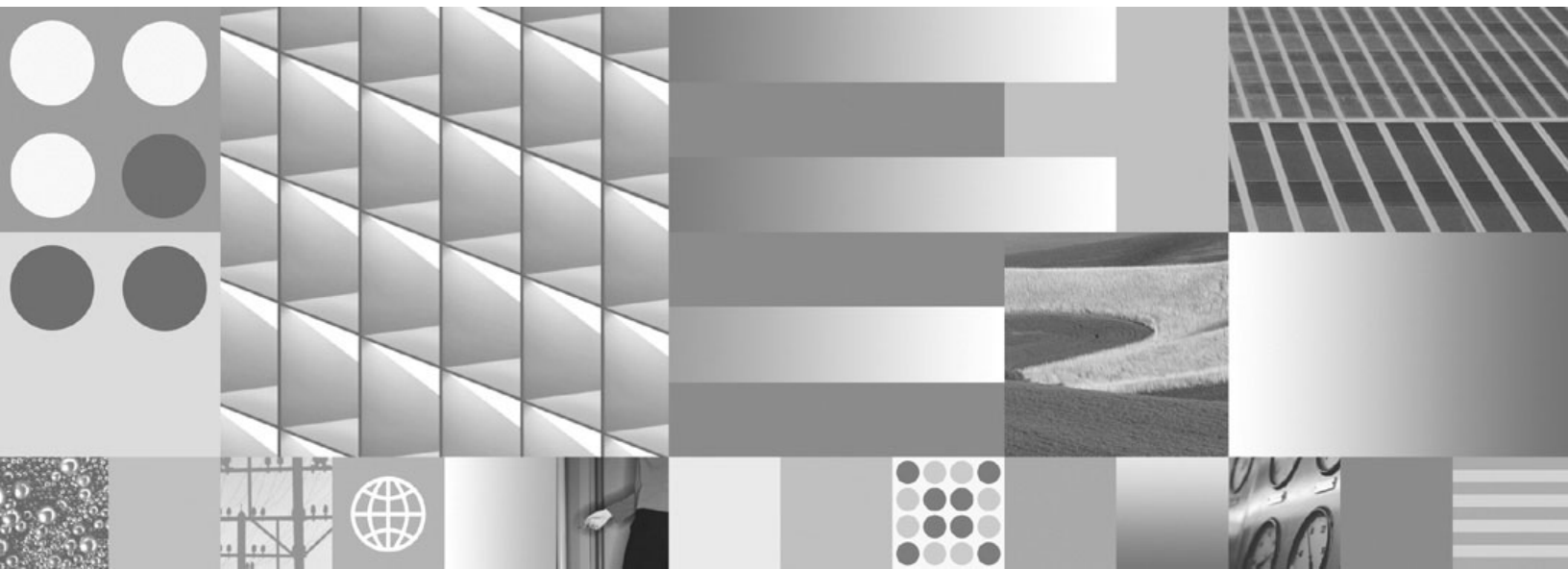


IBM DB2 9.7
for Linux, UNIX, and Windows



Version 9 Release 7



Upgrading to DB2 Version 9.7
Updated November, 2009

Note

Before using this information and the product it supports, read the general information under Appendix C, "Notices," on page 219.

Edition Notice

This document contains proprietary information of IBM. It is provided under a license agreement and is protected by copyright law. The information contained in this publication does not include any product warranties, and any statements provided in this manual should not be interpreted as such.

You can order IBM publications online or through your local IBM representative.

- To order publications online, go to the IBM Publications Center at www.ibm.com/shop/publications/order
- To find your local IBM representative, go to the IBM Directory of Worldwide Contacts at www.ibm.com/planetwide

To order DB2 publications from DB2 Marketing and Sales in the United States or Canada, call 1-800-IBM-4YOU (426-4968).

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2006, 2009.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this book v

Part 1. Upgrading your DB2 environment 1

Chapter 1. Upgrade to DB2 Version 9.7 3

Chapter 2. Planning your DB2 environment upgrade 5

Planning your DB2 servers upgrade 6

Planning your clients upgrade 8

Planning your database applications and routines upgrade 8

Part 2. Upgrading DB2 servers 13

Chapter 3. DB2 servers upgrade. 15

Chapter 4. Upgrade essentials for DB2 servers 17

DB2 command actions to upgrade instances and databases 17

Upgrade restrictions for DB2 servers 19

Best practices for upgrading DB2 servers 22

Disk space requirements for DB2 server upgrades 25

Support changes for 32-bit and 64-bit DB2 servers 27

DB2 server behavior changes 28

Deprecated or discontinued functionality that affects DB2 server upgrades 38

Migration from non-DB2 relational database management systems 40

Chapter 5. Pre-upgrade tasks for DB2 servers 43

Converting type-1 indexes to type-2 indexes 44

Verifying that your databases are ready for upgrade 46

Backing up databases before upgrade. 48

Backing up DB2 server configuration and diagnostic information 49

Increasing table space and log file sizes before upgrade 51

Changing raw devices to block devices (Linux) 53

Upgrading DB2 servers in a test environment 54

 Creating database duplicates 55

 Taking a DB2 server offline before upgrade. 56

Chapter 6. Upgrading a DB2 server (Windows) 59

Upgrading instances 60

Upgrading the DB2 Administration Server (DAS). 62

Upgrading databases 64

Chapter 7. Upgrading a DB2 server (Linux and UNIX) 69

Upgrading instances 70

Upgrading the DB2 Administration Server (DAS). 72

Upgrading databases 73

Chapter 8. Upgrading DB2 servers with specific characteristics. 77

Upgrading DB2 32-bit servers to 64-bit systems (Windows). 77

Upgrading non-root installations 79

Upgrading a DB2 server with multiple DB2 copies 80

Upgrading to a new DB2 server 82

Upgrading a DB2 server using online backups from a previous release 85

Upgrading partitioned database environments. 86

Upgrading DB2 Text Search 87

Upgrading DB2 Data Links Manager environments 90

Upgrading a DB2 server with XML Extender to DB2 Version 9.7. 91

Upgrading DB2 servers in Microsoft Cluster Server environments. 93

Chapter 9. Post-upgrade tasks for DB2 servers 95

Adjusting the log space size in upgraded databases 97

Activating a database after upgrade 98

Managing DB2 server behavior changes 98

Setting up security to manage database auditing in upgraded databases 100

Rebinding packages in upgraded databases 101

Migrating explain tables. 102

Converting XML storage objects to the Version 9.7 format. 103

Ensuring system temporary table spaces page sizes meet requirements. 103

Re-creating write-to-table event monitors 105

Verifying upgrade of DB2 servers 106

Chapter 10. Adopting new Version 9.7 functionality in upgraded databases 107

Chapter 11. Migrating DB2 functionality to DB2 product features 111

Migrating from DB2 Governor to DB2 workload manager 111

Migrating from Query Patroller to DB2 workload manager 113

Migrating from XML Extender to pureXML 116

Chapter 12. Reversing DB2 server upgrade. 117

Part 3. Upgrading clients	119
Chapter 13. Clients upgrade	121
Chapter 14. Upgrade essentials for clients	123
Best practices for upgrading clients	125
Chapter 15. Pre-upgrade tasks for clients	127
Backing up client configuration information	127
Upgrading clients in a test environment	128
Chapter 16. Upgrading to Data Server Client (Windows)	129
Chapter 17. Upgrading to Data Server Runtime Client (Windows)	131
Chapter 18. Upgrading clients (Linux and UNIX)	133
Chapter 19. Upgrading to IBM Data Server Driver Package	135
Chapter 20. Post-upgrade tasks for clients	137
Recataloging nodes and databases using TCP/IP protocol	137
Verifying your client upgrade	138
Part 4. Upgrading applications and routines	141
Chapter 21. Database applications and routines upgrade	143
Chapter 22. Upgrade essentials for database applications	145
Upgrade impact from DB2 API changes	151
Upgrade impact from DB2 command changes	155
Upgrade impact from SQL statement changes	162
Upgrade impact from system catalog changes	170
Chapter 23. Upgrade essentials for routines	177
Chapter 24. Pre-upgrade tasks for database applications and routines	181
Chapter 25. Upgrading database applications	183

Upgrading embedded SQL applications	184
Upgrading CLI applications	185
Upgrading Java applications that use IBM Data Server Driver for JDBC and SQLJ	186
Upgrading Java applications that use DB2 JDBC Type 2 driver	188
Upgrading ADO.NET applications	189
Upgrading scripts	190
Upgrading 32-bit database applications to run on 64-bit instances	191

Chapter 26. Upgrading routines	193
Upgrading C, C++, and COBOL routines	194
Upgrading Java routines	196
Upgrading .NET CLR routines	197
Upgrading SQL procedures	198
Upgrading 32-bit external routines to run on 64-bit instances	199

Chapter 27. Post-upgrade tasks for database applications and routines	201
--	------------

Chapter 28. Adopting new DB2 Version 9.7 functionality in database applications and routines	203
---	------------

Part 5. Appendixes	205
---------------------------	------------

Appendix A. Important references	207
---	------------

Appendix B. Overview of the DB2 technical information	209
DB2 technical library in hardcopy or PDF format	209
Ordering printed DB2 books	212
Displaying SQL state help from the command line processor	213
Accessing different versions of the DB2 Information Center	213
Displaying topics in your preferred language in the DB2 Information Center	213
Updating the DB2 Information Center installed on your computer or intranet server	214
Manually updating the DB2 Information Center installed on your computer or intranet server	215
DB2 tutorials	217
DB2 troubleshooting information	217
Terms and Conditions	218

Appendix C. Notices	219
----------------------------	------------

Index	223
--------------	------------

About this book

The Upgrading to DB2 Version 9.7 guide describes the upgrade process and concepts for each component of your DB2® environment. These components are DB2 servers, clients, database applications, and routines.

Who should use this book

This book is intended for database administrators, system administrators, and system operators who need to upgrade DB2 servers and clients. It is also intended for programmers and other users who need to upgrade database applications and routines.

How this book is structured

This book contains information about how to create an upgrade plan and how to upgrade each component of your DB2 environment:

- Part 1, “Upgrading your DB2 environment,” on page 1
- Part 2, “Upgrading DB2 servers,” on page 13
- Part 3, “Upgrading clients,” on page 119
- Part 4, “Upgrading applications and routines,” on page 141

Part 1. Upgrading your DB2 environment

This part of the book contains the following chapters:

- Chapter 1, “Upgrade to DB2 Version 9.7,” on page 3
- Chapter 2, “Planning your DB2 environment upgrade,” on page 5

Chapter 1. Upgrade to DB2 Version 9.7

Upgrading to a new release of DB2 database products might require upgrading your DB2 environment components if you want them to run on the new release.

Your DB2 environment has several components such as DB2 servers, DB2 clients, database applications, and routines. Upgrading these components requires an understanding of DB2 database products and their upgrade concepts. For example, if you have an existing DB2 environment with DB2 Version 9.5, DB2 Version 9.1, or DB2 UDB Version 8 copies and you want to upgrade them to DB2 Version 9.7, then you must upgrade your DB2 environment.

The upgrade process consists of all the tasks that you must perform to have your DB2 environment running successfully on a new release. 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.

The following information is provided to document the upgrade process for DB2 Version 9.7:

- Upgrade overviews define upgrade concepts and describe the upgrade process for a component.
- Upgrade essentials include the details about upgrade support, restrictions and best practices that you must know to plan your upgrade strategy.
- Pre-upgrade tasks describe all the preparation tasks that you must perform before upgrade.
- Upgrade tasks describe step by step the basic upgrade process for a component and how to upgrade DB2 environment components with special characteristics.
- Post-upgrade tasks describe all the tasks that you must perform after upgrade to have your DB2 server running at the optimum level.

In the upgrade tasks, the term *pre-Version 9.7 DB2 releases* refers to a DB2 Version 9.5, DB2 Version 9.1, and DB2 UDB Version 8 release.

Chapter 2. Planning your DB2 environment upgrade

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 following 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 Version 9.7:

1. Set up a DB2 Version 9.7 test server and create test databases.
2. Test your database applications and routines on a DB2 Version 9.7 test database to determine whether they run successfully. If your application requires a client, use a Version 9.7 client.
3. Upgrade your DB2 servers and clients in a test environment. Determine what the issues are and how to resolve them. Use this information to adjust your upgrade plan.
4. Upgrade your DB2 servers to DB2 Version 9.7 in your production environment. Ensure that they operate as expected.
5. Upgrade your clients to DB2 Version 9.7 in your production environment. Ensure that your clients operate as expected.
6. Test your database applications and routines in the DB2 Version 9.7 upgraded environment to determine whether they run as expected.
7. Make your upgraded environment available to users.
8. Identify the use of deprecated functionality that will eventually become discontinued and new functionality that can improve the functionality and performance of your applications and routines. Plan how to modify your applications and routines.
9. Modify your database applications and routines as planned. Ensure that they run successfully in DB2 Version 9.7.

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 Version 9.7. Include in your new plan any steps related to internal procedures to request access, software installation or other system services within your organization.

Review also the DB2 upgrade portal (formerly know as DB2 migration portal) that provides access to additional resources and up-to-date information about the upgrade process as they become available at <http://www.ibm.com/software/data/db2/upgrade/portal>. These resources include educational material, white papers, and webcasts for upgrade.

Finally, plan to remove the use of deprecated functionality and incorporate new functionality from DB2 Version 9.7. 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.

Planning your DB2 servers upgrade

Planning the upgrade of DB2 servers requires that you review all of the applicable upgrade prerequisites, pre-upgrade tasks, upgrade tasks and post-upgrade tasks.

Procedure

To create an upgrade plan for your DB2 servers:

1. Write the upgrade plan for DB2 servers, using all of the details that apply to your environment:

Table 1. Upgrade plan details for DB2 servers.

Upgrade plan	Details
Prerequisites	<p>Ensure that you:</p> <ul style="list-style-type: none"> • meet the installation requirements for DB2 database products described in <i>Installing DB2 Servers</i>. • resolve any support issues described in upgrade essentials for DB2 servers. • meet all prerequisites for the upgrade task and subtasks, especially obtaining root or Local Administrator access and required DB2 authorization.
Pre-upgrade tasks	<p>Include the following tasks:</p> <ul style="list-style-type: none"> • Upgrade your DB2 server in a test environment to determine any upgrade issues • Verify that databases are ready for DB2 upgrade • Back up your databases • Back up configuration information • Increase table space and log file sizes • If the diaglevel database manager configuration parameter is set to 2 or less, set this parameter to 3 or higher. • Take the DB2 server offline for DB2 upgrade <p>In addition, check the list of pre-upgrade tasks for optional tasks that you might want to perform for your environment such as upgrading your DB2 server in a test environment.</p>

Table 1. Upgrade plan details for DB2 servers. (continued)

Upgrade plan	Details
Upgrade task	<p>You must include these steps:</p> <ul style="list-style-type: none"> • Install DB2 Version 9.7 • Upgrade instances • Upgrade the DAS • Upgrade databases <p>Review the following upgrade tasks to determine the additional steps that are required to upgrade your environment:</p> <ul style="list-style-type: none"> • Upgrading a DB2 server (Windows[®]) • Upgrading a DB2 server (Linux[®] and UNIX[®]) • Upgrading environments with specific characteristics <p>Take note of the time required to upgrade your databases.</p>
Post-upgrade tasks	<p>Include the following tasks:</p> <ul style="list-style-type: none"> • Reset the <i>diaglevel</i> database manager configuration parameter to the value set before the upgrade • Adjust the log space size • Activate your database after upgrade • Manage changes in DB2 server behavior • Rebind packages in upgraded databases • Upgrade DB2 Explain tables • Update the statistics on the system catalog tables. • Verify that your DB2 server upgrade was successful • Back up your databases after the upgrade is complete <p>In addition, check the list of post-upgrade tasks for optional tasks that you might want to perform for your environment. Consider adding the following tasks to your upgrade plan:</p> <ul style="list-style-type: none"> • Tune your DB2 server after the upgrade is completed. See “Tuning database performance” in <i>Troubleshooting and Tuning Database Performance</i> • Remove the use of deprecated functionality in DB2 Version 9.7 • Adopt new functionality in upgraded databases, where appropriate, to improve performance at the DB2 server level. Review manageability, performance, and scalability enhancements in What’s New for DB2 Version 9.7 to determine what new functionality you might want to apply to your environment

2. If you must be able to reverse the upgrade, add details to the plan about the tasks required to reverse a DB2 server upgrade. These details should include any steps required in the upgrade task that enables you to reverse the upgrade.
3. Combine with the upgrade plan for other components such as clients, database applications, and routines to create an overall upgrade plan for your DB2 environment.

Planning your clients upgrade

Planning the upgrade of your clients requires that you review all of the applicable upgrade prerequisites, pre-upgrade tasks, upgrade tasks and post-upgrade tasks.

Procedure

To create an upgrade plan for your clients:

1. Write the upgrade plan for clients, using all the details that apply to your environment:

Table 2. Upgrade plan details for clients.

Upgrade plan	Details
Prerequisites	Ensure that you: <ul style="list-style-type: none">• meet the installation requirements for DB2 database products described in <i>Installing DB2 Servers</i>.• resolve any support issues in upgrade essentials for clients including client and server connectivity.• meet all prerequisites for the upgrade task and subtasks, especially obtaining root or Local Administrator access and required DB2 authorization.
Pre-upgrade tasks	Include the following tasks: <ul style="list-style-type: none">• Upgrade your DB2 servers• Back up your client configuration information In addition, check the list of pre-upgrade tasks for optional tasks that you might want to perform for your environment such as upgrading your clients in a test environment.
Upgrade task	You must include these steps: <ul style="list-style-type: none">• Install Version 9.7 client• Upgrade client instance Review the following upgrade tasks to determine the additional steps that are required to upgrade your environment: <ul style="list-style-type: none">• Upgrading to Data Server Client (Windows)• Upgrading to Data Server Runtime Client (Windows)• Upgrading clients (Linux and UNIX)
Post-upgrade tasks	Include the following tasks: <ul style="list-style-type: none">• Recatalog nodes and databases that use NetBIOS and SNA protocols• Review changes in DB2 server behavior• Verify that upgrade for clients was successful

2. Combine with the upgrade plan for other components such as DB2 servers, database applications, and routines to create an overall upgrade plan for your DB2 environment.

Planning your database applications and routines upgrade

Planning the upgrade of database applications and routines requires that you review all of the applicable pre-upgrade tasks, upgrade prerequisites, upgrade tasks, and post-upgrade tasks.

Procedure

To create an upgrade plan for your database applications and routines:

1. Write the upgrade plan for database applications, using all the details that apply to your environment:

Table 3. Upgrade plan details for database applications.

Upgrade plan	Details
Prerequisites	<p>Ensure that you:</p> <ul style="list-style-type: none"> • meet the installation requirements for DB2 database products described in <i>Installing DB2 Servers</i>. • meet the development software requirements. See “Support for elements of the database application development environment” in <i>Getting Started with Database Application Development</i> • resolve any support issues in upgrade essentials for database applications during the upgrade. • meet all prerequisites for the upgrade task and subtasks, especially obtaining required DB2 authorization.
Pre-upgrade tasks	<p>Include the following tasks:</p> <ul style="list-style-type: none"> • Upgrade your client or install Version 9.7 application driver. • Test your database applications in a DB2 Version 9.7 testing environment. If your applications run successfully, the rest of the upgrade steps are not required. <p>In addition, check the list of pre-upgrade tasks for optional tasks that you might want to perform for your environment. Even if your current operating system and development software are supported, consider including the following tasks to improve application performance:</p> <ul style="list-style-type: none"> • Upgrade your operating system to the latest supported level • Upgrade your development software to the latest supported level
Upgrade task	<p>You must include these steps:</p> <ul style="list-style-type: none"> • Modify your application code to support changes in DB2 Version 9.7 and to remove use of functionality that is discontinued in DB2 Version 9.7. • Modify your application to support changes specific to the development environment. • Rebuild all database applications after completing your modifications. • Test your database applications using DB2 Version 9.7. <p>Review the following upgrade tasks to determine the additional steps that are required by your development environment to upgrade database applications:</p> <ul style="list-style-type: none"> • “Upgrading embedded SQL applications” on page 184 • “Upgrading CLI applications” on page 185 • “Upgrading Java applications that use IBM Data Server Driver for JDBC and SQLJ” on page 186 • “Upgrading Java applications that use DB2 JDBC Type 2 driver” on page 188 • “Upgrading ADO.NET applications” on page 189 • “Upgrading scripts” on page 190 • “Upgrading 32-bit database applications to run on 64-bit instances” on page 191

Table 3. Upgrade plan details for database applications. (continued)

Upgrade plan	Details
Post-upgrade tasks	<p>Perform the recommended post-upgrade tasks for database applications, especially:</p> <ul style="list-style-type: none"> • Tune performance of your database applications. • Remove the use of deprecated functionality in DB2 Version 9.7. • Adopt new functionality introduced in DB2 Version 9.7 for application development where appropriate.

2. Write the upgrade plan for routines, using all the details that apply to your environment:

Table 4. Upgrade plan details for routines.

Upgrade plan	Details
Prerequisites	<p>Ensure that you:</p> <ul style="list-style-type: none"> • meet the development software requirements. See “Support for elements of the database application development environment” in <i>Getting Started with Database Application Development</i>. • resolve any support issues in upgrade essentials for routines during the upgrade. • meet all prerequisites for the upgrade task and subtasks, especially obtaining required DB2 authorization.
Pre-upgrade tasks	<p>Include the following task:</p> <ul style="list-style-type: none"> • Test your routines in a DB2 Version 9.7 testing environment. If your routines run successfully, the rest of the upgrade steps are not required. <p>In addition, check the list of pre-upgrade tasks for optional tasks that you might want to perform for your environment. Even if your development software is supported, consider upgrading your development software to the latest supported level.</p>
Upgrade task	<p>You must include these steps:</p> <ul style="list-style-type: none"> • Modify your routines to support changes in DB2 Version 9.7 and to remove use of functionality that is discontinued in DB2 Version 9.7. • Modify your routines to support changes specific to the development environment. • Rebuild all external routines after completing your modifications. • Retest your routines using DB2 Version 9.7. <p>Review the following upgrade tasks to determine the additional steps that are required by your development environment to upgrade routines:</p> <ul style="list-style-type: none"> • “Upgrading C, C++, and COBOL routines” on page 194 • “Upgrading Java routines” on page 196 • “Upgrading .NET CLR routines” on page 197 • “Upgrading SQL procedures” on page 198 • “Upgrading 32-bit external routines to run on 64-bit instances” on page 199

Table 4. Upgrade plan details for routines. (continued)

Upgrade plan	Details
Post-upgrade tasks	Perform the recommended post-upgrade tasks for routines, especially: <ul style="list-style-type: none"><li data-bbox="721 323 1453 352">• Remove the use of deprecated functionality in DB2 Version 9.7<li data-bbox="721 359 1453 420">• Adopt new functionality introduced in DB2 Version 9.7 for routines where appropriate

3. Combine with the upgrade plan for other components such as clients and DB2 servers to create an overall upgrade plan for your DB2 environment.

Part 2. Upgrading DB2 servers

This part of the book contains the following chapters:

- Chapter 3, “DB2 servers upgrade,” on page 15
- Chapter 4, “Upgrade essentials for DB2 servers,” on page 17
- Chapter 5, “Pre-upgrade tasks for DB2 servers,” on page 43
- Chapter 6, “Upgrading a DB2 server (Windows),” on page 59
- Chapter 7, “Upgrading a DB2 server (Linux and UNIX),” on page 69
- Chapter 8, “Upgrading DB2 servers with specific characteristics,” on page 77
- Chapter 9, “Post-upgrade tasks for DB2 servers,” on page 95
- Chapter 11, “Migrating DB2 functionality to DB2 product features,” on page 111
- Chapter 10, “Adopting new Version 9.7 functionality in upgraded databases,” on page 107
- Chapter 12, “Reversing DB2 server upgrade,” on page 117

Chapter 3. DB2 servers upgrade

Upgrading to DB2 Version 9.7 requires that you upgrade your existing DB2 servers.

Upgrading your DB2 server requires that you install a DB2 Version 9.7 copy and then upgrade all the instances and databases to be able to run them under the DB2 Version 9.7 copy.

You can directly upgrade existing DB2 Version 9.5, DB2 Version 9.1, or DB2 UDB Version 8 instances and databases to DB2 Version 9.7. Learn details, limitations about the upgrade process, and possible issues that you must be aware of in the upgrade essentials section. Refer to the upgrading DB2 server tasks for details on how to upgrade to DB2 Version 9.7. In the upgrading DB2 server topics, the term *pre-Version 9.7 DB2 copy* refers to a DB2 Version 9.5, DB2 Version 9.1, or DB2 UDB Version 8 copy.

On Windows operating systems, you have an option to automatically upgrade an existing pre-Version 9.7 DB2 copy. If you choose to upgrade your existing DB2 copy during installation, you only need to upgrade your databases after installation.

If your DB2 servers are running on a release prior to DB2 UDB Version 8, migrate them first to DB2 UDB Version 8, and then upgrade to DB2 Version 9.7. It is recommended that you migrate to the latest fix pack of DB2 UDB Version 8.2. Refer to the DB2 UDB Version 8 migration roadmap for details on how to migrate to DB2 UDB Version 8.2.

Upgrade to DB2 Version 9.7 is supported for the following DB2 products:

Table 5. DB2 database products supported for upgrade

DB2 Version	DB2 product name
Version 9.5	<ul style="list-style-type: none">• DB2 Enterprise Server Edition• DB2 Workgroup Server Edition• DB2 Personal Edition• DB2 Express™ Edition• DB2 Express-C• DB2 Connect™ Enterprise Edition• DB2 Connect Personal Edition• DB2 Connect Unlimited Edition• DB2 Connect Application Server Edition• DB2 Query Patroller• IBM® Data Server Client• IBM Data Server Runtime Client

Table 5. DB2 database products supported for upgrade (continued)

DB2 Version	DB2 product name
Version 9.1	<ul style="list-style-type: none"> • DB2 Enterprise Server Edition • DB2 Workgroup Server Edition • DB2 Personal Edition • DB2 Express Edition • DB2 Connect Enterprise Edition • DB2 Connect Personal Edition • DB2 Connect Unlimited Edition • DB2 Connect Application Server Edition • DB2 Client • DB2 Runtime Client • DB2 Query Patroller
Version 8	<ul style="list-style-type: none"> • DB2 UDB Enterprise Server Edition • DB2 UDB Workgroup Server Unlimited Edition • DB2 UDB Workgroup Server Edition • DB2 UDB Personal Edition • DB2 Express Edition • DB2 Universal Developer's Edition • DB2 Personal Developer's Edition • DB2 UDB Express Edition • DB2 Connect Unlimited Edition • DB2 Connect Enterprise Edition • DB2 Connect Application Server Edition • DB2 Connect Personal Edition • DB2 Query Patroller • DB2 Administration Client • DB2 Application Development Client • DB2 Runtime Client

For DB2 products not supported, refer to functionality deprecated or discontinued in DB2 database products that impacts upgrade.

Chapter 4. Upgrade essentials for DB2 servers

Upgrading DB2 servers to DB2 Version 9.7 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.

Consider the following factors to develop a complete understanding of upgrading DB2 servers to DB2 Version 9.7:

- “DB2 command actions to upgrade instances and databases”
- “Upgrade restrictions for DB2 servers” on page 19
- “Best practices for upgrading DB2 servers” on page 22
- “Disk space requirements for DB2 server upgrades” on page 25
- “Support changes for 32-bit and 64-bit DB2 servers” on page 27
- “DB2 server behavior changes” on page 28
- “Deprecated or discontinued functionality that affects DB2 server upgrades” on page 38
- “Migration from non-DB2 relational database management systems” on page 40

DB2 command actions to upgrade instances and databases

Learning what actions take place when you invoke the commands to upgrade instances and databases gives you a better understanding of the upgrade process for DB2 servers.

Instance upgrade

When the instance upgrade is called explicitly using the `db2iupgrade` command, or implicitly when you install DB2 Version 9.7 on Windows and select the **Work with Existing** option and then choose a pre-Version 9.7 copy with the **upgrade** action, this command:

- Upgrades an existing instance to a new instance under a DB2 Version 9.7 copy.
- Upgrades instance profile registry variables. The global profile registry variables set by the user are not upgraded.
- Upgrades the database manager configuration (`dbm cfg`) file.
- Sets the `jdk_path` `dbm cfg` parameter appropriately.
- Upgrades the `db2audit.cfg` audit configuration file when the audit facility is enabled.
- Uses the `SSLconfig.ini` SSL configuration file to set the new database manager configuration parameters to the corresponding SSL parameter value in this file and upgrades the instance profile registry setting `DB2COMM=SSL`.
- In a Microsoft® Cluster Server (MSCS) environment, defines a new resource type, updates all DB2 MSCS resources to use the new resource type, removes the old resource type, and brings all resources online.

For a successful instance upgrade, it is essential that all files for all instances exist and that write access is granted. However, review upgrade restrictions for particular scenarios that are not supported.

Database directory upgrade

When you access the database directory the first time, it is implicitly upgraded if necessary. The database directory is accessed when you issue commands such as LIST DATABASE DIRECTORY or UPGRADE DATABASE command.

Database upgrade

When the database upgrade is called explicitly using the UPGRADE DATABASE command, or implicitly using the RESTORE DATABASE command from a pre-Version 9.7 database backup, the following database entities might be 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, the UPGRADE DATABASE command renames all log files in the active log path with the extension .MIG. After you upgrade your databases successfully, you can delete all the S*.MIG files. Refer to Chapter 9, “Post-upgrade tasks for DB2 servers,” on page 95 for details.

The UPGRADE DATABASE command upgrades the 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 a database and rebalancing automatic storage table spaces after you add or drop storage paths from a database.

The UPGRADE DATABASE command automatically collects statistics for all system catalog tables during database upgrade. The following table shows the RUNSTATS command called for the automatic collection of statistics:

Table 6. RUNSTATS command for automatic statistics collection

auto_runstats	User profile	RUNSTATS command
Enabled	Exists	RUNSTATS command with the SET PROFILE parameter using the information in the STATISTICS_PROFILE column in SYSCAT.TABLES. ¹
Enabled	Does not exist	RUNSTATS command with default parameters
Disabled	N/A	RUNSTATS command from the packed descriptor for the most recent call to the RUNSTATS command. ²

Note:

1. If you are upgrading databases from DB2 UDB Version 8 FixPak 8 or lower level (Version 8.2 FixPak 1 or lower level), the automatic collection of statistics does not occur. You have to manually collect statistic after upgrading your databases.
2. If statistics were previously collected for the table, the RUNSTATS command is issued as indicated in the table. If there are no statistics collected for the table, the RUNSTATS command is not issued.

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.

Upgrade restrictions for DB2 servers

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

What is supported?

- Upgrading to DB2 Version 9.7 is supported from DB2 Version 9.5, DB2 Version 9.1, and DB2 UDB Version 8. If you have DB2 UDB Version 7 or earlier, you must upgrade to DB2 UDB Version 8.2 before upgrading to DB2 Version 9.7.
- Upgrading to a DB2 Version 9.7 non-root installation is supported from a DB2 Version 9.5 non-root installation. Upgrading to a DB2 Version 9.7 non-root installation from a pre-Version 9.7 root installation is not supported.
- On Windows operating systems, the **upgrade** action shows for existing DB2 copies that can be upgraded during the installation of DB2 Version 9.7. This action automatically installs DB2 Version 9.7 and upgrades all of your instances and your DB2 Administration Server (DAS) running on the DB2 copy. This action also uninstalls the DB2 copy and any add-on products installed in this copy. If you do not choose the **upgrade** action, you must manually upgrade your instances and your DAS after installation.
- On Linux and UNIX operating systems, the **upgrade** action is not available and you can only install a new copy of DB2 Version 9.7. You have to manually upgrade your instances after installation. You can manually upgrade your existing DAS.
- Instance bit size is determined by the operating system where DB2 Version 9.7 is installed, and support for 32-bit kernels and 64-bit kernels has changed.
- Upgrading from a system with multiple DB2 copies of DB2 Version 9.5, DB2 Version 9.1, DB2 UDB Version 8, or all levels is supported. On Windows operating systems, you must be aware of the restrictions on coexistence of previous versions of the DB2 database products. Refer to “Updating DB2 copies (Windows)” in *Database Administration Concepts and Configuration Reference*.
- Upgrading from a partitioned database environment with multiple database partitions is supported.
- Restoring full database offline backups from pre-Version 9.7 DB2 copies is supported. However, rolling forward of logs from a previous level is not possible. Review “Backup and restore operations between different operating systems and hardware platforms” in *Data Recovery and High Availability Guide and Reference* for complete details about upgrade support using the RESTORE DATABASE command.
- When the DB2 Workload Manager feature is installed and you are upgrading from DB2 Version 9.1 or DB2 UDB Version 8, an upgraded database has three predefined default service classes: the default user class, the default maintenance class, and the default system class. After database upgrade, all connections belong to the default workload. Connections that belong to the default workload are mapped to the

default user class. The default user class has only one service subclass: the default subclass. All activities from the connections in the default user class run in the default subclass.

- In upgraded databases with the **RESTRICT_ACCESS** database configuration parameter set to YES, you must grant the USAGE privilege to non-DBADM users on SYSDEFAULTUSERWORKLOAD. Otherwise, these users are unable to submit any work to the database.
- Index extensions are upgraded as part of the database upgrade. However, you might have to re-create your indexes if you upgraded from a DB2 UDB Version 8 32-bit instance to a DB2 Version 9.7 64-bit instance. Review the post-upgrade tasks for details.

What is unsupported?

DB2 Version 9.7 installation fails if the following situations exist:

- The operating system is not supported. You must upgrade to a supported version of the operating system before you upgrade to DB2 Version 9.7 or upgrade to a new DB2 server that meets the operating system requirements. See “Installation requirements for DB2 database products” in *Installing DB2 Servers*.
- A 32-bit kernel is running on Linux and UNIX operating systems except for Linux on x86. A 64-bit kernel must be installed prior to installing DB2 Version 9.7.
- A copy of DB2 UDB Version 7 or earlier is installed on Windows operating systems.

The db2iupgrade command fails if the following situations exist:

- You do not have authorization to upgrade the instance.
- The instance that you are trying to upgrade is active. Run the db2stop command to stop the instance.
- The instance is already at DB2 Version 9.7 or later. Run the db2iupdt command to update to a different fix pack levels or copies of DB2 Version 9.7.
- You try to upgrade from DB2 Version 9.7 back to DB2 Version 9.5, DB2 Version 9.1, or DB2 UDB Version 8. Chapter 12, “Reversing DB2 server upgrade,” on page 117 is possible, however, you must follow the prerequisites and steps in this procedure.
- The type of instance that you are trying to upgrade to the DB2 Version 9.7 copy is unsupported. The following table describes the upgrade support for each type of instance by DB2 database product:

Table 7. Instance upgrade support for DB2 Version 9.7 database products

Instance type	Node type	Upgrade support
client – default type for DB2 clients ¹	Client	<ul style="list-style-type: none"> • Upgrade to a client, a <i>standalone</i>, a <i>wse</i>, or an <i>ese</i> instance is supported.
standalone – default type for DB2 Personal Edition (PE)	Database server with local clients	<ul style="list-style-type: none"> • Upgrade to a <i>standalone</i>, a <i>wse</i>, or an <i>ese</i> instance is supported. • Upgrade to a <i>client</i> instance is unsupported.

Table 7. Instance upgrade support for DB2 Version 9.7 database products (continued)

Instance type	Node type	Upgrade support
wse – default type for DB2 Workgroup Server Edition (WSE)	Database server with local and remote clients	<ul style="list-style-type: none"> • Upgrade to a <i>wse</i> or an <i>ese</i> instance is supported. • Upgrade to a <i>standalone</i> instance creates a <i>standalone</i> instance² (Linux and UNIX only) • Upgrade to a <i>client</i> instance is unsupported.
ese – default type for DB2 Enterprise Server Edition (ESE)	Partitioned database server with local and remote clients or Enterprise Server Edition with local and remote clients	<ul style="list-style-type: none"> • Upgrade to an <i>ese</i> instance is supported. • Upgrade to a <i>standalone</i> or a <i>wse</i> instance from single database partition environments creates a <i>standalone</i> or <i>wse</i> instance²(Linux and UNIX only) • Upgrade to a <i>client</i> instance is unsupported.

Note:

1. The highest level for each DB2 database product is the default instance type as indicated in Table 7 on page 20 ordered from lower to higher-level. Each instance type supports instance types of a lower-level. For example, the *ese* instance type supports *wse*, *standalone*, and *client*. You can use the `db2icrt` command with the `-s` parameter to create instances of a lower-level. If you do not specify the `-s` parameter, the instance is created using the highest level of instance type supported by the DB2 database product installed.
2. Database manager configuration parameters have default values for the created instance. Previous database manager configuration settings are not retained. If the configuration parameters are available in the new instance, after upgrade, you can restore previous settings. Avoid upgrading from a higher-level instance type to a lower-level instance type if possible.
 - The `db2ckupgrade` command fails and causes the `db2iupgrade` command to fail. The `db2iupgrade` command calls the `db2ckupgrade` command to verify whether cataloged local databases are ready for upgrade to DB2 Version 9.7.
 - DB2 Data Links Manager Version 8 is installed on the DB2 server. DB2 Data Links Manager is unsupported in DB2 Version 9.7. You can upgrade to a standard DB2 Version 9.7 instance without the DB2 Data Links Manager functionality.
 - DB2 Data Warehouse Manager Version 8 and any extensions are installed on the DB2 server. DB2 Data Warehouse Manager is unsupported in DB2 Version 9.7. However, when you run the `db2iupgrade` command, the error message that is generated includes instructions on how to upgrade to a standard DB2 Version 9.7 instance without the DB2 Data Warehouse Manager functionality.

The `UPGRADE DATABASE` command fails if the following situations exist:

- You do not have authorization to upgrade the database.
- A cataloged database does not exist.
- Database upgrade encounters any of the problems described in the reason codes of error message “SQL1704N” in *Message Reference Volume 2*.

- User-defined distinct types (UDTs) are encountered with the names ARRAY, BINARY, CURSOR, DECFLOAT, ROW, VARBINARY, or XML. You must drop these UDTs and re-create them with different names before database upgrade.
- Database objects were created using restricted schema names described in the error message “SQL0553N” in *Message Reference Volume 2*. The list of restricted schema names now includes SYSPUBLIC.
- User objects use the system-defined DATALINK data type. You must drop or alter these objects before database upgrade. Also, if you installed DB2 Net Search Extender (NSE) on your DB2 server, you must drop the UDFs created by NSE for Data Links support. Refer to “Upgrading DB2 Data Links Manager environments” on page 90 for details.
- A database enabled as a high availability disaster recovery (HADR) standby database.

Best practices for upgrading DB2 servers

Consider the following best practices when planning your DB2 server upgrade.

Review changes in existing DB2 database product functionality

Changes in existing functionality introduced in DB2 Version 9.7 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-Version 9.7 releases can also have an impact. Review these changes and plan how to address these changes before the upgrade:

- For upgrade from DB2 Version 9.1 or DB2 UDB Version 8, changed functionality in DB2 Version 9.5
- For upgrade from DB2 UDB Version 8, changes in existing functionality introduced in DB2 Version 9.1

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

Perform hardware and operating system upgrades prior to DB2 database product upgrade

The support for UNIX, Linux and Windows operating systems has changed in DB2 Version 9.7. Review “Installation requirements for DB2 database products” in *Installing DB2 Servers* to determine whether your operating system version is supported and if you need to upgrade your operating system before installing DB2 Version 9.7. Note that newer versions of operating systems can also bring new hardware requirements.

Even when you are not required but decide to upgrade, 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 prior to a DB2 database product upgrade, ensure that your system is operating as expected before attempting the upgrade process.

If you have a DB2 Version 9.1 copy on Windows XP or Windows 2003, first apply a fix pack that supports Windows Vista before you upgrade the operating system to Windows Vista to ensure that your DB2 copy performs as expected after the operating system upgrade. The support for Windows Vista starts from DB2 Version 9.1 Fix Pack 2. If you have a DB2 UDB

Version 8 copy on Windows XP or Windows 2003, first upgrade to DB2 Version 9.7 and then upgrade the operating system to Windows Vista.

If you have a DB2 UDB Version 8.1 32-bit copy on Linux on POWER®, update your current DB2 copy to DB2 UDB Version 8.1 FixPak 7 or higher and then upgrade your operating system to SUSE Linux Enterprise Server (SLES) 10 before installing DB2 Version 9.7.

If you have a DB2 version 9.5 or DB2 Version 9.1 copy on SLES 10, first apply Version 9.5 Fix Pack 4 or later, or Version 9.1 Fix Pack 7 or later before you upgrade the operating system to SLES 11. If you have a DB2 UDB Version 8 copy on SLES 10, first upgrade to DB2 Version 9.7 and then upgrade the operating system to SLES 11.

If you are upgrading a pre-Version 9.7 copy on POWER3™ processor-based systems, first upgrade to POWER4™ processor-based systems before upgrading to DB2 Version 9.7. POWER3 processor-based systems are not supported in DB2 Version 9.7.

Benchmark DB2 server performance

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 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.

Devise a plan to reverse an upgrade

There is no utility to reverse an upgrade or falling back from DB2 Version 9.7 to a pre-Version 9.7 release. See Chapter 12, “Reversing DB2 server upgrade,” on page 117 to learn all the required steps to reverse a database upgrade.

Perform pre-upgrade tasks

There are several pre-upgrade tasks that you should execute for a successful upgrade, such as backing up DB2 configuration parameters settings, increasing table spaces and log files, and verifying that databases are ready for upgrade.

To avoid performance degradation after the upgrade, perform pre-upgrade tasks such as converting type-1 indexes to type-2 indexes. If you do not convert your type-1 indexes before the database upgrade, the type-1 indexes are marked invalid during the database upgrade, and they will be rebuilt on your first access to the table. You cannot access the table until the index rebuild is completed.

Upgrade 32-bit Linux operating systems to 64-bit

If you are upgrading to DB2 Version 9.7 32-bit database product on Linux operating systems, the multithreaded architecture brings new restrictions due to the 32-bit virtual memory address limit such as:

- Agent private memory for all agent threads is now allocated within a single process. The process memory space might not be large enough to allocate the aggregate of all private memory for all agents. You might need to reduce the number of agents configured.
- Support for multiple databases is limited because all database shared memory segments for all databases are allocated in a single process

memory space. You can reduce the memory usage for each database so that you can activate all databases successfully. However, the database server performance is impacted.

Consider upgrading to DB2 Version 9.7 64-bit database product instead, to avoid running into any of the 32-bit kernel limitations.

Upgrade DB2 servers first

Upgrading your DB2 servers before upgrading your data server clients is the traditional approach to avoid any known restrictions and limitations such as support of new DB2 database product functionality, network protocols, and connectivity. These restrictions and limitations are not associated with DB2 Connect.

Upgrading your data server clients first requires that you manage any incompatibilities between releases. If you must upgrade your client due to a software requirement, make sure that the software supports the DB2 database product version that you are running on your DB2 server. In this case, the software will manage any incompatibilities between releases. See “Best practices for upgrading clients” on page 125 for details.

Upgrade database applications and routines

If you upgrade your DB2 server, you might also need to upgrade your database applications and routines to support changes for 64-bit instances, SQL stored procedures, Java™ Virtual Machine (JVM), and development software.

Chapter 22, “Upgrade essentials for database applications,” on page 145 and Chapter 23, “Upgrade essentials for routines,” on page 177 describe the factors that can impact your database application upgrade or routine upgrade. Review these factors and make any necessary changes to your database applications and routines to ensure that they run after the upgrade to DB2 Version 9.7.

In an upgrade testing environment, you can test and verify that your database applications and routines run successfully in DB2 Version 9.7 to find out if you need to upgrade them. You can also upgrade your database applications and routines before you upgrade your production environment.

Upgrading DB2 High Availability Disaster Recovery (HADR) environments

Upgrading a primary database to DB2 Version 9.7 changes the database role from primary to standard. Upgrading standby databases to DB2 Version 9.7 is not supported because these databases are in roll forward pending state. Because of these restrictions, upgrading an HADR environment to DB2 Version 9.7 requires that you stop HADR, upgrade your DB2 server where the primary database resides, and then re-initialize HADR.

The following list includes each of these actions and the topic where is documented:

- Stop the HADR primary or standby databases as in indicated in the Chapter 5, “Pre-upgrade tasks for DB2 servers,” on page 43.
- Upgrade the DB2 server where the primary database resides using one of the following tasks:
 - Upgrading DB2 servers to DB2 Version 9.7 (Windows)
 - Upgrading DB2 servers to DB2 Version 9.7 (Linux and UNIX)

- Re-initialize HADR as indicated in the Chapter 9, “Post-upgrade tasks for DB2 servers,” on page 95

Migrating SQL replication environments

After upgrading your database servers, you can optionally migrate your SQL replication environment to DB2 Version 9.7.

See Migrating to SQL replication Version 9 for details about when to migrate and how to migrate your SQL replication environment.

Upgrading DB2 Spatial Extender

If you had DB2 Spatial Extender installed and you upgraded your spatially-enabled databases to DB2 Version 9.7, see *Upgrading to DB2 Spatial Extender Version 9.7* in *Spatial Extender and Geodetic Data Management Feature User's Guide and Reference* for upgrade details specific to DB2 Spatial Extender.

Upgrading Microsoft Cluster Server environments

In a Microsoft Cluster Server (MSCS) environment, you should install DB2 Version 9.7 as a new copy and then run the db2iupgrade command to upgrade the MSCS instance. See “Upgrading DB2 servers in Microsoft Cluster Server environments” on page 93 for details.

Autonomic computing functionality

If you are upgrading from DB2 Version 9.1 or earlier, DB2 Version 9.7 enables additional autonomic computing functionality introduced in Version 9.5 such as automatic agent configuration and real-time statistics. However, when you upgrade your database to DB2 Version 9.7, agent configuration is not automatic and real-time statistics are not enabled. You should consider adopting this autonomic functionality introduced in DB2 Version 9.5 to gain performance and manageability improvements.

If you are upgrading from DB2 UDB Version 8, DB2 Version 9.7 enables additional autonomic computing functionality introduced in Version 9.1 when you create a database:

- Automatic execution of the configuration advisor.
- Enablement of automatic storage.
- Enablement of the **auto_runstats** and **self_tuning_mem** database configuration parameters.

However, this autonomic computing functionality is not enabled when you upgrade your databases to DB2 Version 9.7. You should consider adopting this autonomic functionality introduced in DB2 Version 9.1 in your upgraded databases.

Disk space requirements for DB2 server upgrades

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 following disk space recommendations are applicable for upgrading to DB2 Version 9.7.

Table space information files

The files SQLSPCS.1 and SQLSPCS.2 contain table space information. During upgrade from DB2 UDB Version 8 to DB2 Version 9.7, these files grow to four times their previous size but the total data size on disk does not

exceed the new size of the SQLSPCS.1 and SQLSPCS.2 files. For example, if these two files have a total size of 512 KB before upgrade, you will need at least 2 MB of free disk space.

System catalog and system temporary table spaces

Ensure that you have sufficient free space on the system catalog and the system temporary table spaces for the databases that you are upgrading. System catalog table space is required for both old and new database catalogs during upgrade. The amount of free space required varies, depending on the complexity of the database, as well as on the number and size of database objects.

System catalog table space (SYSCATSPACE)

Increasing the total size to twice the total of used space is recommended. In other words the amount of free space should be at least the same as the current amount of used space.

Temporary table space (TEMPSPACE1 is the default name)

Increasing the total size to twice the total size of the system catalog table space is recommended.

For the system catalog table space, free pages should be equal to or greater than used pages. Total pages for the system temporary table space should be twice the amount of total pages for the system catalog table space.

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.

Log file space

The database upgrade process makes changes to system catalog objects. All changes to each system catalog object are performed in a single transaction and need adequate log space to contain this transaction. If there is insufficient log space, this transaction is rolled back and upgrade does not complete successfully.

To ensure sufficient log file space is available, you can set the **logsecond** database configuration parameter to twice the current value of **logprimary** and **logsecond** if the file system containing the log files has enough disk free space to increase this parameter. If you already have available a large log file space, it might not be necessary to increase this parameter. Also on partitioned database environments, you only need to increase the log space in the catalog partition.

You must update these database configuration parameters values before you upgrade the instance to DB2 Version 9.7, because you will not be able to update these database configuration parameters until you 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.

Index space

Each index on every populated table requires one additional page per index to use the following functionality:

- Real-time statistics.
- Deferred cleanup roll out for MDC tables.
- Index rebuild on a populated table.

If you have type-1 indexes, they will be marked invalid and recreated as type-2 indexes at restart or at first access after upgrade. Ensure that you have enough disk space to convert your type-1 indexes. Type-2 indexes require one more byte per row. Refer to Space requirements for indexes for details on how to estimate the space needed.

If you have a limited amount of free disk space for indexes, you can get the 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.

Automatic storage files

If you enable automatic storage on an existing database by issuing the ALTER DATABASE statement with the ADD STORAGE ON clause, this statement creates the SQLSGF.1 and SQLSGF.2 files that are required for maintaining automatic storage.

Support changes for 32-bit and 64-bit DB2 servers

DB2 Version 9.1 or later provides support for 32-bit operating systems on Linux on x86 and Windows operating systems, and 64-bit operating systems on UNIX, Linux and Windows operating systems.

Check “Installation requirements for DB2 database products” in *Installing DB2 Servers* for details about supported architectures on each operating system.

You cannot specify the bit size for the instance when you create or upgrade an instance. The bit size for new instances is determined by the operating system where DB2 Version 9.7 is installed. The following table summarizes the DB2 Version 9.7 bit size support that is available for each of the following operating systems:

Table 8. DB2 Version 9.7 32-bit and 64-bit support available per operating system

Operating systems	DB2 Version 9.7 support available
<ul style="list-style-type: none">• 32-bit Windows on x86 and x64 (Using DB2 Version 9.7 32-bit product)• 32-bit Linux on x86	<ul style="list-style-type: none">• 32-bit instances only• 32-bit DB2 server, client, and GUI tools packages• 32-bit IBM Software Development Kit (SDK) for Java

Table 8. DB2 Version 9.7 32-bit and 64-bit support available per operating system (continued)

Operating systems	DB2 Version 9.7 support available
<ul style="list-style-type: none"> • 64-bit kernels of AIX®, HP-UX, or Solaris • 64-bit Windows on x64 • 64-bit Linux kernel on x64, POWER, and zSeries® 	<ul style="list-style-type: none"> • 64-bit instances • 32-bit and 64-bit DB2 libraries available • 64-bit DB2 server and client • 64-bit applications and routines • 32-bit client side application support • 32-bit fenced stored procedures/UDFs only (non-Java) • Java fenced Stored Procedures/UDFs • 64-bit IBM SDK for Java

The changes in 32-bit and 64-bit support can have an impact in your applications depending on the shared library path that you indicated when you linked the DB2 libraries to your applications. If you specified the DB2 installation path, the applications fail to run because the DB2 Version 9.7 copy has a different installation path. However, if you linked the libraries using the library path under the instance home directory, your applications will run successfully in the following cases:

- If you have 32-bit instances and you upgrade to DB2 Version 9.7 on a 32-bit system. You can only upgrade to 32-bit instances on 32-bit Windows or 32-bit Linux on x86.
- If you have 64-bit instances and you upgrade to DB2 Version 9.7 on a 64-bit system. You can only upgrade to a 64-bit instance on a 64-bit system.

If you have 32-bit instances and you upgrade to DB2 Version 9.7 on a 64-bit system, you must manage incompatibilities so that your applications and routines can run successfully. Incompatibilities arise because of discontinued functionality or incorrect shared library path specification. Table 8 on page 27 summarizes the details on the available 32-bit and 64-bit support. For example, 32-bit unfenced stored procedures in any supported language except Java are not supported. By dropping and recreating these stored procedures as fenced you can resolve this issue.

DB2 server behavior changes

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.

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.

Changes that impact all pre-Version 9.7 releases

The following tables describe in detail the upgrade impact of all of the changes to variables, database and database manager configuration parameters, physical design characteristics of databases, and database authorities and privileges:

- New registry variables
- Changes to existing registry variables
- Deprecated and discontinued variables
- New database manager configuration parameters
- Changes to existing database manager configuration parameters
- Deprecated and discontinued database manager configuration parameters
- New database configuration parameters
- Changes to existing database configuration parameters
- Deprecated and discontinued database configuration parameters
- Changes to physical design characteristics of databases
- Changes to authorities and privileges

New registry variables

The following table describes the upgrade impact of the default values of new registry variables:

Table 9. New registry variables

Name	Upgrade impact
DB2_DDL_SOFT_INVALID	<p>By default, this new registry variable is set to ON to enable soft invalidation support on certain DDL statements. See “Automatic invalidation and revalidation of database objects” in <i>Database Administration Concepts and Configuration Reference</i> for a list of the DDL statements.</p> <p>In Version 9.1 Fix Pack 2 or later, if the DB2_WORKLOAD registry variable was set to SAP, soft invalidation support was provided for certain types of views that had no object dependencies.</p> <p>After instance upgrade, if you need to disable soft invalidation, set DB2_DDL_SOFT_INVALID to OFF. However, if the DB2_WORKLOAD registry variable is set to SAP, soft invalidation is still supported for certain types of views regardless of the DB2_DDL_SOFT_INVALID setting.</p>
DB2_FORCE_OFFLINE_ADD_PARTITION	<p>This new registry variable indicates whether the add partition operation using the START DATABASE MANAGER command with the ADD DBPARTITIONNUM parameter is performed online or offline. The default value (FALSE) indicates online mode.</p> <p>After instance upgrade, if you need to enforce the same behavior as in previous releases, set this registry variable to TRUE. This setting allows you to add partitions only in offline mode when you start the instance.</p>
DB2_DEFERRED_PREPARE_SEMANTICS	<p>This new registry variable allows dynamic statements containing untyped parameter markers to use <i>deferred prepare</i> semantics. By default, this variable is set to YES so that any untyped parameter markers derive their data types and length attributes based on the input descriptor from subsequent OPEN or EXECUTE statements. In previous releases, compilation of such dynamic statements would have failed.</p>

Table 9. New registry variables (continued)

Name	Upgrade impact
DB2_LIMIT_FENCED_GROUP	<p>By default, this registry variable is set to OFF to maintain the same behavior as in previous releases so that there is no upgrade impact. However, you should consider setting this registry variable to ON after the upgrade to improve the security for external routines. See “Restricting operating system permission for db2fmp process (Windows)” in <i>Installing DB2 Servers</i> for details.</p> <p>On Windows operating systems, ensure that the DB2 service account for new and existing instances belongs to the DBUSERS group.</p>
DB2_PMAP_COMPATIBILITY	<p>By default, this registry variable is set to ON so that the distribution map maintains its size of 4 096 entries. If you set this registry variable to OFF before upgrading a database, the distribution map size is increased to 32 768 entries and you need to use the new db2GetDistMap and db2GetRowPartNum APIs. See “Upgrade impact from DB2 API changes” on page 151 for details.</p>

Changes to existing registry variables

The following table describes the upgrade impact of changes to existing registry variables:

Table 10. Changes to existing registry variables

Name	Upgrade impact
DB2_EVALUNCOMMITTED and DB2_SKIPDELETED	<p>For statements operating under cursor stability isolation level with currently committed behavior enabled using the cur_commit database configuration parameter, these registry variables are in effect only when currently committed cannot be applied to a scan. Otherwise, the evaluation of predicates is performed on data retrieved by currently committed scans.</p> <p>If currently committed behavior was enabled using the BIND command or the PREPARE statement, these registry variables have no effect. See the row for the cur_commit configuration parameter in Table 13 on page 33 for details.</p>
DB2_GRP_LOOKUP	<p>The setting for DB2_GRP_LOOKUP is not modified by instance upgrade. Due to the changes to the security model in Version 9.7, if this registry variable is not set, ensure that domain users are granted the database authorities and privileges that they required after the upgrade. See “Windows platform security considerations for users” in <i>Database Security Guide</i> for details about recommended settings for the DB2_GRP_LOOKUP registry variable and how to verify the authorities and privileges are granted to domain users.</p>
DB2_LOGGER_NON_BUFFERED_IO	<p>Starting with Version 9.7, the default value for this variable is AUTOMATIC, which means that log files in the active log path might be accessed using non-buffered I/O. The database manager determines which log files benefit from using non-buffered I/O. In Version 9.5 Fix Pack 1 or later, the default was OFF and log files were accessed using only buffered I/O.</p> <p>After upgrading your instance, leaving this variable unset, it has the same effect as the AUTOMATIC setting and you might experience this change in I/O access. The benefits of using non-buffered I/O are reduced memory requirements and more efficient I/O access for log files. Therefore, carefully consider the impact before you decide to disable this feature by setting this variable to OFF.</p>
DB2_SKIPINSERTED	<p>For statements operating under cursor stability isolation level with currently committed behavior enabled, this registry variable has no effect. Read the row for the cur_commit configuration parameter in Table 13 on page 33 for details.</p>

Deprecated and discontinued 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. See “Deprecated registry variables” in *What’s New for DB2 Version 9.7* to determine the upgrade impact of deprecated registry variables.

