPREFETCHSIZE, SGID, BPTIERNUM.
 - Column DATAPRIORITY with value CHAR(1) deprecated and replaced by DATATAG with value SMALLINT.
- [SYSCAT.THRESHOLDS](#) - DATATAGLIST, OVERFLOWPERCENT
- [SYSCAT.TRIGGERS](#) - EVENTUPDATE, EVENTDELETE, EVENTINSERT, SECURE, ALTER_TIME, ENABLED
- [SYSCAT.VARIABLES](#) - NULLS
- [SYSCAT.WORKACTIONS](#) - SECTIONACTUALOPTIONS.
- [SYSCAT.WORKCLASSES](#) - Following columns discontinued: WORKCLASS_DESC, USAGEAUTH, COLLECTAGGACTDATA, COLLECTACTDATA, WORKLOAD_DESC, COLLECTACTPARTITION
- [SYSCAT.WORKLOADS](#) - MAXDEGREE, COLLECTUOWDATA, SECTIONACTUALOPTIONS, COLLECTAGGUOWDATA

Built-in administrative routines and views

- ADMIN_GET_TAB_COMPRESS_INFO_V97 and ADMINTABCOMPRESSINFO - deprecated and replaced by ADMIN_GET_TAB_COMPRESS_INFO and ADMIN_GET_TAB_DICTIONARY_INFO.
- MON_BP_UTILIZATION - returns several new metrics
- MON_GET_BUFFERPOOL - returns additional new columns for asynchronous reading and writing from buffer pool, and I/O servers efficiency metrics
- MON_GET_PKG_CACHE_STMT - returns additional new columns
- MON_GET_TABLESPACE - returns additional new columns
- MON_GET_CONTAINER - returns new column
- SNAP_GET_STORAGE_PATHS_V97 and SNAPSTORAGE_PATHS - deprecated and replaced by ADMIN_GET_STORAGE_PATHS.
- SNAP_GET_TBSP_PART - replaces SNAP_GET_TBSP_PART_V91.

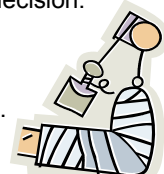
WLM - built-in administrative routines and views

- Replacing version-suffixed routine which are deprecated.
- Routine names now remain same across releases to improve compatibility for applications and scripts across releases.
- [WLM GET SERVICE CLASS WORKLOAD OCCURRENCES](#) - returns additional columns.
- [WLM GET SERVICE SUBCLASS STATS](#) - returns additional columns.
- [WLM GET WORKLOAD STATS](#) - returns additional columns.
- [WLM GET WORKLOAD STATS](#)
- [WLM GET WORKLOAD OCCURRENCE ACTIVITIES](#) – Also returns additional columns.
- [WLM GET SERVICE CLASS AGENTS](#) - returns additional column.
- [WLM GET SERVICE CLASS WORKLOAD OCCURRENCES](#) - returns additional columns.

Upgrade essentials for routines

Implicit casting

- DB2 decides which function is "best fit". A comparison of data types of arguments with defined data types of parameters of functions under consideration forms basis for this decision.
 - Untyped parameter marker or untyped NULL constant argument accepts any parameter type as a best fit.
 - impacts function resolution that involves modified system built-in functions and any new functions created using these arguments.



XML data is passed by reference in SQL routines

- In SQL routines, when assign XML data to input and output parameters of XML type or local variables of XML type, XML data is now passed by reference.
 - Previously, XML data was passed by value in SQL procedures. Some operations using XML data in SQL procedures can return results different from results returned by same operations in previous releases.

Unfenced external routines

- During database upgrade to DB2 10.1, all external unfenced routines that have no dependency on DB2 engine libraries (libdb2e.a or libdb2apie.a) are altered to FENCED and NOT THREADSAFE so can safely run under multithreaded database manager.
 - Running external routines defined as NOT FENCED and THREADSAFE can yield incorrect results, database corruption, or abnormal termination of the database manager.

Java external routines

- IBM Software Developer's Kit (SDK) for Java 1.4.2 is deprecated and might be discontinued in future.
- Starting with DB2 9.5, default JDBC driver to run JDBC routines is the IBM Data Server Driver for JDBC and SQLJ.

Upgrade essentials describe changes in application development support, changes to support new functionality, unsupported functionality, and deprecated functionality that might impact your routines.

The changes described in for database applications could also impact your routines.