If you are upgrading from DB2 Version 9.1 or earlier, consider removing deprecated registry variables in pre-Version 9.7 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-Version 9.7 releases as they do not have the intended effect. See “Changes that impact Version 9.1 or earlier releases” on page 38 for details.

New database manager configuration parameters

The following table describes the upgrade impact of the default values of new database manager configuration parameters:

Table 11. New database manager configuration parameters

Name	Upgrade impact
alternate_auth_enc	<p>This parameter enables AES 256-bit encryption of user IDs and passwords. By default, this parameter is not set which means that the server accepts the encryption algorithm the client proposes so that there is no upgrade impact.</p> <p>During instance upgrade, if the DB2_SERVER_ENCALG registry variable is set, the alternate_auth_enc configuration parameter is set to AES_ONLY or AES_CMP depending on the setting of DB2_SERVER_ENCALG so that your pre-upgrade setting is preserved. After the upgrade, if you want to change how AES 256-bit encryption is used, update the setting of the alternate_auth_enc configuration parameter because the setting of DB2_SERVER_ENCALG is ignored.</p>
diagsize	<p>This parameter enables the DB2 diagnostic rotating logs functionality. During instance upgrade, the diagsize is set to 0 to maintain the same behavior as in previous releases. With this setting, one single diagnostic log file (db2diag.log) and one administration notification log (<i>instance_name.nfy</i>) are used for error and notification logging and these two files grow indefinitely. See Chapter 10, “Adopting new Version 9.7 functionality in upgraded databases,” on page 107 for details on enabling rotating logs functionality.</p>
ssl_svr_keydb, ssl_svr_stash, ssl_svr_label, ssl_svcename	<p>To configure SSL support in a DB2 instance in previous releases, you set values for SSL parameters in the SSLconfig.ini file. If this file exists when you upgrade your instances, these new database manager parameters are set to the corresponding SSL parameter value in SSLconfig.ini. If this file does not exist, these database manager parameters are set to null value (default) which means that the instance is started without SSL protocol support. See “SSL client support expanded and configuration simplified” in <i>What’s New for DB2 Version 9.7</i> for more details.</p>

Table 11. New database manager configuration parameters (continued)

Name	Upgrade impact
ssl_clnt_keydb, ssl_clnt_stash	To configure Secure Sockets Layer (SSL) support in a client in previous releases, you used to set values for SSL parameters in the <code>SSLClientconfig.ini</code> file. If this file exists when you upgrade your instances, these new database manager parameters are set to the corresponding SSL parameter value in <code>SSLClientconfig.ini</code> . If this file does not exist, these database manager parameters are set to null value (default). You must set these database manager parameters to enable SSL protocol support in a client.

Changes to existing database manager configuration parameters

The following table describes the upgrade impact of changes to database manager configuration parameters:

Table 12. Changes to existing database manager configuration parameters

Name	Upgrade impact
authentication, srvcon_auth	If you enabled 256-bit AES encryption for user IDs and passwords, check the alternate_auth_enc parameter which allows you to specify an alternate encryption algorithm for user names and passwords. See the row for the alternate_auth_enc parameter in Table 11 on page 31 for more details.
instance_memory	Starting with Version 9.7 Fix Pack1, the AUTOMATIC setting does not enforce a limit on memory allocated across the instance for DB2 database products without memory usage restrictions. If you are upgrading from Version 8, Version 9.1, or Version 9.5 Fix Pack 4 or earlier and you want to enforce a limit, set instance_memory to a specific value . See “instance_memory - Instance memory configuration parameter” in <i>Database Administration Concepts and Configuration Reference</i> for details.

Deprecated and discontinued database manager configuration parameters

No database manager configuration parameters have been deprecated or discontinued in this release. However, if you are upgrading from DB2 Version 9.1 or earlier, consider removing deprecated database manager configuration parameters in pre-Version 9.7 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-Version 9.7 releases as they do not have the intended effect. See “Changes that impact Version 9.1 or earlier releases” on page 38 for details.

New database configuration parameters

The following table describes the upgrade impact of the default values of new database configuration parameters:

Table 13. New database configuration parameters

Name	Upgrade impact
auto_reval	<p>During database upgrade, this configuration parameter is set to DISABLED to maintain the same invalidation and revalidation behavior for database objects as in previous releases so that there is no upgrade impact.</p> <p>If you create new databases in DB2 Version 9.7, the auto_reval configuration parameter is set to DEFERRED by default so that revalidation deferred semantics are enabled. This setting impacts certain DDL statements and allows you to create views, functions, stored procedures, triggers, and global variables even if they reference objects that do not exist or are invalid. See “Automatic invalidation and revalidation of database objects” in <i>Database Administration Concepts and Configuration Reference</i> for details about these new semantics.</p>
cur_commit	<p>During database upgrade, this configuration parameter is set to DISABLED to maintain the same behavior as in previous releases so that there is no upgrade impact. If you want to enable currently committed behavior on cursor stability scans, you need to set the cur_commit configuration parameter to ON after the upgrade. See Chapter 28, “Adopting new DB2 Version 9.7 functionality in database applications and routines,” on page 203 for details on enabling currently committed behavior.</p> <p>For new databases created in DB2 Version 9.7, the cur_commit configuration parameter is set to ON so that currently committed behavior is enabled on cursor stability scans. See “Currently committed semantics allow for more concurrency” in <i>What’s New for DB2 Version 9.7</i> for details.</p>
dec_to_char_fmt	<p>This configuration parameter controls the character string returned by the CHAR(<i>decimal-expression</i>) scalar function and the CAST specification from decimal to character. If dec_to_char_fmt is set to NEW, the CHAR function returns a fixed-length character string representation of a decimal number without leading zeros and without a decimal separator when the decimal part is zero. If dec_to_char_fmt is set to V95, the character string included leading zeros and a decimal separator when the decimal part was zero. During database upgrade, this configuration parameter is set to V95 so that the function returns the same character string format as in previous releases.</p> <p>For new databases created in DB2 Version 9.7, dec_to_char_fmt is set to NEW. You need to set this parameter to V95 for compatibility with existing applications.</p>
mon_deadlock, mon_locktimeout, mon_lockwait, mon_lw_thresh, mon_act_metrics, mon_obj_metrics, mon_req_metrics, mon_uow_data	<p>These parameters allow you to control the collection of metrics and event monitor data at the database level including the new lock event monitor. During database upgrade, mon_deadlock is set to WITHOUT_HIST and mon_lw_thresh is set to 5 000 000 while the remaining parameters are set to NONE so that there is no change in behavior from previous releases.</p> <p>For new databases created in DB2 Version 9.7, mon_req_metrics, mon_act_metrics, mon_obj_metrics are set to BASE, mon_lw_thresh is set to 5 000 000, and mon_deadlock is set to WITHOUT_HIST. The mon_locktimeout, mon_lockwait, and mon_uow_data parameters are set to NONE. You need to review the setting of these parameters to ensure that you have the intended settings.</p>

Changes to existing database configuration parameters

The following table describes the upgrade impact of changes to existing database configuration parameters:

Table 14. Changes to existing database configuration parameters

Name	Upgrade impact
applheapsz	<p>In databases upgraded from DB2 Version 9.1 or DB2 UDB Version 8, the applheapsz configuration parameter is set to AUTOMATIC to account for changes to the DB2 memory model. In releases prior to DB2 Version 9.5, this parameter indicated the amount of memory for each database agent. Starting with DB2 Version 9.5, this parameter indicates the total amount of memory for an application.</p> <p>Due to optimization enhancements to match MQTs, the requirement for application heap has increased. If this parameter is set to AUTOMATIC, this setting accounts for the new requirements. If you cannot set this parameter to AUTOMATIC or increase its value, reduce the number of MQTs considered for a given query by using optimization profiles. See “Anatomy of an optimization profile” in <i>Troubleshooting and Tuning Database Performance</i>.</p>
dbheap	<p>The database manager can now determine when to apply row compression to temporary tables that meet certain criteria to improve query performance. Memory allocated for database heap is used to create the compression dictionary and released once the dictionary is created. If you are using row compression and temporary tables eligible for compression, ensure that you have enough space to create the dictionary by setting the dbheap parameter to AUTOMATIC. See “Table compression” in <i>Database Administration Concepts and Configuration Reference</i> for details about temporary table compression.</p>
locklist	<p>Increase the locklist parameter to twice the pre-upgrade value. Due to the increase of the lock request block size to twice the size required in previous releases, active locks in the database require twice the amount of memory.</p> <p>The limit for the locklist parameter is now 134 217 728 pages (4 KB).</p>
logbufsz	<p>The default value for this parameter is now 256 pages (4KB). In previous releases it was 8 pages (4KB). After database upgrade, if you set the cur_commit configuration parameter to ON so that currently committed behavior is enabled on cursor stability scans, ensure that this parameter has a value of 256 or higher.</p> <p>A log sequence number (LSN) now uses 8 bytes. In previous releases, LSN was 6 bytes in length. You might need to increase the value of this parameter according to your database logging activity.</p>
logfilsiz, logprimary	<p>A log sequence number (LSN) uses now 8 bytes. In previous releases, LSN was 6 bytes in length. You might need to increase the value of these parameters according to your database logging activity. See “Maximum limit of log sequence numbers has increased” in <i>What’s New for DB2 Version 9.7</i> for details.</p>

Table 14. Changes to existing database configuration parameters (continued)

Name	Upgrade impact
pckcachesz	<p>To support the new access plan reuse and section diagnostic facilities, package cache memory requirements can increase 25 to 40 percent. For certain types of complex queries, the package cache memory requirements have doubled. The impact from the database upgrade should be minimal because of the small size of this cache relative to overall memory requirements. By setting this parameter to AUTOMATIC, the new requirements are taken into account.</p> <p>For upgraded databases, the INLINE LENGTH default value is the maximum size of the LOB descriptor. LOB data is inlined when the length of the LOB data plus the overhead do not exceed the INLINE LENGTH value. Therefore, if the LOB data length plus the overhead is less than the LOB descriptor size for the LOB column, the LOB data is implicitly inlined in a table row after the database upgrade. Storing LOB data as inlined might require that you increase the pckcachesz database configuration parameter. By setting this parameter to AUTOMATIC, the new requirements are taken into account.</p> <p>The maximum limit for pckcachesz on 64-bit operating systems has been changed to 2 147 483 646. Although this parameter retains its value after database upgrade, consider tuning this parameter. See Chapter 28, “Adopting new DB2 Version 9.7 functionality in database applications and routines,” on page 203 for details.</p>

Deprecated and discontinued 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” in *What’s New for DB2 Version 9.7* to determine the upgrade impact of deprecated registry variables.

If you are upgrading from DB2 Version 9.1 or earlier, consider removing deprecated database configuration parameters in pre-Version 9.7 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-Version 9.7 releases as they do not have the intended effect. See “Changes that impact Version 9.1 or earlier releases” on page 38 for details.

Changes to physical design characteristics of databases

The following table describes the upgrade impact of changes in physical design characteristics of databases:

Table 15. Changes to physical design characteristics of databases

Physical characteristic	Upgrade impact
BUFFER POOL	<p>If you choose to store qualifying LOB data in the row of a table instead of in the default storage objects, the buffer pool usage can increase.</p> <p>In upgraded databases, if the LOB data length plus the overhead is less than the LOB descriptor size for the LOB column, the LOB data is implicitly inlined in a table row. As a result, row length increases, a lower number of rows fit in a page, and a higher number of pages are read into the buffer pool.</p>
LONG VARCHAR and LONG VARGRAPHIC data types	<p>The LONG VARCHAR and LONG VARGRAPHIC data types are deprecated and might be removed in a future release. Although the use of these data types is still supported in the current release, consider migrating to other data types that will continue to be supported in future releases such as VARCHAR, VARGRAPHIC, CLOB, or DBCLOB.</p> <p>You can use the new SYSPROC.ADMIN_MOVE_TABLE system-defined procedure to change the column definition to a compatible data type while the data remains online and available for access. See “Moving tables using the ADMIN_MOVE_TABLE procedure” in Data Movement Utilities Guide and Reference.</p>
TIMESTAMP data type	<p>The TIMESTAMP data type now supports optional fractional seconds. You can specify the number of digits in the fractional seconds as an attribute. The number of digits ranges from 0 to 12 with a default of 6. Therefore, the only impact from upgrading your databases is a minor performance overhead in datetime arithmetic operations.</p> <p>However, if you create new tables using the TIMESTAMP data type with a different number of digits in the fractional seconds other than 6 digits, take into account that the string length for internal representation of a timestamp now ranges from 7 and 13 bytes and adjust your applications accordingly. In previous releases, the string length was 10 bytes. See “Datetime values” in <i>SQL Reference</i> for details.</p>
XML data type	<p>In Version 9.7, the XML storage object has a new format to support new functionality such as row compression on XML data and collection of statistics to estimate the inline length for XML columns. If you have tables with XML columns created in a pre-Version 9.7 release and you want to use this new functionality, convert the XML storage object to the Version 9.7 format by re-creating these tables. See “Converting XML storage objects to the Version 9.7 format” on page 103 for details.</p>

Changes to authorities and privileges

New authorities and changes to the authorization required to run DB2 system commands, CLP commands, and SQL statements are introduced in DB2 Version 9.7. The following table summarizes the upgrade impact of changes in authorities and privileges:

Table 16. Changes to authorities and privileges

Name	Upgrade impact
DBADM authority	<p>In DB2 Version 9.7, there are new authorities for access control and data access. For each authorization ID holding DBADM authority, including the SYSADM group, the UPGRADE DATABASE command explicitly grants ACCESSCTRL and DATAACCESS authorities so that existing database administrators maintain the same authority access and privileges as in previous releases. Review “Database authorities” in <i>Database Security Guide</i> for details about these authorities.</p> <p>The UPGRADE DATABASE command also grants EXECUTE privilege on all system-defined routines by explicitly granting the SYSROLE_AUTH_DBADM system role to any authorization ID holding DBADM authority.</p> <p>Revoking DBADM authority now implicitly revokes all these authorities.</p>
EXECUTE privilege	<p>In DB2 Version 9.7, the UPGRADE DATABASE command revokes the EXECUTE privilege from PUBLIC on the audit routines, AUDIT_LIST_LOGS, AUDIT_DELIM_EXTRACT, and AUDIT_ARCHIVE. For each authorization ID holding SECADM authority, the UPGRADE DATABASE command explicitly grants the EXECUTE privilege on the audit routines by granting the SYSROLE_AUTH_SECADM system role.</p>
SECADM authority	<p>In DB2 Version 9.7, SECADM authority is required for security administration and it is the only authority that provides the ability to grant and revoke all authorities and privileges.</p> <p>If the database does not have a user with SECADM authority, the UPGRADE DATABASE command explicitly grants SECADM authority to the user performing this command. If any users in the SYSADM group need SECADM authority, you must explicitly grant it to them.</p> <p>Also, the UPGRADE DATABASE command revokes the EXECUTE privilege from PUBLIC on the audit routines, AUDIT_LIST_LOGS, AUDIT_DELIM_EXTRACT, and AUDIT_ARCHIVE. For each authorization ID holding SECADM authority, the UPGRADE DATABASE command explicitly grants the EXECUTE privilege on the audit routines by granting the SYSROLE_AUTH_SECADM system role.</p>
SYSADM authority	<p>In DB2 Version 9.7, DBADM authority is required for database administration and SECADM authority is required for security administration. If users in the SYSADM group need either authority, you have to explicitly grant it. Also, a user who holds SYSADM authority is no longer able to grant any authorities or privileges, except to grant tablespace privileges.</p> <p>The UPGRADE DATABASE command explicitly grants DBADM authority to the SYSADM group. Therefore, there should be no upgrade impact but you should review all of the changes in authorities and make any necessary changes.</p>
SYSMON authority	<p>In DB2 Version 9.7, the SYSMON authority now enables a user to also run several LIST commands. See “SYSMON authority ” in <i>Database Security Guide</i> for details.</p>

When the database upgrade is called implicitly using the RESTORE DATABASE command from a pre-Version 9.7 database backup, the changes described in this table are also applied to the database that you are restoring.

On Windows operating systems, the setting for **DB2_GRP_LOOKUP** can have an impact on database upgrade. See the row for the **DB2_GRP_LOOKUP** registry variable in Table 10 on page 30 for more details.

See “Upgrade impact from DB2 command changes” on page 155 and “Upgrade impact from SQL statement changes” on page 162 for a summary of DB2 command and SQL statement changes with upgrade impact. See the *Command Reference* and *SQL Reference* for details about all the changes in authorization.

Changes that impact Version 9.1 or earlier releases

If you are upgrading from DB2 Version 9.1 or earlier, also review all of the changes to variables, database and database manager configuration parameters, and physical design characteristics of databases between pre-Version 9.7 releases that might also impact your upgrade:

- DB2 server behavior changes between DB2 Version 9.1 and DB2 Version 9.5
- DB2 server behavior changes between DB2 UDB Version 8 and DB2 Version 9.1

Deprecated or discontinued functionality that affects DB2 server upgrades

You should be aware of functionality that is deprecated or discontinued in DB2 Version 9.7 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 Version 9.7 is unsupported.

To deal with these functionality changes, you must perform additional tasks before or after upgrade. The majority of these tasks are pre-upgrade or post-upgrade tasks for DB2 servers. The following list describes changes that are not included in the pre-upgrade and post-upgrade tasks for DB2 servers:

Control Center tools have been deprecated

The Control Center tools have been deprecated in DB2 Version 9.7 and might be discontinued in a future release. See “Control Center tools and DB2 administration server (DAS) have been deprecated” in *What’s New for DB2 Version 9.7* for a completed list of the tools that have been deprecated.

Start using the Data Source Explorer in IBM Data Studio to perform database administration tasks. See Database administration from the Data Source Explorer for details. Also, visit the Data Studio product page at <http://www.ibm.com/software/data/studio/> for details about product offerings and downloads.

Netscape support has been discontinued

Netscape is no longer a supported Web browser for First Steps and the installation launchpad. If Netscape is set up as your default Web browser, running First Steps will return the DBI1435E error message.

Setup a supported Web browser as the default Web browser before running First Steps or the installation launchpad. See a list of supported Web

browsers in the installation requirements for DB2 servers and clients by operating system at “Installation requirements for DB2 database products” in *Installing DB2 Servers*.

Health Monitor has been deprecated

The Health Monitor has been deprecated in DB2 Version 9.7 and might be discontinued in a future release. Start using Data Studio to monitor the health of your instances and databases. Visit the Data Studio product page at <http://www.ibm.com/software/data/studio/> for details about product offerings and downloads.

Type-1 indexes have been discontinued

Type-1 indexes have been discontinued in DB2 Version 9.7 and are marked invalid during database upgrade. See “Converting type-1 indexes to type-2 indexes” on page 44 for details on how to avoid the overhead of automatic index rebuild after the database upgrade.

Partitioned databases are no longer supported on Windows 32-bit operating systems

Partitioned databases are no longer supported on Windows 32-bit operating systems in DB2 Version 9.7. The need for partitioned databases to run on 32-bit operating systems has been reduced because of the adoption of 64-bit processors.

You can upgrade a partitioned database from a pre-Version 9.7 release on Windows 32-bit operating systems by migrating first to DB2 Version 9.5 64-bit database product and then upgrading to DB2 Version 9.7 64-bit database product. See *Migrating DB2 32-bit servers to 64-bit systems (Windows)* for details.

Support for Red Hat Enterprise Linux (RHEL) 4 and SUSE Linux Enterprise Server (SLES) 9 operating systems has been discontinued

DB2 Version 9.7 is not supported on RHEL 4 and SLES 9. You must upgrade your operating systems to a supported level before attempting to upgrade to DB2 Version 9.7. See “Best practices for upgrading DB2 servers” on page 22 for details about upgrading your operating systems to a newer version.

Deprecated or discontinued commands

Review “Upgrade impact from DB2 command changes” on page 155 to learn what commands are deprecated and discontinued in DB2 Version 9.7 and how to manage this impact on your database applications and routines.

Raw logs

The use of raw devices for database logging has been deprecated since DB2 Version 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. The following example illustrates how to change the **newlogpath** parameter setting to a file system directory:

```
db2 UPDATE DATABASE CONFIGURATION USING newlogpath /disk2/newlogdir
```

The new setting does not become effective until the database is in a consistent state and all users are disconnected from the database. The database manager moves the logs to the new location after the first user connects to the database.

DB2 Products

Certain **Net Search Extender** (NSE) features and commands have been deprecated and might be discontinued in a future release. See “Net Search Extender features and commands have been deprecated” in *What’s New for DB2 Version 9.7* for details on how to start using equivalent features or commands.

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. See *Migration to DB2 Text Search* for details.

The following DB2 add-on products are deprecated or discontinued:

- **DB2 governor** is deprecated. See “Migrating from DB2 Governor to DB2 workload manager” on page 111 for details on how to upgrade to DB2 Version 9.7 using DB2 workload management.
- **Query Patroller** is deprecated. See “Migrating from Query Patroller to DB2 workload manager” on page 113 for details on how to upgrade to DB2 Version 9.7 using workload management.
- **XML Extender** is discontinued. See “Migrating from XML Extender to pureXML” on page 116 for details on how to upgrade your existing database applications to DB2 Version 9.7 using pureXML[®] feature. Also, see “Upgrading a DB2 server with XML Extender to DB2 Version 9.7” on page 91 to upgrade to a DB2 server without XML Extender functionality.

Upgrade from DB2 Version 9.1 or DB2 UDB Version 8

Review the following topics to learn about additional possible impacts on the upgrade of your DB2 server:

- Deprecated or discontinued functionality in DB2 Version 9.5 for upgrade from DB2 Version 9.1 or DB2 UDB Version 8
- Deprecated or discontinued functionality in DB2 Version 9.1 for upgrade from DB2 UDB Version 8

Migration from non-DB2 relational database management systems

Migrating from a non-DB2 relational database management system is a more complex process than migrating from a DB2 database product. Therefore, you should carefully determine what the migration process entails and create a porting plan.

The porting plan should include tasks such as, converting your database objects to create the equivalent database objects in a DB2 database, moving the actual data to the new DB2 database and porting your database applications. Porting your applications refers to converting SQL statements, modifying interface calls, and converting any database specific code to access DB2 databases.

The most common approaches to converting database application code are manual conversion, dynamic call translation, and automated conversion. In general, conversion tools take source code as input and translate data management calls to

equivalent SQL calls. Information from the source and target database, as well as program code, is used to build the new SQL statements.

The IBM Migration Toolkit (MTK) is a conversion tool that is designed to migrate data and the query and procedure language from source database management systems such as Informix® Dynamic Server, Informix Extended Parallel Server (XPS), Microsoft SQL Server, Oracle, and Sybase Enterprise to DB2 database products. MTK runs on AIX, Linux, Solaris, and Windows operating systems. The only language supported is English. MTK is available as a complementary download from the IBM Migration Toolkit Web page.

The most important and frequently accessed resources that IBM offers to assist in all aspects of migration from a non-DB2 relational database management systems are as follows:

- The IBM DB2 porting Web site can help you to find the information that you need to port your application and its data from other database management systems. This Web site describes the common migration steps and provides resources including tools and education. Additional resources are provided for IBM customers and IBM Business Partners.
- The worldwide IBM Innovation Centers for Business Partners offer a wide range of complimentary workshops and technical seminars. Visit the training resources page to find out details and schedules.
- The IBM Virtual Innovation Center™ (VIC) is an online knowledge and enablement center that provides educational courses, live mentoring, online technical support, solution roadmaps, client simulations, answers to FAQs, case studies, and discussion forums.
- The DB2 Migrate Now! end-to-end offering for strategic IBM Business Partners that includes migration tool kits, complementary online education, information, sales teams and other resources to assist you in planning and implementing your migration to DB2 products from Oracle, Sybase, and Microsoft SQL server.
- The developerWorks® Information Management Web site offers technical resources for DB2 Information Management software. It features product information, downloads, learning resources, support, and communities. On this Web site you can find many articles and tutorials that can help you to learn about the functionality of DB2 database products and how to use them in your applications.

Chapter 5. Pre-upgrade tasks for DB2 servers

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 prior to upgrade in order to have a successful upgrade.

Prepare for the upgrade of your DB2 servers by performing the following tasks:

1. 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.
2. Convert type-1 indexes to type-2 indexes because type-1 indexes are discontinued in DB2 Version 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 Version 9.7.
3. Verify that databases are ready for DB2 upgrade to identify any problems before the actual upgrade. You must resolve them before you proceed with the upgrade.
4. Optional: Stop HADR on the primary and standby databases. You can upgrade only the primary database.
5. Back up your databases to be able to upgrade them to a new upgraded system or restore them in the original pre-upgrade system.
6. Back up configuration and diagnostic information to have a record of your current configuration that you can compare with the configuration after the upgrade. You can also use this information to create new instances or databases using the same configuration that you had before upgrade.
7. Archive all of the DB2 log files, either for SQL replication or Q replication if the log files are needed by the Capture or Q Capture programs, or for high availability disaster recovery (HADR) replication if the log files are needed to create a standby database.
8. Review the disk space requirements to ensure that you 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. Depending on the number of database objects, you might require more log space to perform the upgrade.
9. Windows only: 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-Version 9.7 DB2 copy.
You do not need to backup standard code page conversion tables. Upgrading your pre-Version 9.7 DB2 copy removes these tables because standard code page tables are contained in a DB2 Version 9.7 library.
10. Linux only: Change raw devices to block devices.
11. Optional: Upgrade your DB2 server in a test environment to identify upgrade issues and to verify that applications, scripts, tools and routines work as expected before upgrading your DB2 server in the production environment.
12. In DB2 Version 9.7, 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. See “Setting the diagnostic log file error capture level” in *Troubleshooting and Tuning Database Performance*.

13. Take the DB2 server offline for upgrade.

Converting type-1 indexes to type-2 indexes

Type-1 indexes are discontinued in DB2 Version 9.7 and are marked invalid during database upgrade. Consider converting any existing type-1 indexes to type-2 indexes before the upgrade to avoid the overhead of automatic index rebuild after the upgrade.

Prerequisite

Ensure that you have the required authorization for the `db2IdentifyType1`, `INSPECT`, and `REORG INDEXES/TABLE` command. See the Command Reference for details on the required authorization.

About this task

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-Version 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 Version 7 or earlier and that you upgraded all the way through to DB2 Version 9.5 or databases under an instance where the `DB2_INDEX_TYPE2` registry variable was set to `OFF`.

If you decide not to convert your type-1 indexes before the database upgrade, the type-1 indexes are marked invalid during database upgrade. If the `indexrec` database configuration parameter is set to `RESTART`, indexes marked invalid are rebuilt when the database is restarted. Otherwise, the type-1 index rebuild starts on your first access to the table and you might experience an unexpected degradation in response time.

Procedure

To convert type-1 indexes to type-2 indexes:

1. Ensure that you have enough disk space to convert your type-1 indexes. Type-2 indexes require one more byte per row. Refer to Space requirements for indexes for details on how to estimate the space needed.
2. From a DB2 command line prompt, change to the appropriate directory:
 - On UNIX or Linux operating systems, change to the `$DB2DIR/bin` directory where `DB2DIR` is the location that you specified during the DB2 Version 9.7 installation.
 - On Windows operating system, you need to insert the DB2 Version 9.7 product CD in the drive and change to the `\db2\Windows\utilities` directory.
3. Verify if you have type-1 indexes by issuing the `db2IdentifyType1` command:

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

If you have type-1 indexes, you will receive the following message: Type-1 indexes were found in the inspected tables. The `convert-t1-indexes-`

dbname.db2 command file contains REORG INDEXES ALL commands with the **CONVERT** parameter for each identified type-1 index.

If you are running the `db2IdentifyType1` command on DB2 UDB Version 8 databases, this command does not check for type-1 indexes in typed tables.

The running time of the `db2IdentifyType1` command is proportional to the number of tables in the database and the number of database partitions. For databases with a large number of tables, large number of database partitions, or both, use the `-s` or `-t` parameters to run this command on specific schemas or tables until you process all your tables. See the *Command Reference* for additional performance considerations.

4. If you are upgrading from DB2 UDB Version 8, perform the following actions to determine whether your typed tables have type-1 indexes and manually add the REORG INDEXES ALL commands to the command file:
 - a. Generate a list of all the root tables for typed tables by issuing the following query:

```
SELECT DISTINCT H.ROOT_SCHEMA, H.ROOT_NAME,
               T.TBSPACEID, T.TABLEID
FROM SYSCAT.TABLES T, SYSCAT.INDEXES I,
     SYSCAT.HIERARCHIES H, SYSCAT.NAMEMAPPINGS N
WHERE T.TYPE = 'H' AND T.TABSCHEMA = I.TABSCHEMA
     AND T.TABNAME = I.TABNAME AND H.METATYPE='U'
     AND H.ROOT_SCHEMA=N.LOGICAL_SCHEMA AND H.ROOT_NAME=N.LOGICAL_NAME
     AND T.TABSCHEMA=N.IMPL_SCHEMA AND T.TABNAME=N.IMPL_NAME
```

- b. Identify any typed tables with type-1 indexes using the `INSPECT` command as follows:

```
db2 INSPECT CHECK DATABASE RESULTS KEEP sample.log
db2inspf $INSTHOME/sql1lib/db2dump/sample.log sample.out
```

Use the values for `TBSPACEID` and `TABLEID` from the query result in the previous step to match the *Object* and *Tablespace* identifiers in the formatted output from the `db2inspf` command and determine the index type for each root table as shown in the following example:

```
...
Table phase start (ID Signed: 4, Unsigned: 4;
Tablespace ID: 3) :
```

```
Data phase start. Object: 4 Tablespace: 3
The index type is 1 for this table.
Traversing DAT extent map, anchor 96.
Extent map traversal complete.
DAT Object Summary: Total Pages 20 - Used Pages 20
- Free Space 2 %
Data phase end.
```

```
Index phase start. Object: 4 Tablespace: 3
Traversing INX extent map, anchor 160.
Extent map traversal complete.
INX Object Summary: Total Pages 17 - Used Pages 17
Index phase end.
Table phase end.
Tablespace phase end.
```

- c. Add a REORG INDEXES ALL command to the *convert-t1-indexes-database.db2* command file for each root table with type-1 indexes, that you identified in the previous step, as shown in the following example:

```
REORG INDEXES ALL FOR TABLE table-name
ALLOW WRITE ACCESS CONVERT
```

If you decide not to run the INSPECT command because of time and resources limitations, you can simply add a REORG INDEXES ALL command with the CONVERT parameter to the `convert-t1-indexes-dbname.db2` command file for each of the root tables listed in step 4a on page 45. The CONVERT parameter converts only your type-1 indexes and has no effect on your type-2 indexes.

5. If you have type-1 indexes, convert them to type-2 indexes by running the `convert-t1-indexes-dbname.db2` command file:

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

You can edit this command file and add or remove commands to convert type-1 indexes.

Verifying that your databases are ready for upgrade

Before you upgrade your databases, it is important to use the `db2ckupgrade` command to verify that your databases are ready for upgrade.

The `db2ckupgrade` command verifies that a list of conditions are true in order to succeed at the database upgrade. Also, this command writes to the log file, specified with the `-l` parameter, a warning message for a list of conditions that impact database upgrades. See the Command Reference for details about the list of conditions.

The `db2iupgrade` calls the `db2ckupgrade` command. The `db2iupgrade` fails if the `db2ckupgrade` command finds any of the conditions are not true, and returns the error code DBI1205E.

Prerequisites

- Ensure that you have SYSADM authority.
- Ensure that all the local databases that you want to upgrade are cataloged.
- On Linux or UNIX operating systems, uncompress a DB2 Version 9.7 installation image to be able to run the `db2ckupgrade` command.
- Ensure that you meet the installation requirements for DB2 database products. See “Installation requirements for DB2 database products” in *Installing DB2 Servers* .

Restriction

In a partitioned database environment, to verify that your databases are ready for upgrade, you must run the `db2ckupgrade` command on each database partition. If you do not run the `db2ckupgrade` command on each database partition, the `db2iupgrade` command could succeed even when one or more database partitions are not ready for upgrade. However, the database upgrade will fail. The `db2iupgrade` command only runs the `db2ckupgrade` command on the database partition where you issue the `db2iupgrade` command.

Procedure

To verify that your databases are ready for upgrade:

1. Log on to the DB2 server as the DB2 instance owner that you want to upgrade.
2. If the instance owning the databases that you want to verify is not running, start the instance by running the `db2start` command.
3. From the command line prompt, change to the appropriate directory:

- On UNIX or Linux operating systems, change to the *DIRIMG/db2/OS/utilities/db2ckupgrade/bin* directory where *DIRIMG* is the location where you uncompressed the DB2 Version 9.7 installation image or the directory where you mounted the DB2 product DVD, and *OS* is the operating system name of the DB2 server.
 - On Windows operating system, you need to insert the DB2 Version 9.7 product CD in the drive and change to the *\db2\Windows\utilities* directory.
4. Verify that the local databases that are owned by the current instance are ready to be upgraded and generate a log file by running the `db2ckupgrade` command, as follows:

```
db2ckupgrade sample -l db2ckupgrade.log -u adminuser -p password
db2ckupgrade was successful. Database(s) can be upgraded.
```

where *sample* is the database name and `db2ckupgrade.log` is the log file created in the current directory that includes details on errors and warnings. Each time you issue this command, it overwrites the existing log file. You can rename the log file to avoid losing the error details. You must correct these errors before you upgrade.

If you performed the “Converting type-1 indexes to type-2 indexes” on page 44 pre-upgrade task, you can use the `-not1` parameter to skip the check for type-1 indexes. See 7 for details.

When the `db2iupgrade` command runs the `db2ckupgrade` command, the log file specified is the `db2ckupgrade.log` file in the instance home directory for Linux and UNIX operating systems or in the current directory for Windows operating systems.

5. If you created user-defined data types using a name that is a system built-in data type name, drop these user-defined data types and re-create them using a different name that is not restricted. The `db2ckupgrade` command returns the SQL0473N error message when user-defined data types have a name that is a system built-in data type name. If you try to upgrade the database, the `UPGRADE DATABASE` command will fail.
6. If you created database objects using restricted schema names, drop all the database objects that use reserved schema names and re-create them using a schema name that is not restricted. The `db2ckupgrade` command returns the SQL0553N error message when database objects have restricted schema names. If you try to upgrade the database, the `UPGRADE DATABASE` command will fail.
7. If you have type-1 indexes, perform the “Converting type-1 indexes to type-2 indexes” on page 44 pre-upgrade task or run the generated script file. Alternatively, if you omit the `-not1` parameter, you can run the `type1_index_dbname.db2` script file.

The `db2ckupgrade` command returns the SQL1498W warning message and generates the `type1_index_database-name.db2` script file in the same directory as the `db2ckupgrade` log file. The script file contains `REORG INDEXES ALL` statements with the `ALLOW WRITE ACCESS` and `CONVERT` clauses for each identified type-1 index.

If you do not perform the pre-upgrade task or do not run the generated script, the `UPGRADE DATABASE` command marks all type-1 indexes as invalid. The database manager will automatically rebuild the type-1 indexes as type-2 indexes on the first table access after database upgrade and you might experience an unexpected degradation in response time. Access to the table is not allowed until the index rebuild is completed.

8. If you have external unfenced routines on Linux and UNIX that have no dependency on the DB2 engine library in your database, use generated list to determine if you can safely run your routines as unfenced in the new multithreaded database manager after database migration. See “Upgrading C, C++, and COBOL routines” on page 194 for details.

The db2ckupgrade command writes the ADM4100W warning message to the log file and generates a file, in the same directory as the db2ckupgrade log file, called `alter_unfenced_database-name.lst` containing a list of all the external unfenced routines that will be redefined as FENCED and NOT THREADSAFE when you upgrade the database.

9. If you have identifiers called NULL for column names, routine parameter names, or variable names, qualify or delimit with quotes these identifiers in your SQL statements to avoid conflict with the NULL keyword.

The db2ckupgrade command writes the ADM4102W warning message to the log file when a database has identifiers called “NULL”. If you use identifiers called “NULL” that are not fully qualified or delimited with quotes in your SQL statements, the identifier name might resolve to the NULL keyword instead. This would result in a change in behavior from previous releases. See “Upgrade impact from SQL statement changes” on page 162 for details.

10. If workload connection attributes contain asterisks (*), replace the asterisks (*) with another character. The db2ckupgrade command writes the ADM4103W warning message to the log file when workload connection attributes contain asterisks (*).

Starting with DB2 Version 9.7, you can use a single asterisk (*) as a wildcard character and two asterisks (**) to represent one literal asterisk (*) in some workload attributes. The UPGRADE DATABASE command replaces the single asterisk (*) with two asterisks (**) depending the type of connection attribute.

11. If the database is enabled for XML Extender, perform all the steps in “Upgrading a DB2 server with XML Extender to DB2 Version 9.7” on page 91 to completely disable XML Extender and remove XML Extender functionality before upgrading your instance and databases. The db2ckupgrade command writes the ADM4104E warning message to the log file when a database is enabled for XML Extender.

Starting with DB2 Version 9.7, XML Extender is discontinued.

12. Ensure that the log file for db2ckupgrade command contains the following text: `Version of DB2CKUPGRADE being run: VERSION 9.7`. This text confirms that you are running the correct level of the db2ckupgrade command.
13. Start the instance by running the db2start command.

Backing up databases before upgrade

Before you start the upgrade process to DB2 Version 9.7, 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 Version 9.7, you cannot backup databases until you upgrade them.

Prerequisites

- To backup a database, you require SYSADM, SYSCTRL, or SYSMAINT authority.
- Databases must be cataloged. To view a list of all the cataloged databases in the current instance, enter the following command:

Procedure

To perform an offline full back up for each of your local databases:

1. Disconnect all applications and users from the database. To get a list of all database connections for the current instance, issue the LIST APPLICATIONS command. If all applications are disconnected, this command returns the following message:

```
db2 list applications
SQL1611W No data was returned by the Database System Monitor.
SQLSTATE=00000
```

To disconnect all applications and users, use the FORCE APPLICATION command:

```
db2 force application all
```

2. Backup your database using the BACKUP DATABASE command. The following is an example for UNIX operating systems:

```
db2 BACKUP DATABASE sample USER arada USING password TO backup-dir
```

where *sample* is the database alias, the username is *arada*, the password is *password*, and the directory to create back up files is *backup-dir*.

In partitioned database environments, . Refer to “Backing up partitioned databases” in *Data Recovery and High Availability Guide and Reference*.

If you activated and configured DB2 ACS on your databases in DB2 Version 9.5, you can use the **USE SNAPSHOT** parameter to perform a snapshot backup. However, you can only restore a snapshot backup in a DB2 Version 9.5 instance. You cannot use snapshot backup to upgrade to a new server. Refer to Performing a snapshot backup in *Data Recovery and High Availability Guide and Reference*

If you performed a full *offline* database backup recently and you cannot perform another one before upgrade, you can perform an incremental *offline* database backup instead. Refer to “Upgrading to a new DB2 server” on page 82 for details on how to upgrade your database using an incremental *offline* database backup.

3. Optional: Test the integrity of a backup image to ensure that the image can be restored using the db2ckbkp Check Backup command. The following is an example on UNIX operating systems:

```
cd backup-dir
db2ckbkp SAMPLE.0.arada.NODE0000.CATN0000.20051014114322.001
```

```
[1] Buffers processed: #####
```

```
Image Verification Complete - successful.
```

Backing up DB2 server configuration and diagnostic information

Backing up your settings for database and database manager configuration parameters before DB2 server upgrade, allows you to verify DB2 server behavior after upgrade, 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.

Prerequisite

You must have SYSADM authority in order to execute all of the following tasks, although some tasks require lesser authority privileges or none.

Procedure

To back up your DB2 server configuration and diagnostic information:

1. Collect information from your DB2 servers by running the `db2support` command for all your databases that you are going to upgrade in all your instances. This command allows you to collect information about the database system catalog, database and database manager configuration parameters 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
```

The `-cl 0` parameter collects the database system catalog, database and database manager configuration parameters settings, DB2 registry variables settings. The information collected is stored on a compressed zip file in output directory. A summary report in HTML format is included.

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

2. Back up the information about all the packages for your applications associated with each database. Use the following command to list packages associated with your databases and redirect the command output to a file:

```
db2 LIST PACKAGES FOR SCHEMA schema-name  
SHOW DETAIL > /upgrade/sample_pckg.txt
```

The 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.

3. If you enabled the audit facility, back up the audit configuration of your instances by issuing the following command:

```
db2audit describe > audit_instance-name.cfg
```

If you have multiple instances, repeat this command for each instance.

4. Back up all your external routines. See “Backup and restore of external routine library and class files” in *Administrative Routines and Views*. The following example shows how to backup all external routines created using the default path in UNIX operating systems:

```
cp -R $INSTHOME/sql1lib/function $INSTHOME/routine_backup
```

Where INSTHOME is set to the home directory of the 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.

5. Optional: The `db2support` command HTML report includes the database manager configuration parameter settings for the instance that owns the specified database. You can use the GET DATABASE MANAGER CONFIGURATION command to back up your settings for database manager configuration parameters and redirect the command output to a file to save these settings for each instance:

```
db2 GET DBM CFG > dbm_instname.cfg
```

where *instname* is the instance name.

6. Optional: The `db2support` command HTML report includes the database configuration parameter settings for the specified database. You can use the

GET DATABASE CONFIGURATION command to *back up your settings for database configuration parameters* and redirect the command output to a file to save these settings for each database:

```
db2 CONNECT TO database_alias
db2 GET DB CFG FOR database_alias
    SHOW DETAIL > db_database_alias.cfg
```

where *database_alias* is the database alias. The **SHOW DETAIL** clause displays the values calculated by the database manager when configuration parameters are set to AUTOMATIC.

Database configuration parameters can be the same on each database partition in a partitioned database environment. If they are not the same, back up the database configuration parameter settings for each database partition.

7. Optional: The db2support command generates a file with the output of the db2look command for the specified database. However if you need additional information not present in the generated DDL file, you can use this command to save the DDL information for your databases and the statements to re-create your database objects:

```
db2look -d sample -e -o sample_tbs.db2 -l -x
```

8. Optional: The db2support command HTML report includes the environment and registry variable settings for the instance that owns the specified database. You can use the db2set command to back up your DB2 profile registry variables settings and redirect the command output to a file to save these settings:

```
db2set -all > reg_instname.txt
```

If you set DB2 environment variables, use the appropriate system command to list environment variables and their values. For example, on AIX you can issue the following command:

```
set |grep DB2 > env_instname.txt
```

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.

Increasing table space and log file sizes before upgrade

Before you start upgrading your DB2 server, you must ensure that you have a sufficient amount of free space on your system catalog table space and temporary table space, and enough log space to upgrade your databases.

Prerequisite

Ensure that you have SYSCTRL or SYSADM authority to be able to increase the size of table spaces and log space.

Restriction

Additional considerations are required in partitioned database environments to increase table space sizes because table spaces span across database partitions. Also, you only need to increase the log space in the catalog database partition server.

Procedure

To increase the size of your table spaces and log space:

1. Connect to the database you want to upgrade:

```
db2 CONNECT TO sample
```

2. Determine your table space disk usage by issuing the following query:

```
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')"
```

NAME	TYPE	AUTO_RESIZE	CONTAINERS	TOTAL_PGS	USED_PGS	FREE_PGS	MAX_SZ	PG_SZ
SYSCATSPACE	DMS	1	1	8192	7576	612	-1	8192
TEMPSPACE1	SMS	-	1	10	10	0	-	8192

2 record(s) selected.

Take note of the number of containers, total pages, used pages, free pages, MAXSIZE, and page size.

If you are upgrading from Version 8.1, use the following command: db2 LIST TABLESPACES SHOW DETAIL

- Increase the size of the system catalog table spaces using one of the following options:
 - If you have an SMS table space, ensure that you have at least *the same amount of used pages available as free disk space*; in this example, about 60 MB.
 - If you have a DMS table space and the number of used pages is greater than the number of free pages, use the following formula to calculate the number of pages to increase per container:

$$\text{number_of_pages} = (\text{used_pages} - \text{free_pages}) / \text{number_of_containers_in_SYSCATSPACE}$$

Then use the following command to increase the size of all containers in the system catalog table space:

```
db2 "ALTER TABLESPACE SYSCATSPACE EXTEND (ALL number_of_pages)"
```

- If you have a DMS table space with AUTORESIZE enabled and MAXSIZE is set to NONE, ensure that you have at least *twice the amount of used pages* available in free disk space. If MAXSIZE is set to an integer value that is less than twice the amount of used pages, then you need to increase MAXSIZE using the ALTER TABLESPACE statement as shown in the following example:

```
db2 "ALTER TABLESPACE SYSCATSPACE
      MAXSIZE (2*used_pages_in_SYSCATSPACE*page_size/1024) K"
```

The automatic resizing of table spaces is available since DB2 UDB Version 8 FixPak 9.

In our example, the query results in the previous step shows that SYSCATSPACE is a DMS table space with AUTORESIZE enabled and a MAXSIZE value of -1 which indicates unlimited maximum size. Therefore, you must have twice the amount of used pages available in free disk space.

- Increase the size of the temporary table spaces using one of the following options:

-

If you have an SMS table space you only need to ensure that you have at least twice the amount of total pages for the system catalog table space in free disk space; in this example, about 128 MB.

- If you have a DMS table space, use the following formula to calculate the number of pages to increase per container:

$$\text{number_of_pages} = (\text{number_of_total_pages_in_SYSCATSPACE}) / \text{number_of_containers_in_TEMPSPACE1}$$

Use the following command to increase the size of all containers in the temporary table space:

```
db2 "ALTER TABLESPACE TEMPSPACE1 EXTEND (ALL number_of_pages)"
```

- If you have a DMS table space with AUTORESIZE enabled and MAXSIZE is set to NONE, ensure that you have at least twice the amount of total pages for the system catalog table space in free disk space. If MAXSIZE is set to an integer value that is less than twice the amount of total pages for the system catalog table space, then you need to increase MAXSIZE using the ALTER TABLESPACE statement:

```
db2 "ALTER TABLESPACE TEMPSPACE1  
MAXSIZE (2*total_pages_in_SYSCATSPACE*page_size/1024) K"
```

5. Determine the current log space size using the GET DATABASE CONFIGURATION command. The following example shows how to record the values for *logfilsiz*, *logprimary*, and *logsecond* database configuration parameters on Linux and UNIX operating systems:

```
db2 GET DB CFG FOR sample |grep '(LOG[FPS]') | tee logsize.txt  
Log file size (4KB) (LOGFILSIZ) = 1000  
Number of primary log files (LOGPRIMARY) = 3  
Number of secondary log files (LOGSECOND) = 2
```

6. Increase your log space size using the following commands:

```
db2 UPDATE DB CFG FOR sample using LOGSECOND  
(current_value of LOGPRIMARY + current_value of LOGSECOND) * 2
```

If you already have a large log space, you might not need to increase it.

7. Optional: Enable infinite active logging instead of increasing the log space, by setting *logsecond* to -1 and enabling archive logging. Infinite active logging allows an active unit of work to span the primary logs and archive logs, effectively allowing a transaction to use an infinite number of log files. You should be aware that if the upgrade fails, the time to rollback the transactions will depend on how many archived logs need to be retrieved. The following command shows an example on how to enable archive logging to disk and infinite logging:

```
db2 UPDATE DB CFG FOR sample using LOGARCHMETH1 DISK:archive-dir  
db2 UPDATE DB CFG FOR sample using LOGSECOND -1
```

where *archive-dir* is the directory to archive the log files.

All applications must disconnect from this database before the new values become effective.

Changing raw devices to block devices (Linux)

Changing raw (character) devices to block devices on Linux operating systems is required before you upgrade to DB2 Version 9.7.

The previous raw I/O method that required binding the block device to a raw (character) device using the raw utility is deprecated since DB2 Version 9.1, and will be removed in a future release of DB2 database product. This raw I/O method is also deprecated in the Linux operating system and will be removed in a future release of Linux.

The block device method uses Direct I/O to achieve an equivalent performance compared to using the raw (character) device method.

Prerequisite

Ensure the database is offline in order to relocate the containers or change the log file path.

Restriction

In a partitioned database environment, the `db2relocatedb` command must be run against every database partition that requires changes. A different configuration file must be supplied for each database partition, and must include the `NODENUM` value of the database partition being changed.

Procedure

1. Perform a full offline backup of your database.
2. Shut down your database. Also consider putting the database in quiesce mode using the `QUIESCE DATABASE` command as shown in the following example:

```
db2 CONNECT TO sample
db2 QUIESCE DATABASE DEFER FORCE CONNECTIONS
db2 DEACTIVATE DATABASE database-alias
```

3. Use the `raw -a` system command to see which raw bindings you defined. This information will help you determine the block device you should use to replace a raw device for each container on your table spaces.
4. Create a configuration file for the `db2relocatedb` command. Use the clauses `CONT_PATH` and `LOG_DIR` to specify the old value with the new value. For example, you can create the `moveraw.cfg` file with the following content:

```
DB_NAME=SAMPLE
DB_PATH=/databases/SAMPLE
INSTANCE=db2inst1
NODENUM=0
LOG_DIR=/dev/raw/lograw,/dev/sda5
CONT_PATH=/dev/raw/raw1,/dev/sda1
CONT_PATH=/dev/raw/raw2,/dev/sda2
```

5. Execute the `db2relocatedb` command to change the configuration of the database files as shown in the following example:

```
db2relocatedb -f moveraw.cfg
```

6. Activate your database as shown in the following example:

```
db2 ACTIVATE DATABASE database-alias
```

7. Test that your database is functioning as expected. Connect to the database and execute queries on tables created on the table spaces that you relocated.
8. If you put the database in quiesce mode, you can restore the access and activate the database using the `UNQUIESCE DATABASE` command as shown in the following example:

```
db2 CONNECT TO sample
db2 UNQUIESCE DATABASE
```

If you are restoring from a pre-Version 9.7 backup in DB2 Version 9.7, you must do a redirected restore to indicate block devices instead of raw character devices for your containers and log path.

Upgrading DB2 servers in a test environment

Upgrading DB2 servers in a test environment before you upgrade them in your production environment allows you to address any problems during the upgrade process more effectively and to evaluate the impact of changes introduced in DB2 Version 9.7.

You can also verify that applications, scripts, tools and maintenance procedures work properly before upgrading your production environment. In addition, you can assess the disk requirements and the time that it takes to upgrade the database, to solidify your upgrade plan.

Prerequisite

You must have root authority on Linux and UNIX operating systems or Local Administrator authority on Windows. You must also have SYSADM authority.

Procedure

To duplicate your production environment in a test environment, perform the following tasks:

1. Install DB2 Version 9.5, DB2 Version 9.1, or DB2 UDB Version 8. If you already have a DB2 copy, you do not have to create a new one.
2. Create your instance duplicates as test instances.
3. Create your database duplicates in the testing instances. You can duplicate your databases without data to test only database upgrade or using a data subset to test all your application functionality. Database upgrade converts only system catalog objects. Therefore, the volume of data in the tables does not impact the disk requirements or the time that it takes to upgrade the database.
4. Perform the pre-upgrade tasks that apply to your DB2 server.
5. Install DB2 Version 9.7.
6. Upgrade your test instances.
7. Upgrade your test databases. Keep a record of the time it takes to upgrade each database and the size of the system catalog table space, system temporary table space, and log space. The following example shows how to do this on an AIX operating system:

```
time db2 UPGRADE DATABASE nsample | tee upgrade_time.log
db2 connect to nsample
db2 "SELECT SUBSTR(TBSP_NAME,1,15) NAME, TBSP_TYPE TYPE,
      TBSP_AUTO_RESIZE_ENABLED AUTO_RESIZE, TBSP_TOTAL_PAGES TOTAL_PGS,
      TBSP_USED_PAGES USED_PGS, TBSP_FREE_PAGES FREE_PGS,
      TBSP_PAGE_SIZE PG_SZ, TBSP_EXTENT_SIZE EXTENT_SZ,
      TBSP_PREFETCH_SIZE PREFETCH_SZ, TBSP_NUM_CONTAINERS CONTAINERS
FROM SYSIBMADM.TBSP_UTILIZATION
WHERE TBSP_CONTENT_TYPE IN ('ANY','SYSTEMP')" | tee tbs_details.log
db2 GET DB CFG FOR nsample | grep '(LOG[FPS])' | tee log_size.log
```

Use this information in your upgrade plan.

8. If you found any issues upgrading your test databases, find a resolution to these issues before upgrading your production environment. Add the tasks to resolve these issues to your upgrade plan.
9. Perform post-upgrade tasks that apply to your DB2 server.
10. Verify the upgrade was successful.
11. Test your applications, scripts, tools and maintenance procedures by connecting to the test databases that you upgraded to the DB2 Version 9.7 copy if your test databases are populated with data.

Creating database duplicates

Creating production database duplicates in a test environment allows you to test upgrading your databases before you upgrade them in your production environment.

Prerequisite

Ensure that you have SYSCTRL or SYSADM authority.

This procedure uses DDL scripts to create database duplicates. If you have enough resources, you can also create database duplicates by restoring a database backup to create a new database. See “Restoring to a new database” in *Data Recovery and High Availability Guide and Reference* for details.

Procedure

To create a database duplicate for testing database upgrade:

1. Log on as the instance owner on the production database server and use the db2look command to generate DDL scripts with all the existing objects in your databases. The following command shows how to generate the sample.ddl script for the SAMPLE database:

```
db2look -d sample -a -e -m -l -x -f -o sample.ddl
```

Edit the generated DDL scripts and change:

- The database name in the CONNECT statements
- The path of the user table space containers or data and reduce the sizes to a minimum size since we are going to recreate a database with no data or just a data subset

You can use your own DDL scripts to create test databases in the test instance instead of generating DDL scripts.

2. Log on as the instance owner in the test database server and create your database duplicates. The following example shows how to create a database duplicate of the SAMPLE database using the sample.ddl script:

```
db2 CREATE DATABASE NSAMPLE
db2 -tvsvf sample.ddl
db2 UPDATE DBM CONFIGURATION USING diaglevel 4
```

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. A value of 4 captures additional information that can be helpful in problem determination.

3. Adjust the size of the system catalog table space, temporary table space, and log space in your test databases if required.
4. Export data subsets of your production databases and import these data subsets into your test databases. You only need a data subset if you are going to test your applications in your testing environment.
5. Verify that your database duplicates were created successfully by connecting to the them and issue a small query.

Taking a DB2 server offline before upgrade

Before you can continue with the upgrade process, you must take your DB2 server offline by stopping the DB2 license service, stopping all command line processor sessions, disconnecting applications and users, and stopping the database manager.

Prerequisites

You must have SYSADM authority.

Procedure

To take your DB2 server offline:

1. Stop the DB2 license service:

```
db2licd -end
```

2. Disconnect all applications and users. To get a list of all database connections for the current instance, issue the LIST APPLICATIONS command. If all applications are disconnected, this command returns the following message:

```
db2 list applications
SQL1611W No data was returned by the Database System Monitor.
SQLSTATE=00000
```

To disconnect all applications and users, use the FORCE APPLICATION command:

```
db2 force application all
```

3. Stop all command line processor sessions by entering the following command in each session that was running the command line processor.

```
db2 terminate
```

4. When all applications and users are disconnected, stop each database manager instance:

```
db2stop
```

Chapter 6. Upgrading a DB2 server (Windows)

Upgrading a DB2 server on Windows to DB2 Version 9.7 requires that you install a new DB2 Version 9.7 copy and then upgrade your existing instances and databases to this new copy.

If you choose to automatically upgrade your existing pre-Version 9.7 copy during the DB2 Version 9.7 installation, your instances and DB2 administration server (DAS) are upgraded but you still need to upgrade your databases after installation. If you choose to install a new DB2 Version 9.7 copy, you must manually upgrade your instances, your DAS, and databases.

This upgrade task describes the steps for direct upgrade to DB2 Version 9.7 from DB2 Version 9.5, DB2 Version 9.1, or DB2 UDB Version 8. Review upgrading environments with specific characteristics and determine which task applies better to your environment.

Prerequisites

- Ensure that you have Local Administrator authority. See the Prerequisites section in “Installing DB2 servers (Windows)” in *Installing DB2 Servers* for additional authorization details.
- Ensure that you meet the installation requirements for DB2 database products. Refer to “Installation requirements for DB2 database products” in *Installing DB2 Servers*.
- Review upgrade recommendations and disk space requirements.
- Perform pre-upgrade tasks.

Restrictions

- This procedure applies only to upgrade from DB2 32-bit servers when you install the DB2 Version 9.7 32-bit database product or from DB2 64-bit servers when you install the DB2 Version 9.7 64-bit database product. The instance bit size is determined by the operating system and the DB2 Version 9.7 database product that you install, see “Support changes for 32-bit and 64-bit DB2 servers” on page 27 for details.
- If you are performing a response file installation to automatically upgrade a DB2 UDB Version 8 copy with multiple DB2 products installed, your copy must be at DB2 UDB Version 8 FixPak 7 or later.
- Additional upgrade restrictions apply. Review the complete list.

Procedure

To upgrade a DB2 server to DB2 Version 9.7:

1. Log on to the DB2 server as a user with Local Administrator authority.
2. Install DB2 Version 9.7 by running the setup command to launch the DB2 Setup wizard. You have three choices:
 - To automatically upgrade a DB2 copy, all the instances running on the selected DB2 copy, and your DAS, select the **Work with Existing** option on the **Install a Product** panel. Then, in the **Work with Existing** window, choose the DB2 copy name with the **upgrade** action. The selected DB2 copy and add-on products are uninstalled.

You will get a warning that recommends that you run the db2ckupgrade command if you have local databases. If you completed the pre-upgrade

tasks, ignore this warning and continue the upgrade. Otherwise, verify that your databases are ready for DB2 upgrade before continuing with the installation.

- To create a new copy of DB2 Version 9.7, select the **Install New** option on the **Install a Product** panel.
- To create a response file and perform a response file installation, select the **Work with Existing** option on the **Install a Product** panel. Then in the **Work with Existing** window, choose the DB2 copy name with the **upgrade** action. Finally, in the **Select the installation, response file creation, or both** window, select the **Save my installation setting in a response file** option to create a response file for a response file installation. The response file has the required `UPGRADE_PRIOR_VERSIONS` keyword, the DB2 copy name to upgrade, and the installation path.

The result of the response file installation will be the same as in the first choice, all your instances running on the selected DB2 copy and your DAS are automatically upgraded to the DB2 Version 9.7 copy.

3. Install all DB2 add-on products that were installed in the DB2 copy from which you are upgrading.
4. If you installed a new copy of DB2 Version 9.7, upgrade your instances to this new copy.
5. If you want your applications to access the DB2 Version 9.7 copy through the default interface or if you upgraded your existing DB2 UDB Version 8 copy, set the DB2 Version 9.7 copy as the DB2 default copy. See “Changing the default DB2 and default IBM database client interface copy after installation (Windows)” in *Installing DB2 Servers*. You must define a default copy if you upgraded from DB2 UDB Version 8 because there is no default copy defined in your DB2 server.
6. Optional: If you installed a new copy, upgrade the DAS if you want to keep your existing DAS configuration and use new functionality available in DB2 Version 9.7. If your DAS is running on DB2 UDB Version 8, upgrade it to use the Control Center to administer your instances running under DB2 Version 9.1 or later.
7. Upgrade your databases.

After upgrading the DB2 server, perform the recommended post-upgrade tasks such as resetting the diagnostic error level to its pre-upgrade value, adjusting log space size, and rebinding packages. In addition, verify that the upgrade of your DB2 server was successful.

Upgrading instances

As part of the overall process of upgrading your DB2 server to DB2 Version 9.7, you must upgrade your instances. On Linux and UNIX, you must manually upgrade them. On Windows, you must manually upgrade them if you did not choose to automatically upgrade your existing DB2 copy during the DB2 Version 9.7 installation.

Prerequisites

- You must have root authority on Linux and UNIX operating systems or Local Administrator authority on Windows.
- You must install any DB2 add-on products that were installed in the DB2 copy from which you are upgrading.
- Before running the `db2iupgrade` command, the following steps are recommended:

- Verify that databases are ready for DB2 upgrade.
- On Linux and UNIX operating systems, ensure that there is 20 MB of free space in the /tmp directory. The instance upgrade trace file is written to /tmp.

Restriction

- On Linux and UNIX operating systems, you must not set up the instance environment for the root user. Running the db2iupgrade or the db2icrt command when you set up the instance environment is not supported.
- Review the upgrade restrictions for instance upgrade.

About this task

On Linux and UNIX, you must manually upgrade your instances. On Windows, you must manually upgrade them if you did not choose to automatically upgrade your existing DB2 copy during the DB2 Version 9.7 installation.

Procedure

To manually upgrade your existing instances to DB2 Version 9.7 using the db2iupgrade command:

1. Determine if you can upgrade your existing instances to a DB2 Version 9.7 copy that you installed by performing the following actions:
 - Determine the node type. The following examples show how to use the GET DBM CFG command to find out the node type:

Operating system	Example
Linux and UNIX	db2 GET DBM CFG grep 'Node type' Node type = Partitioned database server with local and remote clients
Windows	db2 GET DBM CFG find "Node type" Node type = Partitioned database server with local and remote clients

- Review Table 7 on page 20 to determine the instance type using the node type and whether instance upgrade is supported. In the previous example, the node type is “Partitioned database server with local and remote clients” therefore the instance type is *ese* and you can only upgrade to a DB2 Version 9.7 copy of DB2 Enterprise Server Edition. On Linux and UNIX operating systems, you can upgrade to a DB2 Version 9.7 copy of DB2 Workgroup Server Edition but your instance is recreated with type *wse* using default configuration values.

If you cannot upgrade your instance to any DB2 Version 9.7 copy that you installed, you need to install a copy of the DB2 Version 9.7 database product that supports upgrade of your instance type before you can proceed with the next step.

2. Disconnect all users, stop back-end processes and stop your existing instances by running the following command:

```
db2stop force (disconnects all users and stops the instance)
db2 terminate (terminates back-end process)
```

3. Log on to the DB2 server as root authority on Linux and UNIX operating systems or Local Administrator authority on Windows operating systems:

4. Upgrade your existing instances by running the db2iupgrade command from the target DB2 Version 9.7 copy location. The following table shows how to run the db2iupgrade command to upgrade your instances:

Operating system	Command syntax
Linux and UNIX	\$DB2DIR/instance/db2iupgrade [-u <i>fencedID</i>] <i>InstName</i> ^a
Windows	"%DB2PATH%"\bin\db2iupgrade <i>InstName</i> /u: <i>user,password</i> ^b

Note:

- a. Where *DB2DIR* is set to the location you specified during DB2 Version 9.7 installation, *fencedID* is the user name under which the fenced user-defined functions (UDFs) and stored procedures will run, and *InstName* is the login name of the instance owner. This example upgrades the instance to the highest level for DB2 database product that you installed, use the **-k** option if you want to keep the pre-upgrade instance type.
- b. Where *DB2PATH* is set to the location you specified during DB2 Version 9.7 installation, *user,password* are the user name and password under which the DB2 service will run, and *InstName* is the name of the instance.

If you did not install all DB2 add-on products that were installed in the DB2 copy from which you are upgrading, the instance upgrade fails and returns a warning message. If you plan to install these products later on or you no longer need the functionality provided by these products, use the **-F** parameter to upgrade the instance.

The db2iupgrade command implicitly calls the db2ckupgrade command with the **-not1** parameter to verify that your local databases are ready for upgrade and logs any errors in the db2ckupgrade.log log file. On Linux and UNIX operating systems, the log file is created in the instance home directory. On Windows operating systems, the log file is created in the current directory where you are running the db2iupgrade command. The **-not1** parameter disables the check for type-1 indexes. You should verify that you do not have type-1 indexes in your databases before upgrading the instance, refer to "Converting type-1 indexes to type-2 indexes" on page 44. The db2iupgrade does not run as long as the db2ckupgrade command reports errors. Check the log file if you encounter any errors.

5. Log on to the DB2 server as a user with sufficient authority to start your instance.
6. Restart your instance by running the db2start command:


```
db2start
```
7. Verify that your instance is running on to DB2 Version 9.7 by running the db2level command:


```
db2level
```

The Informational tokens should include a string like "DB2 v9.7.X.X" where X is a digit number.

Upgrading the DB2 Administration Server (DAS)

Upgrading your DB2 Administration Server (DAS) is only necessary to keep your existing DAS configuration. If your DAS is running on DB2 UDB Version 8, upgrading your DAS is necessary to use the Control Center for administrating instances running on DB2 Version 9.1 or later, task management, and task scheduling.

Otherwise, you can drop your existing DAS and create a new DAS in DB2 Version 9.7. See “Creating a DB2 administration server (DAS) ” in Installing DB2 Servers.

On Windows operating systems, if you choose to automatically upgrade your pre-Version 9.7 copy and you have a DAS running under this copy, the DAS is also upgraded along with your instances.

The DB2 administration tools and the DAS have been deprecated in DB2 Version 9.7 and might be discontinued in a future release. If you plan to use the Data Source Explorer in IBM Data Studio to perform database administration tasks, you do not have to upgrade the DAS. Also, you can drop the DAS and the tools catalog database.

Prerequisite

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

Restriction

- You can have only one DAS per computer.

Procedure

To upgrade the DAS:

1. Log on to the DB2 server as root on Linux and UNIX operating systems or Local Administrator authority on Windows.
2. Upgrade your existing DAS by running the `dasmigr` command:

Operating system	Command syntax
Linux and UNIX	<code>\$DB2DIR/instance/dasmigr</code>
Windows	<code>%DB2PATH%\bin\dasmigr</code>

Where `DB2DIR` and `DB2PATH` indicate the location that you specified during DB2 Version 9.7 installation.

If the DAS is running, the `dasmigr` command stops the DAS before upgrade and starts the DAS after upgrade.

3. If you created a tools catalog database and want to use your existing scripts and schedules on the Version 9.7 DB2 Control Center, perform the following steps:
 - Upgrade the instance that owns the tools catalog database.
 - Upgrade the tools catalog database.
 - Log on to the DB2 server as a user with SYSADM authority and run the `db2tdbmgr` command. This stops the scheduler before upgrading the tools catalog database and restarts it after the upgrade. If you run this tool from a remote client, you must stop the scheduler at the server before running this command and restart it after running this command.
 - Verify that the DAS is configured to access the upgraded tools catalog database by running the `GET ADMIN CFG` command to display the current configuration settings for the tools catalog database:

```
db2 GET ADMIN CFG
```

```
Admin Server Configuration
```

```
...
```

```
Tools Catalog Database
```

```
(TOOLSCAT_DB) = tool1sdb
```

```

Tools Catalog Database Instance      (TOOLSCAT_INST) = db2inst1
Tools Catalog Database Schema      (TOOLSCAT_SCHEMA) = cc
Scheduler User ID                    =

```

Use the UPDATE ADMIN CFG command if you need to change any configuration settings for the tools catalog database.

You should upgrade your tools catalog whether you decide to upgrade your DAS or not.

4. If you do not upgrade or do not have a tools catalog database, you can create one in a Version 9.7 instance to use the task scheduling capability. See “CREATE TOOLS CATALOG command ” in *Command Reference*.

You can now use the Control Center for remote administration of DB2 Version 9.7 instances, as well as pre-Version 9.7 instances.

Upgrading databases

After you upgraded your instances to DB2 Version 9.7, you need to upgrade each database under each instance.

Prerequisites

- Ensure that you have SYSADM authority.
- Ensure that all the local databases that you want to upgrade are cataloged.
- Ensure that you backed up your databases as indicated in the pre-upgrade tasks.
- Ensure that you installed DB2 Version 9.7 and upgraded the instance to Version 9.7.

Restrictions

- Review the upgrade restrictions for database upgrade.

Procedure

To upgrade a DB2 database to DB2 Version 9.7:

1. Log on to the DB2 server as the instance owner or a user with SYSADM authority.
2. Optional: Rename or delete the db2diag log files so that new files are created. Also, remove or move to another directory any existing dump files, trap files, and alert log files in the directory indicated by the **diagpath** parameter. By doing this, the files only contain information about the upgrade process that helps you to isolate and understand any problem that might occur during database upgrade.
3. Upgrade the database using the UPGRADE DATABASE command:

```
db2 UPGRADE DATABASE database-alias USER username USING password
```

where *database-alias* is the name or the alias of the database you want to upgrade and the username and password to authenticate a user with SYSADM authority.

4. If the UPGRADE DATABASE command fails and returns the SQL1704N error message with a reason code that describes the cause of the failure, find this SQL error code and determine the action to take from the list of the possible solutions for each reason code. One of the most common causes of upgrade failure is that the log file space is not large enough, in which case the following error is returned:

SQL1704N Database upgrade failed. Reason code "3".

You must increase log file size and execute the UPGRADE DATABASE command again. After the database upgrade is complete reset the value of **logfilesiz**, **logprimary**, and **logsecond** database configuration parameters.

There are additional error codes that are returned by the UPGRADE DATABASE command for specific cases not supported by database upgrade. These cases are described in the upgrade restrictions.

5. If the UPGRADE DATABASE command returns the SQL1243W warning message, you need to drop or rename the SYSTOOLS.DB2LOOK_INFO table. Otherwise, the ALTER TABLE and COPY SCHEMA statements will fail to run. Check if the SYSTOOLS.DB2LOOK_INFO table exists by running the following command:

```
db2 "SELECT tabname, tabschema, definer FROM syscat.tables
     WHERE tabschema = 'SYSTOOLS' AND tabname = 'DB2LOOK_INFO'"
```

If you created this table, rename it by running the RENAME statement:

```
db2 RENAME SYSTOOLS.DB2LOOK_INFO TO new-table-name
```

If you did not create this table, remove it by running the DROP command:

```
db2 DROP TABLE SYSTOOLS.DB2LOOK_INFO
```

6. If the UPGRADE DATABASE command returns the SQL1499W warning message and writes the ADM4100W warning message with all the details to the administration notification log, you have external unfenced routines on Linux or UNIX that have no dependency on the DB2 engine libraries and the UPGRADE DATABASE command redefines your external routines as FENCED and NOT THREADSAFE. Also, the DB2_FENCED option is set to 'Y' for all user-defined wrappers.

This command also generates a script called *alter_unfenced_database-name.db2* with all the SQL statements to redefine external unfenced routines, altered during the database upgrade, as NOT FENCED and THREADSAFE. This script is created in the directory specified by the **diagpath** database manager configuration parameter. If the **diagpath** parameter is not set, the script is created in the INSTHOME/sql1lib/db2dump directory where INSTHOME is the instance home directory.

If you need to define your routines as NOT FENCED and THREADSAFE, refer to "Upgrading C, C++, and COBOL routines" on page 194 for details on how to safely run your routines in the new multithreaded database manager and then use the generated script to redefine your routines.

7. If the UPGRADE DATABASE command returns the SQL1499W warning message and writes the ADM4101W warning message to the administration notification log, take note of the system catalog tables reported in the ADM4101W message so that you collect statistics on these tables as part of the post-upgrade tasks.
8. If the UPGRADE DATABASE command returns the SQL1499W warning message and writes the ADM4102W warning message to the administration notification log, qualify or delimit with quotes the identifiers called NULL in your SQL statements to avoid conflict with the NULL keyword.

If you use identifiers called NULL for column names, routine parameter names, or variable names in an SQL statement that are not fully qualified or delimited with quotes, the identifier name might resolve to the NULL keyword instead. This would result in a change in behavior from previous releases. Refer to Chapter 22, "Upgrade essentials for database applications," on page 145 for details.

9. If the UPGRADE DATABASE command returns the SQL1499W warning message and writes the ADM4106W warning message to the administration notification log, drop all references to the XML Extender user-defined data types and drop all XML Extender database objects under the DB2XML schema. Starting with DB2 Version 9.7, XML Extender is discontinued.
To avoid this error, perform all the steps in “Upgrading a DB2 server with XML Extender to DB2 Version 9.7” on page 91 to completely disable XML Extender and remove XML Extender functionality before upgrading your instance and databases.
10. If the UPGRADE DATABASE command returns the SQL1499W warning message and writes the ADM4105W warning message to the administration notification log, create new MQ functions for the XML data type by running the enable_MQFunctions command with the **-xml** parameter. The set of DB2 WebSphere[®] MQ functions for XML Extender are dropped during database upgrade.
11. If the UPGRADE DATABASE command returns the SQL1499W warning message and writes the ADM9516W warning message to the administration notification log, verify that the **indexrec** configuration parameter is set to RESTART and issue the RESTART DATABASE command to rebuild indexes marked as invalid during database upgrade. Otherwise, index rebuild starts on your first access to the table and you might experience an unexpected degradation in response time.
Type-1 indexes and indexes with an index page that could not be upgraded are marked invalid during database upgrade.
12. If the UPGRADE DATABASE command returns the SQL0473N error message, you need to reverse the database migration and re-create all user-defined data types that use a system built-in data type name with a different name that is not restricted.
To avoid the UPGRADE DATABASE command failure, re-create these user-defined data types during “Verifying that your databases are ready for upgrade” on page 46.
13. If the UPGRADE DATABASE command returns the SQL1700N error message, you need to reverse the database migration and re-create database objects that use restricted schema names with a schema name that is not restricted.
To avoid the UPGRADE DATABASE command failure, re-create these database objects during “Verifying that your databases are ready for upgrade” on page 46.
14. Compare your database configuration settings after upgrade with the configuration settings you had before you upgraded your database. Verify the following settings and database information are the same:
 - Database configuration parameter settings
 - Table spaces information
 - Packages information for your applications only
 You do not need to check package information for system generated packages. The information about system generated packages can change after upgrade.
15. Verify your database upgrade is successful. Connect to the upgraded databases and issue a small query:

```
db2 connect to sample
```

```
Database Connection Information
```

```
Database server          = DB2/AIX64 9.7.0
```

```
SQL authorization ID = TESTDB2
Local database alias = SAMPLE
```

```
db2 "select * from syscat.dbauth"
```

Alternatively, if you have sample files installed, run the `testdata.db2` script:

```
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 DB2 Version 9.7 installation, and `sample` is the database name.

After upgrading a DB2 database, performing the recommended post-upgrade tasks ensures a successful database upgrade.

Chapter 7. Upgrading a DB2 server (Linux and UNIX)

Upgrading a DB2 server to DB2 Version 9.7 on Linux and UNIX requires that you install a new DB2 Version 9.7 copy and then manually upgrade your existing instances and databases to this new copy.

Prerequisites

Before upgrading the DB2 server:

- Ensure that you have root access.
- Ensure that you meet the installation requirements for DB2 database products. Refer to “Installation requirements for DB2 database products” in *Installing DB2 Servers*.
- Review upgrade recommendations and disk space requirements.
- Perform pre-upgrade tasks.

Restrictions

- On Linux and UNIX operating systems except for Linux on x86, your existing 32-bit or 64-bit instances are upgraded to DB2 Version 9.7 64-bit instances. The operating system and DB2 Version 9.7 database product that you installed determines the instance bit size, see “Support changes for 32-bit and 64-bit DB2 servers” on page 27 for details.
- Additional upgrade restrictions apply. Review the complete list.

About this task

This upgrade task describes the steps for direct upgrade to DB2 Version 9.7 from DB2 Version 9.5, DB2 Version 9.1, or DB2 UDB Version 8 regardless of the instance bit size. Review upgrading environments with specific characteristics and determine which task applies better to your environment.

Procedure

To upgrade a DB2 server to DB2 Version 9.7:

1. Log on to the DB2 server as root.
2. Install DB2 Version 9.7. See “Installing DB2 servers using the DB2 Setup wizard (Linux and UNIX)” in *Installing DB2 Servers*. Run the `db2setup` command and select the **Install New** option on the **Install a Product** panel to install a new copy of DB2 Version 9.7.
3. Install all DB2 add-on products that were installed in the DB2 copy from which you are upgrading.
4. Upgrade instances from the same installation path that you indicated during DB2 Version 9.7 installation. Any 32-bit instances are upgraded to DB2 Version 9.7 64-bit instances except for Linux on x86.
5. Optional: Upgrade your DAS if you want to keep your existing DAS configuration and use new functionality available in DB2 Version 9.7. If your DAS is running on DB2 UDB Version 8, upgrade it to use the Control Center to administer your instances running under DB2 Version 9.1 or later.
6. Upgrade databases.

After upgrading the DB2 server, perform the recommended post-upgrade tasks such as resetting the diagnostic error level, adjusting log space size, and rebinding packages. In addition, verify that the upgrade of your DB2 server was successful.

Upgrading instances

As part of the overall process of upgrading your DB2 server to DB2 Version 9.7, you must upgrade your instances. On Linux and UNIX, you must manually upgrade them. On Windows, you must manually upgrade them if you did not choose to automatically upgrade your existing DB2 copy during the DB2 Version 9.7 installation.

Prerequisites

- You must have root authority on Linux and UNIX operating systems or Local Administrator authority on Windows.
- You must install any DB2 add-on products that were installed in the DB2 copy from which you are upgrading.
- Before running the `db2iupgrade` command, the following steps are recommended:
 - Verify that databases are ready for DB2 upgrade.
 - On Linux and UNIX operating systems, ensure that there is 20 MB of free space in the `/tmp` directory. The instance upgrade trace file is written to `/tmp`.

Restriction

- On Linux and UNIX operating systems, you must not set up the instance environment for the root user. Running the `db2iupgrade` or the `db2icrt` command when you set up the instance environment is not supported.
- Review the upgrade restrictions for instance upgrade.

About this task

On Linux and UNIX, you must manually upgrade your instances. On Windows, you must manually upgrade them if you did not choose to automatically upgrade your existing DB2 copy during the DB2 Version 9.7 installation.

Procedure

To manually upgrade your existing instances to DB2 Version 9.7 using the `db2iupgrade` command:

1. Determine if you can upgrade your existing instances to a DB2 Version 9.7 copy that you installed by performing the following actions:
 - Determine the node type. The following examples show how to use the `GET DBM CFG` command to find out the node type:

Operating system	Example
Linux and UNIX	<pre>db2 GET DBM CFG grep 'Node type'</pre> <p>Node type = Partitioned database server with local and remote clients</p>
Windows	<pre>db2 GET DBM CFG find "Node type"</pre> <p>Node type = Partitioned database server with local and remote clients</p>

- Review Table 7 on page 20 to determine the instance type using the node type and whether instance upgrade is supported. In the previous example, the node type is “Partitioned database server with local and remote clients” therefore the instance type is *ese* and you can only upgrade to a DB2 Version 9.7 copy of DB2 Enterprise Server Edition. On Linux and UNIX operating systems, you can upgrade to a DB2 Version 9.7 copy of DB2 Workgroup Server Edition but your instance is recreated with type *wse* using default configuration values.

If you cannot upgrade your instance to any DB2 Version 9.7 copy that you installed, you need to install a copy of the DB2 Version 9.7 database product that supports upgrade of your instance type before you can proceed with the next step.

2. Disconnect all users, stop back-end processes and stop your existing instances by running the following command:


```
db2stop force (disconnects all users and stops the instance)
db2 terminate (terminates back-end process)
```
3. Log on to the DB2 server as root authority on Linux and UNIX operating systems or Local Administrator authority on Windows operating systems:
4. Upgrade your existing instances by running the `db2iupgrade` command from the target DB2 Version 9.7 copy location. The following table shows how to run the `db2iupgrade` command to upgrade your instances:

Operating system	Command syntax
Linux and UNIX	<code>\$DB2DIR/instance/db2iupgrade [-u <i>fencedID</i>] <i>InstName</i>^a</code>
Windows	<code>"%DB2PATH%"\bin\db2iupgrade <i>InstName</i> /u:<i>user,password</i>^b</code>

Note:

- a. Where *DB2DIR* is set to the location you specified during DB2 Version 9.7 installation, *fencedID* is the user name under which the fenced user-defined functions (UDFs) and stored procedures will run, and *InstName* is the login name of the instance owner. This example upgrades the instance to the highest level for DB2 database product that you installed, use the **-k** option if you want to keep the pre-upgrade instance type.
- b. Where *DB2PATH* is set to the location you specified during DB2 Version 9.7 installation, *user,password* are the user name and password under which the DB2 service will run, and *InstName* is the name of the instance.

If you did not install all DB2 add-on products that were installed in the DB2 copy from which you are upgrading, the instance upgrade fails and returns a warning message. If you plan to install these products later on or you no longer need the functionality provided by these products, use the **-F** parameter to upgrade the instance.

The `db2iupgrade` command implicitly calls the `db2ckupgrade` command with the **-not1** parameter to verify that your local databases are ready for upgrade and logs any errors in the `db2ckupgrade.log` log file. On Linux and UNIX operating systems, the log file is created in the instance home directory. On Windows operating systems, the log file is created in the current directory where you are running the `db2iupgrade` command. The **-not1** parameter disables the check for type-1 indexes. You should verify that you do not have type-1 indexes in your databases before upgrading the instance, refer to “Converting type-1 indexes to type-2 indexes” on page 44. The `db2iupgrade` does not run as long as the `db2ckupgrade` command reports errors. Check the log file if you encounter any errors.

5. Log on to the DB2 server as a user with sufficient authority to start your instance.
6. Restart your instance by running the db2start command:


```
db2start
```
7. Verify that your instance is running on to DB2 Version 9.7 by running the db2level command:


```
db2level
```

The Informational tokens should include a string like "DB2 v9.7.X.X" where X is a digit number.

Upgrading the DB2 Administration Server (DAS)

Upgrading your DB2 Administration Server (DAS) is only necessary to keep your existing DAS configuration. If your DAS is running on DB2 UDB Version 8, upgrading your DAS is necessary to use the Control Center for administrating instances running on DB2 Version 9.1 or later, task management, and task scheduling.

Otherwise, you can drop your existing DAS and create a new DAS in DB2 Version 9.7. See "Creating a DB2 administration server (DAS) " in Installing DB2 Servers.

On Windows operating systems, if you choose to automatically upgrade your pre-Version 9.7 copy and you have a DAS running under this copy, the DAS is also upgraded along with your instances.

The DB2 administration tools and the DAS have been deprecated in DB2 Version 9.7 and might be discontinued in a future release. If you plan to use the Data Source Explorer in IBM Data Studio to perform database administration tasks, you do not have to upgrade the DAS. Also, you can drop the DAS and the tools catalog database.

Prerequisite

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

Restriction

- You can have only one DAS per computer.

Procedure

To upgrade the DAS:

1. Log on to the DB2 server as root on Linux and UNIX operating systems or Local Administrator authority on Windows.
2. Upgrade your existing DAS by running the dasmgr command:

Operating system	Command syntax
Linux and UNIX	\$DB2DIR/instance/dasmigr
Windows	%DB2PATH%\bin\dasmigr

Where DB2DIR and DB2PATH indicate the location that you specified during DB2 Version 9.7 installation.

If the DAS is running, the `dasmigr` command stops the DAS before upgrade and starts the DAS after upgrade.

3. If you created a tools catalog database and want to use your existing scripts and schedules on the Version 9.7 DB2 Control Center, perform the following steps:
 - Upgrade the instance that owns the tools catalog database.
 - Upgrade the tools catalog database.
 - Log on to the DB2 server as a user with SYSADM authority and run the `db2tdbmgr` command. This stops the scheduler before upgrading the tools catalog database and restarts it after the upgrade. If you run this tool from a remote client, you must stop the scheduler at the server before running this command and restart it after running this command.
 - Verify that the DAS is configured to access the upgraded tools catalog database by running the `GET ADMIN CFG` command to display the current configuration settings for the tools catalog database:

```
db2 GET ADMIN CFG
```

```
Admin Server Configuration
```

```
...
Tools Catalog Database           (TOOLSCAT_DB) = toolsdb
Tools Catalog Database Instance   (TOOLSCAT_INST) = db2inst1
Tools Catalog Database Schema     (TOOLSCAT_SCHEMA) = cc
Scheduler User ID                 =
```

Use the `UPDATE ADMIN CFG` command if you need to change any configuration settings for the tools catalog database.

You should upgrade your tools catalog whether you decide to upgrade your DAS or not.

4. If you do not upgrade or do not have a tools catalog database, you can create one in a Version 9.7 instance to use the task scheduling capability. See “`CREATE TOOLS CATALOG` command ” in *Command Reference*.

You can now use the Control Center for remote administration of DB2 Version 9.7 instances, as well as pre-Version 9.7 instances.

Upgrading databases

After you upgraded your instances to DB2 Version 9.7, you need to upgrade each database under each instance.

Prerequisites

- Ensure that you have SYSADM authority.
- Ensure that all the local databases that you want to upgrade are cataloged.
- Ensure that you backed up your databases as indicated in the pre-upgrade tasks.
- Ensure that you installed DB2 Version 9.7 and upgraded the instance to Version 9.7.

Restrictions

- Review the upgrade restrictions for database upgrade.

Procedure

To upgrade a DB2 database to DB2 Version 9.7:

1. Log on to the DB2 server as the instance owner or a user with SYSADM authority.
2. Optional: Rename or delete the db2diag log files so that new files are created. Also, remove or move to another directory any existing dump files, trap files, and alert log files in the directory indicated by the **diagpath** parameter. By doing this, the files only contain information about the upgrade process that helps you to isolate and understand any problem that might occur during database upgrade.
3. Upgrade the database using the UPGRADE DATABASE command:


```
db2 UPGRADE DATABASE database-alias USER username USING password
```

where *database-alias* is the name or the alias of the database you want to upgrade and the username and password to authenticate a user with SYSADM authority.

4. If the UPGRADE DATABASE command fails and returns the SQL1704N error message with a reason code that describes the cause of the failure, find this SQL error code and determine the action to take from the list of the possible solutions for each reason code. One of the most common causes of upgrade failure is that the log file space is not large enough, in which case the following error is returned:


```
SQL1704N Database upgrade failed. Reason code "3".
```

You must increase log file size and execute the UPGRADE DATABASE command again. After the database upgrade is complete reset the value of **logfilesiz**, **logprimary**, and **logsecond** database configuration parameters.

There are additional error codes that are returned by the UPGRADE DATABASE command for specific cases not supported by database upgrade. These cases are described in the upgrade restrictions.

5. If the UPGRADE DATABASE command returns the SQL1243W warning message, you need to drop or rename the SYSTOOLS.DB2LOOK_INFO table. Otherwise, the ALTER TABLE and COPY SCHEMA statements will fail to run. Check if the SYSTOOLS.DB2LOOK_INFO table exists by running the following command:

```
db2 "SELECT tabname, tabschema, definer FROM syscat.tables
     WHERE tabschema = 'SYSTOOLS' AND tabname = 'DB2LOOK_INFO'"
```

If you created this table, rename it by running the RENAME statement:

```
db2 RENAME SYSTOOLS.DB2LOOK_INFO TO new-table-name
```

If you did not create this table, remove it by running the DROP command:

```
db2 DROP TABLE SYSTOOLS.DB2LOOK_INFO
```

6. If the UPGRADE DATABASE command returns the SQL1499W warning message and writes the ADM4100W warning message with all the details to the administration notification log, you have external unfenced routines on Linux or UNIX that have no dependency on the DB2 engine libraries and the UPGRADE DATABASE command redefines your external routines as FENCED and NOT THREADSAFE. Also, the DB2_FENCED option is set to 'Y' for all user-defined wrappers.

This command also generates a script called `alter_unfenced_database-name.db2` with all the SQL statements to redefine external unfenced routines, altered during the database upgrade, as NOT FENCED and THREADSAFE. This script is created in the directory specified by the **diagpath** database

manager configuration parameter. If the **diagpath** parameter is not set, the script is created in the `INSTHOME/sql1lib/db2dump` directory where `INSTHOME` is the instance home directory.

If you need to define your routines as `NOT FENCED` and `THREADSAFE`, refer to “Upgrading C, C++, and COBOL routines” on page 194 for details on how to safely run your routines in the new multithreaded database manager and then use the generated script to redefine your routines.

7. If the `UPGRADE DATABASE` command returns the `SQL1499W` warning message and writes the `ADM4101W` warning message to the administration notification log, take note of the system catalog tables reported in the `ADM4101W` message so that you collect statistics on these tables as part of the post-upgrade tasks.

8. If the `UPGRADE DATABASE` command returns the `SQL1499W` warning message and writes the `ADM4102W` warning message to the administration notification log, qualify or delimit with quotes the identifiers called `NULL` in your SQL statements to avoid conflict with the `NULL` keyword.

If you use identifiers called `NULL` for column names, routine parameter names, or variable names in an SQL statement that are not fully qualified or delimited with quotes, the identifier name might resolve to the `NULL` keyword instead. This would result in a change in behavior from previous releases. Refer to Chapter 22, “Upgrade essentials for database applications,” on page 145 for details.

9. If the `UPGRADE DATABASE` command returns the `SQL1499W` warning message and writes the `ADM4106W` warning message to the administration notification log, drop all references to the XML Extender user-defined data types and drop all XML Extender database objects under the `DB2XML` schema. Starting with DB2 Version 9.7, XML Extender is discontinued.

To avoid this error, perform all the steps in “Upgrading a DB2 server with XML Extender to DB2 Version 9.7” on page 91 to completely disable XML Extender and remove XML Extender functionality before upgrading your instance and databases.

10. If the `UPGRADE DATABASE` command returns the `SQL1499W` warning message and writes the `ADM4105W` warning message to the administration notification log, create new MQ functions for the XML data type by running the `enable_MQFunctions` command with the `-xml` parameter. The set of DB2 WebSphere MQ functions for XML Extender are dropped during database upgrade.

11. If the `UPGRADE DATABASE` command returns the `SQL1499W` warning message and writes the `ADM9516W` warning message to the administration notification log, verify that the **indexrec** configuration parameter is set to `RESTART` and issue the `RESTART DATABASE` command to rebuild indexes marked as invalid during database upgrade. Otherwise, index rebuild starts on your first access to the table and you might experience an unexpected degradation in response time.

Type-1 indexes and indexes with an index page that could not be upgraded are marked invalid during database upgrade.

12. If the `UPGRADE DATABASE` command returns the `SQL0473N` error message, you need to reverse the database migration and re-create all user-defined data types that use a system built-in data type name with a different name that is not restricted.

To avoid the `UPGRADE DATABASE` command failure, re-create these user-defined data types during “Verifying that your databases are ready for upgrade” on page 46.

13. If the UPGRADE DATABASE command returns the SQL1700N error message, you need to reverse the database migration and re-create database objects that use restricted schema names with a schema name that is not restricted.

To avoid the UPGRADE DATABASE command failure, re-create these database objects during “Verifying that your databases are ready for upgrade” on page 46.

14. Compare your database configuration settings after upgrade with the configuration settings you had before you upgraded your database. Verify the following settings and database information are the same:
 - Database configuration parameter settings
 - Table spaces information
 - Packages information for your applications only

You do not need to check package information for system generated packages. The information about system generated packages can change after upgrade.

15. Verify your database upgrade is successful. Connect to the upgraded databases and issue a small query:

```
db2 connect to sample
```

```
Database Connection Information
```

```
Database server      = DB2/AIX64 9.7.0
SQL authorization ID = TESTDB2
Local database alias = SAMPLE
```

```
db2 "select * from syscat.dbauth"
```

Alternatively, if you have sample files installed, run the testdata.db2 script:

```
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 DB2 Version 9.7 installation, and sample is the database name.

After upgrading a DB2 database, performing the recommended post-upgrade tasks ensures a successful database upgrade.

Chapter 8. Upgrading DB2 servers with specific characteristics

There are many factors that can impact the overall upgrade process, and the complexity of your environment is one of these factors.

If you installed multiple DB2 product components, if you are upgrading from a 32-bit Windows operating system to a 64-bit Windows operating system, or if you are upgrading from a partitioned database environment, you must perform upgrade tasks that include steps specific to that environment instead of the basic DB2 server upgrade task.

Determine which of the following upgrade tasks apply to your DB2 server, and perform this task:

- “Upgrading DB2 32-bit servers to 64-bit systems (Windows)”
- “Upgrading non-root installations” on page 79
- “Upgrading a DB2 server with multiple DB2 copies” on page 80
- “Upgrading to a new DB2 server” on page 82
- “Upgrading a DB2 server using online backups from a previous release” on page 85
- “Upgrading partitioned database environments” on page 86
- “Upgrading DB2 Text Search” on page 87
- “Upgrading DB2 Data Links Manager environments” on page 90
- “Upgrading a DB2 server with XML Extender to DB2 Version 9.7” on page 91
- “Upgrading DB2 servers in Microsoft Cluster Server environments” on page 93
- “Upgrading DB2 Connect servers” in *Installing and Configuring DB2 Connect Servers*
- “Upgrading Query Patroller” in *Query Patroller Administration and User’s Guide*
- “Upgrading DB2 Net Search Extender” in *Net Search Extender Administration and User’s Guide*
- “Upgrading DB2 Spatial Extender” in *Spatial Extender and Geodetic Data Management Feature User’s Guide and Reference*

Upgrading DB2 32-bit servers to 64-bit systems (Windows)

On the Windows operating systems, there are two ways to upgrade your DB2 32-bit server to a DB2 Version 9.7 64-bit server. One way is to upgrade your existing DB2 32-bit server to DB2 Version 9.7 32-bit server, and then upgrade to DB2 Version 9.7 64-bit server.

The other way is to upgrade to a new computer where DB2 Version 9.7 64-bit database product is installed.

Prerequisites

- Ensure that you have Local Administrator authority.
- Ensure that the DB2 server is running 64-bit windows operating system.
- Review upgrade recommendations and disk space requirements.
- Perform pre-upgrade tasks.

Restrictions

- This procedure is covered by this task and only applies to Windows on x64.
- Additional upgrade restrictions apply. Review the complete list.
- If you are performing a response file installation to automatically upgrade a DB2 UDB Version 8 copy with multiple DB2 products installed, your copy must be at DB2 UDB Version 8 FixPak 7 or later.

Procedure

To upgrade a pre-Version 9.7 DB2 32-bit server to a DB2 Version 9.7 64-bit server:

1. Log on to the DB2 server as a user with Local Administrator authority.
2. If you have multiples copies of DB2 UDB Version 8 32-bit server, DB2 Version 9.1 32-bit server, or DB2 Version 9.5 32-bit server, perform the following actions to have all instances running under one DB2 copy:
 - Update all your instances to run under one DB2 Version 8 32-bit server copy, one DB2 Version 9.1 32-bit server copy, or one DB2 Version 9.5 32-bit server copy. You can only update instances of the same version.
 - If you have instances running on multiple pre-Version 9.7 copies of different version, upgrade all instances to the highest release of pre-Version 9.7 copies. For example, if you have a Version 8 and a Version 9.1 instance, upgrade your Version 8 instance to the DB2 Version 9.1 32-bit server copy.
 - Uninstall all the remaining DB2 server copies except the DB2 server copy where all instances are running. You should have only one DB2 UDB Version 8 32-bit server copy, DB2 Version 9.1 32-bit server copy, or DB2 Version 9.5 32-bit server copy.
3. Install DB2 Version 9.7 32-bit database product and select the **Work with Existing** option on the **Install a Product** panel. See “Installing DB2 servers (Windows)” in *Installing DB2 Servers*. Then in the **Work with an existing** window, choose the DB2 copy name with the **upgrade** action. The selected DB2 copy is removed, and all your instances running on the selected DB2 copy and your DB2 Administration Server (DAS) are automatically upgraded. Do not install additional copies of 32-bit DB2 Version 9.7.

You will get a warning that recommends that you to run the db2ckupgrade command if you have local databases. Ignore this warning and continue the upgrade if you completed the pre-upgrade tasks. Otherwise, verify that your databases are ready for DB2 upgrade before you continue with the installation.
4. Install DB2 Version 9.7 64-bit database product and select the **Work with Existing** option on the **Install a Product** panel. See “Installing DB2 servers (Windows)” in *Installing DB2 Servers*. Then in the **Work with an existing** window, choose the DB2 copy name with the **upgrade** action. This procedure removes DB2 Version 9.7 32-bit database product, and upgrades your existing 32-bit instances to 64-bit instances.
5. If you want your applications to access DB2 Version 9.7 copy through the default interface or if you upgraded your existing DB2 UDB Version 8 copy, set the DB2 Version 9.7 copy as the DB2 default copy. See “Changing the default DB2 and default IBM database client interface copy after installation (Windows)” in *Installing DB2 Servers*.
6. Upgrade your databases.
7. If you want to have your instances running on multiples copies of DB2 Version 9.7, install additional DB2 Version 9.7 copies and issue the db2iupdt command to run an instance under a different DB2 Version 9.7 copy.

After upgrading the DB2 server, perform the recommended post-upgrade tasks such as resetting the diagnostic error level, adjusting log space size, and rebinding packages. In addition, verify that the upgrade of your DB2 server was successful.

Upgrading non-root installations

Upgrading DB2 Version 9.5 non-root installations to DB2 Version 9.7 on Linux and UNIX requires that you install DB2 Version 9.7 as a non-root user and then upgrade your databases to the Version 9.7 non-root installation.

Prerequisites

Before upgrading a non-root installation:

- Ensure that you meet the installation requirements for DB2 database products. See “Installation requirements for DB2 database products” in *Installing DB2 Servers*.
- Review upgrade recommendations and disk space requirements.
- Perform pre-upgrade tasks that apply, especially “Verifying that your databases are ready for upgrade” on page 46. Upgrading the non-root instance verifies that your local databases are ready for upgrade. If this verification fails, the non-root instance upgrade also fails and the DB2 product is not installed.

Restrictions

- You cannot upgrade a DB2 Version 9.5 root installation to a DB2 Version 9.7 non-root installation. You can upgrade databases from a DB2 Version 9.5 root installation to a DB2 Version 9.7 non-root installation by restoring database backups taken in the DB2 Version 9.5 root installation. Use the same process described in “Upgrading to a new DB2 server” on page 82.
- On Linux and UNIX operating systems except for Linux on x86, your existing 32-bit or 64-bit instances are upgraded to DB2 Version 9.7 64-bit instances. The operating system and DB2 Version 9.7 database product that you installed determines the instance bit size, see “Support changes for 32-bit and 64-bit DB2 servers” on page 27 for details.
- Additional upgrade restrictions apply. Review the complete list.

Procedure

To upgrade a non-root installation to DB2 Version 9.7:

1. Log on to the DB2 server as the non-root user for the DB2 Version 9.5 non-root installation.
2. Review Table 7 on page 20 to determine the instance type using the nodetype and the DB2 database product to which you can upgrade the non-root instance. The DB2 database product installation verifies that you can upgrade the non-root instance to the DB2 database product that you select for installation. If this verification fails, the installation fails and you can only end the installation.
3. Stop the non-root instance.
4. Install DB2 Version 9.7 as a non-root user and select the **upgrade** option. See “Installing a DB2 product as a non-root user” in *Installing DB2 Servers*. The **upgrade** option backs up the DB2 Version 9.5 non-root configuration files, installation directory, installs a new DB2 copy, and upgrades the non-root instance. However, the installation directory is not backed up if you specify the **-f nobackup** parameter and the Version 9.5 copy is removed.

The DB2 product installation also verifies the following conditions:

- The directory `INSTHOME/sqllib_v95` does not exist.
- The non-root instance is stopped.
- The local databases running under the non-root instance are ready for upgrade.

If any of these verifications fail and:

- You are running the `db2setup` command, a message box appears indicating the condition that failed. Take the appropriate corrective action and then select the **upgrade** option and continue.
- You are using a response file or running the `db2_install` command, the installer will exit with error. Take the appropriate corrective action and then re-issue the `db2setup` command specifying the response file or the `db2_install` command.

If any of the local databases running under the non-root instance have type-1 indexes, a message box appears giving you the option to continue the installation and ignore this warning or exit the installation. See step 7 on page 47 in “Verifying that your databases are ready for upgrade” on page 46 for details about what happens when you choose to ignore the warning and how to convert type-1 indexes before upgrade. You must specify the `UPGRADE_DBCK_IGNORE_TYPE1` keyword with the `YES` option in the response file or specify the `-f ignoreType1` parameter with the `db2_install` command so that the installer does not check for type-1 indexes and completes processing successfully.

5. If the DB2 database product installation fails and you specified the `-f nobackup` parameter, manually install the DB2 database product and then run the `db2nrupg` command to upgrade the non-root instance as follows:

```
cd $HOME/sqllib/instance
db2nrupg -b BackupDir
```

Where *BackupDir* is the backup directory for the configuration files of the non-root installation prior to upgrade.

6. If the DB2 database product installation fails, review the installation log file to determine the cause and how to resolve the issue before attempting the installation again. By default, the installation log file is located in the `/tmp` directory.
7. Upgrade databases.
8. Enable root-based features by running the `db2rfe` command.
9. If you had additional DB2 products installed in your Version 9.5 non-root copy, install one DB2 product at a time.

After upgrading the non-root installation, perform the recommended post-upgrade tasks such as resetting the diagnostic error level, adjusting log space size, and rebinding packages. In addition, verify that the upgrade of your DB2 server was successful.

Upgrading a DB2 server with multiple DB2 copies

Upgrading a DB2 server with multiple pre-Version 9.7 DB2 copies, requires that you install DB2 Version 9.7 as a new copy and then manually upgrade the instances and databases after installation.

You can have a DB2 server with multiple copies of DB2 database products Version 9.5 and 9.1 installed. On Linux and UNIX, you could also have multiples copies of

DB2 Enterprise Server Edition (ESE) Version 8 on the same DB2 server if have installed several alternate fix packs as a completely new copy of DB2 ESE Version 8.

You can manually upgrade a pre-Version 9.7 instance at any fix pack level by executing the db2iupgrade command from the target DB2 Version 9.7 copy of your choice. After an instance is upgraded to a DB2 Version 9.7 copy, you cannot upgrade it to another DB2 Version 9.7 copy. However, you can update an instance between different DB2 Version 9.7 copies using the db2iupdt command.

Prerequisites

- Ensure that you have root access on Linux and UNIX operating systems or Local Administrator on Windows.
- Ensure that you meet the installation requirements for DB2 database products. The requirements for operating systems have changed.
- Review upgrade recommendations and disk space requirements.
- Perform pre-upgrade tasks.

Restrictions

- This procedure does not apply to upgrade from DB2 32-bit servers to 64-bit systems on Windows. Refer to “Upgrading DB2 32-bit servers to 64-bit systems (Windows)” on page 77 for details.
- On Linux and UNIX operating systems, you must not set up the instance environment for the root user. Running the db2iupgrade or the db2icrt command when you set up the instance environment is not supported.
- Review the upgrade restrictions for DB2 servers.

Procedure

To upgrade a DB2 server with multiple DB2 copies:

1. Log on to the DB2 server as root or a user with Local Administrator authority.
2. Install DB2 Version 9.7 as a new copy of DB2 Version 9.7 by running the DB2 Setup wizard and select the **Install New** option on the **Install a Product** panel. Refer to the following tasks for details:
 - Installing DB2 servers (Windows) in *Installing DB2 Servers*
 - Installing DB2 servers (Linux and UNIX) in *Installing DB2 Servers*

You can install multiple DB2 Version 9.7 copies, if you want to upgrade your existing instances to different DB2 Version 9.7 copies.
3. Upgrade instances using the db2iupgrade command from the installation path of the DB2 Version 9.7 copy of your choice. For example, assume that you have the following DB2 copies and instances on an AIX server and a Windows server:

Table 17. Directory examples for DB2 copies.

Instance name	OS	DB2 copy directory
db2inst1	AIX	/usr/opt/db2_08_FP7/
db2inst2	AIX	/opt/IBM/db2/V9.1
db2inst3	AIX	/home/db2/myV9.5
No instances created	AIX	/opt/IBM/db2/V9.7 /home/db2/myV9.7
DB2	Windows	C:\Program Files\IBM\SQLLIB\ (Version 8.2)

Table 17. Directory examples for DB2 copies. (continued)

Instance name	OS	DB2 copy directory
DB2_91	Windows	C:\Program Files\IBM\SQLLIB_91\
DB2_95	Windows	D:\Program Files\IBM\V_95\
No instances created	Windows	C:\Program Files\IBM\SQLLIB_97\

You can then run the following commands to successfully upgrade your instances to DB2 Version 9.7:

Table 18. Instance upgrade command examples.

Upgrade Instance	Commands
db2inst1	cd /opt/IBM/db2/V9.7/instance ./db2iupgrade -u db2fenc1 db2inst1
db2inst2	cd /opt/IBM/db2/V9.7/instance ./db2iupgrade db2inst2
db2inst3	cd /home/db2/myV9.7/instance ./db2iupgrade db2inst3
DB2	cd C:\Program Files\IBM\SQLLIB_97\BIN db2iupgrade DB2 /u:db2admin1,password1
DB2_91	cd C:\Program Files\IBM\SQLLIB_97\BIN db2iupgrade DB2_91 /u:db2admin2,password2
DB2_95	cd C:\Program Files\IBM\SQLLIB_97\BIN db2iupgrade DB2_95 /u:db2admin3,password3

4. Optional: Upgrade the DB2 Administration Server if you want to keep your existing configuration and to administer your DB2 Version 9.7 instances using the Control Center.
5. Log on to the DB2 server as a user with SYSADM authority.
6. Upgrade databases.

After upgrading the DB2 server, perform the recommended post-upgrade tasks such as resetting the diagnostic error level, adjusting log space size, and rebinding packages. In addition, verify that the upgrade of your DB2 server was successful.

Upgrading to a new DB2 server

If you want to upgrade to a new DB2 Version 9.7 server, recreate your instances and then upgrade your databases by restoring a pre-Version 9.7 database backup. After restoring the database backup, the RESTORE DATABASE command automatically runs the UPGRADE DATABASE command.

Prerequisites

- Ensure that you have root access on Linux and UNIX operating systems or Local Administrator authority on Windows.
- Ensure that you have SYSADM authority.
- Ensure that you meet the “Installation requirements for DB2 database products” in *Installing DB2 Servers*. The requirements for operating systems have changed.
- Review upgrade recommendations and disk space requirements.
- Perform pre-upgrade tasks.

Restrictions

- Review the upgrade restrictions for DB2 servers.

Procedure

To upgrade to a new DB2 Version 9.7 server:

1. Perform a full offline database backup of your existing databases and any other pre-upgrade tasks that apply. If you performed full *offline* database backups recently and you cannot perform another one before upgrade, you can perform an incremental *offline* database backup instead.
2. Log on to the new DB2 server as root on Linux and UNIX operating systems or user with Local Administrator authority on Windows operating systems.
3. Install DB2 Version 9.7 on the new DB2 server.
4. Create your instances on the new DB2 server by running the `db2icrt` command from the DB2 Version 9.7 copy location that you installed in the previous step. See “Creating an instance using `db2icrt`” in *Installing DB2 Servers*. If the new DB2 server has similar resources, then restore the database manager configuration parameter values for each instance using the `UPDATE DBM CFG` command and the values that you saved in the pre-upgrade tasks.
5. Optional: Create a new DB2 Administration Server (DAS) on DB2 Version 9.7. You need a DAS if you want to keep your existing DAS configuration and use new functionality available in DB2 Version 9.7. If your DAS is running on DB2 UDB Version 8, upgrade it to use the Control Center to administer your instances running under DB2 Version 9.1 or later.
6. Transfer pre-Version 9.7 backup files for all the databases that you want to upgrade to the new DB2 server.
7. Log on to the DB2 server as a user with SYSADM authority.
8. Upgrade the database using the `RESTORE DATABASE` command. The following example shows how to restore the sample database on UNIX operating systems:

```
db2 RESTORE DATABASE sample FROM /db2/backups
```

where *sample* is the database name and `/db2/backups` is the directory for the database backup file.

If you performed an incremental *offline* database backup before upgrade, you must have access to the most recent full *offline* database backup and the incremental *offline* database backup and use an automatic incremental restore to upgrade the database. See “Using incremental restore in a test and production environment” in *Data Recovery and High Availability Guide and Reference*. A manual incremental restore will fail because each `RESTORE DATABASE` command tries to upgrade the database before the database is completely recovered. The following example shows how to perform an automatic incremental restore:

```
db2 RESTORE DATABASE sample INCREMENTAL AUTOMATIC  
TAKEN AT timestamp WITHOUT PROMPTING
```

In a partitioned database environment, you must execute the `RESTORE DATABASE` command in all database partitions starting with the catalog partition first.

9. When the database was restored but the database was not upgraded, the `RESTORE DATABASE` command returns the following error and includes the upgrade error message with the reason code:

```
SQL2519N The database was restored but the restored database was not upgraded
to the current release. Error "-1704" with tokens "3" is returned.
SQLSTATE=57011
```

The error message SQL1704N indicates the database upgrade failed. Find this SQL error code in the *Message Reference Volume 2* to read the list of the possible solutions for each reason code. In the previous example, tokens "3" means reason code 3 which indicates that the upgrade failed because the database logs are full. If this error occurs, complete the following steps to upgrade the database:

- a. Increase the size of the log files.
- b. Upgrade the database using the UPGRADE DATABASE command.
- c. If the 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.

- d. After the database upgrade is completed reset the size of the log files to their pre-upgrade values.
10. Optional: Configure your new DB2 server to use the new resources available by running the AUTOCONFIGURE command to calculate the buffer pool sizes, and the database manager and database configuration parameters values. The following example shows how to run this command to only display recommended values for the sample database:

```
db2 CONNECT TO sample
db2 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 you choose not to run this command or not to apply the recommended values, manually configure your DB2 server to use the new resources. Otherwise, your databases might not perform as expected.