Checklist: Best practices for upgrading DB2 servers

- Review changes in existing DB2 database product functionality
- Perform hardware and operating system upgrades before DB2 database product upgrade
- Benchmark DB2 server performance
- Devise a plan to reverse an upgrade
- Perform pre-upgrade tasks
- Determine whether to upgrade DB2 servers or clients first
- Upgrade database applications and routines
- Upgrading DB2 High Availability Disaster Recovery (HADR) environments
 - only primary upgrade supported
- Migrating SQL replication environments
- Upgrading DB2 Spatial Extender
- Upgrading Microsoft Cluster Server environments
- Upgrading from Query Patroller to Workload Manager
 - Query Patroller is discontinued

© Copyright IBM Corporation 2012

117

When planning your DB2 server upgrade, there are a number of best practices to consider. Review these best practices before you start your upgrade.

After upgrading your database servers, you can optionally migrate your SQL replication environment to DB2 Version 10.1.

In a Microsoft Cluster Server (MSCS) environment, install DB2 10.1 as a new copy and then run the db2iupgrade command to upgrade the MSCS instance.

Summary

- Describe changes to DB2 10 packaging and software and operating system requirements.
- Understand DB2 10 changes to DDL, utilities, tools and monitoring.
- Configure a database and instance to implement DB2 10 new features.
- Analyze DB2 10 changes that affect applications and database runtime behavior.
- Develop a DB2 10 upgrade strategy.



© Copyright IBM Corporation 2012

118

Abstract:

Are you responsible for upgrading your database to DB2 10 the newest version for Linux, UNIX and Windows? Attend this presentation to learn about often overlooked but very important tips and tricks to understand before upgrading. Are you looking for details about what will change when you upgrade? Do you know what will not change? What new features will be implemented when you create a new database versus upgrade your existing database. Planning your upgrade is much easier if you know these best practices.

New Course - DB2 10.1 for LUW New Features and Database Migration Considerations

- Course Title: **DB2 10.1 for LUW New Features and Database Migration Considerations**



Course Code: **CL313** (classroom)
3L313 (ILO- Instructor Led Online)

- This course can be customized to meet your needs.
- Check Out IBM Education Offerings at:
<http://www.ibm.com/software/data/education/>

© Copyright IBM Corporation 2012

119

Course Title: DB2 10.1 for LUW New Features and Database Migration Considerations
Course Code: CL313 , 3L313 (ILO)
Course Length : 2 days
Skill Level: Intermediate
Delivery Type: Classroom

Course Description :

Get the technical information about the new functions and features available in DB2 10.1 for Linux, UNIX, and Windows and plan the migration for DB2 Version 9.5, 9.7 or 9.8 database systems to DB2 LUW 10.1.

DB2 Version 10.1 for Linux, UNIX, and Windows continues to deliver database efficiency, simplicity, and reliability. Important new features and enhancements address the needs of your business. Whether your needs are improving business critical reliability and performance, simplifying consolidation and security, increasing business insight, reducing costs, or providing a resilient system for the valuable information assets of your company, DB2 Version 10.1 delivers.

The course will describe the adaptive compression feature of DB2 LUW 10.1 that improves table compression through an advanced row compression technique that uses two levels of compression dictionaries (table-level and page-level) to improve compression ratios, particularly as data changes.

DB2 LUW 10.1 support for Multi-temperature storage allows administrators to define storage groups that can be used to assign data based on performance characteristics and migrate data to a storage group with lower cost devices or lower performance when the demand for that data declines.

The lectures will describe the DB2 LUW 10.1 performance related features including improved query optimizer techniques and functionality including star schema query optimization, improved data and index prefetching, and improved use of statistical views.

We will discuss how to implement system period and application period temporal tables associated with Time Travel Query to assign time-based state information to your data, so you can query the past state of your data.

DB2 LUW 10.1 introduces row and column access control (RCAC), as an additional layer of data security. Row and column access control is sometimes referred to as fine-grained access control or FGAC. We will discuss using RCAC to control access to a table at the row level, column level, or both and how RCAC can be used to complement the table privileges model.

The database high availability feature HADR is significantly enhanced with DB2 LUW 10.1. We will discuss how with DB2 10.1 you can define up to three standby databases. The lecture will also explain how to implement one standby database with the new time-delayed replay feature, which helps to recover from application errors that cause data loss on the primary.

We will cover various migration planning and installation considerations to help move existing DB2 LUW databases to DB2 LUW 10.1 to take advantage of the new features and functions.