11. Restore any external routines that you backed up in the pre-upgrade tasks. See "Backup and restore of external routine library and class files" in *Administrative Routines and Views*.
12. Verify your database upgrade is successful. Connect to the upgraded databases and issue a small query:

```
db2 CONNECT TO sample

      Database Connection Information

      Database server      = DB2/AIX64 9.7.0
      SQL authorization ID = TESTDB2
      Local database alias = SAMPLE

db2 "SELECT * FROM SYSCAT.DBAUTH"
```

Alternatively, if you have sample files installed, run the testdata.db2 script:

```
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 DB2 Version 9.7 installation, and sample is the database name.

After upgrading the DB2 server, perform the recommended post-upgrade tasks such as resetting the diagnostic error level, adjusting log space size, and rebinding packages. In addition, verify that the upgrade of your DB2 server was successful.

Upgrading a DB2 server using online backups from a previous release

You can rebuild your database on a previous release using online database backups from the same release and then upgrade to DB2 Version 9.7.

Prerequisites

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-Version 9.7 database backups of your databases so that you can rebuild your databases using these online backups.

Restrictions

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 the pre-upgrade tasks.

Procedure

To upgrade a DB2 server using online backups from a previous release:

1. Transfer pre-Version 9.7 online database backup files for all the databases that you want to upgrade to the DB2 server.
2. If you do not have a DB2 copy with the same version as the online database backups, install a DB2 copy of the same version. For example, if you performed the online database backups from a DB2 Version 9.1 copy, you must have a DB2 Version 9.1 copy installed on the DB2 server.
3. If you do not have an instance running on the DB2 copy with the same version as the online backups, create an instance under this DB2 copy.
4. Log on to the DB2 server as a user with SYSADM authority.
5. Rebuild your databases using the RESTORE DATABASE command with the REBUILD WITH ALL TABLESPACES IN DATABASE parameter followed by the ROLLFORWARD DATABASE command as shown in the following example:

```
RESTORE DB db-name
        REBUILD WITH ALL TABLESPACES IN DATABASE
        TAKEN AT timestamp-backup;
ROLLFORWARD DB db-name
        TO END OF LOGS AND STOP;
```

You can choose to rebuild your database with just a subset of table spaces. However, you must drop all table spaces in restore pending state after you issue the ROLLFORWARD DATABASE command. You cannot upgrade databases with table spaces in restore pending state.

Refer to “Database rebuild” in Data Recovery and High Availability Guide and Reference for more details.

6. Verify that the databases that you rebuild are in consistent state by issuing the GET DB CFG command as shown in the following example for Windows operating system:

```
db2 GET DB CFG FOR sample | FIND "consistent"
```

```
All committed transactions have been written to disk = YES
```

7. Upgrade the DB2 server using one of the following tasks:
 - Upgrading a DB2 server (Windows)
 - Upgrading a DB2 server (Linux and UNIX)

Upgrading partitioned database environments

Upgrading partitioned database environments requires that you install DB2 Version 9.7 as a new copy in all database partition servers, upgrade the instances and then upgrade the databases.

Prerequisites

- Ensure that you have root access on Linux and UNIX operating systems or Local Administrator authority on Windows.
- Ensure that you have SYSADM authority.
- Review the “Installation requirements for DB2 database products” in *Installing DB2 Servers*. The prerequisites for operating systems have changed.
- Review upgrade recommendations and disk space requirements.
- Perform pre-upgrade tasks.

Restrictions

- The database partition server where the catalog partition resides must be up and running.
- Use only the **Install New** option in the **Install a Product** panel to install DB2 Version 9.7. If you choose the **upgrade** action when you select the **Work with Existing** option on the **Install a Product** panel, the installation process will fail.
- Additional upgrade restrictions apply. Review the complete list.

Procedure

To upgrade DB2 servers in a partitioned database environment:

1. Perform a full offline backup for all database partitions. In Version 9.5, use the BACKUP DATABASE command with the **ON ALL DBPARTITIONNUMS** parameter to back up all partitions. Verify that your databases are ready for upgrade, and perform any other pre-upgrade tasks that apply.
2. Log on as root on Linux and UNIX operating systems or as a user with Local Administrator authority on Windows operating systems.
3. Install DB2 Version 9.7 on each participant database partition server and setup your partitioned database environment. See “Setting up a partitioned database environment” in *Installing DB2 Servers*. Select the **Install New** option in the **Install a Product** panel. Do not select the **Work with Existing** option.
4. Upgrade each instance on the database partition server that owns the instance. The first entry in the db2nodes.cfg file of the instance is the database partition server instance owner.

5. Upgrade each database by running the `UPGRADE DATABASE` command on the catalog partition. If any database partitions are not available, these database partitions are not upgraded. Also, if the `UPGRADE DATABASE` command is stopped, the remaining database partitions are not upgraded. However, you can run the `UPGRADE DATABASE` command again to process these particular database partitions afterward when they are available.

The catalog partition must be available when you issue the `UPGRADE DATABASE` regardless on what database partition you issue this command from.

6. Create a new DB2 Administration Server (DAS) on each database partition server. If you need to keep your existing DAS settings, you can upgrade the DAS on each participating database partition server instead of creating a new DAS.

After upgrading the DB2 server, perform the recommended post-upgrade tasks such as resetting the diagnostic error level, adjusting log space size, and rebinding packages. In addition, verify that the upgrade of your DB2 server was successful.

Upgrading DB2 Text Search

Upgrading DB2 Text Search to DB2 Version 9.7 requires that you install a new DB2 Version 9.7 copy and DB2 Text Search, and then upgrade your existing instances and databases to this new copy.

Prerequisites

- Ensure that you have root authority on Linux and UNIX operating systems or Local Administrator authority on Windows operating systems.
- Ensure that you meet the installation requirements for DB2 database products. See “Installation requirements for DB2 database products” in *Installing DB2 Servers*.
- Review upgrade recommendations and disk space requirements.
- Perform pre-upgrade tasks.

Restrictions

- On Linux and UNIX operating systems, you must not set up the instance environment for the root user. Running the `db2iupgrade` or the `db2icrt` command when you set up the instance environment is not supported.
- This procedure applies to root installations only, refer to “Upgrading non-root installations” on page 79 for non-root installations with DB2 Text Search.
- Additional upgrade restrictions apply. Review the complete list.

About this task

This task describes the procedure to upgrade DB2 Text Search to DB2 Version 9.7 by installing a new DB2 copy. On Windows operating systems, you also have the option to upgrade a DB2 copy with the **upgrade** action in the **Work with Existing** window. However, this option upgrades all the instances under the copy without Text Search functionality.

Procedure

To upgrade to DB2 Text Search Version 9.7:

1. Back up the values for all the properties that are configurable for DB2 Text Search and the server version by issuing the following commands:

```
configTool printAll -configPath configuration-directory
> db2tss_config.out
adminTool version -configPath configuration-directory
>> db2tss_config.out
```

Where *configuration-directory* is one of the following directories:

- For Linux and UNIX operating systems, the *INSTHOME*/sql1lib/db2tss/config directory where *INSTHOME* is the instance home directory.
- For Windows operating systems, the *INSTPROFDIR*\instance-name\db2tss\config directory where *INSTPROFDIR* is the instance profile directory. Refer to DB2INSTPROF for more details.

The DB2 Text Search instance service must be started before you run the admintool command.

2. If you enabled DB2 Text Search for rich text document support and you are upgrading from DB2 Version 9.5 Fix Pack 3 or later, disable rich text document support. See “Disabling DB2 Text Search for rich text document support” in *DB2 Text Search Guide* for details.
3. Log in as the instance owner or a user with SYSADM authority and then stop the DB2 Text Search instance service by using the following command:

```
db2ts STOP FOR TEXT
```

If you have multiple instances configured as Text Search instance service, repeat this step for each instance.

4. Back up the DB2 Text Search *configuration-directory*. Refer to step 1 for the location of the *configuration-directory*. The following example shows how to back up the *configuration-directory* to a new location on Linux and UNIX operating systems:

```
$cp -R INSTHOME/sql1lib/db2tss/config
      INSTHOME/backup/db2tss/config
```

where *INSTHOME* is the instance home directory.

The following example shows how to back up the *configuration-directory* to a new location on Windows operating systems:

```
xcopy INSTPROFDIR\instance-name\db2tss\config backup-dir /E
```

where *INSTPROFDIR* is the instance profile directory.

5. Log on to the DB2 server as root on Linux and UNIX operating systems or user with Local Administrator authority on Windows operating systems.
6. Install a new copy of DB2 Version 9.7 and DB2 Text Search. Refer to “Installing and configuring DB2 Text Search” in *DB2 Text Search Guide* for details. Perform a custom installation. DB2 Text Search is an optional component that is only available when you select a custom installation. It is no longer included as part of a typical installation as in pre-Version 9.7 releases.
7. Upgrade your instances by using one of the following tasks:
 - For instances that you are configuring for Text Search instance services on Windows operating systems, run the db2iupgrade command with the /j parameter as follows:

```
db2iupgrade /j "text_search [[,service-name]][[,port-number]]"
```


- For instances that you are not configuring for Text Search instance services or instances on Linux and UNIX operating systems, perform the upgrading instance task.
8. On Windows operating systems, if you want your applications to access the DB2 Version 9.7 copy through the default interface or if you upgraded your existing DB2 UDB Version 8 copy, set the DB2 Version 9.7 copy as the DB2 default copy. See “Changing the default DB2 and default IBM database client interface copy after installation (Windows)” in *Installing DB2 Servers*. You must define a default copy if you upgraded from DB2 UDB Version 8 because there is no default copy defined in your DB2 server.
 9. Optional: If you installed a new copy, upgrade the DAS if you want to keep your existing DAS configuration and use new functionality available in DB2 Version 9.7. If your DAS is running on DB2 UDB Version 8, upgrade it to use the Control Center to administer your instances running under DB2 Version 9.1 or later.
 10. Upgrade your databases.
 11. Upgrade DB2 Text Search server for your instances by issuing the `configTool upgradeInstance` command as shown in the following examples:
 - For Linux and UNIX operating systems:


```
configTool upgradeInstance
  -installedConfigPath $DB2DIR/cfg/db2tss/config
  -configPath $INSTHOME/sqllib/db2tss/config
```

where *INSTHOME* is the instance home directory and *DB2DIR* is the location of the DB2 Version 9.7 copy.
 - For Windows operating systems:


```
configTool upgradeInstance
  -installedConfigPath "%DB2PATH%\CFG\DB2TSS\CONFIG"
  -configPath "%INSTPROFDIR%\instance-name\DB2TSS\CONFIG"
```

where *DB2PATH* is the location of the DB2 Version 9.7 copy and *INSTPROFDIR* is the instance profile directory.
 12. Review the values for all the properties that are configurable for DB2 Text Search and compare with the values that you backed up to ensure the properties have correct values by using the following command:


```
configTool printAll -configPath configuration-directory
```
 13. If you disabled DB2 Text Search for rich text document support in step 2 on page 88, setup and enable rich text document support by performing the following tasks:
 - Setup DB2 Text Search for rich text document support. See “Setting up DB2 Text Search for rich text document support” in *DB2 Text Search Guide* for details
 - Enable DB2 Text Search for rich text document support. See “Enabling DB2 Text Search for rich text document support” in *DB2 Text Search Guide* for details
 14. Verify that the upgrade was successful by starting the DB2 Text Search instance service and printing the status for all collections, as follows:


```
db2ts "START FOR TEXT"
adminTool status -configPath configuration-directory
```

If you disabled DB2 Text Search for rich text document support in step 2 on page 88, verify that rich text document support is enabled by issuing text search queries and compare with pre-upgrade results.

What to do next

After upgrading the DB2 server, perform the recommended post-upgrade tasks such as resetting the diagnostic error level to its pre-upgrade value, adjusting log space size, and rebinding packages. In addition, verify that the upgrade of your DB2 server was successful.

Upgrading DB2 Data Links Manager environments

Upgrading a DB2 server where Data Links Manager is installed or where Data Links functionality is enabled from DB2 UDB Version 8 to DB2 Version 9.7 is not supported. However, you can upgrade to DB2 Version 9.7 if you remove the Data Links Manager functionality.

Prerequisites

- Ensure that you have root access on Linux and UNIX operating systems or Local Administrator on Windows.
- Ensure that you have SYSADM authority.
- Ensure that you meet the installation requirements for DB2 database products. The requirements for Linux and UNIX operating systems have changed.
- Review upgrade recommendations and disk space requirements.
- Perform pre-upgrade tasks.

Restrictions

- Review the upgrade restrictions for DB2 servers.

Procedure

To upgrade a DB2 server in the Data Links environment to DB2 Version 9.7:

1. Remove Data Links Manager from your databases.
2. If you installed DB2 Net Search Extender (NSE), you need to drop the following UDFs :

```
db2 DROP SPECIFIC FUNCTION DB2EXT.DATALINKCONTENT1;  
db2 DROP SPECIFIC FUNCTION DB2EXT.DATALINKCONTENT2;  
db2 DROP SPECIFIC FUNCTION DB2EXT.DATALINKCONTENT4;  
db2 DROP SPECIFIC FUNCTION DB2EXT.DATALINKCONTENT3;
```

These UDFs are always created by NSE for Data Links support, regardless of Data Links Manager installation. Therefore you need to remove these functions even when Data Links Manager is not installed.

If you plan to upgrade by restoring from a database backup, you must drop these UDFs before you back up the database. You cannot restore from a database backup if these UDFs are defined.

3. Drop all references to the DATALINK data type from tables, distinct types, structured types, user-defined functions (UDFs), methods, and dependent objects.
4. Uninstall Data Links Manager on the DB2 server that you want to upgrade.
5. Update your instances to remove the Data Links functionality by running the db2iupdt command:

```
db2iupdt instance-name
```

6. Optional: Disable use of DB2 Data Links functionality by setting the **datalinks** database manager configuration parameter to NO:

```
db2 UPDATE DBM CFG USING datalinks NO
```

When you upgrade the instance, the **datalinks** parameter is set to NO.

7. Install a new DB2 Version 9.7 copy on your DB2 server. If you are installing DB2 Version 9.7 on Windows and you choose to upgrade your existing DB2 UDB Version 8 copy, proceed to step 10.
8. Upgrade instances from the new DB2 Version 9.7 copy .
9. Optional: Upgrade the DB2 Administration Server if you want to keep your existing configuration and to administer your DB2 Version 9.7 instances using the Control Center.
10. Upgrade databases.

After upgrading the DB2 server, perform the recommended post-upgrade tasks such as resetting the diagnostic error level, adjusting log space size, and rebinding packages. In addition, verify that the upgrade of your DB2 server was successful.

Upgrading a DB2 server with XML Extender to DB2 Version 9.7

Upgrading XML Extender to DB2 Version 9.7 requires that you remove the XML Extender functionality then upgrade to DB2 Version 9.7 without this functionality. Upgrading a pre-Version 9.7 DB2 server where XML Extender is installed or where XML Extender functionality is enabled to DB2 Version 9.7 is not supported because XML Extender is discontinued.

Prerequisites

- Ensure that you have root access on Linux and UNIX operating systems or Local Administrator on Windows.
- Ensure that you have SYSADM authority.
- Ensure that you meet the installation requirements for DB2 database products. The requirements for Linux and UNIX operating systems have changed.
- Review upgrade recommendations and disk space requirements.
- Perform pre-upgrade tasks.

Restrictions

- Review the upgrade restrictions for DB2 servers.

Procedure

To upgrade a pre-Version 9.7 DB2 server with XML Extender functionality to DB2 Version 9.7:

1. Optional: Back up all DAD or DTD files from the db2xml.DTD_REF or db2xml.XML_USAGE table for each database that you enabled for XML Extender. The following example shows how to export the DTD files stored in the DTD_REF table to a specific directory:

```
db2 EXPORT TO dtdfiles.del OF del LOBS TO dir-name
      MODIFIED BY lobsinsefiles
      SELECT CONTENT FROM DB2XML.DTD_REF
```

The following example shows how to export the DAD files stored in the db2xml.XML_USAGE table to a specific directory:

```
db2 EXPORT TO dadfiles.del OF del LOBS TO dir-name
      MODIFIED BY lobsinsefiles
      SELECT DAD FROM DB2XML.XML_USAGE
```

2. Disable all XML columns that you enabled for XML Extender in all databases by using the following command:

```
dxxadm disable_column db_name tab_name column_name
```

Alternatively, you can call the `dxxDisableColumn()` stored procedure. The following example shows how to list the columns that are enabled for XML Extender:

```
db2 SELECT TABLE_SCHEMA, TABLE_NAME, COL_NAME
      FROM DB2XML.XML_USAGE
      WHERE NOT TABLE_SCHEMA='DXX_COLL'
```

3. Optional: Disable all XML collections that you enabled for XML Extender in all databases. Alternatively, you can call the `dxxDisableCollection()` stored procedure. The following example shows how to list all the collections that are enabled for XML Extender:

```
db2 SELECT TABLE_SCHEMA, TABLE_NAME, COL_NAME
      FROM DB2XML.XML_USAGE
      WHERE TABLE_SCHEMA='DXX_COLL' AND TABLE_NAME='DXX_COLLECTION'
```

4. Drop all references to the XMLVARCHAR, XMLCLOB, and XMLFILE user-defined data types from tables, distinct types, structured types, user-defined functions (UDFs), methods, and dependent objects in all databases. The following example shows how to list the columns that use the XML Extender UDTs:

```
db2 SELECT TABSCHEMA, TABNAME, COLNAME
      FROM SYSCAT.COLUMNS
      WHERE TYPESCHEMA='DB2XML' AND NOT TABSCHEMA='DB2XML'
```

Alternatively to dropping the column, you could add a new column using a built-in type to move the data from the XML columns using the UPDATE statement before dropping them. You cannot use the ALTER TABLE statement with the ALTER COLUMN clause to change the data type. You can also use the EXPORT, IMPORT, and LOAD utilities to re-create the table without XML columns.

5. Disable all databases enabled for XML Extender by running the following command for each database:

```
dxxadm disable_db database_name
```

Alternatively, you can call the `dxxDisableDB()` stored procedure.

6. Uninstall XML Extender on the DB2 server that you want to upgrade.
7. Update your instances to remove the XML Extender functionality by running the `db2iupdt` command:

```
db2iupdt instance-name
```
8. Install a new DB2 Version 9.7 copy on your DB2 server. If you are installing DB2 Version 9.7 on Windows and you choose to upgrade your existing pre-Version 9.7 DB2 copy, proceed to step 11,
9. Upgrade instances from the new DB2 Version 9.7 copy.
10. Optional: Upgrade the DB2 Administration Server if you want to keep your existing configuration and to administer your DB2 Version 9.7 instances using the Control Center.
11. Upgrade databases.

What to do next

After upgrading the DB2 server, perform the recommended post-upgrade tasks such as resetting the diagnostic error level, adjusting log space size, and rebinding packages. In addition, verify that the upgrade of your DB2 server was successful.

Refer to “Migrating from XML Extender to pureXML” on page 116 for details on how to migrate to pureXML after the upgrade.

Upgrading DB2 servers in Microsoft Cluster Server environments

Upgrading DB2 servers in Microsoft Cluster Server (MSCS) environments to DB2 Version 9.7 requires that you install DB2 Version 9.7 as a new copy in all nodes and then upgrade your MSCS instances and databases.

Microsoft Cluster Server (MSCS) provides High Availability functions to windows users. During setup of DB2 server failover support on MSCS, a server instance is transformed into an MSCS instance. You can run the `db2iupgrade` command to upgrade your MSCS instance and to upgrade existing pre-Version 9.7 MSCS resources to DB2 Version 9.7 DB2 MSCS resources.

Prerequisites

- Ensure that you have Local Administrator access.
- SYSADM authority is required.
- Review upgrade recommendations and disk space requirements.
- Perform pre-upgrade tasks, especially Back up your databases.

Restrictions

- This procedure applies only to upgrade from DB2 32-bit servers when you install the DB2 Version 9.7 32-bit database product, or from DB2 64-bit servers when you install the DB2 Version 9.7 64-bit database product. The instance bit size is determine by the operating system and the DB2 Version 9.7 database product that you install, see “Support changes for 32-bit and 64-bit DB2 servers” on page 27 for details.
- Use only the **Install New** option in the **Install a Product** panel to install DB2 Version 9.7. If you choose the **upgrade** action when you select the **Work with Existing** option on the **Install a Product** panel, the installation process will fail.
- Additional upgrade restrictions apply. Review the complete list.

Procedure

To upgrade a DB2 server in an MSCS environment to DB2 Version 9.7:

1. Log on to the DB2 server as a user with Local Administrator authority.
2. Install DB2 Version 9.7 in all of the nodes in the MSCS cluster. Run the setup command to launch the DB2 Setup wizard and select the **Install New** option in the **Install a Product** panel. Do not select the **Work with Existing** option.
3. Take the resource for the instance offline using the Cluster Administrator. The resource name is the same as the instance name. Ensure that all the remaining resources in the same group as the instance are online.

For more information about using the Cluster Administrator refer to MSCS documentation.

4. Upgrade your MSCS instances by running the `db2iupgrade` command. This command defines a new resource type called “DB2 Server”, and updates all DB2 MSCS resources to use the new resource type. Having a new resource type during the upgrade eliminates conflict with existing pre-Version 9.7 MSCS resources.

```
$DB2DIR\bin\db2iupgrade /u:user,password MSCS-InstName
```

You must run this command from the node that owns all the instance dependent resources.

5. Stop and restart the cluster service in all of the nodes in the MSCS cluster using the Cluster Administrator.
6. Bring online the group of resources containing the upgraded instance using the Cluster Administrator.
7. Optional: Upgrade your DB2 Administration Server (DAS) if you want to keep your existing DAS configuration and use new functionality available in DB2 Version 9.7. If your DAS is running on DB2 UDB Version 8, upgrade it to use the Control Center to administer your instances running under DB2 Version 9.1 or later. If you choose to create a new DAS, you have to re-configure the DAS settings for your MSCS environment.
8. Upgrade your databases.

After upgrading the DB2 server, perform the recommended post-upgrade tasks such as resetting the diagnostic error level, adjusting log space size, and rebinding packages. In addition, verify that the upgrade of your DB2 server was successful.

Chapter 9. Post-upgrade tasks for DB2 servers

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

Procedure

Perform the following post-upgrade tasks that apply to your DB2 server:

1. If you set the **diaglevel** database manager configuration parameter to 3 or higher as recommended in the pre-upgrade tasks for DB2 servers, reset this parameter to the value set before the upgrade.
2. Adjust the log space size. If you changed your log space setting as recommended in the pre-upgrade tasks for DB2 servers, reset the **logfilesiz**, **logprimary**, and **logsecond** database configuration parameters to their pre-upgrade values. Ensure that the amount of log space that you allocate is adequate for your DB2 server. See “Adjusting the log space size in upgraded databases” on page 97 for details.
3. Ensure that existing libraries for your external routines remain on the original location prior to the upgrade, if necessary, restore these libraries from the backup that you perform in “Backing up DB2 server configuration and diagnostic information” on page 49.
4. Activate your database after upgrade to start up your database and all necessary database services. See “Activating a database after upgrade” on page 98 for details.
5. Manage changes in DB2 server behavior. There are new registry variables, new configuration parameters, and new default values for registry variables and configuration parameters introduced in DB2 Version 9.7 that can impact the behavior of DB2 server. There are also changes in physical design characteristics of databases and changes to security that also have an impact. See “Managing DB2 server behavior changes” on page 98 for details.
6. Set up security to manage database audit in upgraded databases. If you enabled the audit facility in your upgraded databases, grant security administrator (SECADM) authority to allow users to configure and manage database audit using DDL statements. See “Setting up security to manage database auditing in upgraded databases” on page 100 for details.
7. If the automatic collection of statistics failed on certain system catalog tables during database upgrade, update the statistics on those system catalog tables. See “Collecting catalog statistics” in *Troubleshooting and Tuning Database Performance*.
8. Rebind packages in upgraded databases to validate packages and to use updated statistics or new index information. See “Rebinding packages in upgraded databases” on page 101 for details.
9. Migrate DB2 explain tables to retain explain table information that you previously gathered. See “Migrating explain tables” on page 102 for details.
10. If you have tables with XML columns that you created in a pre-Version 9.7 release, convert the XML storage object to the Version 9.7 format by recreating these tables to have access to new functions such as compression on XML data and collection of statistics to estimate the inline length for XML columns. See “Converting XML storage objects to the Version 9.7 format” on page 103 for details.

11. Ensure that you meet system temporary table spaces page sizes requirements to accommodate the largest row size in your result sets from queries or positioned updates, and create a system temporary table space with a larger page size if necessary. See “Ensuring system temporary table spaces page sizes meet requirements” on page 103 for details.
12. If you obtained customized code page conversion tables from the DB2 support service, copy all of the files for those tables from the *DB2OLD/conv* to *DB2DIR/conv*, where *DB2OLD* is the location of your DB2 Version 9.1 or DB2 UDB Version 8 copy and *DB2DIR* is the location of your DB2 Version 9.7 copy. You do not have to copy standard code page conversion tables.
If you upgraded your existing DB2 Version 9.1 or DB2 UDB Version 8 copy on Windows operating systems, you can restore the customized code page conversion tables that you backed up as part of the pre-upgrade tasks for DB2 servers to the **DB2PATH\conv** directory, where **DB2PATH** is the location of your DB2 Version 9.7 copy.
13. If you created write-to-table event monitors in DB2 Version 9.1 or DB2 UDB Version 8, recreate your write-to-table event monitors so that you can successfully activate these monitors after you upgrade to DB2 Version 9.7. See “Re-creating write-to-table event monitors” on page 105 for details.
14. Verify that your DB2 server upgrade was successful. Test your applications and tools to ensure that the DB2 server is working as expected. See “Verifying upgrade of DB2 servers” on page 106 for details.
15. Back up your databases after the DB2 server upgrade is complete. See “Backing up databases before upgrade” on page 48 for details.
16. If you have recoverable databases, the **UPGRADE DATABASE** command renamed all log files in the active log path using the **.MIG** extension. After verifying the database upgrade was successful and backing up your databases, you can delete the **S*.MIG** files that are located in the active log path.

What to do next

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

- If you upgrade a DB2 server running high availability disaster recovery (HADR) replication, initialize HADR replication. See “Initializing high availability disaster recovery (HADR)” in *Data Recovery and High Availability Guide and Reference*. During upgrade to DB2 Version 9.7 in a high availability disaster recovery (HADR) replication environment, a database role is changed from primary to standard. Upgrade of standby databases is not supported because these databases are in roll forward pending state.
- If you are using index extensions or spatial indexes and you upgraded from a DB2 UDB Version 8 32-bit instance to a DB2 Version 9.7 64-bit instance, recreate your index extensions or spatial indexes. If you are a Spatial Extender user, review the upgrading the Spatial Extender environment task for details on how to recreate your spatial indexes. The DB2 Spatial Extender and Geodetic Data Management Feature User’s Guide and Reference is available by at <http://www.ibm.com/software/data/spatial/db2spatial/library.html>.

When your 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 Version 9.7, the statistics collected from your 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. However, you only need these statistics if you are using new functionality.

After updating statistics for your upgraded databases, determine if index or table reorganization is necessary by running the REORGCHK command. Table and index reorganization can help you to improve performance.

At this point, you should resume all of your maintenance activities such as backing up databases and updating statistics. You should also remove any DB2 Version 9.1 or DB2 UDB Version 8 copies that you no longer need.

Adjusting the log space size in upgraded databases

You need to set the appropriate size for log files since it is one of the important factors in tuning your DB2 server. Also, if you increased the log files sizes as a pre-upgrade task, you can restore additional free space to your DB2 server.

Prerequisite

You must have SYSCTRL or SYSADM authority in order to be able to increase the size of table spaces and log space.

Restriction

On a partitioned database environment, you only need to adjust the log space size on the catalog database partition server.

Procedure

1. Connect to the database that you upgraded:

```
db2 CONNECT TO sample
```

where *sample* is the database name.

2. Restore your log file size settings to the values you had before upgrade:

```
db2 UPDATE DB CFG FOR sample using LOGSECOND previous-value
```

where *previous-value* is the setting that you save before upgrade and *sample* is the database name. In the pre-upgrade task, only the **logprimary** and the **logsecond** parameters were changed. If you change the setting for the **logfilsiz** parameter, you should restore the 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
```

where *previous-value* is the setting that you save before upgrade and *sample* is the database name.

3. Optional: If you are upgrading from Version 9.1 or Version 8, increase your log file size settings. The RID for log records has increased in the amount of 2 bytes, depending on the type of log record this could represent less than 2% increase in the log record size.

In general, your current setting for log space should be sufficient to accommodate this change. However, if you have a concern that your log space setting is undersized, monitor the log space usage to find out the appropriate size. The following example increases log file size by 5% to accommodate the log record size increase:

```
db2 UPDATE DB CFG FOR sample using LOGFILSIZ previous-value*1.05
```

where *previous-value* is the setting that you save before upgrade and *sample* is the database name.

4. Disconnect from the database that you upgraded:

```
db2 CONNECT RESET
```

LOGFILSIZ changes only take effect when the database is reactivated. All applications must first disconnect from the database then deactivate and activate the database again.

Activating a database after upgrade

Activating your database allows you to ensure that all database services are running properly and to address any problems that might occur during the database activation. You can also eliminate the overhead on DB2 clients that have to wait until the database manager starts up the database to get a connection to this database.

Prerequisite

Ensure that you have SYSMAINT, SYSCTRL, or SYSADM authority.

To activate your databases after upgrade:

1. Start your database and all necessary database services with the `ACTIVATE DATABASE` command. The following example illustrates the use of this command to activate the `sample` database:

```
db2 ACTIVATE DATABASE sample
```

After this command is executed successfully your database is available for connections.

2. Review the administration notification log or the `db2diag` log files to verify that all database services are running properly and all buffer pools are activated. Address any problems that occurred during the database activation.

Remember that a database, activated by the `ACTIVATE DATABASE` command, stops only when you issue the `DEACTIVATE DATABASE` command or the `db2stop` command. If the database is activated when the first connection is established, then the database is stopped when the last connection is closed.

Managing DB2 server behavior changes

The changes in DB2 registry variables, configuration parameters, and database physical design characteristics can have an upgrade impact. Review these changes to manage the upgrade impact.

After upgrading your DB2 server, compare the values of your registry variables and configuration parameters to their values before upgrade. If you find any differences, take the time to understand them because they could alter the behavior or performance of your applications. However, consider carefully whether to disable any new functionality because it provides support for new resources needed by the database manager. You should disable new functionality only if you experience negative performance or unwanted behavior.

To manage DB2 server behavior changes:

1. Review the information about new, changed, deprecated, and discontinued registry variables, and based on the upgrade impact, choose the appropriate settings:
 - New registry variables
 - Changed registry variables
 - There are no registry variables that have been deprecated or discontinued in this release. However, if you are upgrading from DB2 Version 9.1 or earlier, consider removing registry variables that have been deprecated or discontinued in pre-Version 9.7 releases:
 - Deprecated registry variables in DB2 Version 9.5
 - Discontinued registry variables in DB2 Version 9.5
 - Deprecated registry variables in DB2 Version 9.1
 - Discontinued registry variables in DB2 Version 9.1
2. Set your DB2 global profile registry variables. The variables that you set at the global profile level, using the `db2set` command with the `-g` option, are not upgraded. The global profile variables apply to all instances pertaining to a specific DB2 copy. Therefore, after upgrading your instances, use the configuration information that you saved in the pre-upgrade tasks to restore the values of your global profile registry variables for every DB2 Version 9.7 copy.
3. Review the information about new, changed, and deprecated database manager configuration parameters, and based on the upgrade impact, choose the appropriate settings:
 - New database manager configuration parameters
 - Changed database manager configuration parameters
 - There are no database manager configuration parameters that have been deprecated or discontinued in this release. However, if you are upgrading from DB2 Version 9.1 or earlier, consider removing database manager configuration parameters that have been deprecated in pre-Version 9.7 releases:
 - Deprecated database manager configuration parameters in DB2 Version 9.5
 - Deprecated database manager configuration parameters in DB2 Version 9.1
4. Review the information about new, changed, deprecated, and discontinued database configuration parameters, and based on the upgrade impact, choose the appropriate settings:
 - New database configuration parameters
 - Changed database configuration parameters
 - There are no database manager configuration parameters that have been deprecated or discontinued in this release. However, if you are upgrading from DB2 Version 9.1 or earlier, consider removing database manager configuration parameters that have been deprecated or discontinued in pre-Version 9.7 releases:
 - Deprecated and discontinued database configuration parameters in DB2 Version 9.5
 - Deprecated and discontinued database configuration parameters in DB2 Version 9.1
5. Review the changes in database physical design characteristics and security, and based on the upgrade impact, modify database objects accordingly:
 - Physical design characteristics of databases
 - Authorities and privileges

If you change the settings of any database manager configuration parameters that are not dynamic, you might need to restart the instance so the new settings take effect.

Setting up security to manage database auditing in upgraded databases

Security administrator (SECADM) authority is now required to configure and manage database auditing using SQL statements, the SYSADM authority is not a requirement anymore. Grant SECADM authority to the users who manage the database auditing in upgraded databases.

Prerequisites

- To grant SECADM authority, you must have SECADM authority. If there is no user authorization ID with SECADM authority in your existing databases, the user upgrading the database is granted SECADM authority during upgrade.
- To run the db2audit command, you must have SYSADM authority.

Restriction

You only have to perform this procedure if you upgraded your DB2 server from DB2 Version 9.1 or DB2 UDB Version 8.

Database and instance-level auditing are separate since DB2 Version 9.5. You can configure database auditing only by using DDL statements. You can continue to use the db2audit command to configure instance auditing.

When you upgrade an instance, the audit configuration file is converted to the DB2 Version 9.7 format.

When you upgrade a database, the instance-level configuration settings for auditing are used to create an audit policy in the database. If the audit facility is enabled at the instance level, the audit policy is associated with the upgraded database to enable auditing. Otherwise, the audit policy is not associated. These actions ensure that you observe the same audit behavior on your database after upgrading to DB2 Version 9.7.

Procedure

To setup security to manage database auditing in upgraded databases:

1. Grant SECADM authority to the users who manage the audit facility by using the GRANT command. The following sample commands show how to grant SECADM authority to a user:

```
db2 CONNECT TO SAMPLE
db2 GRANT SECADM ON DATABASE TO USER <user-id>
```

2. Verify that the DB2AUDIT_CFG_MIGR audit policy was created for your databases during upgrade by querying the SYSCAT.AUDITPOLICIES system catalog view. The following sample query determines whether this audit policy was created:

```
db2 "SELECT * FROM SYSCAT.AUDITPOLICIES A
      WHERE A.AUDITPOLICYNAME = 'DB2AUDIT_CFG_MIGR'"
```

If the DB2AUDIT_CFG_MIGR audit policy was not created during upgrade, create it using the CREATE AUDIT POLICY statement.

3. Verify that the DB2AUDIT_CFG_MIGR audit policy was associated with the upgraded databases by querying the SYSCAT.AUDITUSE system catalog view. The following sample query determines whether the audit policy was associated with the SAMPLE database:

```
db2 "SELECT * FROM SYSCAT.AUDITUSE U
      WHERE U.OBJECTNAME = 'SAMPLE'"
```

If the database upgrade fails to associate the DB2AUDIT_CFG_MIGR audit policy to the upgraded databases, use the AUDIT statement to associate this policy to your database.

4. Optional: If you want to extract all audit records from the original audit log file that you had before upgrade and place the contents into a new audit log file in the new default location, run the db2audit command with the **extract** parameter. The original audit log file from a previous release remains in the same location that it was in prior to database upgrade.

The default location for the audit logs in DB2 Version 9.7 is:

- INSTHOME/sql1lib/security/auditdata on Linux and UNIX operating systems, where INSTHOME is the instance home directory.
- INSTHOME\security\auditdata on Windows operating systems

where INSTHOME is the instance home directory that stores user data and instance directories.

Now, you can use the following DDL statements to manage database auditing:

- CREATE AUDIT POLICY
- ALTER AUDIT POLICY
- AUDIT

Rebinding packages in upgraded databases

During database upgrade, all packages for user applications and routines are marked as invalid. You must rebind invalidated packages to take advantage of changes in the DB2 server and new statistics.

Prerequisite

Ensure that you have DBADM authority.

Restriction

This procedure only applies to Embedded SQL database applications programmed in C, C++, COBOL, FORTRAN, and REXX.

About this task

Packages will be implicitly rebound the first time an application uses them after upgrading your database. To eliminate this overhead, you can rebind invalid packages by running the REBIND command or the db2rbind command after the upgrade process is complete. You must explicitly rebind inoperative packages.

Procedure

To rebind packages in upgraded databases:

1. Log on as a user with DBADM authority.
2. Rebind all invalid packages in each database by running the db2rbind command:

```
db2rbind database-name -l logfile all -u userid -p password
```

The all clause rebinds valid and invalid packages. Review the *logfile* file and address any issues rebinding any database packages.

3. Verify that your DB2 server upgrade was successful. Test your applications and tools to ensure the server is working as expected.

After you have rebound all your database packages, you will automatically be able to take advantage of optimizer improvements. Refer to Chapter 22, “Upgrade essentials for database applications,” on page 145 for details on the optimizer improvements available in this release.

Migrating explain tables

If you must maintain explain table information that you previously gathered in your existing DB2 copies from previous releases, migrate your explain tables to DB2 Version 9.7.

You can manually migrate your explain tables after you upgrade your database, or you can later re-create the explain tables and gather new information.

If you plan to collect sections actuals, you must migrate your explain tables or re-create them in Version 9.7.

Prerequisite

Ensure that you have DBADM authority. Additional authorization details are available in the *Command Reference*.

Procedure

To migrate the explain tables:

1. Run the db2exmig command:

```
db2exmig -d dbname -e explain_schema [-u userid password]
```

where:

- *dbname* represents the database name. This parameter is required.
- *explain_schema* represents the schema name of the explain tables to be migrated. This parameter is required.
- *userid* and *password* represent the current user’s ID and password. These parameters are optional.

The explain tables belonging to the user ID that is running db2exmig, or that is used to connect to the database, are migrated. The explain tables migration tool renames the existing explain tables, creates a new set of tables using the EXPLAIN.DDL file, and copies the contents of the existing explain tables to the new tables. Finally, it drops the existing explain tables. The db2exmig command preserves any user added columns on the explain tables.

2. Use Visual Explain to see a graphical display of a query access plan or the db2expln command to see the access plan information in the migrated explain tables.

Converting XML storage objects to the Version 9.7 format

If you have tables with XML columns that you created in a pre-Version 9.7 release and want to use certain new functions, you must convert the XML storage objects to the Version 9.7 format by re-creating the tables.

Before you begin

- Ensure that you have authorization to create tables and access to the existing tables.

About this task

The following new functions require the XML storage object to be in the Version 9.7 format:

- Row compression on tables with XML columns
- Collection of statistics to estimate the inline length for XML columns
- Upgrade from a single-partition database environment to a multi-partition database environment

Procedure

To convert XML storage objects to the Version 9.7 format :

1. Generate a list of all the tables with XML columns with XML storage object in pre-Version 9.7 format by issuing the following query:

```
SELECT TABSCHEMA, TABNAME FROM SYSIBMADM.ADMINTABINFO WHERE XML_RECORD_TYPE=1 GROUP BY (TABSCHEMA, TABNAME)
```

Use the GROUP BY clause for partitioned database environments.
2. Re-create the tables in the list that generated in the previous step by using one of the following methods:
 - Re-create tables using the ADMIN_MOVE_TABLE procedure. See “Moving tables using the ADMIN_MOVE_TABLE procedure” in Data Movement Utilities Guide and Reference for details and restrictions. This method is best suited for a small number of tables and re-creates tables while the data remains online.
 - Perform an offline procedure to re-create the tables and objects dependent on those tables. This method is best suited for a large number of tables.

For example, you can have a procedure that uses the db2move command with the -co COPY action. See “Examples of schema copy using the db2move utility” in *Database Administration Concepts and Configuration Reference*.

Another example of an offline procedure involves creating a copy of the table definition using the db2look command and then performing a load from cursor.

Ensuring system temporary table spaces page sizes meet requirements

The use of larger record identifiers (RID) increases the row size in your result sets from queries or positioned updates. If the row size in your result sets is close to the maximum row length limit for your existing system temporary table spaces, you might have to create a system temporary table space with a larger page size.

Prerequisite

Ensure that you have SYSCTRL or SYSADM authority to create a system temporary table space if required.

Procedure

To ensure that the maximum page size of your system temporary table space is large enough for your queries or positioned updates:

1. Determine the maximum row size in your result sets from queries or positioned updates. Monitor your queries or calculate the maximum row size using the DDL statement that you used to create your tables.
2. Determine the page size for each of your system temporary table spaces and the page size of the table spaces where the tables referenced in the queries or updates were created by issuing the following query:

```
db2 "SELECT CHAR(TBSP_NAME,20) TBSP_NAME, TBSP_CONTENT_TYPE, TBSP_PAGE_SIZE
    FROM SYSIBMADM.SNAPTbsp"
```

TBSP_NAME	TBSP_CONTENT_TYPE	TBSP_PAGE_SIZE
SYSCATSPACE	ANY	8192
TEMPSPACE1	SYSTEMP	8192
USERSPACE1	LARGE	8192
IBMDB2SAMPLEREL	LARGE	8192
SYSTOOLSPACE	LARGE	8192
SYSTOOLSTMPSPACE	USRTEMP	8192

6 record(s) selected.

You can identify the system temporary table spaces in the output by looking for table spaces that have the TBSP_CONTENT_TYPE column with a value of SYSTEMP.

If you are upgrading from Version 8.1, use the following command:

```
db2 LIST TABLESPACES SHOW DETAIL
```

3. Check whether the largest row size in your result sets fits into your system temporary table space page size:

```
maximum_row_size > maximum_row_length - 8 bytes (structure overhead in
                                                    single partition)
maximum_row_size > maximum_row_length - 16 bytes (structure overhead in DPF)
```

where maximum_row_size is the maximum row size for your result sets, and maximum_row_length is the maximum length allowed based on the largest page size of all of your system temporary table spaces. Review the "SQL and XML limits" in *SQL Reference, Volume 1* to determine the maximum row length per table space page size.

If the maximum row size is less than the calculated value then your queries will run in the same manner that they did in DB2 UDB Version 8, and you do not have to continue with this task.

4. Create a system temporary table space that is at least one page size larger than the table space page size where the tables were created if you do not already have a system temporary table with that page size. For example, on the Windows operating systems, if you created your table in a table space with 8 KB page size, create the additional system temporary table space using an 16 KB page size:

```
db2 CREATE SYSTEM TEMPORARY TABLESPACE tmp_tbsp
    PAGESIZE 16K
    MANAGED BY SYSTEM
    USING ('d:\tmp_tbsp','e:\tmp_tbsp')
```


If your table space page size is 32 KB, you can reduce the information that you are selecting in your queries or split the queries to fit in the system temporary table space page. For example, if you select all columns from a table, you can instead select only the columns that you really required or a substring of certain columns to avoid exceeding the page size limitation.

Re-creating write-to-table event monitors

If you created write-to-table event monitors in a pre-Version 9.7 DB2 copy, re-create your write-to-table event monitors and target tables so that you can start collecting data for new or changed Version 9.7 monitor elements after you upgrade your databases.

Version 9.7 target tables now include new columns for new monitor elements, changed column data types, or longer column lengths for existing monitor elements. Activating existing write-to-table event monitors after database upgrade results in lost data because data cannot be collected in your existing target tables.

Prerequisite

Ensure that you have DBADM authority.

To re-create write-to-table event monitors:

1. Identify which are the target tables for each write-to-table event monitor that you created in a pre-Version 9.7 DB2 copy by querying the SYSCAT.EVENTTABLES view as shown in the following example:

```
SELECT TABSCHEMA, TABNAME FROM SYSCAT.EVENTTABLES
WHERE EVMONNAME = 'write-to-table-event-monitor-name'
```

2. Rename or drop the existing target tables that you identified in the previous step by issuing one of the following statements for each target table:

```
RENAME TABLE target-table-name TO new-target-table-name
or
DROP TABLE target-table-name
```

You only need to rename the target tables if you want to keep the existing data that you collected.

3. Drop the write-to-table event monitors by issuing the following statement for each event monitor:

```
DROP EVENT MONITOR write-to-table-event-monitor-name
```

4. Create your write-to-table event monitors.
5. If you created your write-to-table event monitors without the **AUTOSTART** command parameter, activate the write-to-table event monitor to start collecting data by issuing the SET EVENT MONITOR STATE statement as shown in the following example:

```
SET EVENT MONITOR write-to-table-event-monitor-name 1
```

If you have applications that query target tables, you need to modify your applications to manage the changes.

Verifying upgrade of DB2 servers

When the upgrade of your DB2 server is complete, it is a good measure to run some tests on the new upgraded environment to verify that the DB2 server is working as expected. These tests can consist of batch programs that you usually run against the DB2 server or any programs or scripts that you run for benchmarks.

If you have DB2 command scripts with SQL statements, you can use the `db2batch` benchmark tool command to execute the statements in these scripts, and gather performance information details and statistics such as CPU time and elapsed time. This tool can work in both a single partition database and in a multiple partition database.

Prerequisite

Ensure that you have the same authority level that is required to run the SQL statements in your script.

Procedure

To verify that your DB2 server upgrade was successful:

1. Log on to the DB2 server as a user with the same authority level that is required to run the SQL statements in the script.
2. Prepare a script with SQL statements that you frequently run. If you installed the sample files, you can also run any of the sample CLP scripts.
3. Run your script using the `db2batch` command. The following example shows you how to run this tool with the `testdata.db2` sample script:

```
cd samplefile-dir-clp
db2batch -d sample -f testdata.db2 -o r 0 p 3
```

where *samplefile-dir-clp* is `DB2DIR/samples/clp` on Linux and UNIX and `DB2DIR\samples\clp` on Windows, `DB2DIR` represents the location for your DB2 Version 9.7 copy, `sample` is the database name, and the option `-o r 0 p 3` indicates to print 0 fetched rows to the output and to report elapsed time, CPU time, and summary of monitoring information for each statement in the `testdata.db2` script.

The following text is an extract of the summary table output generated by the command in the previous example:

Summary Table:

Type	Number	Total Time	Min Time	Max Time	Arithmetic Mean	Geometric Mean
Statement	1	0.281284	0.281284	0.281284	0.281284	0.281284
Statement	2	0.073158	0.073158	0.073158	0.073158	0.073158
Statement	3	0.000823	0.000823	0.000823	0.000823	0.000823
Statement	4	0.155366	0.155366	0.155366	0.155366	0.155366

```
* Total Entries:          4
* Total Time:             0.510630 seconds
* Minimum Time:          0.000823 seconds
* Maximum Time:          0.281284 seconds
* Arithmetic Mean Time:  0.127658 seconds
* Geometric Mean Time:   0.040271 seconds
```

Chapter 10. Adopting new Version 9.7 functionality in upgraded databases

After upgrading your DB2 server, enhance the functionality and improve the performance of your upgraded databases by adopting new Version 9.7 functionality.

Prerequisite

You must upgrade your DB2 server to Version 9.7.

Procedure

Perform any of the following steps to adopt the specified Version 9.7 functionality in your upgraded DB2 environment:

- **Enable automatic storage in existing databases** by issuing the following statement:

```
ALTER DATABASE database-name ADD STORAGE ON storage-location
```

After enabling your databases for automatic storage, enable your existing DMS table spaces for automatic storage. One way to do this enablement is to keep existing containers intact and use automatic storage for future growth and reduction operations by issuing the ALTER TABLESPACE statement:

```
ALTER TABLESPACE tablespace-name MANAGED BY AUTOMATIC STORAGE
```

If you want to convert existing containers in your table spaces to use automatic storage, perform a redirected restore to re-create existing DMS table spaces as automatic storage table spaces. See “Existing databases and table spaces can now use automatic storage” in *What’s New for DB2 Version 9.7*.

In addition, you can now drop storage paths from an automatic storage database as well as add them. After altering the database storage paths, you can optionally rebalance the data in the automatic storage table spaces to better utilize data striping and increase I/O throughput. The following example shows how to rebalance an automatic storage table space:

```
ALTER TABLESPACE tablespace-name REBALANCE
```

The following SQL statement generates a list of all the regular and large automatic storage table spaces for the currently connected database:

```
SELECT TBSP_NAME  
FROM SYSIBMADM.SNAPTbsp  
WHERE TBSP_USING_AUTO_STORAGE = 1 AND TBSP_CONTENT_TYPE IN ('ANY', 'LARGE')  
ORDER BY TBSP_ID
```

All of these enhancements provide greater control over your automatic storage databases and table spaces.

- If you are using DMS table spaces in databases with or without automatic storage enabled, **start using new DMS table spaces created in Version 9.7 or migrate existing DMS table spaces**. Newly created DMS table spaces have reclaimable storage enabled by default. You can trigger the extent movement operation to relocate the maximum number of extents in them and reduce the high water mark by using the following commands:
 - For automatic storage DMS table spaces, use the ALTER TABLESPACE statement with the REDUCE clause.

- For non-automatic storage DMS table spaces, use the ALTER TABLESPACE statement with the LOWER HIGH WATER MARK. Then issue the ALTER TABLESPACE statement with the REDUCE clause to alter the size of the containers.

Existing DMS table spaces that you created in pre-Version 9.7 releases have reclaimable storage disabled. They can coexist with DMS table spaces that use reclaimable storage. To enable reclaimable storage in your existing DMS table spaces, you need migrate them using in one of the following methods:

- Re-create DMS table spaces.
- Create new DMS table spaces and call the new SYSPROC.ADMIN_MOVE_TABLE procedure to move the data to a new DMS table space with reclaimable storage enabled while the data remains online and available for access.

- If you have disk space limitations on the directory indicated by the **diagpath** configuration parameter, **control the total size of DB2 diagnostic (db2diag) and administration notification log files** by setting the **diagsize** database manager configuration parameter to a value and restart the instance.

After you restart the instance, all the messages are written to db2diag rotating log files (db2diag.N.log) and rotating administration notification logs (*instance.N.nfy*) whose total size is limited by the value in **diagsize**. To avoid losing information because of the log file rotation, you must specify an adequate value between 1 GB and the amount of free space in the directory indicated by **diagpath** minus 5 GB.

- **Use SYSTEM sampling to reduce the cost of collecting statistics on statistical views** as shown in the following example:

```
RUNSTATS ON TABLE view-name
      WITH DISTRIBUTION TABLESAMPLE SYSTEM (sampling-rate)
```

When you issue the RUNSTATS with the TABLESAMPLE SYSTEM option on statistical views, you will not receive the SQL20288N error code. If the RUNSTATS cannot use the SYSTEM sampling, it collects statistics for the view using BERNOULLI sampling with the same sampling rate and seed as specified in the command and returns the SQL2317W warning message to indicate the change in sampling style.

You can continue to use BERNOULLI sampling as in previous releases. DB2 Version 9.7 now provides improvements on the performance of BERNOULLI sampling.

- **Use access plan reuse and statement optimization guidelines enhancements to influence the optimizer** and obtain consistent query execution plans for the same query. For static statements, you indicate to the query compiler to reuse existing access plans for the statements in a package by issuing the following statement:

```
ALTER PACKAGE schema-name.package-id ACCESS PLAN REUSE YES
```

Also, you can indicate to the query compiler to reuse existing access plans for the statements in a package by issuing the BIND statement as shown in the following example:

```
BIND filename ACTION REPLACE APREUSE YES
```

- **Enable data row compression on tables that use replication.** You can now enable data row compression on tables that use data replication. Ensure that your server can manage the additional resources required for data row compression and that applications are modified to receive more data if you are using the db2ReadLog API in your application. See “Upgrade impact from DB2 API changes” on page 151 to learn how to manage the additional resources required for your application.

- If you frequently delete data from MDC tables, use the new RECLAIM EXTENTS ONLY option with **automatic reorganization** to free empty extents in MDC tables to DMS table spaces so that these extents can be reused by other tables or database objects by performing the following steps:
 1. Set the **auto_maint**, **auto_tbl_maint** and **auto_reorg** database configuration parameters to ON.
 2. Configure an automated maintenance policy for reclaiming extents by calling the AUTOMAINT_SET_POLICY or AUTOMAINT_SET_POLICYFILE system-defined procedure using maintenance type AUTO_REORG and an the maintenance policy specification in XML as parameters. Refer to the DB2AutoReorgPolicySample.xml sample file in DB2 samples directory for details on how to specify a maintenance policy in XML.

Reclaiming empty extents allows you to improve performance for prefetching data because there will be less extents to bring from disk to memory. However, if there are no empty extents for reuse when you insert data, new extents need to be reallocated. Although you can manually reclaim empty extents, using automatic reorganization allows the database manager to determine when this action should take place to obtain performance gains.

- **Use partitioned indexes in partitioned tables to optimize performance** when you **add a new range or remove an existing range from a partitioned table** by issuing the ALTER TABLE statement with the ATTACH PARTITION or DETACH PARTITION clauses. Index maintenance is not required when you issue the SET INTEGRITY statement on partitioned tables that use partitioned indexes. To convert existing nonpartitioned indexes in a partitioned table, you must drop and re-create them as partitioned indexes.

In addition, partitioned index definitions in the table that you plan to attach and the partitioned table should match before you issue the ALTER TABLE statement with the ATTACH PARTITION to obtain this performance gain and avoid index maintenance.

See “Migrating existing indexes to partitioned indexes” in *Partitioning and Clustering Guide* for details.

- **Use the pureXML feature in partitioned database environments** to benefit from integrated XML data storage by:
 - Creating tables with one or more XML columns with the CREATE TABLE statement and the DISTRIBUTE BY clause.
 - Adding one or more XML columns to existing tables that have a distribution key with the ALTER TABLE statement and the ADD COLUMN clause.
 - Adding a distribution key to existing tables that have one or more XML columns with the ALTER TABLE statement and the DISTRIBUTE BY clause. First re-create these tables to convert the XML storage object to the new Version 9.7 format. See Re-creating tables with XML columns for details.
- **Use the pureXML feature in partitioned tables** to benefit from integrated XML data storage in partitioned tables or **use table partitioning on tables storing XML data** to benefit from partition elimination and easy roll-in and roll-out capabilities in table partitioning by:
 - Creating a partitioned table with one or more XML columns with the CREATE TABLE statement.
 - Adding an XML column to an existing partitioned table with the ALTER TABLE statement and the ADD COLUMN clause.
 - Migrate an existing table with one or more XML columns to a partitioned table.

- Rolling in partitioned table data with the ALTER TABLE statement and the ATTACH PARTITION clause.
- Rolling out partitioned table data with the ALTER TABLE statement and the DETACH PARTITION clause.
- **Use the pureXML feature in MDC tables** to benefit from integrated XML data storage or **use MDC on tables storing XML data** to benefit from better query performance by:
 - Creating tables with one or more XML columns with the CREATE TABLE statement and the ORGANIZE BY clause.
 - Adding one or more XML columns to existing MDC tables with the ALTER TABLE statement and the ADD COLUMN clause.
 - Migrating an existing table with one or more XML columns to an MDC table.
- **Enable the read capability on HADR Standby databases** to use them for read-only workloads and disaster recovery. In Version 9.7, you can connect to HADR Standby databases to run read-only workloads by setting the **DB2_HADR_ROS** registry variable to ON.

If you upgraded your DB2 server from DB2 Version 9.1 or earlier, adopt functionality introduced in pre-Version 9.7 releases in your upgraded DB2 environment. See the following topics for details:

- Enabling new DB2 Version 9.5 functionality in migrated databases in the *Migration Guide (Version 9.5)*.
- Enabling new DB2 Version 9.1 functionality in migrated databases in the *Migration Guide (Version 9.1)*.

Chapter 11. Migrating DB2 functionality to DB2 product features

Migrating DB2 functionality to specific DB2 product features requires that you understand how the product feature works and how to implement equivalent functionality using a product feature.

The following migration tasks provides guidelines on how to implement workload management and XML data store features:

- “Migrating from DB2 Governor to DB2 workload manager”
- “Migrating from Query Patroller to DB2 workload manager” on page 113
- “Migrating from XML Extender to pureXML” on page 116
- Migrating from Net Search Extender to DB2 Text Search in *DB2 Text Search Guide*

Migrating from DB2 Governor to DB2 workload manager

Migrating from DB2 Governor to DB2 workload manager (WLM) requires that you set up your database for coexistence of DB2 Governor and DB2 WLM, re-examine your goals, and implement a workload management solution.

Prerequisites

- Review your overall approach to workload management in light of the DB2 WLM capabilities provided to determine the best implementation. Refer to Workload management roadmap for a number of resources that are available to get you started with DB2 WLM, including “Best Practices: DB2 Workload Management.”
- Review the Chapter 11. Query Patroller and DB2 Governor in *DB2 Workload Manager for Linux, UNIX, and Windows* available at <http://www.redbooks.ibm.com/redpieces/abstracts/sg247524.html> for details about migration from DB2 Governor to DB2 WLM.
- If your existing workload management solution includes Query Patroller, also review “Migrating from Query Patroller to DB2 workload manager” on page 113.

About this task

There is no tool to automatically migrate your Governor configuration to DB2 WLM because the type of controls and mechanisms available are different between the two. When a query is running, the Governor watches for certain thresholds during the query execution which can trigger certain events. In DB2 WLM, a number of control mechanisms are available, in addition to the control of thresholds, which enable you to approach the same workload management problems in different but more effective ways.

This task provides guidelines to implement an efficient workload management solution and assist users migrating from DB2 Governor to DB2 WLM.

Important: With the new workload management features introduced in DB2 Version 9.5, the DB2 governor utility has been deprecated in Version 9.7 and might

be removed in a future release. For more information, see the “DB2 Governor and Query Patroller have been deprecated” topic in the *What’s New for DB2 Version 9.7* book.

Procedure

To migrate from DB2 Governor to DB2 WLM:

1. Upgrade the data server where the Governor is installed to DB2 Version 9.7 so that you have an environment where DB2 WLM and the Governor can coexist. Use one of the following tasks:

- Upgrading DB2 servers (Windows)
- Upgrading DB2 servers (Linux and UNIX)

After the upgrade, there is a default workload created to identify all the user database activities and map them to a default user service class which is the execution environment. The Governor can only intercept and manage queries assigned and executing in the default user service class. If there are workloads defined to route user activities to service classes other than the default user service class, the Governor cannot manage those activities.

2. Limit the use of DB2 WLM to control work in the default user service class to avoid potential conflicts between the Governor and DB2 WLM.
3. Re-examine your workload management goals. Understanding them is critical to implement a workload management solution.
4. Identify the work that runs on the data server and maps to your goals. Take advantage of the additional identification options at your disposal in DB2 WLM.
5. Manage the work that you identified by assigning resources and imposing controls to meet your goal metrics. Using any of the following approaches might result in a more simple and effective implementation:
 - Use DB2 service classes to separate and isolate competing workloads from each other or group database activities. Then change the agent, buffer pool, and prefetch priority options each service class receives to affect their individual response times. Try this approach first instead of creating concurrency thresholds.
 - Take note of the AUTHID and APPLNAME parameter values in the Governor control file and create a workload specifying the SESSION_USER and APPLNAME connection attributes using the AUTHID and APPLNAME parameter values.
 - If you cannot separate work by its source using workloads, map all incoming work to a common service super class and use a DB2 work action set to separate work by different characteristics and assign it to different service sub classes. At this point, manipulate the resources available to each service class to achieve your goals.
 - If you do not achieve the desired results by setting the priority options each service class receives alone, selectively apply other features of DB2 WLM as needed until you achieve your goals, such as the application of DB2 thresholds.
 - When you use DB2 thresholds, ensure that the threshold violations event monitor is created and activated; otherwise, you will not know when and what thresholds are being violated.
 - If you create thresholds to map to the same workloads the Governor was watching, consider all the thresholds available in DB2 WLM. Some of the DB2 Governor reactive rules will find a direct functional equivalent in DB2

workload management thresholds, like those controlling maximum execution time, the maximum number of rows returned, or the maximum connection idle time. Others are unique to workload management or to the DB2 Governor and require you to rethink your approach to controlling work in current workload management terms. Note that DB2 Governor rules can apply to already running queries, whereas changes to DB2 WLM thresholds apply only to new queries.

Consider all the different threshold actions available in DB2 WLM. You can choose a more forgiving action when a resource threshold is exceeded than ending the activity, such as letting the threshold continue execution or remapping it to a service subclass with different resource controls, and you can use the information logged in the threshold violations event monitor to further investigate the activity.

- For the rowsdel limit, you can create a threshold using the SQLROWSRETURNED condition to indicate what action should be taken when the limit of number of data rows returned to the application is exceeded.
 - For the rowsread limit, you can create a threshold using the SQLROWSREAD or SQLROWSREADINSC condition to indicate what action should be taken when the limit of number of data rows read during query evaluation is exceeded.
 - For the cpu limit, you can create a threshold using the CPUTIME or CPUTIMEINSC condition to indicate what action should be taken when the limit for the amount of combined user and system CPU time consumed by an activity is exceeded.
 - For the idle limit, you can create a threshold using the CONNECTIONIDLETIME condition to indicate what action should be taken when the maximum connection idle time is exceeded.
 - For the uowtime limit, you can create a threshold using the UOWTOTALTIME condition to indicate the length of time a unit of work is allowed to run.
 - If you are using connection pooling, DB2 WLM has the client attributes available for proper identification and management of queries. The application at the middle tier could either call the sqleseti API or WLM_SET_CLIENT_INFO procedure to set one of the client attributes before it issues the SQL.
 - If your data server runs on the AIX operating system, consider using AIX WLM for a more granular control of processor resource.
6. Monitor options to ensure that you are meeting your goals.

Migrating from Query Patroller to DB2 workload manager

Migrating from Query Patroller to DB2 workload management (WLM) requires that you set up your database for coexistence of Query Patroller and DB2 WLM, re-examine your goals, and implement a workload management solution.

Prerequisites

- Ensure that you have root access.
- Ensure that you meet the installation requirements for Query Patroller. See “Installation requirements for DB2 database products” in *Installing DB2 Servers*.
- Review your overall approach to workload management in light of the DB2 WLM capabilities provided to determine the best implementation. Refer to

Workload management roadmap for a number of resources that are available to get you started with DB2 WLM, including “Best Practices: DB2 Workload Management.”

- Review the Chapter 11. Query Patroller and DB2 Governor in *DB2 Workload Manager for Linux, UNIX, and Windows* available at <http://www.redbooks.ibm.com/redpieces/abstracts/sg247524.html> for details about migration from Query Patroller to DB2 WLM
- If your existing workload management solution includes DB2 Governor, review also “Migrating from DB2 Governor to DB2 workload manager” on page 111.

Restriction

- There is no equivalent in DB2 WLM for the bypass options in Query Patroller.

About this task

This task provides guidelines to implement an efficient workload management solution and assist users migrating from Query Patroller to DB2 WLM.

Important: With the new workload management features introduced in DB2 Version 9.5, the DB2 governor utility has been deprecated in Version 9.7 and might be removed in a future release. For more information, see the “DB2 Governor and Query Patroller have been deprecated” topic in the *What’s New for DB2 Version 9.7* book.

Procedure

To migrate from Query Patroller to DB2 WLM:

1. Upgrade the data server where Query Patroller is installed to DB2 Version 9.7 so that you have an environment where DB2 WLM and Query Patroller can coexist so that you can reduce the migration impact and risk. Use one of the following tasks:
 - Upgrading DB2 servers (Windows)
 - Upgrading DB2 servers (Linux and UNIX)After the upgrade, there is a default workload created to identify all the user database activities and map them to a default user service class which is the execution environment. Query Patroller can only intercept and manage queries assigned and executing in the default user service class. If there are workloads defined to route user activities to service classes other than the default user service class, Query Patroller cannot be able to manage those activities.
2. Limit the use of DB2 WLM to control work in the default user service class to avoid potential conflicts between Query Patroller and DB2 WLM.
3. Re-examine your workload management goals. Understanding them is critical to implement a workload management solution.
4. Identify the work that runs on the data server and maps to your goals. Take advantage of the additional identification options at your disposal in DB2 WLM.
5. Manage the work that you identified by assigning resources and imposing controls to meet your goal metrics. You can map Query Patroller management functions to a DB2 WLM equivalent using one of the following approaches:
 - a. If you have an effective Query Patroller implementation, you can map Query Patroller management functions to WLM by using the `qpwlmmig.pl` script provided in the `INSTHOME/sql1lib/samples/perl` (Linux and UNIX) or `DB2PATH\samples\perl` (Windows) directory. This Perl script allows you to

generate a DB2 script containing DDL statements to create the database objects that best emulate the behavior of your Query Patroller implementation using WLM. Modify the script if necessary and run it to create an initial WLM setup.

- b. You can moderate the transition between Query Patroller and DB2 WLM by gradually creating service classes. Use any of the following approaches for a simple and effective implementation:
 - Use DB2 service classes to separate and isolate competing workloads from each other or to group database activities. Then change the agent, buffer pool, and prefetch priority options each service class receives to affect their individual response times. Try this approach first instead of creating concurrency thresholds.
 - If you cannot separate work by its source using workloads, map all incoming work to a common service super class and use a DB2 work action set to separate work by different characteristics and assign it to different service sub classes. At this point, manipulate the resources available to each service class to achieve your goals.
 - If you do not achieve the desired results by setting the priority options each service class receives alone, selectively apply other features of DB2 WLM as needed until you achieve your goals such as the application of DB2 thresholds, including concurrency thresholds.
 - When you use DB2 thresholds, ensure that the threshold violations event monitor is created and activated; otherwise, you will not know when and what thresholds are being violated.
 - To implement query classes, you can use work classes and work action sets to identify the DML work and then set up thresholds on the work action set based on the range of query cost.
 - Identify large queries and consider the option of collecting detailed information, or remapping these queries to service subclasses with different resource controls, before taking the more severe action of stopping execution. When collecting information for later analysis, you can limit the scope of what you collect to a specific service class.
 - If you are using connection pooling, DB2 WLM has the client attributes available for proper identification and management of queries. The application at the middle tier could either call the `sqleseti` API or `WLM_SET_CLIENT_INFO` procedure to set one of the client attributes before it issues the SQL.
 - If your data server runs on the AIX operating system, consider using AIX WLM for a more granular control of processor resource.
 - If your data server runs on the Linux operating system, consider using WLM for control of processor resource. Linux kernel version 2.6.26 or later is required.
6. If you used historical analysis functions in Query Patroller, Review Exercise 10: Generating historical data and reports to learn how to use the DB2 WLM Historical Analysis Tool sample. DB2 samples include a set of Perl scripts that provides functionality similar to the Query Patroller historical analysis functions using information captured by the DB2 WLM activity event monitor. You can modify the scripts to produce additional historical analysis reports to suit your needs.
7. Monitor options to ensure that you are meeting your goals.

Migrating from XML Extender to pureXML

Migrate your database applications that use XML Extender to use the pureXML feature so that they can run in DB2 Version 9.7. XML Extender is discontinued in DB2 Version 9.7.

DB2 Version 9.7 supports native XML data storage. This support includes a new XML type, XQuery, XML indexes, and a series of SQL/XML functions.

Prerequisite

A pre-Version 9.7 DB2 server where XML Extender is installed.

Procedure

To migrate your applications from the XML Extender to the new native XML storage support:

1. Upgrade your DB2 server where XML Extender is installed to DB2 Version 9.7.
2. Optional: Convert your databases to Unicode databases. See "Converting non-Unicode databases to Unicode" in *Globalization Guide*. Although XML type support is provided for non-Unicode databases in DB2 Version 9.7, using a Unicode database eliminates the overhead of character conversion from the database code page to the Unicode code page and preserves the data integrity because there is no character conversion.
3. Add XML type columns to your tables. Use the ALTER TABLE command:

```
db2 ALTER TABLE table_name
      ADD column_name XML [NOT NULL]
```

You only need to perform this step if you stored entire XML documents in its native format in a column of data type CLOB, VARCHAR, XMLCLOB, XMLVARCHAR, or XMLFILE.

4. Register your XML schemas in the XML Schema repository (XSR). See "Registering and enabling XML schemas for decomposition" in *pureXML Guide*.
5. Import XML documents into the table with the new XML data type column.
6. Convert your application to use annotated XML schema decomposition to store content from XML documents in table columns, and the new SQL/XML functions to construct or publish XML using the new XML data type.

Details on all these migration steps and examples of application migration are available in the XML application migration series at http://www.ibm.com/developerworks/views/db2/libraryview.jsp?search_by=viper+migration+series.

Chapter 12. Reversing DB2 server upgrade

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.

Prerequisites

- Ensure that you have SYSADM authority, as well as root on Linux and UNIX operating systems or Local Administrator authority on Windows operating systems.
- Perform the following steps before upgrading your DB2 server:
 - Review upgrade recommendations and disk space requirements.
 - Take an offline full backup of all databases that you are going to upgrade.
 - Back up all database manager configuration parameter values for each instance and all database configuration parameter values for each database.
 - Perform other pre-upgrade tasks that apply to your environment.
- Keep your existing pre-Version 9.7 DB2 UDB Version copy during upgrade of your DB2 server. To do this, select the **Install New** option to create a new copy when installing DB2 Version 9.7. Do not select the **Work with an existing** option and then choose a pre-Version 9.7 copy with the **upgrade** action that is available on Windows operating systems.
- Keep all the S*.MIG files in the active log path in case you want to rollforward through these log files after reversing the upgrade. For recoverable databases, the UPGRADE DATABASE command renames log files in the active log path with the extension .MIG.

Restrictions

- This procedure applies only to DB2 server upgrade. It does not include DB2 clients.
- In partitioned database environments you must perform this procedure on all participating database partition servers. If you have several database partitions on a partition server, execute tasks at the database level, such as backup and restore, on each database partition.
- Additional upgrade restrictions apply. Review the complete list.

Procedure

To reverse a DB2 server upgrade, you need to perform the following steps:

1. Log on to the DB2 server as a user with SYSADM authority.
2. Drop all databases in DB2 Version 9.7 by running the DROP DATABASE command.
3. Log on to the DB2 server as root on Linux and UNIX operating systems or a user with Local Administrator authority on Windows operating systems.

4. Drop your DB2 Version 9.7 instances by running the db2idrop command. This command does not remove the database files; you need to drop your databases before dropping your instances.
5. If you upgraded your pre-Version 9.7 instances to DB2 Version 9.7, re-create your instances in the pre-Version 9.7 by running the db2icrt. Then restore the database manager configuration parameter values for each instance using the UPDATE DATABASE MANAGER CONFIGURATION command.
6. For each pre-Version 9.7 instance, log on to the DB2 server as the instance owner and restore your upgraded databases from a pre-Version 9.7 offline full backup by running the RESTORE DATABASE command. You cannot upgrade your databases from DB2 Version 9.7 to pre-Version 9.7 release.

If you recreated the instances using the same instance owner they had prior to upgrade and you did not upgrade a database to a DB2 Version 9.7 instance, the database is still in pre-Version 9.7 release and you can access it by just re-cataloging it.

7. If you have recoverable databases and you want to rollforward through the log files you had before the upgrade, rename all the S*.MIG files in the active log path using the .LOG extension and issue the ROLLFORWARD DATABASE command as shown in the following example on Windows operating system:

```
cd E:\DB2_01\NODE0000\SQL00001\SQLLOGDIR
dir S*.MIG
...
25/02/2008 10:04 AM          12,288 S0000000.MIG
25/02/2008 10:10 AM          12,288 S0000001.MIG
25/02/2008 09:59 AM      4,104,192 S0000002.MIG
25/02/2008 10:10 AM      4,104,192 S0000003.MIG
25/02/2008 10:19 AM      4,104,192 S0000004.MIG
          5 File(s)      12,337,152 bytes
          2 Dir(s)      4,681,842,688 bytes free

rename S*.MIG S*.LOG
dir S*.LOG
...
25/02/2008 10:04 AM          12,288 S0000000.LOG
25/02/2008 10:10 AM          12,288 S0000001.LOG
25/02/2008 09:59 AM      4,104,192 S0000002.LOG
25/02/2008 10:10 AM      4,104,192 S0000003.LOG
25/02/2008 10:19 AM      4,104,192 S0000004.LOG
          5 File(s)      12,337,152 bytes
          2 Dir(s)      4,681,842,688 bytes free

db2 ROLLFORWARD DB sample TO END OF LOGS AND STOP
```

Part 3. Upgrading clients

This part of the book contains the following chapters:

- Chapter 13, “Clients upgrade,” on page 121
- Chapter 14, “Upgrade essentials for clients,” on page 123
- Chapter 15, “Pre-upgrade tasks for clients,” on page 127
- Chapter 16, “Upgrading to Data Server Client (Windows),” on page 129
- Chapter 17, “Upgrading to Data Server Runtime Client (Windows),” on page 131
- Chapter 18, “Upgrading clients (Linux and UNIX),” on page 133
- Chapter 20, “Post-upgrade tasks for clients,” on page 137

Chapter 13. Clients upgrade

Upgrading to DB2 Version 9.7 might require the upgrade of your clients.

Upgrading a client involves installing a Version 9.7 client copy and then upgrading the client instance. A client instance allows you to connect your application to a database and keeps the information about your client configuration, your cataloged nodes, and your cataloged databases.

The current level of client that you have installed determines the way to proceed with upgrade to DB2 Version 9.7. You can directly upgrade to Version 9.7 clients from Version 8, Version 9.1, or Version 9.5 clients. If you have Version 7 or earlier clients, migrate to any Version 8 client first.

Review Chapter 14, “Upgrade essentials for clients,” on page 123 for details about upgrade support and options available for clients.

Chapter 14. Upgrade essentials for clients

Upgrading clients to DB2 Version 9.7 requires an understanding of upgrade concepts, upgrade options, upgrade restrictions, upgrade recommendations, and connectivity between clients and DB2 servers.

After you have a complete understanding of what upgrading your clients involves, you can create your own plan to successfully upgrade your clients to DB2 Version 9.7.

In the upgrading client topics, the term *pre-Version 9.7 clients* refers to Version 9.5, Version 9.1, and Version 8 clients.

Upgrade options for clients

The upgrade options vary depending on the type of client that you want to install. The following table describes the upgrade options for each type of Version 9.7 client:

Table 19. Upgrade options for Version 9.7 clients

Upgrading from	Upgrading to	Upgrade support details
<ul style="list-style-type: none"> • Version 8 DB2 Administration Client • Version 8 DB2 Application Development Client • Version 9.1 DB2 Client • Version 9.5 Data Server Client (Windows)	Version 9.7 Data Server Client(Windows)	You have two options: <ul style="list-style-type: none"> • Install the Version 9.7 Data Server Client, and choose a pre-Version 9.7 client copy with the upgrade action in the Work with Existing window. The client instance is then automatically upgraded for you. • Install a new copy of the Version 9.7 Data Server Client, and then manually upgrade existing client instances.
<ul style="list-style-type: none"> • Version 8 DB2 Run-Time Client • Version 8 DB2 Run-Time Client Lite • Version 9.1 DB2 Runtime Client • Version 9.5 Data Server Runtime Client (Windows)	Version 9.7 Data Server Runtime Client(Windows)	<ul style="list-style-type: none"> • Install the Version 9.7 Data Server Runtime Client as a new copy, and then manually upgrade your existing client instance.
All Version 9.5, 9.1, or Version 8 clients (Linux or UNIX)	All Version 9.7 clients (Linux or UNIX)	<ul style="list-style-type: none"> • Install a new copy of any Version 9.7 client, and then manually upgrade your existing client instance.

When you upgrade a client instance, the bit size is determined by the operating systems where you installed the Version 9.7 client. Refer to Table 8 on page 27 for details.

Upgrade restrictions for clients

Review “Upgrade restrictions for DB2 servers” on page 19 for information regarding instance upgrade and operating system support. These restrictions also apply to clients and can impact their upgrade.

If you installed a Version 8 client on the same system as a DB2 Version 9.7 server or if you installed a Version 9.7 client on the same system as a DB2 Version 8 server, connections to the databases on the DB2 server from the client cataloged using a local node are not supported. You should upgrade both the DB2 server and the client to DB2 Version 9.7. If you do not upgrade the Version 8 client or the DB2 Version 8 server, you can only connect to the databases that are cataloged using TCP/IP nodes. Review the “Recataloging nodes and databases using TCP/IP protocol” on page 137 post-upgrade task for details.

Also, the trusted context capability supports only the TCP/IP protocol. Any connections to upgraded databases that you cataloged using a local node are unable to use this capability unless you recatalog the nodes using the TCP/IP protocol.

Connectivity support between clients and DB2 servers

In DB2 Version 9.7, the following support for connectivity between clients and DB2 servers is available:

Table 20. DB2 Version 9.7 connectivity support

Client	DB2 server	Client connectivity support
32-bit or 64-bit Version 9.7 clients	32-bit or 64-bit DB2 Version 9.7 server	Any Version 9.7 clients can establish 32-bit or 64-bit connections.
32-bit or 64-bit Version 9.7 clients	32-bit or 64-bit DB2 Version 9.5 server	Only DB2 Version 9.5 or earlier functionality is available.
32-bit or 64-bit Version 9.7 clients	32-bit or 64-bit DB2 Version 9.1 server	Only DB2 Version 9.1 or earlier functionality is available.
32-bit or 64-bit Version 9.7 clients	32-bit or 64-bit DB2 Version 8.2 server	Only DB2 Version 8.2 or earlier functionality is available.
32-bit or 64-bit Version 9.5 clients	32-bit or 64-bit DB2 Version 9.7 server	Only DB2 Version 9.5 or earlier functionality is available.
32-bit or 64-bit Version 9.1 clients	32-bit or 64-bit DB2 Version 9.7 server	Only DB2 Version 9.1 or earlier functionality is available.
32-bit or 64-bit Version 8.2 clients	32-bit or 64-bit DB2 Version 9.7 server	Only DB2 Version 8.2 or earlier functionality is available.

Connections to DB2 Version 9.7 servers from a client release prior to Version 8.2 are not supported.

Besides connectivity support, if you issue DB2 commands or SQL statements from a client to a DB2 server with a different version, you must be aware of incompatibilities between releases that can arise from changes in default behavior or restrictions lifted for these commands or SQL statements.

For example, if you issue the DESCRIBE command with the INDEXES FOR TABLE parameter from a Version 9.7 client, a pre-Version 9.7 DB2 server

lists only relational indexes while a Version 9.7 DB2 server lists indexes over XML data and text search indexes in addition to relational indexes. Refer to “Upgrade impact from DB2 command changes” on page 155 and “Upgrade impact from SQL statement changes” on page 162 for details.

Upgrade from Version 9.1 or Version 8 clients

If you are upgrading from Version 9.1 clients or earlier, review additional upgrade support changes that can also impact your upgrade:

- Changes between Version 9.5 and Version 9.1 clients.
- Changes between Version 9.1 and Version 8 clients.

Best practices for upgrading clients

Consider the following best practices when planning your client upgrade.

Upgrade clients after upgrading DB2 servers

In general, upgrade clients after you upgrade your DB2 servers. Supported pre-Version 9.7 clients can connect to Version 9.7 DB2 servers. However, the functionality introduced in releases after the pre-Version 9.7 client release is not available. If you plan to use this functionality in your applications, upgrade your clients to DB2 Version 9.7 or install new Version 9.7 client copies. See “Supported combinations of client and server versions” in *Installing IBM Data Server Clients* for details.

You can upgrade your clients before you upgrade your DB2 servers. However, you must ensure that your applications are able to manage any incompatibilities between releases. Review the following topics to determine if any incompatibilities apply to your application, and take necessary actions to manage these incompatibilities:

- Chapter 22, “Upgrade essentials for database applications,” on page 145 for changes to DB2 APIs, DB2 commands, and SQL statements
- “DB2 server behavior changes” on page 28 for changes on default values for existing registry variables, database and database manager configuration parameters
- “Deprecated or discontinued functionality that affects DB2 server upgrades” on page 38 for discontinued functionality not supported by Version 9.7 clients
- “Changed functionality” in *What’s New for DB2 Version 9.7* for additional changes between releases

Upgrade your clients in a test environment

Upgrading clients in a test environment allows you to determine if the upgrade can be successful and to address any problems that might occur during the upgrade process. You can also test your database applications and determine if you must upgrade them to run successfully in DB2 Version 9.7.

If you are upgrading your clients first, upgrading clients in a test environment allows you to determine and manage any incompatibilities between releases to successfully run your applications on pre-Version 9.7 DB2 servers using Version 9.7 clients

Install a new client copy instead of upgrading existing client

If you have software that requires a pre-Version 9.7 client, install the Version 9.7 client as a new copy and keep your existing client copy to

satisfy the software requirement. Then create a Version 9.7 client instance and keep your existing client instance with its configuration. You can select the option to create a new client instance during the installation, or you can manually create the client instance after installation.

Perform pre-upgrade and post-upgrade tasks

Perform the pre-upgrade and post-upgrade tasks for clients to ensure a successful upgrade.

Chapter 15. Pre-upgrade tasks for clients

Before you upgrade your clients, you should complete certain tasks to help ensure that your upgrade is successful.

Prepare for the upgrade of your clients by performing the following tasks:

1. Review the upgrade essentials for clients to determine which factors might impact your client upgrade.
2. Review the supported and non-supported client configurations.
3. Plan your upgrade strategy. For example, you might need to upgrade your DB2 server first, then your clients.
4. Optional: Upgrade your DB2 servers.
5. Back up your client configuration information.
6. Optional: Upgrade your clients in a test environment to identify upgrade issues and to verify that applications, scripts, tools and routines work as expected before upgrading your production environment.

Backing up client configuration information

Before you upgrade your client, back up the database manager configuration parameter settings of your client instance and the information details about all of your cataloged databases. With this information, you can restore your previous client configuration and cataloged databases after upgrade, if necessary.

Prerequisites

Ensure that you have SYSADM or SYSCTRL authority to run the db2cfexp command.

Restrictions

This procedure describes how to back up the configuration information for only one client. If you have different configuration settings on each client, you need to back up the configuration information for each client.

Procedure

To back up your client configuration information:

1. Back up your database manager configuration parameter settings by using the GET DATABASE MANAGER CONFIGURATION command to list your settings for the parameters and redirect the command output to a file as shown in the following example:

```
db2 GET DBM CFG > D:\upgrade\dbm_client.cfg
```

2. Back up the information of cataloged databases by running the db2cfexp command to create a configuration profile:

```
db2cfexp cfg_profile BACKUP
```

The BACKUP option creates the cfg_profile file as a configuration profile of the client instance that contains all of the instance configuration information, including the registry profile settings and information of a specific nature relevant only to this client instance. You can also use the DB2 Configuration Assistant to export your configuration profile.

Upgrading clients in a test environment

Upgrading clients in a test environment before you upgrade them in your production environment allows you to address problems during the upgrade process more effectively and to evaluate the impact of changes introduced in DB2 Version 9.7.

Prerequisite

- You must have root authority on Linux and UNIX operating systems or Local Administrator authority on Windows. You must also have SYSADM authority.

Restriction

- On Linux and UNIX operating systems, you must not set up the instance environment for the root user. Running the `db2iupgrade` or the `db2icrt` command when you set up the instance environment is not supported.

Procedure

To duplicate your production environment in a test environment, perform the following tasks:

1. Install the same client and version that you have in your production environment in a test system.
2. Re-create your client instance by running the `db2icrt` command with the `-s` option:

Operating system	DB2 command
Windows	<code>"%DB2PATH%"\bin\db2icrt -s client <i>InstName</i></code>
Linux and UNIX	<code>\$DB2DIR/instance/db2icrt -s client <i>InstName</i></code>

where `DB2PATH` and `DB2DIR` are set to the location of the client copy that you installed in the previous step, and `InstName` is the name of the instance.

3. Perform the pre-upgrade tasks that apply to your client.
4. Install a Version 9.7 client that you can upgrade to depending on the client that you are upgrading from. Select the **Install New** option to install a new copy. Refer to Table 19 on page 123 to determine what client product to install.
5. Upgrade your client instance by running the `db2iupgrade` command:

Operating system	DB2 command
Windows	<code>"%DB2PATH%"\bin\db2iupgrade <i>InstName</i></code>
Linux and UNIX	<code>\$DB2DIR/instance/db2iupgrade <i>InstName</i></code>

where `DB2PATH` and `DB2DIR` are set to the location of the Version 9.7 client copy that you installed in the previous step, and `InstName` is the name of the instance.

6. If you found any issues upgrading your test client instance, resolve these issues and add the tasks to resolve these issues to your upgrade plan.
7. Perform post-upgrade tasks that apply to your client.
8. Verify the client upgrade was successful.
9. Test your applications, scripts, tools and maintenance procedures using the Version 9.7 client.

Chapter 16. Upgrading to Data Server Client (Windows)

Upgrading an existing client copy to DB2 Version 9.7 requires that you install a Version 9.7 Data Server Client copy and then upgrade your client instance to retain your client configuration and to connect to all your previously cataloged databases.

Prerequisites

- Ensure that you have SYSADM, SYSCTRL, or SYSMANT authority and Local Administrator authority to run the db2iupgrade and the db2icrt commands.
- Review supported connectivity between DB2 clients and DB2 servers in upgrade essentials for DB2 clients.
- Perform pre-upgrade tasks for DB2 clients.

Restriction

- The bit size of the client instance is determined by the operating system where you install a Version 9.7 client. The instance is 32-bit only in 32-bit Windows on x86 or x64. The instance is 64-bit only in 64-bit Windows on x64. Refer to Table 8 on page 27 for details.
- If you are performing a response file installation to automatically upgrade a DB2 UDB Version 8 copy with multiple DB2 products installed, your copy must be at DB2 UDB Version 8 FixPak 7 or later.

About this task

When you install a Version 9.7 Data Server Client, you can choose to automatically upgrade an existing pre-Version 9.7 client copy. Your existing client instances are upgraded to a new Version 9.7 Data Server Client copy and the existing pre-Version 9.7 client copy is removed. You can also choose to install a new copy of Version 9.7 Data Server Client and then manually upgrade your existing client instance after installation.

Procedure

To upgrade from an existing client copy to a Version 9.7 Data Server Client on Windows:

1. Install Version 9.7 Data Server Client by running the setup command to launch the DB2 Setup wizard. You have three choices:
 - Select the **Work with Existing** option on the **Install a Product** panel. Then in the **Work with an existing DB2 copy** window, select a client copy name with action **upgrade**. The selected DB2 copy is removed and your client instance is upgraded. You can choose this option, if you have an existing copy of Version 8 Administration Client, Version 8 Application Development Client, Version 9.1 Client, or Version 9.5 Data Server Client.
 - Select the **Install New** option in the **Install a Product** panel. You should choose this option to create a new copy of Version 9.7 Data Server Client and keep your existing client copy. After installation, you must manually upgrade the client instance to run on the Version 9.7 Data Server Client copy:
 - Log on to the system as a user with Local Administrator authority.
 - Run the db2iupgrade command:

```
"%DB2PATH%\bin\db2iupgrade InstName
```

where DB2PATH is set to the location that you specified during the Version 9.7 Data Server Client installation and *InstName* is the name of the instance.

- Select the **Work with Existing** option on the **Install a Product** panel. Then in the **Work with Existing** window, choose the client copy name with the **upgrade** action. Finally, in the **Select the installation, response file creation, or both** window, select the **Save my installation setting in a response file** option to create a response file for a response file installation. The response file has the required UPGRADE_PRIOR_VERSIONS keyword, the client copy name to upgrade, and the installation path.

The result of the response file installation will be the same as in the first choice, all your client instances running on the selected client copy are automatically upgraded to the Version 9.7 Data Server Client copy. Using a response file installation to upgrade your clients can help you automate the upgrade process when you have a large number of clients.

2. If you want your applications to use the Version 9.7 Data Server Client copy through the default interface or if you upgraded your existing Version 8 client copy, set the Version 9.7 Data Server Client copy as the DB2 default copy. See “Changing the default DB2 and default IBM database client interface copy after installation” in *Installing DB2 Servers*.
3. Optional: You can create a new Version 9.7 client instance instead of upgrading the existing client instance. You only need to create a new Version 9.7 client instance when you want to keep multiple client copies running on the same machine, or create a testing environment. To create a new Version 9.7 client instance, run the db2icrt command with the option -s:

```
"%DB2PATH%"\bin\db2icrt -s client InstName
```

To create the same client connectivity environment you had, including the database manager configuration parameter and DB2 profile registry settings, run the db2cfimp command with the configuration profile that you save in the pre-upgrade tasks.

4. Compare the upgraded database manager configuration parameter values with the pre-upgrade values to ensure the changed values are compatible with your database applications.

After upgrading your client, perform the recommended post-upgrade tasks for DB2 clients, especially verifying upgrade for clients to ensure that your client upgrade was successful.

Chapter 17. Upgrading to Data Server Runtime Client (Windows)

Upgrading an existing Runtime Client copy to DB2 Version 9.7 requires that you install a Version 9.7 Data Server Runtime Client copy and then upgrade your client instance to retain your client configuration and to connect to all your previously cataloged databases

After you install a Version 9.7 Data Server Runtime Client copy, you can manually upgrade your existing client instance from a Version 8 DB2 Run-Time, Version 8 DB2 Run-Time Client Lite copy, Version 9.1 DB2 Runtime Client copy, or a Version 9.5 Data Server Runtime Client.

Prerequisites

- Ensure that you have SYSADM, SYSCTRL, or SYSMANT authority and Local Administrator authority to run the db2iupgrade and the db2icrt commands.
- Review supported connectivity between clients and DB2 servers in upgrade essentials for clients.
- Perform pre-upgrade tasks for clients.

Restrictions

- The bit size of the client instance is determined by the operating systems where you install Version 9.7 client. The instance is 32-bit only in 32-bit Windows on x86 or x64. The instance is 64-bit only in 64-bit Windows on x64. Refer to Table 8 on page 27 for details.

Procedure

To upgrade from a Version 8 DB2 Run-Time, Version 8 DB2 Run-Time Client Lite, or a Version 9.1 DB2 Runtime Client copy to Version 9.7 Data Server Runtime Client on Windows:

1. Install Version 9.7 Data Server Runtime Client. See “Installing IBM data server clients (Windows)” in *Installing IBM Data Server Clients*. Run the setup command to launch the DB2 Setup wizard.
2. If you want your applications to use the Version 9.7 Data Server Runtime Client copy through the default interface or if you upgraded your existing Version 8 client copy, set the Version 9.7 Data Server Runtime Client copy as the DB2 default copy. See “Changing the default DB2 and default IBM database client interface copy after installation” in *Installing DB2 Servers*.
3. Log on to the system as a user with Local Administrator authority.
4. Upgrade your existing client instance by running the db2iupgrade command:

```
"%DB2PATH%" \bin\db2iupgrade InstName
```

where DB2PATH is set to the location that you specified during the Version 9.7 Data Server Runtime Client installation and *InstName* is the name of the instance.

5. Optional: You can create a new Version 9.7 client instance instead of upgrading an existing client instance. You only need to create a new Version 9.7 client instance when you want to keep multiple client copies running on the same machine. To create a new Version 9.7 client instance, run the db2icrt command with the option -s:

```
"%DB2PATH%"bin\db2icrt -s client InstName
```

To create the same client connectivity environment you had, including the database manager configuration parameter and DB2 profile registry settings, run the `db2cfimp` command with the configuration profile that you saved in the pre-upgrade tasks.

6. Compare the upgraded database manager configuration parameter values with the pre-upgrade values to ensure the changed values are compatible with your database applications.

After upgrading your client, perform the recommended post-upgrade tasks for clients, especially verifying upgrade for clients to ensure that your client upgrade was successful.

Chapter 18. Upgrading clients (Linux and UNIX)

Upgrading existing clients to Version 9.7 requires that you install a Version 9.7 client copy and then upgrade your existing client instances to retain your client configuration and to connect to all your previously cataloged databases.

Prerequisites

- Ensure that you have root access.
- Ensure that you have SYSADM, SYSCTRL, or SYSMANT authority and root access to run the db2iupgrade and the db2icrt commands.
- Ensure that you meet the installation requirements for DB2 database products. Some operating systems require a 64-bit kernel.
- Review supported connectivity between clients and DB2 servers in upgrade essentials for clients.
- Perform pre-upgrade tasks for clients.

Restrictions

- You can only upgrade from a Version 8 DB2 Administration Client, Version 8 DB2 Application Development Client, Version 9.1 DB2 Client, or Version 9.5 Data Server Client to a Version 9.7. Data Server Client.
- You can only upgrade from a Version 8 DB2 Run-Time Client, Version 8 DB2 Run-Time Client Lite, Version 9.1 DB2 Runtime Client, or Version 9.5 Data Server Runtime Client to a Version 9.7 Data Server Runtime Client.
- On Linux and UNIX except for Linux on x64, your existing 32-bit or 64-bit client instances are upgraded to Version 9.7 64-bit client instances. The bit size of the client instance is determined by the operating system where you install the Version 9.7 client. Refer to Table 8 on page 27 for details.
- On Linux and UNIX operating systems, you must not set up the instance environment for the root user. Running the db2iupgrade or the db2icrt command when you set up the instance environment is not supported.

Procedure

To upgrade existing clients to Version 9.7 clients:

1. Install the appropriate Version 9.7 client as a new copy by running the db2setup command and select Install New on the Install a Product panel:
 - If you are upgrading from a Version 8 DB2 Administration Client, Version 8 DB2 Application Development Client, Version 9.1 DB2 Client, , or Version 9.5 Data Server Client, install a new Version 9.7 Data Server Client.
 - If you are upgrading from a Version 8 DB2 Run-Time Client, Version 8 DB2 Run-Time Client Lite, Version 9.1 DB2 Runtime Client, or Version 9.5 Data Server Runtime Client, install a new Version 9.7 Data Server Runtime Client copy.
2. Log on to the system as root.
3. Upgrade your existing client instances by running the db2iupgrade command:
`$DB2DIR/instance/db2iupgrade InstName`

where

DB2DIR

is set to the location that you specified during the Version 9.7 client installation. The default installation path for UNIX is /opt/IBM/db2/V9.7 and for Linux is /opt/ibm/db2/V9.7.

InstName

is the login name of the client instance owner.

4. Optional: You can also create a new Version 9.7 client instance instead of upgrading the existing client instance. You only need to create a new Version 9.7 client instance when you want to keep multiple client copies running on the same machine. To create a new Version 9.7 client instance, run the db2icrt command with the option -s:

```
$DB2DIR/instance/db2icrt -s client InstName
```

where

DB2DIR

is set to the location you specified during the Version 9.7 client installation.

InstName

Is the login name of the instance owner.

To create the same client connectivity environment you had, including the database manager configuration parameter and DB2 profile registry settings, run the db2cfimp command with the configuration profile that you backed up in the pre-upgrade tasks.

5. Compare the upgraded database manager configuration parameter values with the pre-upgrade values to ensure the changed values are compatible with your database applications.

After upgrading your client, perform the recommended post-upgrade tasks for clients, especially verifying upgrade for clients to ensure that your client upgrade was successful.

Chapter 19. Upgrading to IBM Data Server Driver Package

Upgrading to IBM Data Server Driver Package (DSDRIVER) requires that you install a Version 9.7 DSDRIVER and optionally set the default client interface.

Prerequisites

- Review supported connectivity between DB2 clients and DB2 servers in Chapter 14, “Upgrade essentials for clients,” on page 123.

Procedure

1. If you have installed a Version 9.5 DSDRIVER or Version 9.5 Data Server Client, back up the `db2dsdriver.cfg` DSDRIVER configuration file. The following table shows examples on how to back up this configuration file :

Operating system	Commands to back up configuration file
Linux and UNIX	<code>cd DB2DIR/cfg</code> <code>cp db2dsdriver.cfg BACKUPDIR/db2dsdriver.cfg</code>
Windows	<code>cd DB2DIR\cfg</code> <code>copy db2dsdriver.cfg BACKUPDIR\db2dsdriver.cfg</code>

Where `BACKUPDIR` is the directory to backup the configuration file and `DB2DIR` is the installation directory for the Version 9.5 DSDRIVER or Version 9.5 Data Server Client.

2. Install a Version 9.7 DSDRIVER copy. See “Installation methods for IBM data server clients” in *Installing IBM Data Server Clients* for details.
3. If you have installed a Version 9.5 Data Server Client copy, you can use this existing Data Server Client copy to configure the Version 9.7 DSDRIVER copy by issuing the following command:
`db2dsdcfgfill [-i instance-name | -p instance-directory | -o output-dir]`
4. If you have installed a Version 9.5 DSDRIVER and you want to have the same configuration on the Version 9.7 DSDRIVER copy, restore the `db2dsdriver.cfg` configuration file that you backed up in step 1 to the `DB2DIR2/cfg` directory. Where `DB2DIR2` is the installation directory for the Version 9.7 DSDRIVER. The following example shows how to restore a configuration file:

Operating system	Commands to back up configuration file
Linux and UNIX	<code>copy BACKUPDIR/db2dsdriver.cfg DB2DIR2/cfg</code>
Windows	<code>copy BACKUPDIR\db2dsdriver.cfg DB2DIR2\cfg</code>

5. If you want your applications to use the Version 9.7 DSDRIVER copy through the default interface, set the Version 9.7 DSDRIVER copy as the DB2 client interface default. See “Changing the default DB2 and default IBM database client interface copy after installation” in *Installing DB2 Servers*.

If you did not have a Version 9.1 or Version 9.5 DSDRIVER installed, the Version 9.7 DSDRIVER copy is set as the client interface default.

What to do next

After upgrading your IBM Data Server Driver Package, perform only the post-upgrade tasks for DB2 clients that apply.

Chapter 20. Post-upgrade tasks for clients

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

Perform the following post-upgrade tasks that apply to your clients:

1. Recatalog nodes and databases if you cataloged them using the NetBIOS and SNA protocols in DB2 UDB Version 8. Starting DB2 Version 9.1, the NetBIOS and SNA protocols are discontinued.
2. Manage changes in DB2 server behavior by modifying your settings where required. There are new registry variables, new configuration parameters, and new default values for registry variables and configuration parameters introduced in DB2 Version 9.7 that can impact the behavior of your application.
3. Verify that upgrading your clients was successful.

Recataloging nodes and databases using TCP/IP protocol

Recataloging nodes using the TCP/IP protocol is required when the communication protocol that you used to catalog these nodes is no longer supported or when you want to use a feature that supports only TCP/IP protocol. If you change the node name, you must recatalog the databases as well to use the new node name.

The NetBIOS and SNA protocols are discontinued since DB2 Version 9.1. You must recatalog, using a valid protocol, any nodes that you cataloged with the NetBIOS and SNA protocols. If you try to connect to any databases cataloged on a node that uses the NetBIOS or SNA protocol, your connection request returns an error because these protocols are invalid.

If you have a Version 8 client installed on the same system as a DB2 Version 9.7 server or a Version 9.7 client installed on the same system as a DB2 Version 8 server, connections to the databases on the DB2 server from the DB2 client cataloged using a local node are not supported. If you do not upgrade the Version 8 client or DB2 Version 8 server to DB2 Version 9.7, recatalog local nodes as TCP/IP nodes.

If you want to use the trusted context capability on upgraded databases that are cataloged using a local node, recatalog the nodes using the TCP/IP protocol.

Prerequisites

- Ensure that you have SYSADM or SYSCTRL authority.
- Ensure that you have network connectivity from the client to the DB2 server.

Restriction

The only protocols available in DB2 Version 9.7 are TCP/IP, Named Pipes, and SSL.

Procedure

To recatalog nodes and databases specifying the TCP/IP protocol:

1. Determine which are the local nodes that you want to recatalog or the nodes use the NetBIOS or SNA protocol by issuing the LIST NODE DIRECTORY command:

```
db2 LIST NODE DIRECTORY show detail > node_list.log
```

Redirect the output of this command to a file and keep it, because the information is useful to recatalog your nodes.

2. Remove the local nodes that you want to recatalog and all nodes that use NetBIOS or SNA protocol from the node directory by issuing the UNCATALOG NODE command:

```
db2 UNCATALOG NODE node-name
```

3. Determine which databases use the nodes that you uncataloged in the previous step by issuing the LIST DATABASE DIRECTORY command:

```
db2 LIST DATABASE DIRECTORY show detail > database_list.log
```

4. If you are going recatalog your nodes using a different node name, remove all databases using those nodes by issuing the UNCATALOG DATABASE command:

```
db2 UNCATALOG DATABASE database-name
```

5. Recatalog your nodes specifying TCP/IP as the protocol by issuing the CATALOG TCPIP NODE command. If you use the original node name, you do not have to recatalog your databases.

```
db2 CATALOG TCPIP NODE new-node REMOTE host-name  
SERVER instance-svcename REMOTE_INSTANCE instance-name
```

You can determine the value of *instance-svcename* by looking at the value of the **svcename** database manager configuration parameter for that instance.

6. If you did not recatalog your nodes using the original node names, recatalog your databases using the new node name by issuing the CATALOG DATABASE command.

```
db2 CATALOG DATABASE db-name [AS alias-db-name]  
AT NODE new-node
```

Verifying your client upgrade

When the upgrade of your client is complete, it is a good practice to run some tests in the new upgraded environment to verify that your client is working as expected. These tests can consist of running batch programs that connect to databases in a DB2 server or any programs or scripts that you use for benchmarking.

Prerequisites

- Ensure that you have network connectivity from the client to the DB2 server.
- Ensure that the DB2 servers and instances are up and running.

Procedure

To verify that your client upgrade is successful:

1. Test connecting to all cataloged databases. The following example tests a connection to a remote database by issuing the CONNECT command:

```
db2 CONNECT TO sample USER mickey USING mouse
```

```
Database Connection Information
```

Database server = DB2/AIX64 9.7.0
SQL authorization ID = MICKEY
Local database alias = SAMPLE

You need to specify a user ID and password when connecting to a remote database.

2. If you experience problems connecting to your cataloged database, use the db2cfimp tool and the configuration profile that you saved by performing the saving DB2 clients configuration pre-upgrade task to re-create the same client connectivity environment you had prior to upgrade.
3. Run your client database applications or scripts that connect to your databases to ensure they are working as expected.

Part 4. Upgrading applications and routines

This part of the book contains the following chapters:

- Chapter 21, “Database applications and routines upgrade,” on page 143
- Chapter 22, “Upgrade essentials for database applications,” on page 145
- Chapter 23, “Upgrade essentials for routines,” on page 177
- Chapter 24, “Pre-upgrade tasks for database applications and routines,” on page 181
- Chapter 25, “Upgrading database applications,” on page 183
- Chapter 26, “Upgrading routines,” on page 193
- Chapter 27, “Post-upgrade tasks for database applications and routines,” on page 201
- Chapter 28, “Adopting new DB2 Version 9.7 functionality in database applications and routines,” on page 203

Chapter 21. Database applications and routines upgrade

Upgrading to DB2 Version 9.7 involves the upgrade of your database applications and routines if changes in DB2 Version 9.7 impact your database applications and routines.

Upgrading your applications and routines involves the following actions:

- Test whether your applications and routines perform as expected in a DB2 Version 9.7 testing environment. You do not need to upgrade your applications and routines if they run successfully.
- If your applications or routines have errors running in DB2 Version 9.7, you should:
 - Review upgrade essentials for database applications to identify any changes in DB2 Version 9.7 that can impact your applications.
 - Review upgrade essentials for routines to identify any changes in DB2 Version 9.7 that can impact your routines.
 - Plan how to modify your applications and routines to handle these changes. Determine the steps that you must perform by reviewing the Upgrading database applications or Upgrading routines tasks.
 - Modify your applications and routines according to your plan.
 - Test your applications and routines in a DB2 Version 9.7 testing environment.
- Verify that your applications and routines perform as expected in your DB2 Version 9.7 production environment before deploying them.

If your applications and routines use any functionality that is deprecated in DB2 Version 9.7, you should plan how to remove this functionality from your application code in the near future.

Also, you should consider adopting new functionality available in DB2 Version 9.7 to enhance functionality and improve performance.

Chapter 22. Upgrade essentials for database applications

Changes in application development support, new functionality, discontinued functionality, and deprecated functionality might impact your database applications, scripts and tools after you upgrade them to DB2 Version 9.7.

Operating system support

A complete list of supported operating systems is available at “Installation requirements for DB2 database products” in *Installing DB2 Servers*. If your current version of operating system is unsupported, you must upgrade it before you install DB2 Version 9.7.

In UNIX operating systems, only 64-bit kernels are supported. Your 32-bit instances are upgraded to DB2 Version 9.7 64-bit instances.

If you upgrade to the latest version of your operating system or you install a 64-bit kernel, rebuild all database applications and external routines after you upgrade to DB2 Version 9.7 so that they use the new runtime libraries in the operating system.

Development software support

Development software support has also changed. To improve performance and avoid technical support issues, rebuild your applications with the latest version of your development software. Review the changes in support for development software requirements. See “Support for elements of the database application development environment” in *Getting Started with Database Application Development*

Application drivers

The IBM Data Server Driver for JDBC and SQLJ includes the `db2jcc.jar` class file for applications that use JDBC 3.0 methods or earlier and the `db2jcc4.jar` class file for applications that use JDBC 4.0 methods or earlier. The JDBC 4.0 `java.sql.DatabaseMetaData.getDriverName` method returns the IBM Data Server Driver for JDBC and SQLJ name instead of the IBM DB2 JDBC Universal Driver Architecture name. To manage the behavioral differences between the driver that supports JDBC 4.0 in Version 9.7 and previous releases of this driver, upgrade Java applications that use IBM Data Server Driver for JDBC and SQLJ. See “Upgrading Java applications that use IBM Data Server Driver for JDBC and SQLJ” on page 186 for details.

The DB2 JDBC Type 2 driver has been deprecated since DB2 Version 9.1. You should modify your Java applications and external routines to use the IBM Data Server Driver for JDBC and SQLJ with type 2 connections. To manage the behavioral differences between the driver that supports JDBC 3.0 in Version 9.7 and the DB2 JDBC Type 2 driver, upgrade your Java applications that use DB2 JDBC Type 2 driver. See “Upgrading Java applications that use DB2 JDBC Type 2 driver” on page 188 for details.

See “Java software support for DB2 products” in *Installing DB2 Servers* for details about the versions of IBM Data Server Driver for JDBC and SQLJ that are delivered with every DB2 database product version and fix packs.

DB2 CLI applications, DB2 CLP interface, and .Net Data Provider clients support Secure Sockets Layer (SSL). The IBM Global Security Kit (GSKit) provides encryption services for the Secure Sockets Layer (SSL) support.

Refer to “Configuring Secure Sockets Layer (SSL) support in non-Java DB2 clients” in *Database Security Guide* for details about how to enable SSL in a client including how to download and install the GSKit.

Implicit LOB data inlining

For upgraded databases, the `INLINE LENGTH` default value is the maximum size of the LOB descriptor for the corresponding LOB column. LOB data is inlined when the length of the LOB data plus the overhead is less than the `INLINE LENGTH` value. Therefore, if the LOB data length plus the overhead is less than the LOB descriptor size for the LOB column, the LOB data is implicitly inlined in a table row after the database upgrade.

If you make extensive use of LOBs in your database applications, you can increase performance for SQL statements that access the LOB data by increasing `INLINE LENGTH` to an adequate value because no additional I/O is required to access the LOB data when it is inlined in a table row. Refer to Chapter 28, “Adopting new DB2 Version 9.7 functionality in database applications and routines,” on page 203 for details.

XQuery expressions and XML data types

After upgrading to DB2 Version 9.7, the XQuery string data type is used for values of elements or attributes that are not cast in an XQuery expression. Type annotations in existing XML documents that you validated are no longer used to do implicit casting. If you validate new XML documents to insert them in an XML data type column, these XML documents are stored without type annotations.

For XQuery expressions that depend on data types based on type annotations from validated XML documents, you need to explicitly cast elements and attributes in all XQuery expressions from validated XML documents. Without explicit type casting, XQuery expressions that used implicit casting or casting to other types will fail after the upgrade.

These examples shows in *italics font style* the explicit type casting that was added to the following XQuery expressions:

- In this example, the string values are cast to integer to compare values as integers.

```
xquery
let $x1 := db2-fn:sqlquery("SELECT xmlcol FROM xq.sch1
                           WHERE DOCID=1")/prodA/sale/xs:integer(price),
    $x2 := db2-fn:sqlquery("SELECT xmlcol FROM xq.sch2
                           WHERE DOCID=1")//xs:integer(clearance)
return (<b1>{ $x1 > $x2 }</b1>)
```

- In this example, the string value *\$i* is cast to date to perform a date comparison.

```
xquery
for $i in
  (db2-fn:sqlquery("SELECT xmlcol FROM xq.basicSchemaTypes
                   WHERE DOCID=1")//date)[1]
return if (xs:date($i) eq xs:date('1967-08-13'))
  then $i else (<notOK1/>)
```

- In this example, the string value returned by `db2-fn:xmlcolumn` is cast to a `dateTime` data type and the day is extracted from the `dateTime` value.

```
xquery
for $datetime in
  (db2-fn:xmlcolumn('XQ.BASICSCHEMATYPES.XMLCOL')//dateTime)[1]
  cast as xs:dateTime
return $datetime cast as xs:gDay
```

DB2 APIs and DB2 commands

Review the following topics to determine if you have applications and scripts that are impacted by changes to DB2 APIs and DB2 commands in DB2 Version 9.7:

- DB2 API functions
- DB2 command line processor (CLP) and system commands

SQL statements

Review the changes to SQL statements in DB2 Version 9.7 to determine if you have applications and scripts that are impacted by these changes and how to manage these changes. Introduction of new functionality such as an untyped NULL keyword in expressions and a DEFAULT keyword in procedure parameters requires that you modify your applications to adapt to these changes.

System catalog views and system-defined administrative routines and views

After database upgrade to DB2 Version 9.7, the system catalog views under the SYSCAT schema remain compatible with catalog views that you defined in DB2 Version 9.1. However, there are new columns, increases in column length, or columns with changed data types in some of the system catalog views.

SQL administrative routines include changes such as new parameters and new columns returned. Also, some routines are replaced with system-defined administrative routines and views. In addition, all of the system-defined table functions with names that start with SNAPSHOT_ have been deprecated since DB2 Version 9.1.

Review the following topics to determine if you have applications and scripts that are impacted by changes to system catalog views and system-defined administrative routines and views:

- System catalog
- “Deprecated system-defined administrative routines and their replacement routines or views” in *Administrative Routines and Views*

Optimizer and query execution plans

For new databases created in DB2 Version 9.7, the **cur_commit** configuration parameter is set to ON so that currently committed semantics is enabled on cursor stability scans. Under the new currently committed semantics, only committed data is returned, as was the case previously with the cursor isolation level, but now a read operation does not wait for a write operation to release row locks. A returned result set operating under cursor stability isolation level might be different than in previous releases. Refer to Chapter 10, “Adopting new Version 9.7 functionality in upgraded databases,” on page 107 for details on enabling currently committed behavior.

The optimizer now calculates execution plans for star join queries using different cardinality estimates than in previous releases. With these new estimates, the optimizer might choose a better execution plan for certain queries to improve their performance and stability.

The MQT matching process now considers additional situations that can result in the optimizer choosing a different execution plan for queries that match an MQT. In upgraded databases, you could experience improvements in queries matching GROUP BY MQTs that use the DISTINCT clause and queries that use DATE predicates right after the upgrade without any actions on your part. However, exploiting these features further and exploiting other improvements such as use of view MQTs or optimization guidelines to force the optimizer to choose a specific MQT, require implementation after the upgrade. Refer to Chapter 28, “Adopting new DB2 Version 9.7 functionality in database applications and routines,” on page 203 for details on how to use these new features.

The optimizer now pushes down relational predicates (for filters and XPath extractions) into XQuery query blocks. Therefore, enabling early data filtering and better potential index usage. In partitioned database environments, early data filtering potentially reduces the amount of data transfer between partitions. Consequently, you will notice new query access paths, improved performance, and reduced memory usage for combined SQL/XQuery queries. See “Compiler rewrite example: Predicate pushdown for combined SQL/XQuery statements” in *Troubleshooting and Tuning Database Performance*

Scan sharing is introduced in DB2 Version 9.7 to allow a scan to read the buffer pool pages of another scan. This behavior increases concurrency, reduces query response times, and increases system throughput without requiring hardware upgrades. The SQL compiler determines eligibility for scan sharing automatically. At run time, an eligible scan may or may not participate in sharing, based on considerations in effect that were not known at compile time. See “Scan sharing” in *Troubleshooting and Tuning Database Performance*

The optimizer now calculates more realistic CPU costs of passing data from one database agent to another in partitioned database environments. This will enable the optimizer to consistently select plans that minimize CPU costs associated with data transmission, thus improving the performance of some SQL statements. Refer to “Optimizer’s cost model improved for queries in DPF environment” in *What’s New for DB2 Version 9.7* for details.

Rebind any statically bound packages after upgrade to take advantage of optimizer improvements.

Database packages

When you upgrade a database, all packages for user applications and routines are placed into an invalid state. Packages are also placed into an invalid state if they depend on database objects that you dropped, such as tables, views, aliases, indexes, triggers, referential constraints, and table check constraints. If you drop a UDF, your package is placed into an inoperative state.

Although invalid packages are automatically rebound by the database manager the first time that an application needs to access them, rebind your database packages to control when rebinding occurs and resolve any possible issues. See the Optimizer enhancements section for additional advantages of manually rebinding your database packages.

32-bit and 64-bit DB2 server support

On Linux and UNIX operating systems excluding Linux on x86, DB2 Version 9.7 requires 64-bit kernels and supports only 64-bit instances. Therefore, when you upgrade to DB2 Version 9.7, your DB2 UDB Version 8 32-bit instances are upgraded to 64-bit instances. See 32-bit and 64-bit DB2 server support changes for details.

The following table indicates applications that run after you upgrade to DB2 Version 9.7 depending on the embedded shared library path:

Table 21. Embedded shared library paths used in database applications

Application	Operating system	Upgrading from	Embedded shared library paths
32-bit	32-bit or 64-bit (Linux and UNIX operating systems)	32-bit instance	\$INSTHOME/sqllib/lib ¹ \$INSTHOME/sqllib/lib32
64-bit	64-bit (Linux and UNIX operating systems)	64-bit instance	\$INSTHOME/sqllib/lib ² \$INSTHOME/sqllib/lib64
32-bit	Windows	32-bit instance	DB2PATH\lib
32-bit	Windows	64-bit instance	DB2PATH\lib\Win32
64-bit	Windows	64-bit instance	DB2PATH\lib

Note:

1. \$INSTHOME/sqllib/lib is a symbolic link to \$INSTHOME/sqllib/lib32.
2. \$INSTHOME/sqllib/lib is a symbolic link to \$INSTHOME/sqllib/lib64.

where *INSTHOME* is your instance home directory, and *DB2PATH* is the directory of your DB2 Version 9.7 copy.

During DB2 Version 9.7 installation, statements are added to the *db2profile* and *db2cshrc* file to set the environment variables for the library search path. These environment variables specify additional locations where DB2 shared libraries can be loaded at application run time, allowing your application to run after you upgrade to DB2 Version 9.7 if you did not specify the correct shared library path. The following table shows the settings that you should have for the library search path environment variables:

Table 22. Environment variable settings for library search paths

Environment variable and Operating system	Application	Variable value
<ul style="list-style-type: none"> • LIBPATH (AIX operating system) • LD_LIBRARY_PATH (HP-UX, Linux, and Solaris operating systems) 	32-bit	INSTHOME/sqllib/lib32 ¹

Table 22. Environment variable settings for library search paths (continued)

Environment variable and Operating system	Application	Variable value
<ul style="list-style-type: none"> • LIBPATH (AIX operating system) • LD_LIBRARY_PATH (HP-UX, Linux, and Solaris operating systems) 	64-bit	INSTHOME/sql1lib/lib64
LIB (Windows operating systems)	32-bit application running on a 64-bit instance	DB2PATH\lib\Win32 ²
LIB (Windows operating systems)	32-bit or 64-bit	DB2PATH\lib

Note:

1. You should indicate this setting at the beginning of the LIBPATH variable.

These environment variables specify additional locations where DB2 shared libraries can be loaded at application runtime, allowing your application to run after you upgrade to DB2 Version 9.7 if you did not specify the correct shared library path.

DB2 server behavior

In general, the DB2 server behavior is compatible between releases. However, there are changes in behavior to support new functionality or improve the performance of existing functionality. Review “DB2 server behavior changes” on page 28 to determine the impact of these behavior changes on your applications.

After upgrading your DB2 server, compare your registry variable and configuration parameter values to your values before upgrade, and change any values according to the needs of your applications.

Client connectivity support

Your applications can use pre-Version 9.7 clients to access databases in DB2 Version 9.7 servers. However, your applications are restricted to the functionality available for that client. Review Chapter 14, “Upgrade essentials for clients,” on page 123 to learn details about client connectivity and to identify changes in support that can impact your DB2 clients.

Web Objects Runtime Framework (WORF)

Starting with DB2 Version 9.7, WORF has been discontinued. You can replace WORF with a new feature within IBM Data Studio that allows you to create Web services without writing document access definition extension (DADX) files. See Upgrading Web applications that were developed for the Web Object Runtime Framework (WORF) for details on how to upgrade you existing Web applications.

DB2 Embedded Application Server (EAS)

DB2 EAS has been discontinued and the DB2WebServices application is not available in DB2 Version 9.7. Use Data Studio to re-create your Web services and redeploy your Web services after upgrading to DB2 Version 9.7. If you have Web applications that were developed for WORF, see the previous section for details about how to migrate these Web applications.

The DB2 administration tools have been deprecated and the DB2 Web Tools have been discontinued since DB2 Version 9.5. Alternatively, install

Data Studio and use the Data Source Explorer to perform database administration tasks. See Database administration from the Data Source Explorer for details. Also, visit the Data Studio product page at <http://www.ibm.com/software/data/studio/> for details about product offerings and downloads.

Upgrade of applications from DB2 Version 9.1 or DB2 UDB Version 8

If you are upgrading from DB2 Version 9.1 or earlier, review changes in application driver support, 32-bit and 64-bit DB2 server support, and discontinued functionality between pre-Version 9.7 releases that might also impact your applications and scripts:

- Changes between DB2 Version 9.5 and DB2 Version 9.1 that impact applications.
- Changes between DB2 Version 9.1 and DB2 UDB Version 8 that impact applications.

Upgrade impact from DB2 API changes

The changes in DB2 Version 9.7 to DB2 APIs can impact your existing applications after you upgrade to DB2 Version 9.7.

The changes to DB2 APIs include new parameters, modifications to existing parameters, and deprecated or discontinued APIs. The following table lists the changes that impact your existing applications:

Table 23. Changes to DB2 APIs

DB2 API	Summary of changes with upgrade impact
db2DatabaseUpgrade	This API upgrades a pre-Version 9.7 DB2 database to the current release. It uses the new db2DatabaseUpgradeStruct structure as parameter. Use this API instead of the sqlmgdb or sqlmgdb API.
db2Export, db2Import, or db2Load	<p>Due to changes in the DB2 authorization model, the SYSADM group is no longer authorized to run these APIs and the DBADM authority no longer provides access to data. The new DATAACCESS authority now provides access to data. The UPGRADE DATABASE command grants DBADM authority to the SYSADM group and grants DATAACCESS authority to users holding DBADM authority including the SYSADM group so that there is no upgrade impact. However, for these APIs, you should review all of the changes in authorization and grant required authorization to users.</p> <p>If you create databases in DB2 Version 9.7, grant the required authorization to users that need to run these APIs. You can grant DBADM authority to the SYSADM group and grant DATAACCESS authority to users holding DBADM authority to maintain the same authorization as in previous releases.</p> <p>The CREATE and REPLACE_CREATE parameters of the IMPORT command are deprecated. Use DDL scripts that you developed or generated with the db2look command to create the table before you call the db2Import API specifying another supported parameter in the action string such as INSERT, INSERT_UPDATE, or REPLACE.</p>
db2GetDistMap	Due to the size increase of the distribution map from 4 096 to 32 768 entries, use this new API to read the distribution map. This API requires the new db2DistMapStruct structure as parameter. Read the row in this table for the sqlugtpi API for more details.

Table 23. Changes to DB2 APIs (continued)

DB2 API	Summary of changes with upgrade impact
db2GetRowPartNum	<p>Due to the size increase of the distribution map from 4 096 to 32 768 entries, use this new API to read database partition number and database partition server number for a row. This API requires the new db2RowPartNumStruct structure as parameter. Read the row in this table for the sqlugrpn API for more details.</p>
db2HistoryGetEntry	<p>A log sequence number (LSN) now uses 8 bytes. Therefore, every occurrence of the SQLU_LSN datatype in the db2HistoryData structure has been replaced with the new db2LSN datatype. After upgrading your DB2 servers and clients to Version 9.7, calls to this API in your applications continue to work until the value of the LSN exceeds the 6-byte maximum limit for pre-Version 9.7 releases. If the LSN reaches the 6-byte limit, you will receive the SQL2032N error message.</p> <p>Modify your application code to read the new db2LSN datatype in the modified structure and rebuild your applications using the DB2 Version 9.7 libraries before the LSN reaches the maximum limit. Refer to “Log sequence number changes affecting API and application behavior” in <i>Administrative API Reference</i> for details.</p>
db2Prune	<p>Although there are no changes to the db2PruneStruct structure, if you call this API with the piString field of this structure pointing to a string representing an LSN, the length of the LSN string can now be variable and have a maximum length of 16 characters excluding the NULL terminator. In previous releases, the LSN string was exactly 12 characters in size excluding the NULL terminator. Modify your application code to increase the maximum length for an LSN string and rebuild your applications using the DB2 Version 9.7 libraries. Also, use the new db2LsnToString API for conversions between the db2LSN structure and the LSN string.</p>
db2Runstats, sqlabndx, sqlaprep, or sqlarbnd	<p>Due to changes in the DB2 authorization model, the SYSADM group is no longer authorized to run these APIs. The UPGRADE DATABASE command grants DBADM authority to the SYSADM group so that there is no upgrade impact. However, for these APIs, you should review all of the changes in authorization and grant required authorization to users.</p> <p>If you create databases in DB2 Version 9.7, grant the required authorization to users that need to run these APIs. You can grant DBADM authority to the SYSADM group to maintain the same authorization as in previous releases.</p>

Table 23. Changes to DB2 APIs (continued)

DB2 API	Summary of changes with upgrade impact
db2ReadLog and db2ReadLogNoConn	<p>An LSN now uses 8 bytes. Therefore, every occurrence of the SQLU_LSN datatype in the db2ReadLogStruct, db2ReadLogInfoStruct, db2ReadLogNoConnStruct, and db2ReadLogNoConnInfoStruct structures has been replaced with the new db2LSN datatype. If you call any of these APIs in your applications, modify your application code to read the new db2LSN datatype in these modified structures and rebuild your applications using the DB2 Version 9.7 libraries before upgrading your DB2 servers and clients. Refer to “Log sequence number changes affecting API and application behavior” in <i>Administrative API Reference</i> for details.</p> <p>If you set the iFilterOption parameter to DB2READLOG_FILTER_ON when you call the db2ReadLog API in your applications, modify your application code to read the new db2ReadLogFilterData structure that is prefixed to log records. If you set the iFilterOption parameter to DB2READLOG_FILTER_OFF, although log records continue to be prefixed with an LSN as in previous releases, modify your application code to read the new db2LSN datatype.</p>
Db2Reorg	<p>The db2ReorgStruct structure has been modified to support partitioned indexes. A new pPartitionName field to indicate a data partition name has been added to the db2ReorgTable and db2ReorgIndexesAll structures. Depending on the value of the versionNumber field, a new partitionNameLen field has also been added to indicate the length in bytes of pPartitionName. Although the value of this parameter is only considered when the DB2REORG_ON_DATA_PARTITION flag is specified, rebuild your applications using the DB2 Version 9.7 libraries to start using the modified structure.</p>
db2Runstats	<p>When you call this API to update statistics on tables with LOB columns, statistics for the average length of column and number of NULLs in a column are now collected.</p> <p>Refer to the db2Runstats API row for additional authorization changes that also apply to this API.</p>
sqlbctsq, sqlbgtss, sqlbotsq, and sqlbtcq	<p>These APIs and related SQLB_TBSPQRY_DATA data structure have been deprecated and might be removed in a future release. Start using queries with the MON_GET_TABLESPACE or the MON_GET_CONTAINER table functions instead. These table functions return more information than was provided by the deprecated APIs.</p>

Table 23. Changes to DB2 APIs (continued)

DB2 API	Summary of changes with upgrade impact
sqlbftpq, sqlbmtsq, and sqlbstpq	<p>The data type for the lifeLSN field in the SQLB_TBSPQRY_DATA structure has changed to sqluint64 from char[6]. Calls to this API in your applications continue to work after upgrading your DB2 servers and clients until the value of the LSN exceeds the 6-byte maximum limit. Modify your application code to read the sqluint64 datatype in this modified structure and rebuild your applications using the DB2 Version 9.7 libraries before you reach this maximum limit.</p> <p>These APIs and related SQLB_TBSPQRY_DATA data structure have been deprecated and might be removed in a future release. Start using queries with the MON_GET_TABLESPACE or the MON_GET_CONTAINER table functions instead. These table functions return more information than was provided by the deprecated APIs.</p>
sqlbstsc	<p>You can now use this API to enable automatic storage for table spaces. All existing containers are redefined to use the storage paths defined in the database. A value of 0 for the NumContainers parameter along with a NULL value for the pContainerData parameter enables the table space for automatic storage.</p>
sqlcrea	<p>For new databases created on DB2 Version 9.7, the new cur_commit configuration parameter is set to ON so that currently committed semantics is enabled on cursor stability scans. A returned result set operating under cursor stability isolation level set might be different than in previous releases. Refer to “Currently committed semantics improve concurrency” in <i>Troubleshooting and Tuning Database Performance</i> for details on how currently committed behavior works.</p>
sqlmgdb and sqlgmdb	<p>These APIs are deprecated. Use the new db2DatabaseUpgrade API.</p>
sqladau	<p>The sqladau API is discontinued in DB2 Version 9.7. Use the AUTH_LIST_AUTHORITIES_FOR_AUTHID table function to get the information that the sqladau API provided.</p>
sqlgtpi	<p>Starting with DB2 Version 9.7, this API is deprecated. If you set the DB2_PMAP_COMPATIBILITY registry variable to OFF and you call this API, you will receive the SQL2768N error message because of the size increase of the distribution map to 32 768 entries. Use the new db2GetDistMap API.</p>
sqlurlog	<p>If you call this API from an application running on a pre-Version 9.7 client that connects to a Version 9.7 database, you will receive the SQL1198N error message. Calls to the sqlurlog API from an application running on a Version 9.7 client that connects to a Version 9.7 database, continue to work until the value of the LSN exceeds the 6-byte maximum limit for pre-Version 9.7 releases. If the LSN reaches this 6-byte limit, you will receive the SQL2650N error message</p>
sqlgrpn	<p>Starting with DB2 Version 9.7, this API is deprecated. If you set the DB2_PMAP_COMPATIBILITY registry variable to OFF and you call this API, you will receive the SQL2768N error message because of the size increase of the distribution map to 32 768 entries. Use the new db2GetRowPartNum API to read the database partition number and database partition server number for a row. See DB2_PMAP_COMPATIBILITY in Table 9 on page 29 for details.</p>

Upgrade impact from DB2 command changes

The changes in DB2 Version 9.7 to DB2 command line processor (CLP) and system commands can impact your existing applications and scripts after you upgrade to DB2 Version 9.7.

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

Table 24. Changes to DB2 CLP and system commands

Command	Summary of changes with upgrade impact
db2advis	This command can now make recommendations to convert standard tables with XML columns to multidimensional clustering (MDC) tables, and to use compression on indexes.
db2cat	For non-partitioned indexes, the command output now includes the new column <code>indexTbsp</code> to indicate the tablespace ID. For partitioned indexes, the command output includes the new column <code>indexTbsp</code> to indicate the tablespace ID and the new column <code>indexObjId</code> to indicate the index object ID. In base tables, the dependent MQTs output now includes a new field called MQT Flags.
db2ckmig	This command is deprecated and might be removed in a future release. Use the <code>db2ckupgrade</code> command instead.
db2ckupgrade	This command replaces the <code>db2ckmig</code> command. This command checks for type-1 indexes and generates a script file using the <code>REORG TABLE</code> command to convert type-1 indexes to type-2 indexes. Type-1 indexes are not supported in DB2 Version 9.7. See “Converting type-1 indexes to type-2 indexes” on page 44 for details. This command now requires that the instance owning the databases that you want to verify is running. You no longer have to stop the instance to run this command. If the instance is not started, the <code>db2ckupgrade</code> command returns the <code>SQL1032N</code> error message.
db2dart	The <code>/DD</code> parameter now includes inline length data as part of the formatted table data.
db2expln, db2exmig, db2jdbcbind, db2sqljbind, db2sqljcustomize and db2rbind	Due to changes in the DB2 authorization model, the <code>SYSADM</code> group is no longer authorized to perform these commands. The <code>UPGRADE DATABASE</code> command grants <code>DBADM</code> authority to the <code>SYSADM</code> group so that there is no upgrade impact. However, for these commands, you should review all of the changes in authorization and grant any required authorization to users. If you create databases in DB2 Version 9.7, you have to grant the required authorization to users that need to run these commands or grant <code>DBADM</code> authority to the <code>SYSADM</code> group to maintain the same authorization as in previous releases.
db2gpmap	The output generated by this command is larger due to the increase of the distribution map size.

Table 24. Changes to DB2 CLP and system commands (continued)

Command	Summary of changes with upgrade impact
db2icrt	<p>On Linux and UNIX operating systems, these commands now write to a new file called <i>INSTHOME</i>/sql11ib/log/db2instance.log, where <i>INSTHOME</i> is the home directory for the instance owner. This file acts as a history file that records the instance activities such as create (db2icrt), drop (db2idrop), update (db2iupdt) and upgrade (db2iupgrade) and is only intended for DB2 support use.</p>
db2idrop	<p>In partitioned database environments, issue the db2idrop command from any database partition only once. The db2idrop command updates the global registry for all available database partitions and removes the instance from the registry. If a database partition is not available, you will receive the DBI1165E error message and the DBI1383I warning message.</p> <p>On Linux and UNIX operating systems, these commands now write to a new file called <i>INSTHOME</i>/sql11ib/log/db2instance.log, where <i>INSTHOME</i> is the home directory for the instance owner. This file acts as a history file that records the instance activities such as create (db2icrt), drop (db2idrop), update (db2iupdt) and upgrade (db2iupgrade) and is only intended for DB2 support use.</p>
db2imigr	<p>This command is deprecated and might be removed in a future release. Use the db2iupgrade command instead.</p>
db2iupgrade and db2iupdt	<p>The db2iupgrade command replaces the db2imigr command.</p> <p>If you issue the db2iupgrade or db2iupdt command with the -u parameter to indicate a different fenced ID than the previously defined fenced ID, this different fenced ID overwrites the previously defined one. In previous releases, the -u parameter was ignored.</p> <p>In a partitioned database environment, the db2iupgrade and db2iupdt commands update the global registry for all available database partitions. This means if the instance is updated or upgraded successfully, the command removes the instance from the DB2 registry under the DB2 copy location where the instance was running and adds it under the DB2 copy location from which you are running the command. If a database partition is not available, you will receive the DBI1165E error message and the DBI1383I warning message.</p> <p>On Linux and UNIX operating systems, these commands now write to a new file called <i>INSTHOME</i>/sql11ib/log/db2instance.log, where <i>INSTHOME</i> is the home directory for the instance owner. This file acts as a history file that records the instance activities such as create (db2icrt), drop (db2idrop), update (db2iupdt) and upgrade (db2iupgrade) and is only intended for DB2 support use.</p>
db2look	<p>Although you can run this command with the same authorization that was required before upgrading your data server, now you might require SECADM authority, ACCESSCTRL authority, DATAACCESS authority, or all of them in some cases. The -x parameter now generates the authorization DDL for all of the new database authorities.</p>

Table 24. Changes to DB2 CLP and system commands (continued)

Command	Summary of changes with upgrade impact
db2pd	<p>The command output now includes information about scan sharing for tables and block indexes on MDC tables. The information that this command returns differs for individual shared scans, sharing sets, or sharing groups.</p> <p>For automatic storage table spaces, the -storagepaths parameter output now shows the new PathID and PathState columns to indicate the storage path identifier and current state. Also, the -tablespaces parameter output now shows the new PathDropped and PathID columns to indicate whether one or more containers reside on a storage path that have been dropped and the storage path identifier for the container.</p> <p>For temporary tables, the -tcbstats parameter output now shows three new SchemaNm, StoredBytes, and BytesSaved columns. Also, the -tablespaces parameter output now shows the new Max HWM column to indicate the maximum HWM for DMS table spaces since the instance was started. For SMS table spaces, the HWM and Max HWM columns do not display any value.</p> <p>For partitioned tables, the -tcbstats parameter output now shows the new PartID column to indicate the data partition ID in the TCB Index Information section and the TCB Index Stats section.</p>
db2relocatedb	<p>You can now specify in the configuration file a location for the mirrorlogpath, failarchpath, logarchmeth1, logarchmeth2, or overflowlogpath database configuration parameters. If you have set any of these database configuration parameters in the database that you want to relocate, you can specify a new location in the configuration file for any of these parameters. If you do not specify any of the new keywords, the db2relocatedb command maintains the original location as it did in previous releases.</p>
db2secv82	<p>The db2secv82 command is now discontinued. Use the db2extsec command instead to set the permissions for DB2 objects such as files, directories, network shares, registry keys, and services.</p>
db2uiddl	<p>The db2uiddl command is now discontinued. This command generated a script with CREATE UNIQUE INDEX statements to convert unique indexes created on your database before DB2 UDB Version 5. If you ran the db2uiddl command after you upgraded your databases to a pre-Version 9.7 DB2 release, you do not have to run this command again before your databases are upgraded to DB2 Version 9.7.</p> <p>If you are converting type-1 indexes to type-2 indexes, you are also converting the unique indexes created on your database before DB2 UDB Version 5 and you do not have to run the db2uiddl command.</p>
db2_deinstall	<p>If you specify the -F TEXT_SEARCH parameter and you have one or more instances configured as DB2 Text Search instance services on the DB2 copy that you are uninstalling, this command returns the DBI1325E error message.</p>
installFixPack	<p>If you have one or more instances configured as DB2 Text Search instance services on the DB2 copy that you updating, this command issues the db2ts STOP FOR TEXT command for each instance to stop the Text Search instance service. If stopping the Text Search instance service fails, the installFixPack command returns DBI1325E error message.</p>

Table 24. Changes to DB2 CLP and system commands (continued)

Command	Summary of changes with upgrade impact
BIND, DECOMPOSE XML DOCUMENT, DESCRIBE, EXPORT, IMPORT, LOAD, PRECOMPILE, REBIND, REGISTER XMLSCHEMA, REGISTER XSROBJECT, RUNSTATS, and UPDATE XMLSCHEMA	<p>Due to changes in the DB2 authorization model, the SYSADM group is no longer authorized to run these commands. Additionally, in some cases where DBADM authority used to be required, DATAACCESS authority is now sufficient. The UPGRADE DATABASE command grants DBADM authority to the SYSADM group and grants DATAACCESS authority to users holding DBADM authority including the SYSADM group so that there is no upgrade impact. However, for these commands, you should review all of the changes in authorization and grant any required authorization to users.</p> <p>If you create databases in DB2 Version 9.7, grant the required authorization to users that need to run these commands. You can grant DBADM authority to the SYSADM group and grant DATAACCESS authority to users holding DBADM authority to maintain the same authorization as in previous releases.</p>
CREATE DATABASE	<p>For new databases created on DB2 Version 9.7, the new cur_commit configuration parameter is set to ON so that currently committed behavior is enabled on cursor stability scans. A returned result set operating under cursor stability isolation level set might be different than in previous releases. See “Currently committed semantics improve concurrency” in <i>Troubleshooting and Tuning Database Performance</i> for details on how currently committed behavior works.</p> <p>When you create a database in DB2 Version 9.7, the DB2DETAILDEADLOCK event monitor is created by default as in previous releases. However, it is deprecated and might be discontinued in a future release. You should disable DB2DETAILDEADLOCK and remove it. Use the new lock event monitor instead before the DB2DETAILDEADLOCK event monitor is discontinued. See “Monitoring locking data” in <i>Database Monitoring Guide and Reference</i> for details.</p>
DEACTIVATE DATABASE	<p>If you enable the read capability on HADR standby databases, this command sets the database in super exclusive mode to prevent new connections to the database and waits for all users to disconnect before stopping the database. When the last user disconnects, the database is stopped.</p> <p>If you must stop the database immediately, you can issue the FORCE APPLICATIONS ALL command after the DEACTIVATE DATABASE command to end all existing connections. In previous releases, the DEACTIVATE DATABASE command stopped HADR standby databases because no user connections were allowed.</p>
DECOMPOSE XML DOCUMENTS	<p>If you issue this new command from a Version 9.7 client, the command will complete successfully when you connect to Version 9.7 databases but it will return SQL0440N error message when you connect to Pre-Version 9.7 databases. This command calls the XDB_DECOMP_XML_FROM_QUERY system-defined procedure available in Version 9.7. Pre-Version 9.7 databases do not have the correct version of this procedure which causes this command to fail.</p> <p>When you connect to Pre-Version 9.7 databases from Version 9.7 clients, use the DECOMPOSE XML DOCUMENT command instead. The DECOMPOSE XML DOCUMENTS command decomposes XML documents stored in a binary or XML column while the DECOMPOSE XML DOCUMENT command decomposes XML documents stored in files.</p>

Table 24. Changes to DB2 CLP and system commands (continued)

Command	Summary of changes with upgrade impact
DESCRIBE	<p>The DESCRIBE command with the INDEXES FOR TABLE parameter now lists indexes over XML data and text search indexes in addition to relational indexes. The output displays additional columns, a lists for partitioned indexes, and a list for non-partitioned indexes. Refer to the <i>Command Reference</i> for details about the additional columns.</p> <p>If you use the new RELATIONAL DATA INDEXES FOR TABLE parameter to list only relational indexes, the output displays the same columns as in previous releases.</p> <p>The DESCRIBE INDEXES command output now shows an new INDEX PARTITIONING column to indicate whether an index is partitioned or non-partitioned.</p> <p>The DESCRIBE DATA PARTITIONS command with SHOW DETAIL parameter now shows the new IndexTblSpId column to indicate the table space identifier where index data is stored.</p> <p>Refer to the BIND command row for additional authorization changes that also apply to the DESCRIBE command.</p>
EXPORT	<p>The Worksheet file format (WSF) is deprecated and support will be removed in a future release. Start using a supported file format instead of WSF before support is removed.</p> <p>Refer to the BIND command row for additional authorization changes that also apply to the EXPORT command.</p>
GET AUTHORIZATIONS	<p>The GET AUTHORIZATIONS command is discontinued in DB2 Version 9.7. Use the AUTH_LIST_AUTHORITIES_FOR_AUTHID table function instead.</p>
GET DB CFG	<p>For HADR standby databases with the read capability enabled, this command output now shows the value DATABASE – READ ENABLED for Rollforward pending to indicate that the database is in rollforward pending state and has the read capability enabled. In previous releases, this command showed the value DATABASE for Rollforward pending.</p>
GET DBM CFG	<p>This command now list the values of new parameters to configure Secure Sockets Layer (SSL) support. See Table 11 on page 31 for details.</p>
GET SNAPSHOT	<p>For the TABLESPACES parameter, this command now indicates whether a DMS table space is using reclaimable storage.</p>

Table 24. Changes to DB2 CLP and system commands (continued)

Command	Summary of changes with upgrade impact
IMPORT and LOAD	<p>You can import or load from files exported in previous releases provided that you did not export columns with user-defined and system-defined data types that are unsupported in DB2 Version 9.7. See “Verifying that your databases are ready for upgrade” on page 46 for a list of reserved and unsupported data types.</p> <p>You must also manage changes to the IMPORT and LOAD command that impact importing or loading files that you exported in previous releases. Refer to the <i>Command Reference</i> for details about changes to the IMPORT and LOAD command.</p> <p>The Worksheet file format (WSF) is deprecated and support will be removed in a future release. Start using a supported file format instead of WSF before support is removed. After the tables are exported using a supported file format, continue to use the IMPORT and LOAD command to populate your tables.</p> <p>The CREATE and REPLACE_CREATE modes of the IMPORT command are deprecated. Use DDL scripts that you developed or generated with the db2look command to create the table before you issue the IMPORT command.</p> <p>Refer to the BIND command row for additional authorization changes that also apply to the DESCRIBE command.</p> <p>If you issue the LOAD command with the REPLACE mode and the RESETDICTIONARY keyword on a table that has XML data in a Version 9.7 XML storage object and row compression enabled, this command now builds a compression dictionary for the XML data in addition to the dictionary for the table data. The compression dictionary for the XML data is stored in the XML storage object. The automatic compression dictionary creation (ADC) now builds a compression dictionary for the XML data as part of the table data population operations performed by the INSERT, IMPORT with mode INSERT, LOAD with mode INSERT, and REDISTRIBUTE DATABASE PARTITION GROUP commands.</p>
INSPECT	<p>If you issue the INSPECT command with the ROWCOMPESTIMATE TABLE parameter on a table has XML data in a Version 9.7 XML storage object and row compression enabled, this command now builds a compression dictionary for the XML data in addition to the dictionary for the table data. The compression dictionary for the XML data is stored in the XML storage object.</p>
LIST TABLESPACE CONTAINERS and LIST TABLESPACES	<p>These commands and related APIs have been deprecated and might be removed in a future release. Start using the MON_GET_TABLESPACE or the MON_GET_CONTAINER table functions instead. These table functions return more information than was provided by the deprecated commands and APIs. See Upgrade impact from DB2(r) API changes for details about the deprecated APIs.</p>
MIGRATE DATABASE	<p>This command is deprecated. Use the UPGRADE DATABASE command instead.</p>

Table 24. Changes to DB2 CLP and system commands (continued)

Command	Summary of changes with upgrade impact
REDISTRIBUTE DATABASE PARTITION GROUP	<p>If you issue this command without the NOT ROLLFORWARD RECOVERABLE parameter, ADC now builds a compression dictionary for the XML data in a Version 9.7 XML storage object on all database partitions without a dictionary as part of the table data population operations performed by this command provided that row compression is enabled. After the compression dictionary is built, XML data is compressed as well as table data. If you issue this command with the NOT ROLLFORWARD RECOVERABLE parameter, ADC now builds a compression dictionary for the XML data in a Version 9.7 XML storage object on new database partitions without a dictionary as part of the table data population operations performed by this command. ADC will not build a compression dictionary on existing database partitions that receive new data.</p>
REORG INDEXES/TABLE	<p>The LONGLOBDATA keyword now converts existing LOB data into inlined LOB data in addition to reorganizing long fields and LOB data. In previous releases, this parameter was used to reorganize long fields and LOB data. See Chapter 28, “Adopting new DB2 Version 9.7 functionality in database applications and routines,” on page 203 for details about inlined LOB data.</p> <p>The CONVERT keyword is deprecated because type-1 indexes are discontinued. You should convert your type-1 indexes to type-2 indexes before upgrading your data server. See “Converting type-1 indexes to type-2 indexes” on page 44 for details.</p> <p>If you specify the INPLACE keyword without NOTRUNCATE TABLE, you might want to adjust your outage window because the time locks are held on the table is reduced during the truncate table phase. The truncate table phase is faster than in previous releases, especially on sparsely populated tables.</p> <p>If you issue the REORG TABLE command with the LONGLOBDATA keyword and the KEEPDICTIONARY or RESETDICTIONARY keyword on a table has XML data in a Version 9.7 XML storage object and row compression is enabled, this command now builds a compression dictionary for the XML data in addition to the dictionary for the table data. The compression dictionary for the XML data is stored in the XML storage object object.</p>
REORGCHK	<p>The output of this command now includes table statistics, index statistics, and recommendations for table or index reorganization at the partition level. The output displays one line for each table and one line for each data partition only for partitioned tables.</p>
RUNSTATS	<p>DB2 Version 9.7 now supports SYSTEM sampling on statistical views and provides improvements on the performance of BERNOULLI sampling. See Chapter 10, “Adopting new Version 9.7 functionality in upgraded databases,” on page 107 for details.</p> <p>When you run this command on tables with LOB columns, it now collects statistics for the average length of column and number of null values in a column. Refer to the <i>Command Reference</i> for additional details.</p> <p>Refer to the BIND command row for additional authorization changes that also apply to the RUNSTATS command.</p>

Table 24. Changes to DB2 CLP and system commands (continued)

Command	Summary of changes with upgrade impact
SET WORKLOAD	Due to changes in the DB2 authorization model, the SYSADM group is no longer authorized to perform this command. If you do not have ACCESSCTRL, DATAACCESS, WLMADM, SECADM, or DBADM authority, you will receive an error when running this command.

On the Windows Vista operating system, to perform administration tasks that require Local Administrator authority, you must run your scripts from a DB2 command prompt with full administrator privileges. Launch the **Command Window - Administrator** shortcut to get a DB2 command prompt with full administrator privileges. If extended security is enabled on the Windows Vista operating system, you also need to log on to the system with a user that is a member of the DB2ADMNS group to launch this shortcut.

Upgrade impact from SQL statement changes

The changes to SQL statements in DB2 Version 9.7 can impact your existing applications and scripts after you upgrade to DB2 Version 9.7.

The changes to SQL statements include new default behaviors and modifications to statement output. In addition, some statements are discontinued. The following table lists the changes that impact applications and scripts:

Table 25. Changes to SQL statements

SQL statement	Summary of changes with upgrade impact
ALTER FUNCTION, ALTER HISTOGRAM TEMPLATE, ALTER METHOD, ALTER NICKNAME, ALTER PROCEDURE, ALTER SEQUENCE, ALTER SERVER, ALTER TABLE, ALTER TYPE (Structured), ALTER USER MAPPING, ALTER VIEW, ALTER WRAPPER, and ALTER XSROBJECT	<p>Due to changes in the DB2 authorization model, the SYSADM group is no longer authorized to run these statements. The UPGRADE DATABASE command grants DBADM authority to the SYSADM group so that there is no upgrade impact. However, for these statements, you should review all of the changes in authorization and grant any required authorization to users.</p> <p>If you create databases in DB2 Version 9.7, grant the required authorization to users that need to run these statements or explicitly grant DBADM authority to the SYSADM group to maintain the same authorization as in previous releases.</p> <p>Soft invalidation is supported on ALTER FUNCTION and ALTER VIEW statements when the <code>DB2_DDL_SOFT_INVALID</code> registry variable is set to ON. Refer to "Automatic invalidation and revalidation of database objects" in <i>Database Administration Concepts and Configuration Reference</i> for details about soft invalidation semantics.</p>

Table 25. Changes to SQL statements (continued)

SQL statement	Summary of changes with upgrade impact
ALTER SERVICE CLASS, ALTER THRESHOLD, ALTER WORK ACTION SET, ALTER WORK CLASS SET, ALTER WORKLOAD, CREATE HISTOGRAM TEMPLATE, CREATE SERVICE CLASS, CREATE THRESHOLD, CREATE WORK ACTION SET, CREATE WORK CLASS SET, and CREATE WORKLOAD	<p>Due to changes in the DB2 authorization model, the SYSADM group is no longer authorized to run these statements. The UPGRADE DATABASE command grants DBADM authority to the SYSADM group so that there is no upgrade impact. However, for these statements, you should review all of the changes in authorization and grant any required authorization to users.</p> <p>However, if you create databases in DB2 Version 9.7, grant the required authorization to users that need to run these statements or explicitly grant DBADM or WLMADM authority to the SYSADM group to maintain the same authorization as in previous releases.</p>
ALTER DATABASE	<p>The ADD STORAGE ON clause can now be issued with the ALTER DATABASE statement to enable automatic storage on existing databases. In previous releases, the SQL20317N error message was returned. This statement creates the SQLSGF.1 and SQLSGF.2 files that are required for maintaining automatic storage. Refer to Chapter 10, "Adopting new Version 9.7 functionality in upgraded databases," on page 107 for details on how to enable automatic storage.</p>

Table 25. Changes to SQL statements (continued)

SQL statement	Summary of changes with upgrade impact
ALTER TABLE	<p>The COMPRESS clause now enables compression for data in the table and the XML storage object. If you issue the ALTER TABLE statement with the ADD COLUMN clause to add an XML column, the compression of the XML data depends on whether data row compression is enabled for the table.</p> <p>If you issue the ALTER TABLE statement with the COMPRESS YES clause in a table with XML columns created in a pre-Version 9.7 release, only table data compression is supported. To convert the XML storage object to the new Version 9.7 format that supports compression on XML data, re-create the table. See “Converting XML storage objects to the Version 9.7 format” on page 103 for details.</p> <p>The COMPRESS and DATA CAPTURE CHANGES clauses can now be used together. SQL replication now supports tables with data row compression enabled as source and target tables. You will not receive an error if you specify both clauses with this statement.</p> <p>The ALTER TABLE statement with the ATTACH PARTITION clause now requires that columns defined with structured, XML, or LOB data type in the partitioned table and the table that you attaching have the same INLINE LENGTH value. If you issue this statement in a table with XML columns, maintenance is performed in all XML column path indexes as part of the statement processing. Starting with Version 9.7 Fix Pack 1, the ALTER TABLE statement with the DETACH PARTITION clause proceeds concurrently with certain queries. See “Data partition detach phases” in <i>Partitioning and Clustering Guide</i> for details.</p> <p>If you issue the ALTER TABLE statement with the DETACH PARTITION clause in a table with XML columns, XML column path indexes for the separate table are created as part of the statement processing.</p> <p>XML columns are now supported in partitioned tables. The ALTER TABLE statement with the ADD COLUMN, ATTACH PARTITION, or DETACH PARTITION clause no longer returns SQL1242N error message with reason code 2 when attaching or detaching a table with XML columns to partitioned tables, or adding XML columns to partitioned tables.</p> <p>You can now add columns with the XML type to MDC tables. In previous releases, the SQL1242N error message with reason code 1 was returned.</p> <p>Refer to the ALTER FUNCTION statement row for additional authorization changes that also apply to this command.</p>

Table 25. Changes to SQL statements (continued)

SQL statement	Summary of changes with upgrade impact
CALL, CREATE ALIAS, CREATE EVENT MONITOR, CREATE FUNCTION, CREATE INDEX, CREATE INDEX EXTENSION, CREATE METHOD, CREATE NICKNAME, CREATE PROCEDURE, CREATE SCHEMA, CREATE SEQUENCE, CREATE SERVER, CREATE TABLE, CREATE TRANSFORM, CREATE TRIGGER, CREATE TYPE, CREATE TYPE MAPPING, CREATE VARIABLE, FLUSH EVENT MONITOR, FLUSH OPTIMIZATION PROFILE CACHE, FLUSH PACKAGE CACHE, RENAME, SET EVENT MONITOR STATE, and SET INTEGRITY	<p>Due to changes in the DB2 authorization model, the SYSADM group is no longer authorized to run these statements. The UPGRADE DATABASE command grants DBADM authority to the SYSADM group so that there is no upgrade impact. However, for these statements, you should review all of the changes in authorization and grant any required authorization to users.</p> <p>If you create databases in DB2 Version 9.7, grant the required authorization to users that need to run these statements or explicitly grant DBADM authority to the SYSADM group to maintain the same authorization as in previous releases.</p> <p>Soft invalidation is supported on CREATE ALIAS, CREATE FUNCTION, CREATE NICKNAME, CREATE PROCEDURE, CREATE TRIGGER, and CREATE VARIABLE statements when the DB2_DDL_SOFT_INVALID registry variable is set to ON. Refer to “Automatic invalidation and revalidation of database objects” in <i>Database Administration Concepts and Configuration Reference</i> for details about soft invalidation semantics.</p> <p>If you create databases in DB2 Version 9.7, the auto_reval configuration parameter is set to DEFERRED by default so that revalidation deferred semantics are enabled. This setting allows you to create views, functions, stored procedures, triggers, and global variables even if they reference objects that do not exist or are invalid.</p>

Table 25. Changes to SQL statements (continued)

SQL statement	Summary of changes with upgrade impact
CREATE VIEW, DECLARE CURSOR, DECLARE GLOBAL TEMPORARY TABLE, DELETE, INSERT, MERGE, REFRESH TABLE, SELECT INTO, SET variable, and UPDATE	<p>Due to changes in the DB2 authorization model, the SYSADM group is no longer authorized to run these statements and in some cases the DBADM authority requirement is replaced by the DATAACCESS authority. The UPGRADE DATABASE command grants DBADM authority to the SYSADM group and grants DATAACCESS authority to users holding DBADM authority so that there is no upgrade impact. However, for these statements, you should review all of the changes in authorization and grant any required authorization to users.</p> <p>If you create databases in DB2 Version 9.7, grant the required authorization to users that need to run these statements. You can explicitly grant DBADM authority to the SYSADM group, DATAACCESS authority to users holding DBADM authority, or both to maintain the same authorization as in previous releases.</p> <p>Soft invalidation is supported on CREATE VIEW statement when the DB2_DDL_SOFT_INVAL registry variable is set to ON. Refer to “Automatic invalidation and revalidation of database objects” in <i>Database Administration Concepts and Configuration Reference</i> for details about soft invalidation semantics.</p>
CALL	<p>If you specified DEFAULT as a parameter or variable name, DEFAULT now resolves to the DEFAULT keyword. In previous releases, DEFAULT resolved to a variable or parameter called DEFAULT. Delimit in double quotes the word DEFAULT in upper case letters when you want to resolve to a variable or parameter called DEFAULT as shown in this example:</p> <pre>CALL (P1,"DEFAULT");</pre> <p>Refer to the CALL statement row for additional authorization changes that also apply to this command.</p>
COMMENT	<p>Due to changes in the DB2 authorization model, the SYSADM group is no longer authorized to run these statements. The UPGRADE DATABASE command grants DBADM authority to the SYSADM group so that there is no upgrade impact. However, for these statements, you should review all of the changes in authorization and grant any required authorization to users. Also, WLMADM authority is required to comment on workload management objects.</p> <p>If you create databases in DB2 Version 9.7, grant the required authorization to users that need to run these statements or explicitly grant DBADM authority to the SYSADM group to maintain the same authorization as in previous releases.</p>
CONNECT	<p>If you enable the read capability on HADR standby databases, you can connect to them except when the database is switching roles on takeover. In previous releases, the SQL1776N error message was returned.</p>

Table 25. Changes to SQL statements (continued)

SQL statement	Summary of changes with upgrade impact
CREATE INDEX	<p>When you create indexes in tables with data row compression enabled, indexes will have compression enabled by default. If data row compression is disabled, index compression is disabled. Use the COMPRESS NO clause if you must disable index compression.</p> <p>Online index create is now supported when you create indexes on an XML column. This support requires additional log space and index table space. The additional requirements are the same for online index create on relational indexes.</p> <p>When you create indexes for partitioned tables, by default, they are created as partitioned indexes. If you must create nonpartitioned indexes, use the NOT PARTITIONED clause. Partitioned indexes are not supported for spatial indexes and unique indexes with index key columns that are not a superset of the range-partitioning key columns.</p> <p>Refer to the CALL statement row for additional authorization changes that also apply to this command.</p>
CREATE PROCEDURE (SQL) statement	<p>If you declare variables called TRUE or FALSE in an SQL procedure, this statement returns the SQL0408N error message and fails to create the procedure. Modify the procedure code and change the variable name before you re-issue the CREATE PROCEDURE statement.</p> <p>In SQL procedures, when you assign XML data to input and output parameters of XML type or local variables of XML type, the XML data is now passed by reference. In previous releases, the XML data was passed by value. Therefore, some operations using XML data can return results that are different from the results returned by the same operations in previous releases.</p>

Table 25. Changes to SQL statements (continued)

SQL statement	Summary of changes with upgrade impact
CREATE TABLE	<p>The COMPRESS clause now enables compression for data in the table and the XML storage object. If you issue the ALTER TABLE statement with the ADD COLUMN clause to add an XML column, the compression of the XML data depends on whether data row compression is enabled for the table.</p> <p>The COMPRESS and DATA CAPTURE CHANGES clauses can now be used together. SQL replication now supports tables with data row compression enabled as source and target tables. You will not receive an error if you specify both clauses with this statement.</p> <p>You can now specify columns using the XML type when creating partitioned tables. In previous releases, the SQL1242N error message with reason code 2 was returned. XML data placement on a partitioned table follows the long data placement rules. The XML storage objects and the XML regions indexes are partitioned in the same manner as table data.</p> <p>You can now specify columns with the XML type and use the ORGANIZE BY clause in the CREATE TABLE statement. If you specify columns with the XML type in the ORGANIZE BY clause, you will receive the SQL0350N error message. In previous releases, the SQL1242N error message with reason code 1 was returned.</p> <p>XML columns are now supported in partitioned tables. The CREATE TABLE statement no longer returns SQL1242N error message with reason code 4 when you define a distribution key and XML columns in the same table. However, XML columns cannot be used to define the distribution key.</p> <p>When you create a partitioned table, the system generated indexes for primary keys and unique constraints are always partitioned by default. If you create a partitioned table with XML columns, the XML regions indexes are always partitioned, and the XML column paths indexes are always non-partitioned. Refer to SQL Reference for details.</p> <p>The NOT LOGGED option only applies to LOB data that is not inlined. In upgraded databases, the LOB data is implicitly inlined when the length is less than the LOB descriptor size. In this case, the NOT LOGGED option does not apply to implicitly inlined LOB data.</p> <p>The COMPRESS YES clause now enables index compression in addition to data row compression when you create new indexes on the table. Use the CREATE INDEX statement with the COMPRESS NO clause if you must disable index compression.</p> <p>Refer to the CALL statement row for additional authorization changes that also apply to this command.</p>
CREATE TABLESPACE	<p>Whether a database has automatic storage enabled or not, reclaimable storage is the default when you create a new DMS table space. Reclaimable storage is a new functionality that allows you to reduce the table space high water mark and reclaim space. Refer to Chapter 10, "Adopting new Version 9.7 functionality in upgraded databases," on page 107 for details about reclaimable storage in upgraded databases.</p>

Table 25. Changes to SQL statements (continued)

SQL statement	Summary of changes with upgrade impact
DECLARE GLOBAL TEMPORARY TABLE	<p>Data row compression is enabled by default in temporary tables. However, rows are only compressed when the database manager determines that there is a performance gain. Consequently, index compression is also enabled for indexes that you create on temporary tables.</p> <p>The DISTRIBUTE BY clause replaces the PARTITIONING KEY clause. For compatibility with previous releases, you can still use the PARTITIONING KEY clause.</p> <p>Refer to the CREATE VIEW statement row for additional authorization changes that also apply to this command.</p>
DROP	<p>Due to changes in the DB2 authorization model, the SYSADM group is no longer authorized to run this statement. Also, you might be required to hold SECADM, SQLADM, or WLMADM authority to drop certain objects. Refer to the <i>SQL Reference</i> for details about all the changes in authorization.</p> <p>Soft invalidation is supported on the DROP ALIAS, DROP FUNCTION, DROP TRIGGER, DROP VARIABLE, and DROP VIEW statements when the DB2_DDL_SOFT_INVAL registry variable is set to ON. Refer to “Automatic invalidation and revalidation of database objects” in <i>Database Administration Concepts and Configuration Reference</i> for details about soft invalidation semantics.</p> <p>Starting with Version 9.7 Fix Pack 1, the DROP WORKLOAD statement has a new optional clause RESTRICT. Dropping a workload that has activity thresholds or a work action sets applied to it now returns the SQL2090N error message. The RESTRICT clause is the default. You must drop any activity thresholds or a work action sets applied to the workload before dropping the workload.</p>
GRANT and REVOKE	<p>The authorization to run these statements has changed. You are now required to hold SECADM authority to grant ACCESSCTRL, DATAACCESS, DBADM, or SECADM authority, and to hold ACCESSCTRL or SECADM authority to grant other authorities or privileges on global variables, indexes, packages, routines, schemas, sequences, servers, tables, views, nicknames, workloads, and XSR objects. The same changes apply to the REVOKE statements.</p> <p>The UPGRADE DATABASE command explicitly grants DBADM authority to the SYSADM group and explicitly grants ACCESSCTRL and DATAACCESS authorities to users holding DBADM authority. After the upgrade, grant SECADM authority to users that need to grant ACCESSCTRL, DATAACCESS, DBADM, or SECADM authority.</p>
Prefix operator + (unary plus) and prefix operator - (unary minus)	<p>The result data type of a prefix operator + or prefix operator - with an untyped parameter marker as an operand is now DECFLOAT(34). In previous releases, the result data type was DOUBLE.</p> <p>Applications expecting a DOUBLE might receive the SQL0440N error message. Use the DOUBLE scalar function to explicitly cast the result data type of a prefix operator + or prefix operator - with an untyped parameter marker as an operand.</p>

Table 25. Changes to SQL statements (continued)

SQL statement	Summary of changes with upgrade impact
SET variable	<p>If you specified DEFAULT as a value to assign to a variable, DEFAULT now resolves to the DEFAULT keyword. If the usage of the DEFAULT keyword is invalid, it now returns the SQL0584N error message. In previous releases, DEFAULT as a value resolved to a variable or parameter called DEFAULT and only returned SQL0584N when such a variable or parameter was not defined. Delimit in double quotes the word DEFAULT in uppercase letters when you want to resolve to variables or parameters called DEFAULT as shown in this example:</p> <pre>CREATE PROCEDURE FOO(IN DEFAULT INTEGER) BEGIN DECLARE V0 INTEGER DEFAULT 1; SET V0 = "DEFAULT"; RETURN V0; END%</pre> <p>Refer to the CREATE VIEW statement row for additional authorization changes that also apply to this command.</p>
Untyped NULL keyword in expressions	<p>You can now specify an untyped NULL keyword anywhere in an expression. If you use identifiers called NULL in SQL statements without being fully qualified or delimited such as column names or parameter names, the identifier specification could resolve to the keyword NULL instead of the identifier name. To avoid conflict with the untyped NULL keyword, fully qualify or delimit columns named NULL in SQL statements as shown in the following example:</p> <pre>SELECT MY_TAB.NULL FROM MY_TAB SELECT "NULL" FROM MY_TAB</pre>

Refer to the *SQL Reference, Volume 2* guide for details about any of the statements.

Upgrade impact from system catalog changes

In DB2 Version 9.7, system catalog objects are modified to support new functionality. These changes can impact your existing applications and scripts after you upgrade to DB2 Version 9.7.

System catalog views

In general, modifications to the existing catalog views consist of new columns, changed column data types, or longer column lengths. The following table lists the system catalog view changes between DB2 Version 9.7 and DB2 Version 9.5 that impact applications and scripts:

Table 26. Changes to system catalog views

View name	Summary of changes with upgrade impact
SYSCAT.ATTRIBUTES	<p>The SCALE column now has a value for TIMESTAMP data type to indicate the number of digits of fractional seconds.</p> <p>New ATTR_TPEMODULENAME, SOURCE_TPEMODULENAME, TARGET_TPEMODULENAME, TPEMODULENAME, columns are added.</p>
SYSCAT.BUFFERPOOLS	New column NUMBLOCKPAGES is added.

Table 26. Changes to system catalog views (continued)

View name	Summary of changes with upgrade impact
SYSCAT.CONSTDEP, SYSCAT.INDEXDEP, SYSCAT.INDEXEXTENSIONDEP, SYSCAT.PACKAGEDEP, SYSCAT.TABDEP, SYSCAT.TRIGDEP, SYSCAT.XSROBJECTDEP	New columns BMODULEID and BMODULENAME are added.
and SYSSTAT.COLUMNS	New column PCT_INLINED is added. The AVGCOLLEN column returns has a value of -1 only when statistics have not been collected. It no longer returns -1 when the column data type is a long field or LOB as in previous releases. The SCALE column now has a value for TIMESTAMP data type to indicate the number of digits of fractional seconds. The INLINE_LENGTH column in SYSCAT.COLUMNS returns the LOB maximum size for inline length. During database upgrade, this column is set to the maximum LOB descriptor size.
SYSCAT.DATAPARTITIONS	New columns ACTIVE_BLOCKS INDEX_TBSPACEID, AVGCOMPRESSEDROWSIZE, AVGWCOMPRESSORATIO, AVGWROWSIZE, CARD, FPAGES, NPAGES, OVERFLOW, PCTPAGESSAVED, PCTROWSCOMPRESSED, and STATS_TIME are added.
SYSCAT.DATATYPEDEP	New columns BMODULEID, BMODULENAME, TPEMODULEID, and TPEMODULENAME are added.
SYSCAT.DATATYPES	The SCALE column now has a value for TIMESTAMP data type to indicate the number of digits of fractional seconds. New columns ARRAY_LENGTH, ARRAYINDEXTYPESHEMA, ARRAYINDEXTYPENAME, ARRAYINDEXTYPEID, ARRAYINDEXTYPELENGTH, PUBLISHED, SOURCEMODULEID, SOURCEMODULENAME, TPEMODULEID, TPEMODULENAME, and VALID are added.
SYSCAT.DBAUTH	New columns ACCESSCTRLAUTH, DATAACCESSAUTH, EXPLAINADMAUTH, SQLADMAUTH, and WLMADMAUTH are added.
SYSCAT.HISTOGRAM TEMPLATEUSE	New column WORKLOADNAME is added.
SYSCAT.INDEXEXTENSION PARMS	The SCALE column now has a value for TIMESTAMP data type to indicate the number of digits of fractional seconds.
SYSCAT.INDEXES, SYSSTAT.INDEXES	New columns AVGLEAFKEYSIZE, AVGNLEAFKEYSIZE, LASTUSED, COMPRESSION, and PCTPAGESSAVED are added.
SYSCAT.PACKAGES	New columns ANONBLOCK, ALTER_TIME, APREUSE, CONCURRENTACCESSRESOLUTION, EXTENDEDINDICATOR, DBPARTITIONNUM, PKGID, LASTUSED, and VALID are added. The PKG_CREATE_TIME column has been renamed CREATE_TIME. The PKG_CREATE_TIME column is still available for compatibility with previous releases.
SYSCAT.PARTITIONMAPS	The size of the distribution map in the SYSCAT.PARTITIONMAPS view has increased from 4 096 entries to 32 768 entries. During database upgrade, the size of the distribution map is increased to 32 768 entries. Use the new db2GetPmap API to read the distribution map. See “Upgrade impact from DB2 API changes” on page 151 for details.
SYSCAT.ROUTINEDEP	New columns BMODULEID, BMODULENAME, ROUTINEMODULEID, and ROUTINEMODULENAME are added.

Table 26. Changes to system catalog views (continued)

View name	Summary of changes with upgrade impact
SYSCAT.ROUTINEPARMS	The SCALE column now has a value for TIMESTAMP data type to indicate the number of digits of fractional seconds. New columns DEFAULT, ROUTINEMODULEID, ROUTINEMODULENAME, TARGET_TYPEMODULENAME, and TYPEMODULENAME are added.
SYSCAT.ROUTINES, SYSSAT.ROUTINES	New columns DIALECT, MODULEROUTINEIMPLEMENTED, PUBLISHED, RETURN_TYPEMODULE, ROUTINEMODULEID, and ROUTINEMODULENAME are added.
SYSCAT.SEQUENCES	New columns BASE_SEQNAME and BASE_SEQSCHEMA are added. This catalog view now returns rows that represent an alias for a sequence in addition to rows that represent a sequence.
SYSCAT.SERVICECLASSES	New columns BUFFERPOOLPRIORITY, COLLECTACTMETRICS, and COLLECTREQMETRICS are added.
SYSCAT.TABDETACHEDDEP	New columns BMODULEID and TABMODULENAME are added.
SYSCAT.TABLES	New columns LASTUSED, LOGGED, ONCOMMIT, and ONROLLBACK are added.
SYSCAT.TABLESPACES	New column DATAPRIORITY is added. The DATATYPE column value U now indicates created or declared temporary tables. In previous releases, the U value indicated only declared temporary tables.
SYSCAT.THRESHOLDS	New columns CHECK_INTERVAL, REMAPSCID, and VIOLATIONRECORDLOGGED are added.
SYSCAT.VARIABLEDEP	New BMODULEID, VARMODULEID, BMODULENAME, and VARMODULENAME columns are added.
SYSCAT.VARIABLES	The SCALE column now has a value for TIMESTAMP data type to indicate the number of digits of fractional seconds. New columns PUBLISHED, VARMODULEID, VARMODULENAME, VALID, and READONLY are added.
SYSCAT.WORKLOADS	New columns COLLECTACTMETRICS, COLLECTDEADLOCK, COLLECTLOCKTIMEOUT, COLLECTLOCKWAIT, and LOCKWAITVALUE are added.

System-defined built-in routines

Changes to system-defined built-in routines include new routines, new parameters and changes in behavior. The following table lists the new routines and changes to existing routines between DB2 Version 9.7 and DB2 Version 9.5 that impact applications and scripts:

Table 27. Changes to system-defined built-in routines

Routine name	Summary of changes with upgrade impact
ADD_MONTHS, EXTRACT, LAST_DAY, MONTHS_BETWEEN, NEXT_DAY, and TRUNC_TIMESTAMP	If you have user-defined functions with the same names as these new built-in functions and they are not fully qualified in your queries, the resolution path might result in these new built-in functions being called. Fully qualify your user-defined function calls with the schema name or start using these new built-in functions instead.
ARRAY_DELETE, ARRAY_FIRST, ARRAY_LAST, ARRAY_NEXT, ARRAY_PRIOR, CURSOR_ROWCOUNT	If you have user-defined functions with the same names as these new built-in functions and they are not fully qualified in your queries, the resolution path might result in these new built-in functions being called. Fully qualify your user-defined function calls with the schema name or start using these new built-in functions instead.

Table 27. Changes to system-defined built-in routines (continued)

Routine name	Summary of changes with upgrade impact
DAYNAME, DECFLOAT_FORMAT, INITCAP, INSTR, LPAD, LOCATE_IN_STRING, MONTHNAME, RPAD, TO_CLOB, TO_NUMBER, TO_TIMESTAMP	If you have user-defined functions with the same names as these new built-in functions and they are not fully qualified in your queries, the resolution path might result in these new built-in functions being called. Fully qualify your user-defined function calls with the schema name or start using these new built-in functions instead.
CHAR (decimal-expression)	<p>If the dec_to_char_fmt database configuration parameter is set to NEW, the CHAR function now returns a fixed-length character string representation of a decimal number without leading zeros and without a decimal separator when the decimal part is zero.</p> <p>In previous releases the character string included leading zeros and a decimal separator when the decimal part was zero. The dec_to_char_fmt database configuration parameter must be set to V95 to obtain the same result as in previous releases. See Table 13 on page 33 for details.</p>
DOUBLE_PRECISION	If you have user-defined functions with the same names as these new built-in functions and they are not fully qualified in your queries, the resolution path might result in these new built-in functions being called. Fully qualify your user-defined function calls with the schema name or start using these new built-in functions instead.
SYSIBM.DOUBLE	<p>The SYSIBM.DOUBLE function returns a double-precision floating-point representation of a number or a string representation of a number.</p> <p>When you call this function with a <i>string-expression</i> argument, the function now removes only leading and trailing blanks from the argument string-expression. In previous releases this function removed leading and trailing blanks, tabs, carriage returns, new lines, and line feeds.</p> <p>If you are using the default setting for SQL path, function calls to DOUBLE resolves to SYSIBM.DOUBLE over SYSFUN.DOUBLE. The SYSFUN.DOUBLE is still available. If you rely on the previous release behavior for this function, fully qualify references to SYSFUN.DOUBLE.</p>
LONG_VARGRAPHIC, LONG_VARCHAR	LONG_VARGRAPHIC and LONG_VARCHAR scalar functions are deprecated. Although the use of these scalar functions is still supported in the current release, consider using other scalar functions such as CHAR, VARCHAR, and CLOB. The LONG VARCHAR and LONG VARGRAPHIC data types are deprecated and might be removed in a future release. See Table 15 on page 36 for details.

System-defined administrative routines and views

Changes to system-defined administrative routines and views include new columns returned and new routines and views. In addition, all of the

administrative routines with names that start with SNAPSHOT have been deprecated since DB2 Version 9.1. The following table lists the administrative routine and view changes between DB2 Version 9.7 and DB2 Version 9.5 that impact applications and scripts:

Table 28. Changes to system-defined administrative routines and views

Routine or view name	Summary of changes with upgrade impact
ADMIN_GET_TAB_COMPRESS_INFO and ADMINTABCOMPRESSINFO	<p>The ADMIN_GET_TAB_COMPRESS_INFO table function is deprecated. The ADMINTABCOMPRESSINFO view now calls ADMIN_GET_TAB_COMPRESS_INFO_V97 and returns the new additional columns. Use the new ADMIN_GET_TAB_COMPRESS_INFO_V97 function or the ADMINTABCOMPRESSINFO view which returns the new column OBJECT_TYPE. The value of the EXPAND_DICT_SIZE column now includes the sum of the current and historical dictionary sizes.</p> <p>Both ADMIN_GET_TAB_COMPRESS_INFO_V97 and ADMINTABCOMPRESSINFO can now return additional rows for information on compression dictionaries for XML data stored in the XML storage object.</p>
ADMIN_GET_TAB_INFO_V95	<p>The ADMIN_GET_TAB_INFO_V95 table function is deprecated. The ADMINTABINFO view now calls ADMIN_GET_TAB_INFO_V97 and returns the new additional columns. Use the new ADMIN_GET_TAB_INFO_V97 function or the ADMINTABINFO view which returns the new columns RECLAIMABLE_SPACE, XML_DICTIONARY_SIZE, and XML_RECORD_TYPE. Also, the description of the DICTIONARY_SIZE column is changed to indicate that the value of this column now includes the sum of the current and historical dictionary sizes.</p>
ADMIN_EST_INLINE_LENGTH	<p>This new function provides a method to estimate the inline length of XML, CLOB, BLOB and DBCLOB columns. XML columns are only supported when they are created in DB2 Version 9.7 because new statistics are collected to estimate the inline length. This function returns the value -2 for XML columns created in previous releases because it cannot estimate the inline length without the new statistics.</p> <p>To convert the XML storage object to the new Version 9.7 format that supports this function, re-create the table. The new SYSPROC.ADMIN_MOVE_TABLE system-defined procedure allows you to re-create the table while the data remains online and available for access. See “Moving tables using the ADMIN_MOVE_TABLE procedure” in Data Movement Utilities Guide and Reference.</p>
AUDIT_ARCHIVE, AUDIT_DELIM_EXTRACT, AUDIT_LIST_LOGS	<p>In DB2 Version 9.7, the UPGRADE DATABASE command revokes the EXECUTE privilege from PUBLIC on the audit routines, AUDIT_LIST_LOGS, AUDIT_DELIM_EXTRACT, and AUDIT_ARCHIVE. For each authorization ID holding SECADM authority, the UPGRADE DATABASE command explicitly grants the EXECUTE privilege on the audit routines by granting the SYSROLE_AUTH_SECADM system role. You need to explicitly grant the EXECUTE privilege on these audit routines to any users that do not hold SECADM authority but need to call these routines.</p>
DBCFCG, GET_DB_CONFIG	<p>Selecting from the DBMCFG view or the GET_DBM_CONFIG table function now returns new database configuration manager parameters listed in Table 13 on page 33.</p>
DBMCFG, GET_DBM_CONFIG	<p>Selecting from the DBMCFG view or the GET_DBM_CONFIG table function now returns new database configuration manager parameters listed in Table 11 on page 31.</p>

Table 28. Changes to system-defined administrative routines and views (continued)

Routine or view name	Summary of changes with upgrade impact
REBIND_ROUTINE_PACKAGE	In DB2 Version 9.7, the <i>options</i> parameter of VARCHAR(1024) data type replaces the <i>resolve</i> parameter of VARCHAR(12) data type. For compatibility with previous releases, 'ANY' and 'CONSERVATIVE' are accepted as valid values for <i>options</i> . 'ANY' is a synonym for 'RESOLVE ANY' and 'CONSERVATIVE' is a synonym for 'RESOLVE CONSERVATIVE'. If you are using parameter markers or host variables when you call this procedure from your applications, adjust the length of your variables.
REORGCHK_IX_STATS	This function has been modified and now returns the new column DATAPARTITIONNAME.
REORGCHK_TB_STATS	This function has been modified and now returns the new column DATAPARTITIONNAME.
SNAP_GET_STORAGE_PATHS, SNAPSTORAGE_PATHS	The SNAP_GET_STORAGE_PATHS function is deprecated. The new SNAP_GET_STORAGE_PATHS_V97 function replaces SNAP_GET_STORAGE_PATHS and returns the new DB_STORAGE_PATH_WITH_DPE and DB_STORAGE_PATH_STATE columns. The SNAPSTORAGE_PATHS view now calls SNAP_GET_STORAGE_PATHS_V97 and returns the new additional columns.
SNAP_GET_TBSP_PART_V91, SNAPTbsp_PART	The SNAP_GET_TBSP_PART_V91 function is deprecated. The new SNAP_GET_TBSP_PART_V97 function replaces SNAP_GET_TBSP_PART_V91 and returns the new TBSP_PATHS_DROPPED column. The SNAPTbsp_PART view now calls SNAP_GET_TBSP_PART_V97 and returns the new additional column.
XDB_DECOMP_XML_FROM_QUERY	This new procedure can process multiple XML documents stored in binary or XML columns in one call and does not require serialization of the XML documents. The new DECOMPOSE XML DOCUMENTS command calls this new procedure. Pre-Version 9.7 databases do not have the correct version of this procedure, this command fails to run on such databases. See Chapter 28, "Adopting new DB2 Version 9.7 functionality in database applications and routines," on page 203 for details on how to use this new procedure.
WLM_GET_ACTIVITY_DETAILS	This table function is deprecated. Use the new MON_GET_ACTIVITY_DETAILS table function instead. This new function replaces WLM_GET_ACTIVITY_DETAILS and returns additional columns WL_WORK_ACTION_SET_ID and WL_WORK_CLASS_ID to support new WLM functionality.
WLM_GET_SERVICE_CLASS_AGENTS	This table function is deprecated. Use the new WLM_GET_SERVICE_CLASS_AGENTS_V97 table function instead. This new function replaces WLM_GET_SERVICE_CLASS_AGENTS and returns the new column EVENT_OBJECT_NAME.
WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES	This table function is deprecated. Use the new WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES_V97 table function instead. This new function replaces WLM_GET_SERVICE_CLASS_AGENTS and returns the new column ADDRESS.
WLM_GET_SERVICE_SUBCLASS_STATS	This table function is deprecated. Use the new WLM_GET_SERVICE_SUBCLASS_STATS_V97 table function instead. This new function replaces WLM_GET_SERVICE_SUBCLASS_STATS and returns additional column UOW_TOTAL_TIME_TOP to support new WLM functionality.

Table 28. Changes to system-defined administrative routines and views (continued)

Routine or view name	Summary of changes with upgrade impact
WLM_GET_WORKLOAD_STATS	This table function is deprecated. Use the new WLM_GET_WORKLOAD_STATS_V97 table function instead. This new function replaces WLM_GET_WORKLOAD_STATS and returns additional column UOW_TOTAL_TIME_TOP to support new WLM functionality.
WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES	This table function is deprecated. Use the new WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES_V97 table function instead. This new function replaces WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES and returns the new column EXECUTABLE_ID.

Review the list of the deprecated administrative routines and their replacement routines or views in “Deprecated SQL administrative routines and their replacement routines or views” in *Administrative Routines and Views* to determine additional changes that might impact your applications and scripts.

System catalog changes between pre-Version 9.7 releases

If you are upgrading from DB2 Version 9.1 or DB2 UDB Version 8, the following additional system catalog changes between pre-Version 9.7 releases can also impact your applications and scripts:

- System catalog changes between DB2 Version 9.5 and DB2 Version 9.1.
- System catalog views and system-defined routines changes between DB2 Version 9.1 and DB2 UDB Version 8.

Chapter 23. Upgrade essentials for routines

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 Chapter 22, “Upgrade essentials for database applications,” on page 145 could also impact your routines.

Development software support

The information about development software support in Chapter 22, “Upgrade essentials for database applications,” on page 145 applies to external stored procedures and user-defined functions (UDFs).

Implicit casting

After function invocation, the database manager must decide which function in a group of like-named functions is the “best fit”. A comparison of the data types of the arguments with the defined data types of the parameters of the functions under consideration forms the basis for this decision. An untyped parameter marker or an untyped NULL constant argument accepts any parameter type as a best fit.

This change to support implicit casting impacts function resolution that involves modified system built-in functions and any new functions that you create using these arguments.

XML data is passed by reference in SQL routines

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

Refer to “XML data is passed by reference in stored procedures” in *What’s New for DB2 Version 9.7* for details about this difference in behavior.

SQL procedures

SQL procedures that you created in DB2 UDB Version 8.1 will run in DB2 Version 9.7 if you upgrade from a DB2 UDB Version 8 32-bit instance to a DB2 Version 9.7 32-bit instance, provided that they do not reference any unsupported functionality. This also applies if you upgrade from a DB2 UDB Version 8 64-bit instance to a DB2 Version 9.7 64-bit instance. However, if you upgrade from a DB2 UDB Version 8.1 32-bit instance to a DB2 Version 9.7 64-bit instance, your SQL procedures do not run because the 64-bit DB2 engine cannot load the 32-bit libraries associated to these procedures. You must drop and re-create these SQL procedures.

If you created SQL procedures in DB2 UDB Version 8.2 or later and upgrade your databases to DB2 Version 9.7, your SQL procedures will function successfully provided that they do not reference any unsupported functionality.

Unfenced external routines

During database upgrade to DB2 Version 9.7 on Linux and UNIX operating systems, all external unfenced routines that have no dependency on the

DB2 engine libraries (libdb2e.a or libdb2apie.a) are altered to FENCED and NOT THREADSAFE so you can safely run these routines under the new multithreaded database manager. Running external routines defined as NOT FENCED and THREADSAFE in the new multithreaded database manager that are not thread safe can yield incorrect results, database corruption, or abnormal termination of the database manager. Refer to “Upgrading C, C++, and COBOL routines” on page 194 for details about how to manage this change.

32-bit external routines

The implementation for LOB locators depends on DB2 database product installed. Also, you can use LOB locators only in unfenced routines. If you upgrade from a DB2 UDB Version 8 32-bit instance to a DB2 Version 9.7 64-bit instance, you must rebuild 32-bit external routines that use LOB locators as 64-bit unfenced routine libraries.

Support for default function entry points in external routine libraries is deprecated in DB2 Version 9.1. If you upgraded from a DB2 UDB Version 8 32-bit instance on AIX or Windows operating systems, you should specify an explicit entry point for your routine library.

31-bit external routines (Linux on zSeries)

All upgrade considerations for 32-bit external routines also apply to 31-bit external routines running on a DB2 database on Linux on zSeries.

Java external routines

DB2 Version 9.7 installs the 32-bit IBM Software Developer’s Kit (SDK) for Java 6 by default on Linux on x86 and Windows (when DB2 Version 9.7 32-bit product is installed) operating systems. For all other supported operating systems, DB2 Version 9.7 installs a 64-bit SDK for Java 6.

If you upgrade an instance to DB2 Version 9.7, the **jdk_path** database manager configuration parameter is set to the installation path of SDK for Java 6. The following table indicates the possible values for **jdk_path** :

Table 29. jdk_path database manager configuration parameter settings

DB2 Version 9.7 instance	Operating System	jdk_path value
32-bit instance	Linux	INSTHOME/sql11ib/java/jdk32
64-bit instance	Linux and UNIX	INSTHOME/sql11ib/java/jdk64
32-bit or 64-bit instance	Windows	DB2PATH\java\jdk

In DB2 Version 9.7 64-bit instances, Java external routines require that the **jdk_path** parameter is set to a 64-bit SDK for Java installation path to run successfully. A DB2 Version 9.7 64-bit instance cannot load a 32-bit JVM.

The IBM Software Developer’s Kit (SDK) for Java 1.4.2 is deprecated and might be discontinued in a future release.

Starting with DB2 Version 9.5, the default JDBC driver to run JDBC routines is the IBM Data Server Driver for JDBC and SQLJ. See “Upgrading Java routines” on page 196 for details on how to manage this change.

Upgrade of routines from DB2 Version 9.1 or DB2 UDB Version 8

If you are upgrading from DB2 Version 9.1 or earlier, review the changes in routine development support, changes to support new functionality, unsupported functionality, and deprecated functionality that might also impact your routines:

- Changes between DB2 Version 9.5 and DB2 Version 9.1 that impact routines .
- Changes between DB2 Version 9.1 and DB2 UDB Version 8 that impact routines.

Chapter 24. Pre-upgrade tasks for database applications and routines

Before you upgrade your database applications and routines, you should perform certain tasks to help you ensure a successful upgrade.

Prepare for the upgrade of your database applications and routines by performing the following tasks:

1. Review upgrade essentials for database applications to determine which changes might impact your database applications.
2. Review upgrade essentials for routines to determine which changes might impact your routines.
3. Plan your upgrade strategy.
4. Upgrade your operating system to a supported level if necessary.
5. Upgrade your development software to a supported level if necessary.
6. Perform benchmark tests on your database applications and routines in your production environment and save these baseline results to compare with benchmark test results after the upgrade.
7. Optional: Upgrade your client or install a Version 9.7 application driver if your application requires one. Although DB2 Version 9.7 server provides connectivity support for earlier clients, using a Version 9.7 client eliminates any limitations and incompatibilities between releases.
8. Test your database applications in a DB2 Version 9.7 testing environment. If testing is successful, you do not need to upgrade your applications. However, review the upgrading database applications task and consider performing any steps that can help you improve performance.
9. Test your routines in a DB2 Version 9.7 testing environment. If testing is successful, you do not need to upgrade your routines. However, review the upgrading routines task and consider performing any steps that can help you improve performance.

Chapter 25. Upgrading database applications

Upgrading your existing database applications to DB2 Version 9.7 involves managing the changes between DB2 Version 9.7 and previous releases that impact these applications and verifying that these applications function as expected. Managing these changes might require that you modify your applications code and rebuild your applications.

You only need to modify your application code to manage changes in DB2 Version 9.7 that impact your applications, to remove the use of deprecated or discontinued functionality in DB2 Version 9.7, or to use new functionality.

Prerequisites

- Ensure that you have access to a DB2 Version 9.7 server, including instances and databases. The DB2 server can be part of a testing environment.
- Ensure that you meet the installation requirements for DB2 database products.
- Ensure that the development software is at a version level that is supported by DB2 database products.
- Perform the pre-upgrade tasks for database applications.

Restriction

This procedure only applies to database applications programmed in C, C++, COBOL, FORTRAN, Java, Perl, PHP, REXX, and .NET languages.

Procedure

To upgrade your database applications to DB2 Version 9.7:

1. If you identified changed DB2 commands, changed SQL statements, and changed system catalog views and built-in functions that impact your applications, edit your application code or scripts to modify:
 - DB2 CLP and system command syntax
 - SQL statements syntax
 - SQL statements using catalog views and SQL Administrative views and routines
 - SQL statements using target tables for write-to-table event monitors
 - User defined routine names that are not fully qualified with a schema name
 - DB2 API calls
 - Application programming interface calls such as JDBC, ODBC and CLI
 - If your applications or scripts read from the command output, modify them to read the changed output format.
2. If you identified changes specific to the development environment that impact your applications, modify them to support these changes. Upgrade your:
 - Embedded SQL applications
 - CLI applications
 - Java applications that use the IBM Data Server Driver for JDBC and SQLJ or that use the DB2 JDBC Type 2 driver
 - ADO and .NET applications

- Scripts that use DB2 CLP commands and SQL statements
 - 32-bit database applications to run on 64-bit instances
3. Rebuild all changed database applications programmed in C/C++, COBOL, FORTRAN, and REXX, using the appropriate DB2 build file and specifying the appropriate DB2 shared library path as shown in Table 21 on page 149.
 4. Test your database applications to verify your changes and to ensure that they run as expected using DB2 Version 9.7.

After upgrading your database applications, perform the recommended post-upgrade tasks for database applications to ensure that your upgrade was successful.

Upgrading embedded SQL applications

Upgrading your existing embedded SQL applications to DB2 Version 9.7 involves managing the changes between DB2 Version 9.7 and previous releases that impact these applications and verifying that these applications function as expected.

Prerequisites

- Ensure that you have access to a DB2 Version 9.7 server, including instances and databases. The DB2 server can be part of a testing environment.
- Ensure that the C, C++, COBOL, FORTRAN, or REXX development software is at a version level that is supported by DB2 database products.
- Perform previous steps in the upgrading database applications task.

Restriction

This procedure only applies to database applications programmed in C, C++, COBOL, FORTRAN, and REXX.

Procedure

To upgrade your embedded SQL applications to DB2 Version 9.7:

1. If you modified the library path environment variables, ensure that those variables include the correct DB2 shared library path for your applications as shown in Table 22 on page 149. The environment variables listed in this table specify additional paths to enable your applications to find the appropriate DB2 shared library at runtime (in most cases).

On the Linux operating system: if you link an application using the RPATH link option without also specifying the RUNPATH link option, the LD_LIBRARY_PATH environment variable will be ignored at application run time, which can cause your application to fail.
2. Test your embedded SQL applications in a DB2 Version 9.7 testing environment. If testing is successful, you do not need to perform any additional steps.
3. If you bound your embedded applications using the BIND command with the BLOCKING ALL or BLOCKING UNAMBIGUOUS clause to enable the blocking of cursors for LOB columns, ensure that the **instance_memory** or **database_memory** database configuration parameters are set to AUTOMATIC or increase their numeric value to account for the extra memory usage. If you cannot increase these database configuration parameters, you have the following options:

- Rebind them using the BIND command specifying **BLOCKING NO** or precompile them using the PRECOMPILE command specifying the **SQLRULES STD** command parameter. The **BLOCKING NO** clause disables blocking of all cursors in the application. The **SQLRULES STD** command parameter might have other effects than disabling blocking cursors.
 - Modify the application source code and declare the cursor with the FOR UPDATE clause to disable blocking.
4. To explicitly specify the correct DB2 shared library path for your applications, do one of the following:
 - If the application source code is available, rebuild the application. Specify the required DB2 shared library path as shown in Table 21 on page 149. This is the best option.
 - Create a wrapper script to run your application. In the wrapper script, explicitly set the library path environment variable to the required DB2 shared library path as shown in Table 22 on page 149.
 - If you do not have the original source code available, run the db2chglbpath command to update the embedded runtime library path within the binary code of your application. This command is provided as-is and should therefore be considered a last resort.

After upgrading your embedded SQL applications, perform the remaining steps in the upgrading database applications task.

Upgrading CLI applications

Upgrading your existing CLI applications to DB2 Version 9.7 involves managing the changes between DB2 Version 9.7 and previous releases that impact these applications, such as operating system support changes, development software support changes, the bit-width of the application, and the bit-width of the DB2 instance on which you deploy the applications.

Prerequisites

- Ensure that you have access to a DB2 Version 9.7 server, including instances and databases. The DB2 server can be part of a testing environment.
- Ensure that the C and C++ development software is a version that is supported by DB2 database products.
- Perform previous steps in the Chapter 25, “Upgrading database applications,” on page 183 task.

Restriction

This procedure only applies to database applications programmed in C or C++ using the CLI interface.

Procedure

To upgrade your CLI applications to DB2 Version 9.7:

1. If you modified the library path environment variables, ensure that those variables include the correct DB2 shared library path for your applications, as shown in Table 22 on page 149. You can use the environment variables listed in this table to specify additional paths that enable your applications to find the appropriate DB2 shared library at run time (in most cases).

On Linux operating systems only: If you link an application using the RPATH link option without also specifying the RUNPATH link option, the

`LD_LIBRARY_PATH` environment variable is ignored at application run time, which can cause your application to fail.

2. If you have set the `CLISchema` configuration keyword in your `db2cli.ini` file, set the `SysSchema` configuration keyword instead. The `CLISchema` configuration keyword is discontinued since DB2 Version 9.5.

`SysSchema = alternative schema`

3. Test your CLI applications in a DB2 Version 9.7 testing environment. If testing is successful, you do not need to perform the remaining steps.
4. If you set the **BlockLobs** CLI configuration keyword to 1 and your application gets the error message `SQL0973N`, perform one of the following actions:
 - Set the **database_memory** configuration parameter to `AUTOMATIC`. This is the best option.
 - Reset the **BlockLobs** CLI configuration keyword to 0.
 - Bind LOB values directly to buffers instead of using LOB locators.

Your client requires more memory to receive LOB data because this cursor blocking setting using the **BlockLobs** keyword sends all the LOB values immediately to your client after the row data is sent.

5. Review “CLI and ODBC function summary” in *Call Level Interface Guide and Reference, Volume 2* to determine if you are using any of the deprecated functions in ODBC 3.0 and modify your application to use the replacement function instead. Although this version of DB2 CLI continues to support these functions, using the replacement functions ensures that your applications conform to the latest standards.
6. Explicitly specify the correct DB2 shared library path for your applications by performing one of the following actions:
 - If the application source code is available, rebuild the applications. Specify the required DB2 shared library path as shown in Table 21 on page 149. This is the best option.
 - Create a wrapper script to run your applications. In the wrapper script, explicitly set the library path environment variable to the required DB2 shared library path as shown in Table 22 on page 149.
 - If you do not have the original source code available, run the `db2chglbpath` command to update the embedded runtime library path within the binary code of your applications. This command is provided as-is and should therefore be considered a last resort.

After upgrading your CLI applications, perform the remaining steps in the Chapter 25, “Upgrading database applications,” on page 183 task.

Upgrading Java applications that use IBM Data Server Driver for JDBC and SQLJ

Upgrading Java applications that use previous releases of the IBM Data Server Driver for JDBC and SQLJ involves managing the changes between different releases of this driver and the changes in DB2 Version 9.7 that can impact these applications.

Prerequisites

- Review the upgrade essentials for applications to identify key changes that might impact your Java database applications.

- Ensure that you have access to a DB2 Version 9.7 server, including instances and databases. The DB2 server can be part of a testing environment.
- Ensure that the Java application development software and IBM Data Server Driver for JDBC and SQLJ are at a version level that is supported by DB2 database products.
- Perform the previous steps in the upgrading database applications task.

Restrictions

- The minimum supported Java SDK for the IBM Data Server Driver for JDBC and SQLJ Version 3.57 is Java SDK 1.4.2.
- The minimum supported Java SDK for the IBM Data Server Driver for JDBC and SQLJ Version 4.7 is Java SDK 6.
- This procedure applies only to Java applications using the IBM Data Server Driver for JDBC and SQLJ.

Procedure

To upgrade your Java database applications using the IBM Data Server Driver for JDBC and SQLJ to DB2 Version 9.7:

1. Install the version of the IBM Data Server Driver for JDBC and SQLJ that corresponds to the version and fix pack level of your DB2 copy. See “Java software support for DB2 products” in *Installing DB2 Servers* for a complete list of supported drivers.
 - If you use methods in JDBC 4.0 or earlier specifications in your applications, install IBM Data Server Driver for JDBC and SQLJ Version 4.7 or later.
 - If you use methods in JDBC 3.0 or earlier specifications in your applications, install IBM Data Server Driver for JDBC and SQLJ Version 3.57 or later
2. If you are upgrading applications that use the IBM DB2 Driver for JDBC and SQLJ before Version 3.57, update your applications to manage the following differences between this driver and the IBM Data Server Driver for JDBC and SQLJ that you installed:
 - The IBM Data Server Driver for JDBC and SQLJ Version 4.0 or later returns a different result set than previous releases of this driver for the `ResultSetMetaData.getColumnNames` and `ResultSetMetaData.getColumnLabels` methods to conform to the JDBC 4.0 standard. If you need these methods to return the same result set returned with the IBM DB2 Driver for JDBC and SQLJ before Version 4.0, you can set the `useJDBC4ColumnNameAndLabelSemantics` property to `DB2BaseDataSource.NO` in the `Connection` or `DataSource` object.
 - The IBM Data Server Driver for JDBC and SQLJ allows you to invoke the `commit ()` or `rollback ()` methods if the connection is in auto-commit mode and your application does not receive an exception anymore.
 - If the JNDI store is not available due to JNDI bind or lookup failures, then the IBM Data Server Driver for JDBC and SQLJ attempts a connection to the standard server and port properties of a `datasource` even when the `datasource` is configured to use JNDI for client reroute primaries and alternates. The driver now accumulates warnings to indicate these failures with the original message from the exception appended. In previous releases, the driver did not use this information and threw exceptions.
3. If you are upgrading applications that use IBM DB2 Driver for JDBC and SQLJ before Version 3.1, update your applications to manage the following differences between this driver and the IBM Data Server Driver for JDBC and SQLJ that you installed:

- If your applications connect to a DB2 server that supports progressive streaming, also known as dynamic data format, retrieving LOBs using progressive streaming is enabled by default starting with IBM DB2 Driver for JDBC and SQLJ Version 3.2 to provide improved performance to your Java database applications. You need to manage any changes in semantic that might impact your applications. Refer to LOBs in JDBC applications with the IBM Data Server Driver for JDBC and SQLJ in *Developing Java Applications* for details.
 - If your application connects to a DB2 server that supports progressive streaming, and you want to continue using LOB locators instead of LOB retrieval using progressive streaming, set the `progressiveStreaming` property to `DB2BaseDataSource.NO` in the `Connection` or `DataSource` object.
 - As of Version 3.0, you need to set the `sendDataAsIs` property to indicate if you want the driver to do the data type conversion or not. To maintain the conversion of input parameter values to the target column data types, which was the default behavior before IBM DB2 Driver for JDBC and SQLJ Version 3.0, set the `sendDataAsIs` property to `false`. If you set the `sendDataAsIs` property to `true`, the driver converts to the data type indicated by the `setXXX` method regardless of the information in the `Connection` or `DataSource` object.
 - If you use the JDBC 1.0 method to update or delete data on a database server that supports multiple-row FETCH and you intent to update or delete a single row, modify your applications to use the method described in *Specifying updatability, scrollability, and holdability for ResultSets in JDBC applications in Developing Java Applications* to avoid updating or deleting multiple rows.
4. If you changed your Java application source code, rebuild your Java application. Refer to one of the following tasks in *Developing Java Applications* for details on how to rebuild them:
- Building JDBC applications
 - Building SQLJ applications

Upon completion of this task, your Java application should perform successfully using DB2 Version 9.7.

After upgrading your Java applications, perform the remaining steps in the upgrading database applications task.

Upgrading Java applications that use DB2 JDBC Type 2 driver

The DB2 JDBC Type 2 driver is deprecated. Although your Java applications that use the DB2 JDBC Type 2 driver will function successfully with DB2 Version 9.7, upgrading those applications to the IBM Data Server Driver for JDBC and SQLJ as soon as possible will help you to avoid lack of support in future releases.

Prerequisites

- Ensure that you have access to a DB2 Version 9.7 server, including instances and databases. The DB2 server can be part of a testing environment.
- Ensure that the Java application development software is at a version level that is supported by DB2 database products.
- Perform the previous steps in the upgrading database applications task.

Restriction

- The minimum supported Java SDK is Java SDK 1.4.2.

Procedure

To upgrade your Java database applications to DB2 Version 9.7:

1. Install the IBM Data Server Driver for JDBC and SQLJ. Refer to “Installing the IBM Data Server Driver for JDBC and SQLJ” in *Developing Java Applications*.
2. Update your Java applications to use the IBM Data Server Driver for JDBC and SQLJ. Refer to “Connecting to a data source using the DriverManager interface with the IBM Data Server Driver for JDBC and SQLJ” in *Developing Java Applications*.
3. Review the information in the following topics in *Developing Java Applications* to identify behavioral differences between drivers that might impact your Java application:
 - “Driver support for JDBC APIs”
 - “JDBC differences between the IBM Data Server Driver for JDBC and SQLJ and other DB2 JDBC drivers”
 - “SQLJ differences between the IBM Data Server Driver for JDBC and SQLJ and other DB2 JDBC drivers”
4. Modify your Java applications to resolve any issues created by the behavioral differences that you identified in the previous step. These changes might include modifying existing method calls and removing the use of unsupported functionality in DB2 Version 9.7.
5. If you changed your Java application source code in any of the previous steps, rebuild your Java applications. Refer to one of the following tasks in *Developing Java Applications* for details on how to rebuild them:
 - “Building JDBC applications”
 - “Building SQLJ applications”

After upgrading your Java applications, perform the remaining steps in the upgrading database applications task.

Upgrading ADO.NET applications

Upgrading your existing ADO.NET applications to DB2 Version 9.7 involves managing the changes between DB2 Version 9.7 and previous releases that impact these applications and verifying that these applications function as expected.

You do not have to upgrade ADO.NET applications that use the OLE DB .NET Data Provider or the ODBC .NET Data Provider to run with DB2 Version 9.7. However, upgrading these applications to the Data Server Provider for .NET can be beneficial for the following reasons:

- The Data Server Provider for .NET has a far more extensive set of APIs than the OLE DB and ODBC .NET data providers.
- Access to the DB2 database development productivity tools integrated with Visual Studio.
- Use of the Data Server Provider for .NET can bring significant performance improvements.

Prerequisites

- Ensure that you have access to a DB2 Version 9.7 server, including instances and databases. The DB2 server can be part of a testing environment.

- Ensure that a supported version of the Microsoft .NET Framework software is installed on the DB2 database client computer. See “Supported .NET development software” in *Developing ADO.NET and OLE DB Applications* .
- Perform the previous steps in the upgrading database applications task.

Procedure

To upgrade your ADO.NET applications to DB2 Version 9.7:

1. Review the support for the Data Server Provider for .NET and how to program your applications to use the Data Server Provider for .NET and determine what changes to make on your ADO.NET applications.
2. Rebuild your ADO.NET applications to use the Data Server Provider for .NET.

After upgrading your ADO.NET applications, perform the remaining steps in the upgrading database applications task.

Upgrading scripts

Upgrading your existing scripts that use DB2 command line processor (CLP) commands, DB2 system commands or SQL statements involves managing the changes between DB2 Version 9.7 and previous releases related to SQL statements, DB2 CLP and system commands, SQL Administrative views and routines, built-in functions, and catalog views.

Prerequisites

- Ensure that you have access to a DB2 Version 9.7 server, including instances and databases.
- Ensure that a DB2 Version 9.7 client is installed.
- Perform the previous steps in the upgrading database applications task.

Restriction

This procedure only applies to scripts that use DB2 CLP commands, DB2 system commands or SQL statements.

Procedure

To upgrade your scripts with DB2 CLP commands to DB2 Version 9.7:

1. Run your scripts to detect any incompatibilities with DB2 Version 9.7. If your scripts run successfully, you do not need to perform any additional steps. However, consider performing the remaining steps to remove deprecated functionality in DB2 Version 9.7 before they become discontinued or to use new command functionality.
2. Remove the DB2 CLP and system commands that display or update registry variables and configuration parameters that are deprecated or discontinued:
 - Deprecated and discontinued registry variables
 - Deprecated and discontinued database manager configuration parameters.
 - Deprecated and discontinued database configuration parameters
3. If your scripts perform snapshot or event monitoring, you need to modify your scripts to remove references to discontinued monitor elements or use a new name when they have been replaced by a new monitor element.
4. Determine the upgrade impact from system catalog changes. Using the changed views and routines requires that you:
 - Change the view names on your queries.

- Change column names in your queries for columns that have been renamed in the view or routine.
 - Remove column names from your queries for columns that are not available in the view or result sets from routines.
 - Replace * in your queries for a specific list of column names that you want to receive as a result set because the changed view result set has additional columns.
 - Change routines names and parameter names, and indicate new additional parameters.
 - Modify your script to process additional columns in a result set when calling a changed routine or querying a changed view that returns additional columns.
5. Test your scripts to ensure that they run as expected using DB2 Version 9.7.

After upgrading your scripts, perform the remaining steps in the upgrading database applications task.

Upgrading 32-bit database applications to run on 64-bit instances

Upgrading 32-bit database applications from a DB2 Version 8 32-bit instance to a DB2 Version 9.7 64-bit instance requires that your 32-bit database applications are linked to the appropriate shared library path to run them successfully.

You do not need to modify your 32-bit database applications if you linked them to the \$INSTHOME/sqllib/lib32 shared library path on Linux and UNIX or the DB2PATH\lib\Win32 shared library path on Windows, where INSTHOME is the instance home directory and DB2PATH is the location of the DB2 copy.

Prerequisites

- Ensure that you have access to a DB2 UDB Version 8 32-bit instance that you upgraded to a DB2 Version 9.7 64-bit instance that includes 32-bit shared libraries.
- Ensure that the development software is at a version level that is supported by DB2 database products.
- Perform the previous steps in the upgrading database applications task.

Restrictions

- This procedure applies only to 32-bit database applications programmed in C/C++, COBOL, FORTRAN, and REXX.

Procedure

To upgrade 32-bit database applications to run on a DB2 Version 9.7 64-bit instance:

1. Ensure that the library path environment variables include the correct DB2 shared library path for 32-bit libraries as shown in Table 22 on page 149, so that the correct library can be loaded at runtime.
2. Test your 32-bit applications in a DB2 Version 9.7 testing environment. If testing is successful, you do not need to perform any additional steps. However, consider performing step 4 on page 192 or 5 on page 192 if they apply to your applications, to improve their support by using the correct client and shared library path.
3. Perform any other steps in the following upgrade tasks that apply to your applications:

- Embedded SQL applications
 - CLI applications
 - Java applications that use the IBM Data Server Driver for JDBC and SQLJ or that use the DB2 JDBC Type 2 driver
 - ADO and .NET applications
4. Specify the correct library path by linking or rebuilding your 32-bit applications using the DB2 shared library paths for 32-bit libraries shown in Table 21 on page 149.
 5. Optional: If you no longer have the source code to rebuild your applications or the use of environmental variables is not possible, you can run the `db2chglbpath` command to change the DB2 shared library path to `$INSTHOME/sql1lib/lib32` on your application binary file as long as it has an embedded runtime path. The embedded runtime path can be changed to a new path with the same length or less.
 6. Test your 32-bit applications to ensure that they run as expected using DB2 Version 9.7.

After upgrading your 32-bit database applications, perform the remaining steps in the upgrading database applications task.

Chapter 26. Upgrading routines

Upgrading your existing routines to DB2 Version 9.7 involves managing the changes between DB2 Version 9.7 and previous releases that impact these routines and verifying that they function as expected. Managing these changes might require that you modify your routine code, rebuild your external routines, recreate your external routines in the database, and recreate SQL routines.

Test your routines in a DB2 Version 9.7 testing environment. If they run successfully, you are not required to change them. You only need to modify your routines to manage any changes between releases, to remove the use of discontinued or deprecated functionality in DB2 Version 9.7, or to use new functionality.

Prerequisites

- Review upgrade essentials for routines to identify any changes that apply to your routines.
- Ensure that you have access to upgraded DB2 Version 9.7 databases. These can be test databases.
- Ensure that you meet the installation requirements for DB2 database products. See “Installation requirements for DB2 database products” in *Installing DB2 Servers*.
- Ensure that the development software is at a version level that is supported by DB2 database products.
- Perform the pre-upgrade tasks for routines.
- Ensure that you have the necessary authorizations and privileges to use the ALTER FUNCTION or ALTER PROCEDURE statements. The authorizations allowed are listed in the *SQL Reference, Volume 2*.

Restriction

This procedure only applies to SQL routines and external routines programmed in C/C++, COBOL (procedures only), Java, and .NET languages.

Procedure

To upgrade your routines to DB2 Version 9.7 databases:

1. If you identified changes in DB2 Version 9.7 that impact your routines, edit your routine code and modify:
 - SQL statement syntax
 - SQL statements using SQL Administrative views and routines, built-in routines, and catalog views
 - User defined routine names that are not fully qualified with a schema names
 - Application programming interface calls such as JDBC and CLI
2. If you identified changes specific to the development environment that impact your routines, modify them to support these changes. Upgrade your:
 - C, C++, and COBOL routines
 - Java routines.
 - .NET CLR routines.

- SQL stored procedures, if you created your SQL procedures in DB2 Version 8.1 and you upgraded from a 32-bit DB2 Version 8 instance to a DB2 Version 9.7 64-bit instance.
 - 32-bit external routines to run on 64-bit instances.
3. Rebuild all changed external routine libraries or if you performed operating system or development software upgrades.
 4. Test your routines to verify your changes and to ensure that the routines run as expected using DB2 Version 9.7.

After upgrading your routines, perform the recommended post-upgrade tasks for routines.

Upgrading C, C++, and COBOL routines

Upgrading your existing C, C++, or COBOL routines to DB2 Version 9.7 involves managing the changes between DB2 Version 9.7 and previous releases that impact these routines and verifying that they function as expected.

Prerequisites

- Ensure that you have access to a DB2 Version 9.7 server, including instances and databases. The DB2 server can be part of a testing environment.
- Ensure that the C, C++, or COBOL routine development software are at a version level that is supported by DB2 database products by reviewing the following requirements:
 - “Support for external routine development in C” in *Administrative Routines and Views*
 - “Support for external routine development in C++” in *Administrative Routines and Views*
 - “Support for external procedure development in COBOL” in *Administrative Routines and Views*
- Ensure that you have the necessary authorizations and privileges to use the ALTER FUNCTION or ALTER PROCEDURE statements. The authorizations allowed are listed in the *SQL Reference, Volume 2*.
- Perform the previous steps in the upgrading routines task.

Restriction

This procedure only applies to external routines programmed in C/C++, and COBOL (procedures only).

Procedure

To upgrade a C, C++, or COBOL routine to DB2 Version 9.7, do the following:

1. If you upgraded to a DB2 Version 9.7 64-bit instance, change your routine libraries or routine definitions according to the following table:

Table 30. Upgrading C, C++, and COBOL routines to a Version 9.7 64-bit instance

Routine definition	Action
<i>unfenced</i> 32-bit routine library that use the DB2 engine library	<p>Rebuild the routine source code into a 64-bit library using the DB2 Version 9.7 bldrtn script and redeploy the library to the DB2 server. If LOB locators are referenced in the routine, you must rebuild your routines. You can determine most of the routines that reference lob locators by executing the following query:</p> <pre>SELECT DISTINCT a.routineschema, a.routinename, a.specificname FROM syscat.routines a, syscat.routineparms b WHERE a.specificname = b.specificname AND b.locator = 'Y' AND a.fenced = 'N'</pre> <p>An advantage of this approach is that using a 64-bit library results in better routine runtime performance than using a 32-bit library.</p>
<i>fenced</i> 32-bit routine library	<ul style="list-style-type: none"> Rebuild the routine source code into a 64-bit library using the DB2 Version 9.7 bldrtn scripts and redeploy the library to the DB2 server. If you cannot rebuild your routines, define the routine as not threadsafe using the ALTER PROCEDURE or ALTER FUNCTION statement with the NOT THREADSAFE clause.
upgraded from a Version 8 32-bit instance (AIX and Windows)	<p>You should specify a library entry point for any routine that relies on a default entry point using the ALTER PROCEDURE or ALTER FUNCTION statement. For example, to explicitly specify the entry point for an existing procedure use the following statement:</p> <pre>ALTER SPECIFIC PROCEDURE <i>schema-name.specific-name</i> EXTERNAL NAME '<i>library-name!function-name</i>'</pre> <p>where <i>library-name</i> is the library to be loaded and <i>function-name</i> is the explicit entry point for the function associated with the routine.</p>

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

- If you are using the cursor blocking and found any differences in the behavior of your C, C++, or COBOL routines, review the “Upgrading embedded SQL applications” on page 184 task to learn how to manage those differences.
- For routines that you did not rebuild but that you modified, rebind the routine packages to the target DB2 database.
- Determine if the external routines that were altered during database upgrade or the external routines that use the DB2 engine libraries can safely run as NOT FENCED and THREADSAFE. If you have external unfenced routines in your database, the UPGRADE DATABASE command performs the following actions:
 - Returns the SQL1349W warning message and writes the ADM4100W message to the administration notification log.
 - Redefines all your external unfenced routines that have no dependency on the DB2 engine library as FENCED and NOT THREADSAFE.
 - Creates a CLP script called alter_unfenced_dbname.db2 in the directory specified by the DIAGPATH database manager configuration parameter to redefine the affected routines as NOT FENCED and THREADSAFE.

If you can safely run the external routines altered by database upgrade as NOT FENCED and THREADSAFE, you can redefine them as NOT FENCED and THREADSAFE using the original CLP script or a modified version with just specific routines that you want to redefine. If you can run them as FENCED

and NOT THREADSAFE and the performance degradation that you experience is acceptable, you do not need to redefine your routines .

After upgrading your C, C++, or COBOL routines, perform the remaining steps in the upgrading routines task.

Upgrading Java routines

Upgrading your existing Java routines to DB2 Version 9.7 involves managing the changes between DB2 Version 9.7 and previous releases that impact these routines and ensure that these routines function as expected.

Prerequisites

The following prerequisites must be met to perform this task:

- Ensure that you have access to a DB2 Version 9.7 server, including instances and databases. The DB2 server can be a test system.
- Ensure that the Java routine development software is at a version level that is supported by DB2 database products. See “Supported Java routine development software” in *Developing User-defined Routines (SQL and External)*.
- Ensure that you are using supported DB2 drivers for JDBC and SQLJ APIs. See “Supported drivers for JDBC and SQLJ” in *Developing Java Applications*.
- Ensure that you have the necessary authorizations and privileges to use the ALTER FUNCTION or ALTER PROCEDURE statements. The authorizations allowed are listed in the *SQL Reference, Volume 2*.
- Perform the previous steps in the upgrading routines task.

Procedure

To upgrade your Java routines:

1. Ensure the **jdk_path** database manager configuration parameter specifies the installation path of the IBM Software Developer’s Kit (SDK) for Java that is installed on your DB2 server. Determine the current value of this parameter by issuing the following command:

```
db2 GET DBM CFG
```

By default the **jdk_path** database manager configuration parameter value is set during instance upgrade to the values shown in Table 29 on page 178 which are the installation path of SDK for Java 6.

If you must use an SDK for Java other than the one installed in your DB2 Version 9.7 copy, set this configuration parameter to the installation path of an SDK for Java with the same bit width as the DB2 instance by updating the **jdk_path** parameter:

```
db2 UPDATE DBM CFG USING jdk_path <SDKforJava-path>
```

However, setting the **jdk_path** parameter to the installation path of SDK for Java 1.4.2 is not recommended because SDK for Java 1.4.2 is deprecated and might be discontinued in a future release.

2. Set the DB2_USE_DB2JCCT2_JROUTINE registry variable to indicate the default JDBC driver to run Java routines. By default this registry variable is not set, which means the default JDBC driver is the IBM Data Server Driver for JDBC and SQLJ. This setting gives you access to functionality particular to this driver and use of XML parameters. Use the db2set command with the **-g** parameter to set the default JDBC driver for all instances running under the

same DB2 Version 9.7 copy:

Default driver	Command to set default driver
IBM DB2 JDBC Type 2 driver	db2set -g DB2_USE_DB2JCCT2_JROUTINE=NO
IBM Data Server Driver for JDBC and SQLJ	db2set -g DB2_USE_DB2JCCT2_JROUTINE=YES

Use the **-i** parameter instead of the **-g** parameter, to apply the registry variable setting to a specific instance.

3. Test your Java routines in your DB2 Version 9.7 database. If testing is successful and your Java routine perform as expected, you do not need to perform any additional steps.
4. If you are using the IBM Data Server Driver for JDBC and SQLJ and found any differences in the behavior of your Java routines, review “Upgrading Java applications that use IBM Data Server Driver for JDBC and SQLJ” on page 186 to learn how to manage those differences.
5. If the pre-upgrade value of the **jdk_path** parameter was the installation path of SDK for Java 1.4.2, manage any differences in behavior between SDK for Java 1.4.2 and SDK for Java 6.
6. Explicitly define your Java routines as fenced using the ALTER FUNCTION or ALTER PROCEDURE statement with the FENCED clause. All Java routines run as fenced, regardless of how you defined them, but defining your Java routine definitions as fenced improves routine manageability and maintenance.
7. Optional: If your Java routine class is included within a JAR file that has been installed into a DB2 instance using a specific JAR file ID, ensure that the Java class is resolved more quickly by the DB2 database manager by specifying the JAR file ID as part of the EXTERNAL NAME clause in the routine definition. Use the ALTER PROCEDURE or ALTER FUNCTION statement to update the EXTERNAL NAME clause if required.
8. If you created projects in the Development Center to develop your Java routines, upgrade any existing projects to the Data Studio using the upgrade wizard.

After upgrading your Java routines, perform the remaining steps in the upgrading routines task.

Upgrading .NET CLR routines

Upgrading your existing .NET CLR routines involves managing the changes between DB2 Version 9.7 and previous releases that impact these routines and verifying that they function as expected.

Prerequisites

- Review the upgrade essentials for routines to identify key changes that might apply to your .NET CLR routines.
- Ensure that you have access to a DB2 Version 9.7 server, including instances and databases. The DB2 server can be part of a testing environment.
- Ensure that a supported version of the Microsoft .NET Framework software is installed on the DB2 server.
- Perform the previous steps in the upgrading routines task.

Procedure

To upgrade your .NET CLR routines to DB2 Version 9.7:

1. Connect to the DB2 Version 9.7 database in which you defined the .NET CLR routines.
2. If you created your .NET CLR routines with execution control mode UNSAFE and you are upgrading from pre-Version 9.7 32-bit instance to Version 9.7 64-bit instance, rebuild their source code using the compile and link options specified in bldrtn.bat, the DB2 sample script for building .NET CLR routines.
If you upgraded your .NET Framework, you should also rebuild your .NET CLR routines.
3. Deploy the routine assembly to the DB2 server in the same location specified by the EXTERNAL clause in the routine definition. The routines should function successfully, with no differences in between previous releases and DB2 Version 9.7.

After upgrading your .NET CLR routines, perform the remaining steps in the upgrading routines task.

Upgrading SQL procedures

SQL procedures that you created using DB2 UDB Version 8.1 FixPak 6 or earlier might require manual upgrade to DB2 Version 9.7. SQL procedures that you created in DB2 UDB Version 8.2 (Version 8.1 FixPak 7) or later do not require upgrade.

If you upgraded from a DB2 UDB Version 8 instance to a DB2 Version 9.7 instance with the same bit size, your routines will run successfully in DB2 Version 9.7. However, if you created your SQL procedures in DB2 UDB Version 8.1 and upgraded from a 32-bit instance to a DB2 Version 9.7 64-bit instance, you must drop and re-create those SQL procedures as part of the manual upgrade process.

Prerequisites

- Ensure that you have access to your upgraded database on DB2 Version 9.7.
- Ensure that you have the necessary authorizations and privileges to use the CREATE PROCEDURE and DROP PROCEDURE statements. You can find the complete list of authorizations and privileges required in the *SQL Reference, Volume 2*.
- Perform the previous steps in the upgrading routines task.

Restriction

This procedure applies only to SQL procedures that were created in DB2 UDB Version 8.1 before FixPak 7 (also known as Version 8.2).

Procedure

To manually upgrade your SQL procedures to DB2 Version 9.7:

1. Connect to the upgraded database.
2. Run the following query to identify the SQL procedures that you need to re-create:

```
SELECT procschema, specificname
FROM syscat.procedures
WHERE language = 'SQL' AND fenced = 'N' AND
      substr(IMPLEMENTATION, 10,6) = 'pgsjmp'
```

Take note of the schema and specific name values returned by this query, because you will need this information to perform subsequent steps.

3. Run the db2look tool to generate a DDL script for all your database objects:

```
db2look -d sample -e -o db2look.sql -a
```

where `sample` is the database name, the `-e` option generates DDL statements for database objects, the `-o db2look.sql` option indicates the output file that will contain the DDL statements, and the `-a` option indicates all objects created by all users.

Edit the `db2look.sql` file to keep only the DDL statements necessary to create the SQL procedures that you identified in step 2 on page 198.

4. For each SQL stored procedures that you identified in step 2 on page 198, use the DROP PROCEDURE statement indicating the schema name and specific name to uniquely identify each procedure:

```
DROP SPECIFIC PROCEDURE <schema-name>.<specific-name>
```

Alternatively, if you have a DDL script that drops and re-creates your SQL procedures, edit it to drop and re-create only the SQL procedures identified in step 2 on page 198, and run it. Then proceed to step 6.

5. Re-create the SQL procedures identified in step 2 on page 198 using the CREATE PROCEDURE statement. Alternatively, you can run your own DDL script or the `db2look.sql` file that you created in step 3.
6. Test your SQL procedures to ensure that they run as expected under DB2 Version 9.7. You can use the Data Studio or the Command Line Processor (CLP) interface to test them. The following example illustrates how to invoke an SQL procedure using the CLP :

```
CONNECT TO sample
```

```
Database Connection Information
```

```
Database server      = DB2/AIX64 9.7.0
SQL authorization ID = TESTDB2
Local database alias = SAMPLE
```

```
CALL <schema-name>.<procedure-name> ( [<parameter-list>] )
```

7. If you created projects in the Development Center to develop your SQL procedures, upgrade any existing projects to the Data Studio using the upgrade wizard. You can use projects that you created in the Developer Workbench without having to upgrade them.

After upgrading your SQL procedures, perform the remaining steps in the upgrading routines task.

Upgrading 32-bit external routines to run on 64-bit instances

Upgrading your existing 32-bit external routines to a DB2 Version 9.7 64-bit instance involves managing the changes between DB2 Version 9.7 and previous releases that impact these routines and verifying that these routines function as expected.

Prerequisites

- Ensure that you have access to a DB2 Version 9.7 64-bit instance that includes 32-bit shared libraries.
- Ensure that the development software is at a version level that is supported by DB2 database products.

- Ensure that you have the necessary authorizations and privileges to use the ALTER FUNCTION or ALTER PROCEDURE statements. The authorizations allowed are listed in the *SQL Reference, Volume 2*.
- Perform the previous steps in the upgrading routines task.

Restrictions

- This procedure applies only to 32-bit external routines programmed in C and COBOL.
- This procedure indicates only the changes that are required to run 32-bit external routines on a 64-bit instance that includes 32-bit shared libraries.

Procedure

To upgrade 32-bit external routines to run on a DB2 Version 9.7 64-bit instance:

1. Ensure that the library path environment variables include the correct DB2 shared library path for 32-bit libraries as shown in Table 22 on page 149, so that the correct library can be loaded at runtime.
2. Test your routines in a DB2 Version 9.7 testing environment. If testing is successful, you do not need to perform any additional steps. However, consider performing the remaining steps in this task if they apply to your routine for better support by using the correct library path and development software.
3. Specify the correct library path by linking or rebuilding your 32-bit external routines using the DB2 shared library paths for 32-bit libraries shown in Table 21 on page 149. If you upgraded from a DB2 UDB Version 8 32-bit instance to a DB2 Version 9.7 64-bit instance, you must rebuild 32-bit external routines that use LOB locators as 64-bit routine libraries.
4. Optional: If you no longer have the source code to rebuild your routine library or you cannot use environmental variables, use the db2chglbpath command to change the DB2 shared library path to \$INSTHOME/sql1lib/lib32 on your routine binary file as long as it has an embedded runtime path. The embedded runtime path can be changed to a new path with the same length or less.
5. Perform any other steps in the “Upgrading C, C++, and COBOL routines” on page 194 task that apply to your routines.
6. Determine if the external routines that were altered during database upgrade or the external routines that use the DB2 engine libraries can safely run as NOT FENCED and THREADSAFE. If you have external unfenced routines in your database, the UPGRADE DATABASE command performs the following actions:
 - Returns the SQL1349W warning message and writes the ADM4100W message to the administration notification log.
 - Redefines all your external unfenced routines that have no dependency on the DB2 engine library as FENCED and NOT THREADSAFE.
 - Creates a CLP script called alter_unfenced_dbname.db2 in the directory specified by the DIAGPATH database manager configuration parameter to redefine the affected routines as NOT FENCED and THREADSAFE.

If you can safely run the external routines altered by database upgrade as NOT FENCED and THREADSAFE, you can redefine them as NOT FENCED and THREADSAFE using the original CLP script or a modified version with just specific routines that you want to redefine. If you can run them as FENCED and NOT THREADSAFE and the performance degradation that you experience is acceptable, you do not need to redefine your routines .

After upgrading your 32-bit external routines, perform the remaining steps in the upgrading routines task.

Chapter 27. Post-upgrade tasks for database applications and routines

After upgrading your database applications and routines, you should perform several post-upgrade tasks to ensure that your database applications and routines perform as expected and at their optimum levels.

Perform the following post-upgrade tasks that apply to your database applications and routines:

1. Perform benchmark tests on your database applications and routines in your production environment and compare with the baseline results that you saved before the upgrade.
2. Tune your database applications. Review important guidelines related to:
 - Character conversion
 - Optimization class
 - Isolation level
 - Locks and concurrency
 - Parallel processing for applications
 - Query optimization

See related concepts for information about additional factors that can affect application performance.

3. Tune your routines. Review important guidelines related to:
 - Stored procedures
 - SQL procedures

In addition, review guidelines on improving the performance of database applications that also apply to routines, such as the guidelines on optimization classes, locks, concurrency, and query tuning.

4. Remove dependencies on functionality that is deprecated in DB2 Version 9.7 in your database applications and routines before that functionality becomes discontinued.
5. Adopt new DB2 Version 9.7 functionality in database applications, where appropriate, to improve performance or add new functionality. Check the Sample files to understand how the new functionality works.

Chapter 28. Adopting new DB2 Version 9.7 functionality in database applications and routines

After upgrading to DB2 Version 9.7, enhance the functionality and improve the performance of your database applications by adopting new DB2 Version 9.7 functionality.

Prerequisite

You must upgrade your DB2 server to DB2 Version 9.7.

For applications that access upgraded databases, perform any of the following steps to adopt the specified DB2 Version 9.7 functionality:

- **Use optimization guidelines or view MQTs to improve MQT matching.** Try this new functionality in a testing environment before you implement it in your production environment.
 - Use the new MQTENFORCE element for optimization guidelines to choose an MQT regardless of its cost estimate.
 - Use a View MQT to create an MQT on views containing a complex query. Any queries on the view containing a complex query can be matched to the View MQT. In previous releases, a query on a view with a construct such as OUTER JOIN or UNION ALL could not be matched to an MQT.
- **Enable statement concentrator to improve performance for dynamic SQL statements that are similar.** The database server modifies these statements so that they share the same access plan. See “Statement concentrator reduces compilation overhead” in *Troubleshooting and Tuning Database Performance*.

The following example shows how to enable statement concentrator at the data server level:

```
UPDATE DB CFG FOR dbname
      USING stmt_conc LITERALS
```

After you enable statement concentrator, the following statements share the same access plan:

```
SELECT FIRSTNAME, LASTNAME FROM EMPLOYEE WHERE EMPNO='000020'
and
SELECT FIRSTNAME, LASTNAME FROM EMPLOYEE WHERE EMPNO='000070'
```

You can also enable this functionality at the application level using the statementConcentrator Connection or DataSource property or the setDBStatementConcentrator method in JDBC. See “DB2Connection interface” in *Developing Java Applications* for details.

- If the value of the **pckcachesz** database configuration parameter is close to the upper limit in pre-Version 97 releases running on 64-bit operating systems, tune this parameter or set to AUTOMATIC to enable self tuning. In Version 9.7 the upper limit for this parameter has been increased to 2 147 483 646.

Having enough memory to cache the sections for static or dynamic SQL or XQuery statements might improve performance, especially when you issue the same statement multiple times from an application.

- **If you want to increase concurrency for the cursor stability isolation level or you are migrating Oracle applications, enable currently committed behavior.** To enable this behavior at the database level, perform the following steps:
 1. Set the **cur_commit** configuration parameter to ON by issuing the following statement:

```
UPDATE DB CFG FOR database-name
  USING cur_commit ON
```

2. Increase the **locklist** parameter to twice its value by issuing the following statement:

```
UPDATE DB CFG FOR database-name
  USING locklist pre-upgrade-value*2
```

3. For applications with high-volume transaction processing, you might need to increase log space or log buffer size.
- **If your application requires a temporary table that remains defined after the end of a session or you are migrating Oracle applications, use created temporary tables (CGTTs)** which differ from declared temporary tables (DGTs) in that their definitions are stored in the system catalog and shared by multiple connections. The data stored in an instance of a CGTT is only available to the session that inserted it.

See “Creating and connecting to created temporary tables” in *Database Administration Concepts and Configuration Reference* for details on how to use CGTTs.

- **If you are migrating Oracle applications, use new compatible scalar functions** such as DAYNAME, DECFLOAT_FORMAT, INITCAP, INSTR, MONTHNAME, LOCATE_IN_STRING, LPAD, RPAD, TO_CLOB, TO_NUMBER, or TO_TIMESTAMP to simplify the migration process.
- If you use the DECOMPOSE XML DOCUMENT command or call the system-defined procedures to decompose XML documents, **use the new DECOMPOSE XML DOCUMENTS command to decompose multiple XML documents with one invocation only**. This command does not require serialization of the XML documents and decomposes XML documents stored in a binary or XML columns.

See “Multiple XML document decomposition examples” in *pureXML Guide* for details on how to use the DECOMPOSE XML DOCUMENTS command.

If you upgraded from DB2 Version 9.1 or earlier, adopt functionality introduced in DB2 Version 9.5 in your database applications and routines. See Enabling new DB2 Version 9.5 functionality in database applications and routines in the *Migration Guide (Version 9.5)* for details.

Part 5. Appendixes

Appendix A. Important references

The following list of references can help you with the upgrade of your DB2 environment.

DB2 operating system requirements Web page

You can find the operating system and hardware requirements for DB2 Version 9.7 installation in “Installation requirements for DB2 database products” in *Installing DB2 Servers*. The most up to date version of this topic is available in the DB2 Information Center at <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.qb.server.doc/doc/r0025127.html>.

DB2 Information Center

You can find the information in this book in the online DB2 Information Center at <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp>. Refer to the “Upgrading” topic under the “Database fundamentals” section. The title for the most high level topic is “Upgrading to DB2 Version 9.7”. The online DB2 Information Center also contains information about upgrade-related topics such as DB2 database product installation. You can also find other information referenced in this book.

DB2 DB2 Version 9.7 manuals in PDF format

DB2 DB2 Version 9.7 manuals in PDF format are available for complimentary download at <http://www.ibm.com/support/docview.wss?rs=71&uid=swg27009474>.

DB2 upgrade portal

The DB2 upgrade portal (formerly know as DB2 migration portal) at <http://www.ibm.com/software/data/db2/upgrade/portal> provides you with a single place for accessing up-to-date information about the upgrade process and additional resources as they become available.

DB2 database product education

The Information Management Training Web site at <http://www.ibm.com/software/data/education/> offers a wide variety of training options and the list of skills resources and communities to help you find the educational resources that are right for you. Review the list of complimentary DB2 database product self-study courses that can help you build skills at your own pace at <http://www.ibm.com/software/data/education/selfstudy.html>.

developerWorks Information Management Web site

The developerWorks Information Management Web site at <http://www.ibm.com/developerworks/data> offers technical resources for DB2 Information Management software. It features product information, downloads, learning resources, support, forums, and newsletters. On this Web site you can find many articles and tutorials that can help you to learn about new functionality of DB2 database products and how to use them in your applications.

This Web site also offers portals of learning resources such as New to DB2, Migrate to DB2, and DBA Central. Follow the **Migrate to DB2** link to

access resources that can help you migrate from Microsoft SQL Server, Oracle, Sybase, and other database platforms to DB2 Database products.

DB2 forums

The DB2 forums are places to exchange ideas and share solutions with your peers in the IBM DB2 product community. In addition, DB2 forums include forums that are mirrors to DB2 newsgroups, such as the `ibm.software.db2.udb` and `ibm.software.db2.udb.beta` newsgroups. The DB2 forums are hosted by developerWorks at http://www.ibm.com/developerworks/forums/db2_forums.jsp.

Appendix B. Overview of the DB2 technical information

DB2 technical information is available through the following tools and methods:

- DB2 Information Center
 - Topics (Task, concept and reference topics)
 - Help for DB2 tools
 - Sample programs
 - Tutorials
- DB2 books
 - PDF files (downloadable)
 - PDF files (from the DB2 PDF DVD)
 - printed books
- Command line help
 - Command help
 - Message help

Note: The DB2 Information Center topics are updated more frequently than either the PDF or the hardcopy books. To get the most current information, install the documentation updates as they become available, or refer to the DB2 Information Center at ibm.com.

You can access additional DB2 technical information such as technotes, white papers, and IBM Redbooks® publications online at ibm.com. Access the DB2 Information Management software library site at <http://www.ibm.com/software/data/sw-library/>.

Documentation feedback

We value your feedback on the DB2 documentation. If you have suggestions for how to improve the DB2 documentation, send an e-mail to db2docs@ca.ibm.com. The DB2 documentation team reads all of your feedback, but cannot respond to you directly. Provide specific examples wherever possible so that we can better understand your concerns. If you are providing feedback on a specific topic or help file, include the topic title and URL.

Do not use this e-mail address to contact DB2 Customer Support. If you have a DB2 technical issue that the documentation does not resolve, contact your local IBM service center for assistance.

DB2 technical library in hardcopy or PDF format

The following tables describe the DB2 library available from the IBM Publications Center at www.ibm.com/shop/publications/order. English and translated DB2 Version 9.7 manuals in PDF format can be downloaded from www.ibm.com/support/docview.wss?rs=71&uid=swg2700947.

Although the tables identify books available in print, the books might not be available in your country or region.

The form number increases each time a manual is updated. Ensure that you are reading the most recent version of the manuals, as listed below.

Note: The *DB2 Information Center* is updated more frequently than either the PDF or the hard-copy books.

Table 31. DB2 technical information

Name	Form Number	Available in print	Last updated
<i>Administrative API Reference</i>	SC27-2435-01	Yes	November, 2009
<i>Administrative Routines and Views</i>	SC27-2436-01	No	November, 2009
<i>Call Level Interface Guide and Reference, Volume 1</i>	SC27-2437-01	Yes	November, 2009
<i>Call Level Interface Guide and Reference, Volume 2</i>	SC27-2438-01	Yes	November, 2009
<i>Command Reference</i>	SC27-2439-01	Yes	November, 2009
<i>Data Movement Utilities Guide and Reference</i>	SC27-2440-00	Yes	August, 2009
<i>Data Recovery and High Availability Guide and Reference</i>	SC27-2441-01	Yes	November, 2009
<i>Database Administration Concepts and Configuration Reference</i>	SC27-2442-01	Yes	November, 2009
<i>Database Monitoring Guide and Reference</i>	SC27-2458-00	Yes	August, 2009
<i>Database Security Guide</i>	SC27-2443-01	Yes	November, 2009
<i>DB2 Text Search Guide</i>	SC27-2459-01	Yes	November, 2009
<i>Developing ADO.NET and OLE DB Applications</i>	SC27-2444-00	Yes	August, 2009
<i>Developing Embedded SQL Applications</i>	SC27-2445-01	Yes	November, 2009
<i>Developing Java Applications</i>	SC27-2446-01	Yes	November, 2009
<i>Developing Perl, PHP, Python, and Ruby on Rails Applications</i>	SC27-2447-00	No	August, 2009
<i>Developing User-defined Routines (SQL and External)</i>	SC27-2448-01	Yes	November, 2009
<i>Getting Started with Database Application Development</i>	GI11-9410-01	Yes	November, 2009
<i>Getting Started with DB2 Installation and Administration on Linux and Windows</i>	GI11-9411-00	Yes	August, 2009

Table 31. DB2 technical information (continued)

Name	Form Number	Available in print	Last updated
<i>Globalization Guide</i>	SC27-2449-00	Yes	August, 2009
<i>Installing DB2 Servers</i>	GC27-2455-01	Yes	November, 2009
<i>Installing IBM Data Server Clients</i>	GC27-2454-00	No	August, 2009
<i>Message Reference Volume 1</i>	SC27-2450-01	No	November, 2009
<i>Message Reference Volume 2</i>	SC27-2451-01	No	November, 2009
<i>Net Search Extender Administration and User's Guide</i>	SC27-2469-01	No	November, 2009
<i>Partitioning and Clustering Guide</i>	SC27-2453-01	Yes	November, 2009
<i>pureXML Guide</i>	SC27-2465-01	Yes	November, 2009
<i>Query Patroller Administration and User's Guide</i>	SC27-2467-00	No	August, 2009
<i>Spatial Extender and Geodetic Data Management Feature User's Guide and Reference</i>	SC27-2468-00	No	August, 2009
<i>SQL Procedural Languages: Application Enablement and Support</i>	SC27-2470-00	Yes	August, 2009
<i>SQL Reference, Volume 1</i>	SC27-2456-01	Yes	November, 2009
<i>SQL Reference, Volume 2</i>	SC27-2457-01	Yes	November, 2009
<i>Troubleshooting and Tuning Database Performance</i>	SC27-2461-01	Yes	November, 2009
<i>Upgrading to DB2 Version 9.7</i>	SC27-2452-01	Yes	November, 2009
<i>Visual Explain Tutorial</i>	SC27-2462-00	No	August, 2009
<i>What's New for DB2 Version 9.7</i>	SC27-2463-01	Yes	November, 2009
<i>Workload Manager Guide and Reference</i>	SC27-2464-00	Yes	August, 2009
<i>XQuery Reference</i>	SC27-2466-01	No	November, 2009

Table 32. DB2 Connect-specific technical information

Name	Form Number	Available in print	Last updated
<i>Installing and Configuring DB2 Connect Personal Edition</i>	SC27-2432-01	Yes	November, 2009
<i>Installing and Configuring DB2 Connect Servers</i>	SC27-2433-01	Yes	November, 2009

Table 32. DB2 Connect-specific technical information (continued)

Name	Form Number	Available in print	Last updated
DB2 Connect User's Guide	SC27-2434-01	Yes	November, 2009

Table 33. Information Integration technical information

Name	Form Number	Available in print	Last updated
Information Integration: Administration Guide for Federated Systems	SC19-1020-02	Yes	August, 2009
Information Integration: ASNCLP Program Reference for Replication and Event Publishing	SC19-1018-04	Yes	August, 2009
Information Integration: Configuration Guide for Federated Data Sources	SC19-1034-02	No	August, 2009
Information Integration: SQL Replication Guide and Reference	SC19-1030-02	Yes	August, 2009
Information Integration: Introduction to Replication and Event Publishing	GC19-1028-02	Yes	August, 2009

Ordering printed DB2 books

If you require printed DB2 books, you can buy them online in many but not all countries or regions. You can always order printed DB2 books from your local IBM representative. Keep in mind that some softcopy books on the *DB2 PDF Documentation DVD* are unavailable in print. For example, neither volume of the *DB2 Message Reference* is available as a printed book.

Printed versions of many of the DB2 books available on the *DB2 PDF Documentation DVD* can be ordered for a fee from IBM. Depending on where you are placing your order from, you may be able to order books online, from the IBM Publications Center. If online ordering is not available in your country or region, you can always order printed DB2 books from your local IBM representative. Note that not all books on the *DB2 PDF Documentation DVD* are available in print.

Note: The most up-to-date and complete DB2 documentation is maintained in the DB2 Information Center at <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7>.

To order printed DB2 books:

- To find out whether you can order printed DB2 books online in your country or region, check the IBM Publications Center at <http://www.ibm.com/shop/publications/order>. You must select a country, region, or language to access publication ordering information and then follow the ordering instructions for your location.
- To order printed DB2 books from your local IBM representative:

1. Locate the contact information for your local representative from one of the following Web sites:
 - The IBM directory of world wide contacts at www.ibm.com/planetwide
 - The IBM Publications Web site at <http://www.ibm.com/shop/publications/order>. You will need to select your country, region, or language to the access appropriate publications home page for your location. From this page, follow the "About this site" link.
2. When you call, specify that you want to order a DB2 publication.
3. Provide your representative with the titles and form numbers of the books that you want to order. For titles and form numbers, see "DB2 technical library in hardcopy or PDF format" on page 209.

Displaying SQL state help from the command line processor

DB2 products return an SQLSTATE value for conditions that can be the result of an SQL statement. SQLSTATE help explains the meanings of SQL states and SQL state class codes.

To start SQL state help, open the command line processor and enter:

```
? sqlstate or ? class code
```

where *sqlstate* represents a valid five-digit SQL state and *class code* represents the first two digits of the SQL state.

For example, ? 08003 displays help for the 08003 SQL state, and ? 08 displays help for the 08 class code.

Accessing different versions of the DB2 Information Center

For DB2 Version 9.7 topics, the *DB2 Information Center* URL is <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/>.

For DB2 Version 9.5 topics, the *DB2 Information Center* URL is <http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/>.

For DB2 Version 9.1 topics, the *DB2 Information Center* URL is <http://publib.boulder.ibm.com/infocenter/db2luw/v9/>.

For DB2 Version 8 topics, go to the *DB2 Information Center* URL at: <http://publib.boulder.ibm.com/infocenter/db2luw/v8/>.

Displaying topics in your preferred language in the DB2 Information Center

The DB2 Information Center attempts to display topics in the language specified in your browser preferences. If a topic has not been translated into your preferred language, the DB2 Information Center displays the topic in English.

- To display topics in your preferred language in the Internet Explorer browser:
 1. In Internet Explorer, click the **Tools** —> **Internet Options** —> **Languages...** button. The Language Preferences window opens.
 2. Ensure your preferred language is specified as the first entry in the list of languages.
 - To add a new language to the list, click the **Add...** button.

- Note:** Adding a language does not guarantee that the computer has the fonts required to display the topics in the preferred language.
- To move a language to the top of the list, select the language and click the **Move Up** button until the language is first in the list of languages.
3. Refresh the page to display the DB2 Information Center in your preferred language.
- To display topics in your preferred language in a Firefox or Mozilla browser:
 1. Select the button in the **Languages** section of the **Tools** —> **Options** —> **Advanced** dialog. The Languages panel is displayed in the Preferences window.
 2. Ensure your preferred language is specified as the first entry in the list of languages.
 - To add a new language to the list, click the **Add...** button to select a language from the Add Languages window.
 - To move a language to the top of the list, select the language and click the **Move Up** button until the language is first in the list of languages.
 3. Refresh the page to display the DB2 Information Center in your preferred language.

On some browser and operating system combinations, you must also change the regional settings of your operating system to the locale and language of your choice.

Updating the DB2 Information Center installed on your computer or intranet server

A locally installed DB2 Information Center must be updated periodically.

Before you begin

A DB2 Version 9.7 Information Center must already be installed. For details, see the “Installing the DB2 Information Center using the DB2 Setup wizard” topic in *Installing DB2 Servers*. All prerequisites and restrictions that applied to installing the Information Center also apply to updating the Information Center.

About this task

An existing DB2 Information Center can be updated automatically or manually:

- Automatic updates - updates existing Information Center features and languages. An additional benefit of automatic updates is that the Information Center is unavailable for a minimal period of time during the update. In addition, automatic updates can be set to run as part of other batch jobs that run periodically.
- Manual updates - should be used when you want to add features or languages during the update process. For example, a local Information Center was originally installed with both English and French languages, and now you want to also install the German language; a manual update will install German, as well as, update the existing Information Center features and languages. However, a manual update requires you to manually stop, update, and restart the Information Center. The Information Center is unavailable during the entire update process.

Procedure

This topic details the process for automatic updates. For manual update instructions, see the “Manually updating the DB2 Information Center installed on your computer or intranet server” topic.

To automatically update the DB2 Information Center installed on your computer or intranet server:

1. On Linux operating systems,
 - a. Navigate to the path where the Information Center is installed. By default, the DB2 Information Center is installed in the `/opt/ibm/db2ic/V9.7` directory.
 - b. Navigate from the installation directory to the `doc/bin` directory.
 - c. Run the `ic-update` script:

```
ic-update
```
2. On Windows operating systems,
 - a. Open a command window.
 - b. Navigate to the path where the Information Center is installed. By default, the DB2 Information Center is installed in the `<Program Files>\IBM\DB2 Information Center\Version 9.7` directory, where `<Program Files>` represents the location of the Program Files directory.
 - c. Navigate from the installation directory to the `doc\bin` directory.
 - d. Run the `ic-update.bat` file:

```
ic-update.bat
```

Results

The DB2 Information Center restarts automatically. If updates were available, the Information Center displays the new and updated topics. If Information Center updates were not available, a message is added to the log. The log file is located in `doc\eclipse\configuration` directory. The log file name is a randomly generated number. For example, `1239053440785.log`.

Manually updating the DB2 Information Center installed on your computer or intranet server

If you have installed the DB2 Information Center locally, you can obtain and install documentation updates from IBM.

About this task

Updating your locally-installed *DB2 Information Center* manually requires that you:

1. Stop the *DB2 Information Center* on your computer, and restart the Information Center in stand-alone mode. Running the Information Center in stand-alone mode prevents other users on your network from accessing the Information Center, and allows you to apply updates. The Workstation version of the DB2 Information Center always runs in stand-alone mode. .
2. Use the Update feature to see what updates are available. If there are updates that you must install, you can use the Update feature to obtain and install them

Note: If your environment requires installing the *DB2 Information Center* updates on a machine that is not connected to the internet, mirror the update site to a local file system using a machine that is connected to the internet and has the *DB2 Information Center* installed. If many users on your network will be

installing the documentation updates, you can reduce the time required for individuals to perform the updates by also mirroring the update site locally and creating a proxy for the update site.

If update packages are available, use the Update feature to get the packages. However, the Update feature is only available in stand-alone mode.

3. Stop the stand-alone Information Center, and restart the *DB2 Information Center* on your computer.

Note: On Windows 2008, Windows Vista (and higher), the commands listed later in this section must be run as an administrator. To open a command prompt or graphical tool with full administrator privileges, right-click the shortcut and then select **Run as administrator**.


Procedure

To update the *DB2 Information Center* installed on your computer or intranet server:

1. Stop the *DB2 Information Center*.
 - On Windows, click **Start** → **Control Panel** → **Administrative Tools** → **Services**. Then right-click **DB2 Information Center** service and select **Stop**.
 - On Linux, enter the following command:

```
/etc/init.d/db2icdv97 stop
```
 2. Start the Information Center in stand-alone mode.
 - On Windows:
 - a. Open a command window.
 - b. Navigate to the path where the Information Center is installed. By default, the *DB2 Information Center* is installed in the *Program_Files\IBM\DB2 Information Center\Version 9.7* directory, where *Program_Files* represents the location of the Program Files directory.
 - c. Navigate from the installation directory to the *doc\bin* directory.
 - d. Run the *help_start.bat* file:

```
help_start.bat
```
 - On Linux:
 - a. Navigate to the path where the Information Center is installed. By default, the *DB2 Information Center* is installed in the */opt/ibm/db2ic/V9.7* directory.
 - b. Navigate from the installation directory to the *doc/bin* directory.
 - c. Run the *help_start* script:

```
help_start
```
- The systems default Web browser opens to display the stand-alone Information Center.
3. Click the **Update** button (). (JavaScript™ must be enabled in your browser.) On the right panel of the Information Center, click **Find Updates**. A list of updates for existing documentation displays.
 4. To initiate the installation process, check the selections you want to install, then click **Install Updates**.
 5. After the installation process has completed, click **Finish**.
 6. Stop the stand-alone Information Center:
 - On Windows, navigate to the installation directory's *doc\bin* directory, and run the *help_end.bat* file:

help_end.bat

Note: The help_end batch file contains the commands required to safely stop the processes that were started with the help_start batch file. Do not use Ctrl-C or any other method to stop help_start.bat.

- On Linux, navigate to the installation directory's doc/bin directory, and run the help_end script:

```
help_end
```

Note: The help_end script contains the commands required to safely stop the processes that were started with the help_start script. Do not use any other method to stop the help_start script.

7. Restart the *DB2 Information Center*.

- On Windows, click **Start** → **Control Panel** → **Administrative Tools** → **Services**. Then right-click **DB2 Information Center** service and select **Start**.
- On Linux, enter the following command:

```
/etc/init.d/db2icdv97 start
```

Results

The updated *DB2 Information Center* displays the new and updated topics.

DB2 tutorials

The DB2 tutorials help you learn about various aspects of DB2 products. Lessons provide step-by-step instructions.

Before you begin

You can view the XHTML version of the tutorial from the Information Center at <http://publib.boulder.ibm.com/infocenter/db2help/>.

Some lessons use sample data or code. See the tutorial for a description of any prerequisites for its specific tasks.

DB2 tutorials

To view the tutorial, click the title.

“pureXML” in *pureXML Guide*

Set up a DB2 database to store XML data and to perform basic operations with the native XML data store.

“Visual Explain” in *Visual Explain Tutorial*

Analyze, optimize, and tune SQL statements for better performance using Visual Explain.

DB2 troubleshooting information

A wide variety of troubleshooting and problem determination information is available to assist you in using DB2 database products.

DB2 documentation

Troubleshooting information can be found in the *DB2 Troubleshooting Guide* or the Database fundamentals section of the *DB2 Information Center*. There you will find information about how to isolate and identify problems using

DB2 diagnostic tools and utilities, solutions to some of the most common problems, and other advice on how to solve problems you might encounter with your DB2 database products.

DB2 Technical Support Web site

Refer to the DB2 Technical Support Web site if you are experiencing problems and want help finding possible causes and solutions. The Technical Support site has links to the latest DB2 publications, TechNotes, Authorized Program Analysis Reports (APARs or bug fixes), fix packs, and other resources. You can search through this knowledge base to find possible solutions to your problems.

Access the DB2 Technical Support Web site at http://www.ibm.com/software/data/db2/support/db2_9/

Terms and Conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

Personal use: You may reproduce these Publications for your personal, non commercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these Publications, or any portion thereof, without the express consent of IBM.

Commercial use: You may reproduce, distribute and display these Publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these Publications, or reproduce, distribute or display these Publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the Publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the Publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Appendix C. Notices

This information was developed for products and services offered in the U.S.A. Information about non-IBM products is based on information available at the time of first publication of this document and is subject to change.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
3-2-12, Roppongi, Minato-ku, Tokyo 106-8711 Japan

The following paragraph does not apply to the United Kingdom or any other country/region where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web

sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information that has been exchanged, should contact:

IBM Canada Limited
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario
L6G 1C7
CANADA

Such information may be available, subject to appropriate terms and conditions, including, in some cases, payment of a fee.

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

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

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements, or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

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

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious, and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application

programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (*your company name*) (*year*). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. *_enter the year or years_*. All rights reserved.

Trademarks

IBM, the IBM logo, and ibm.com[®] are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

The following terms are trademarks or registered trademarks of other companies

- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
- Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- Intel[®], Intel logo, Intel Inside[®], Intel Inside logo, Intel[®] Centrino[®], Intel Centrino logo, Celeron[®], Intel[®] Xeon[®], Intel SpeedStep[®], Itanium[®], and Pentium[®] are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
- Microsoft, Windows, Windows NT[®], and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Index

Special characters

- .NET
 - common language runtime (CLR) routines
 - upgrading 197

Numerics

- 32-bit applications
 - upgrading to run on 64-bit instances 191
- 32-bit external routines
 - upgrading to run on 64-bit instances 199
- 32-bit servers
 - upgrading to 64-bit systems 77
- 64-bit instances
 - upgrading 32-bit applications 191
 - upgrading 32-bit external routines 199

A

- ACTIVATE DATABASE command
 - post-upgrade tasks for DB2 servers 98
- administrative routines
 - upgrade impact 170
- administrative views
 - upgrade impact 170
- ADO.NET applications
 - upgrading 189
- applications
 - migrating
 - XML Extender 116
 - post-upgrade tasks
 - new functionality adoption 203
 - overview 201
 - removing deprecated functionality 201
 - tuning 201
 - pre-upgrade tasks
 - overview 181
 - reviewing upgrade essentials 181
 - testing 181
 - upgrading clients 181
 - upgrading operating systems and development software 181
 - upgrade impact
 - catalog view changes 170
 - DB2 APIs 151
 - DB2 command changes 155
 - SQL statements 162
 - system built-in routine changes 170
 - system-defined administrative routine and view changes 170
 - upgrading
 - planning 8, 145
 - process 143, 183
- automatic reclaim reorganization
 - upgraded databases 107
- automatic storage
 - upgraded databases 107
- autonomic computing
 - enabling functionality after upgrade 22

B

- BACKUP DATABASE command
 - pre-upgrade tasks for DB2 servers 48
- backups
 - client configuration 127
 - databases
 - pre-upgrade tasks for DB2 servers 48
 - DB2 server configuration 49
- books
 - ordering 212
- built-in routines
 - upgrade impact 170

C

- call level interface (CLI)
 - applications
 - upgrading 185
- catalog views
 - upgrade impact 170
- clients
 - post-upgrade tasks
 - managing server changes 137
 - overview 137
 - recataloging nodes 137
 - verifying upgrade 138
 - pre-upgrade tasks
 - backing up configuration 127
 - overview 127
 - reviewing upgrade essentials 127
 - upgrading DB2 servers 127
 - upgrading in test environments 128
 - upgrading
 - best practices 125
 - Data Server Client (Windows) 129
 - Data Server Runtime Client (Windows) 131
 - Linux and UNIX 133
 - overview 121, 123
 - planning 8
- command line processor (CLP)
 - scripts
 - upgrade impact 155
 - upgrading 190
- commands
 - dasmigr
 - upgrading DAS 63, 72
 - db2ckupgrade
 - pre-upgrade tasks for DB2 servers 46
 - db2exmig
 - post-upgrade tasks for DB2 servers 102
 - db2IdentifyType1
 - converting type-1 indexes to type-2 indexes 44
 - db2iupgrade
 - failure causes 19
 - overview 17
 - upgrading instances 60, 70
 - db2tdbmgr
 - upgrading DAS 63, 72
- deprecated
 - upgrade impact 38

- commands (*continued*)
 - discontinued
 - upgrade impact 38
 - UPGRADE DATABASE
 - upgraded database entities 17
 - upgrading databases 64, 73
- configuration
 - backups
 - clients 127
 - pre-upgrade tasks for DB2 servers 49
- configuration parameters
 - saving settings before upgrading DB2 servers 49
 - upgrade impact 28, 98
- Control Center
 - deprecated functionality upgrade impact 38
- CREATE TABLESPACE statement
 - adjusting system temporary table spaces page sizes 103

D

- dasmigr command
 - upgrading DAS 63, 72
- Data Links Manager environment
 - upgrading 90
- database applications
 - adopting new functionality 203
 - upgrade process 143
 - upgrade support 145
 - upgrading 183
- database auditing
 - post-upgrade 100
- database duplicates
 - testing DB2 server upgrade 56
- databases
 - auditing
 - post-upgrade tasks for DB2 servers 100
 - new functionality adoption after upgrade 107
 - physical design characteristic changes upgrade impact 28
 - pre-upgrade tasks 46
 - upgrading
 - procedure 64, 73
- DB2 administration server (DAS)
 - upgrading 63, 72
- DB2 environment
 - migrating
 - see DB2 environment, upgrading 3
 - upgrading 3
- DB2 Governor
 - migrating to DB2 workload manager 111
- DB2 Information Center
 - languages 213
 - updating 214, 215
 - versions 213
- DB2 JDBC Type 2 Driver
 - upgrading Java applications 188
- DB2 servers
 - changes 28
 - post-upgrade tasks for clients 137
 - falling back to a previous release 117
 - post-upgrade tasks
 - activating databases 98
 - activating services 98
 - adjusting log space 97
 - adjusting system temporary table space page sizes 103
 - converting type-1 indexes to type-2 indexes 44
 - managing server changes 98
 - migrating explain tables 102

- DB2 servers (*continued*)
 - post-upgrade tasks (*continued*)
 - overview 95
 - rebinding packages 101
 - setting up database auditing 100
 - verifying upgrade 106
 - pre-upgrade tasks
 - backing up configuration 49
 - backing up databases 48
 - changing raw devices to block devices (Linux) 53
 - increasing log space 51
 - increasing table space sizes 51
 - overview 43
 - taking servers offline 56
 - upgrading test environments 55
 - verifying databases 46
 - reversing upgrade 117
 - upgrade impact
 - behavior changes 28
 - deprecated functionality 38
 - discontinued functionality 38
 - registry variables 28
 - upgrading
 - 32-bit to 64-bit 77
 - alternate fix pack installations 80
 - best practices 22
 - databases 64, 73
 - DB2 administration server (DAS) 63, 72
 - instances 60, 70
 - Linux 69
 - multiple DB2 copies 80
 - new server 82
 - partitioned database environments 86
 - planning 6
 - process 15
 - support 17
 - UNIX 69
 - using online database backups 85
 - Windows 59
- DB2 Spatial Extender
 - upgrading
 - best practices 22
- DB2 Text Search
 - upgrading 87
- DB2 workload manager
 - DB2 Governor
 - migrating 111
 - migrating
 - Query Patroller 113
- DB2_USE_DB2JCCT2_JROUTINE variable
 - upgrading Java routines 196
- db2batch command
 - verifying upgrade 106
- db2ckupgrade command
 - pre-upgrade tasks for DB2 servers 46
- db2exmig command
 - post-upgrade tasks for DB2 servers 102
- db2IdentifyType1 command
 - converting type-1 indexes to type-2 indexes 44
- db2iupgrade command
 - failures 19
 - upgrading instances 17, 60, 70
- db2rbind command
 - post-upgrade tasks for DB2 servers 101
- db2support command
 - pre-upgrade tasks for DB2 servers 49

- db2tdbmgr command
 - upgrading DAS 63, 72
- deprecated functionality
 - removing 201
 - upgrade impact 38
- Direct I/O (DIO)
 - changing raw devices to block devices (Linux) 53
- discontinued functionality
 - upgrade impact 38
- disk space
 - requirements 25
- documentation
 - overview 209
 - PDF files 209
 - printed 209
 - terms and conditions of use 218

E

- embedded SQL applications
 - upgrading 184
- explain tables
 - migrating 102

F

- FORTRAN language
 - applications
 - upgrading 184

H

- help
 - configuring language 213
 - SQL statements 213

I

- IBM data server clients
 - IBM Data Server Client 129
- IBM Data Server Driver for JDBC and SQLJ
 - upgrading Java applications 186
- IBM Data Server Driver Package
 - upgrading 135
- IBM Data Server Runtime Client
 - upgrading (Windows) 131
- instances
 - 32-bit and 64-bit upgrade support 27
 - upgrading 19, 60, 70

J

- Java
 - applications
 - upgrading (DB2 JDBC Type 2 Driver) 188
 - upgrading (IBM Data Server Driver for JDBC and SQLJ) 186
 - routines
 - upgrading 196
- jdk_path configuration parameter
 - routines
 - upgrading 196

L

- Linux
 - changing raw devices to block devices 53
 - upgrading
 - clients 133
 - DB2 servers 69
 - non-root installations 79
- logs
 - space requirements
 - adjusting 97
 - increasing 51
 - upgrading DB2 servers 25

M

- Microsoft Cluster Server (MSCS)
 - upgrading 93
- Microsoft SQL Server
 - migrating 40
- migration
 - applications
 - overview 143
 - XML Extender 116
 - clients 121
 - DB2 Governor to DB2 workload manager 111
 - DB2 servers 15
 - explain tables 102
 - Microsoft SQL Server 40
 - non-DB2 relational databases 40
 - Oracle 40
 - overview 3
 - Query Patroller to DB2 workload manager 113
 - routines 143
 - Sybase 40
 - XML data type 116
 - XML Extender 116
 - XML Extender to XML data store 111
- multiple DB2 copies
 - upgrading DB2 servers 80

N

- Net Search Extender (NSE)
 - upgrade impact from UDFs 90
 - upgrading 77
- NetBIOS
 - discontinued functionality 137
- non-root installations
 - upgrading 79
- notices 219

O

- O_DIRECT 53
- online database backups
 - upgrading DB2 servers 85
- Oracle
 - migrating 40
- ordering DB2 books 212

P

- partitioned database environments
 - upgrading 86

- partitioned indexes
 - upgraded databases 107
- partitioned tables
 - XML data
 - upgraded databases 107
- post-upgrade tasks
 - applications
 - new functionality adoption 203
 - removing deprecated functionality 201
 - tuning 201
 - clients
 - managing server changes 137
 - overview 137
 - recataloging nodes 137
 - verifying upgrade 138
 - converting XML storage objects to Version 9.7 103
 - DB2 servers
 - activating databases 98
 - activating services 98
 - converting type-1 indexes to type-2 indexes 44
 - database auditing setup 100
 - log spaces adjustments 97
 - managing behavior changes 98
 - migrating explain tables 102
 - new functionality adoption 107
 - overview 95
 - re-creating write-to-table event monitors 105
 - rebinding packages 101
 - system temporary table space page size adjustments 103
 - verifying upgrade 106
 - routines
 - new functionality adoption 203
 - removing deprecated functionality 201
 - tuning 201
- pre-upgrade tasks
 - applications
 - overview 181
 - clients
 - backing up configuration 127
 - overview 127
 - upgrading in test environments 128
 - DB2 servers
 - backing up configuration 49
 - backing up databases 48
 - changing raw devices to block devices (Linux) 53
 - increasing log space 51
 - overview 43
 - taking servers offline 56
 - upgrading in test environments 55
 - verifying databases are ready to upgrade 46
 - routines
 - overview 181
- problem determination
 - information available 217
 - tutorials 217

Q

- Query Patroller
 - migrating to DB2 workload management 113
 - upgrading
 - documentation 77

R

- raw devices
 - changing to block devices 53
- raw I/O
 - changing raw devices to block devices (Linux) 53
- raw logs
 - deprecated functionality
 - upgrade impact 38
- read-only workloads
 - HADR Standby databases after upgrade 107
- REBIND command
 - post-upgrade tasks for DB2 servers 101
- rebinding
 - packages
 - post-upgrade tasks for DB2 servers 101
- recataloging nodes
 - NetBIOS protocol
 - post-upgrade tasks for clients 137
 - SNA protocol
 - post-upgrade tasks for clients 137
- references
 - upgrades 207
- registry variables
 - saving settings before upgrading DB2 servers 49
 - upgrade impact 28
 - upgrading 98
- REORG INDEXES command
 - converting type-1 indexes to type-2 indexes 44
- RESTORE DATABASE command
 - upgrading DB2 servers 82
- reversing upgrades
 - DB2 servers 117
- REXX language
 - applications
 - embedded SQL (upgrading) 184
- routines
 - planning upgrade 8
 - post-upgrade tasks
 - new functionality adoption 203
 - overview 201
 - removing deprecated functionality 201
 - tuning 201
 - pre-upgrade tasks
 - overview 181
 - reviewing upgrade essentials 181
 - testing 181
 - upgrading development software 181
 - upgrading operating systems 181
 - upgrading
 - .NET 197
 - 32-bit external routines 199
 - C 194
 - COBOL 194
 - Java 196
 - overview 143
 - procedure 193
 - support 177

S

- scenarios
 - upgrading DB2 servers 77
- scripts
 - upgrade impact
 - DB2 command changes 155
 - SQL statement changes 162

- scripts (*continued*)
 - upgrade support 145
 - upgrading 190
- SECADM (security administrator) authority
 - setting up database auditing 100
- SNA (Systems Network Architecture)
 - discontinued functionality
 - post-upgrade tasks for clients 137
- SQL
 - administrative routines
 - upgrading 190
 - administrative views
 - upgrading 190
 - replication environments
 - upgrading 22
- SQL procedures
 - upgrading 198
- SQL statements
 - help
 - displaying 213
 - upgrade impact 162
 - upgrading 190
- statistical views
 - upgraded databases 107
- stored procedures
 - upgrade support 177
 - upgrading 193
- Sybase
 - migrating 40
- system built-in routines
 - upgrade impact 170
- system catalogs
 - views
 - upgrade impact 170
- system commands
 - scripts
 - upgrade impact 155
 - upgrading 190
- system temporary table spaces
 - page sizes
 - larger RID 103
 - post-upgrade tasks for DB2 servers 103
- system-defined administrative routines
 - upgrade impact 170
- system-defined administrative views
 - upgrade impact 170

T

- table spaces
 - requirements
 - upgrading DB2 servers 25
- tables
 - XML columns
 - converting to Version 9.7 format 103
- taking servers offline
 - pre-upgrade tasks 56
- TCP/IP
 - recataloging nodes 137
- terms and conditions
 - publications 218
- test environments
 - upgrading clients 128
 - upgrading DB2 servers
 - creating database duplicates 56
 - procedure 55

- tools catalog database
 - upgrading 63, 72
- troubleshooting
 - online information 217
 - tutorials 217
- tuning
 - applications 201
 - routines 201
- tutorials
 - list 217
 - problem determination 217
 - troubleshooting 217
 - Visual Explain 217
- type-1 indexes
 - discontinued
 - upgrade impact 38

U

- UNIX
 - upgrading
 - clients 133
 - DB2 servers 69
 - non-root installations 79
- updates
 - DB2 Information Center 214, 215
- UPGRADE DATABASE command
 - failures 19
 - upgraded database entities 17
 - upgrading databases 64, 73
- upgraded databases
 - new functionality adoption 107
- upgrades
 - .NET CLR routines 197
 - 32-bit applications 191
 - 32-bit servers 27
 - 64-bit servers 27
 - applications
 - ADO .NET 189
 - C 184
 - catalog view changes 170
 - CLI 185
 - COBOL 184
 - DB2 API changes 151
 - DB2 command changes 155
 - DB2 Version 9.7 3
 - embedded SQL 184
 - FORTRAN 184
 - Java using DB2 JDBC Type 2 driver 188
 - Java using IBM Data Server Driver for JDBC and SQLJ 186
 - overview 143, 145
 - planning 8
 - post-upgrade tasks 201
 - pre-upgrade tasks 181
 - procedure 183
 - REXX 184
 - SQL statement changes 162
 - system built-in routine changes 170
 - system-defined administrative routine and view changes 170
 - autonomic computing functionality 22
 - best practices
 - clients 125
 - DB2 servers 22
 - C applications 184
 - C routines 194

- upgrades (*continued*)
 - clients
 - DB2 Version 9.7 3
 - Linux 133
 - overview 121, 123
 - planning 8
 - post-upgrade tasks 137
 - pre-upgrade tasks 127
 - test environments 128
 - UNIX 133
 - COBOL applications 184
 - COBOL routines 194
 - database applications 183
 - databases 64, 73
 - DB2 Administration Server (DAS) 63, 72
 - DB2 Data Links Manager 90
 - DB2 environments 3
 - DB2 servers
 - 32-bit to 64-bit Windows 77
 - adjusting log space 97
 - alternate fix pack installations 80
 - best practices 22
 - complex environments 77
 - configuration parameter changes 28
 - configuration parameters 98
 - database physical characteristic changes 28
 - DB2 Version 9.7 3
 - discontinued functionality 19
 - duplicate databases for test environments 56
 - Linux 69
 - log space requirements 25
 - multiple DB2 copies 80
 - new 82
 - overview 15, 17
 - partitioned database environments 86
 - performance 22
 - physical characteristics 98
 - planning 6
 - post-upgrade tasks 95
 - pre-upgrade tasks 43
 - registry variable changes 28
 - registry variables 98
 - restrictions 19
 - table space requirements 25
 - taking servers offline 56
 - test environments 55
 - UNIX 69
 - using online database backups 85
 - Windows 59
 - DB2 Spatial Extender 22
 - DB2 Text Search 87
 - DB2 Version 9.7 3
 - development software
 - pre-upgrade tasks for applications and routines 181
 - HADR 19
 - IBM Data Server Driver Package 135
 - instance type 19
 - instances
 - 32-bit upgrade support 27
 - 64-bit upgrade support 27
 - procedure 60, 70
 - Microsoft Cluster Server (MSCS) 93
 - Net Search Extender (NSE) UDFs 90
 - non-root installations
 - Linux 79
 - UNIX 79

- upgrades (*continued*)
 - operating systems
 - pre-upgrade tasks for applications and routines 181
 - planning
 - applications 8
 - clients 8
 - DB2 environments 5
 - DB2 servers 6
 - DB2 upgrade portal 5
 - routines 8
 - references 207
 - routines
 - 32-bit external routines 199
 - C 194
 - COBOL 194
 - DB2 Version 9.7 3
 - Java 196
 - overview 143, 177
 - planning 8
 - post-upgrade tasks 201
 - pre-upgrade tasks 181
 - procedure 193
 - SQL procedures 198
 - scripts
 - overview 145
 - procedure 190
 - SQL replication environments 22
 - tools catalog database 63, 72
 - Windows
 - IBM Data Server Client 129
 - IBM Data Server Runtime Client 131
 - XML Extender 91
 - upgrading to DB2 Version 9.7
 - details v
 - upgrading applications and routines 141
 - upgrading clients 119
 - upgrading DB2 environments 1
 - upgrading DB2 servers 13
 - user-defined routines
 - upgrading 177, 193

V

- verification
 - upgrades
 - post-upgrade tasks for clients 138
 - post-upgrade tasks for DB2 servers 106

W

- Web sites
 - DB2 Migrate Now! 40
 - developerWorks - Information Management 40
 - IBM Virtual Innovation Center 40
- Windows
 - upgrading
 - DB2 servers 59
 - IBM Data Server Client 129
 - IBM Data Server Runtime Client 131
- write-to-table event monitors
 - re-creating after upgrade 105

X

- XML
 - migrating applications from XML Extender 116

- XML data
 - partitioned database environments 107
 - partitioned tables 107
- XML data type
 - migrating applications from XML Extender 116
- XML Extender
 - migrating applications from 116
 - upgrading 91
- XML storage objects
 - converting to Version 9.7 format 103



Printed in USA

SC27-2452-01



Spine information:

IBM DB2 9.7 for Linux, UNIX, and Windows **Version 9 Release 7**

Upgrading to DB2 Version 9.7

