

IBM DB2 10.5
for Linux, UNIX, and Windows

Upgrading to DB2 Version 10.5



IBM DB2 10.5
for Linux, UNIX, and Windows

Upgrading to DB2 Version 10.5



Note

Before using this information and the product it supports, read the general information under Appendix C, "Notices," on page 181.

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 <http://www.ibm.com/shop/publications/order>
- To find your local IBM representative, go to the IBM Directory of Worldwide Contacts at <http://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 IBM Corporation 2006, 2013.

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 database environment 1

Chapter 1. Upgrade to DB2 Version 10.5	3
--	---

Chapter 2. Planning your DB2 environment upgrade 5

Understanding upgrade paths	6
Planning your DB2 servers upgrade	6
Planning your clients upgrade	10
Planning your database applications and routines upgrade	11

Part 2. Upgrading DB2 servers . . . 15

Chapter 3. DB2 servers upgrade.	17
---	----

Chapter 4. Upgrade essentials for DB2 servers 19

DB2 command actions to upgrade instances and databases	19
Upgrade restrictions for DB2 servers	21
DB2 server behavior changes	23
Deprecated or discontinued functionality that affects DB2 server upgrades	27
Disk space requirements for DB2 server upgrades	27
Support changes for 32-bit and 64-bit DB2 servers	29
Best practices for upgrading DB2 servers	30
Migration from non-DB2 relational database management systems	33

Chapter 5. Pre-upgrade tasks for DB2 servers 35

Verifying that your databases are ready for upgrade	36
Backing up databases before or after upgrade	38
Backing up DB2 server configuration and diagnostic information	40
Increasing table space and log file sizes before upgrade	41
Changing raw devices to block devices (Linux)	44
Gathering pre-upgrade diagnostic information.	45
Upgrading DB2 servers in a test environment	46
Creating database duplicates	47
Taking a DB2 server offline for upgrade or for converting to a DB2 pureScale environment	48

Chapter 6. Upgrading a DB2 server (Windows) 49

Upgrading DB2 Version 10.1 or DB2 Version 9.7 instances	50
---	----

Upgrading the DB2 Administration Server (DAS).	53
Upgrading databases	54
Upgrading a server to DB2 Version 10.5 pureScale	57
Scenario 1	58
Scenario 2	58
Scenario 3	59

Chapter 7. Upgrading a DB2 server (Linux and UNIX) 61

Upgrading DB2 Version 10.1 or DB2 Version 9.7 instances	62
Upgrading the DB2 Administration Server (DAS).	64
Upgrading databases	65
Upgrading a server to DB2 Version 10.5 pureScale	69
Scenario 1	69
Scenario 2	70
Scenario 3	70

Chapter 8. Upgrading DB2 servers with specific characteristics. 73

Upgrading DB2 32-bit servers to 64-bit systems (Windows).	73
Upgrading non-root installations	74
Upgrading a DB2 server with multiple DB2 copies	76
Upgrading to a new DB2 server	78
Upgrading a DB2 server by using online backups from a previous release	81
Upgrading partitioned database environments.	82
Upgrading a DB2 pureScale server	83
Upgrading DB2 Version 9.8 instances.	85
Upgrading databases	86
Upgrading DB2 Text Search environments	90
Upgrading DB2 Text Search for administrator or root installation	90
Upgrading DB2 Text Search for non-root installation (Linux and UNIX)	93
Upgrading a multi-partition instance without DB2 Text Search	94
Upgrading DB2 servers in Microsoft Cluster Server environments.	95

Chapter 9. Post-upgrade tasks for DB2 servers 97

Adjusting adaptive compression settings	99
Adjusting the log space size in upgraded databases	100
Activating a database after upgrade	101
Managing DB2 server behavior changes	101
Rebinding packages in upgraded databases	103
Upgrading explain tables	103
Verifying upgrade of DB2 servers	104

Chapter 10. Adopting new Version 10.5 functionality in upgraded databases	107
--	------------

Chapter 11. Migrating DB2 functionality to DB2 database product features	109
Migrating from DB2 Governor to DB2 workload manager	109

Chapter 12. Reversing DB2 server upgrade.	113
--	------------

Part 3. Upgrading clients 115

Chapter 13. Clients upgrade	117
--	------------

Chapter 14. Upgrade essentials for clients	119
Best practices for upgrading clients	121

Chapter 15. Pre-upgrade tasks for clients	123
Backing up client configuration information	123
Upgrading clients in a test environment	124

Chapter 16. Upgrading to Data Server Client (Windows).	127
---	------------

Chapter 17. Upgrading to Data Server Runtime Client (Windows).	129
---	------------

Chapter 18. Upgrading clients (Linux and UNIX).	131
--	------------

Chapter 19. Upgrading to IBM Data Server Driver Package	133
--	------------

Chapter 20. Post-upgrade tasks for clients	135
Verifying your client upgrade	135

Part 4. Upgrading applications and routines 137

Chapter 21. Database applications and routines upgrade.	139
--	------------

Chapter 22. Upgrade essentials for database applications.	141
Upgrade impact from DB2 API changes	143
Upgrade impact from DB2 command changes	143

Upgrade impact from SQL statement changes	145
Upgrade impact from system catalog changes	145

Chapter 23. Upgrade essentials for routines.	149
---	------------

Chapter 24. Pre-upgrade tasks for database applications and routines	151
---	------------

Chapter 25. Upgrading database applications	153
Upgrading embedded SQL applications	154
Upgrading CLI applications	155
Upgrading Java applications that use IBM Data Server Driver for JDBC and SQLJ.	157
Upgrading ADO.NET applications	158
Upgrading scripts	159

Chapter 26. Upgrading routines	161
Upgrading C, C++, and COBOL routines	162
Upgrading Java routines.	164
Upgrading .NET CLR routines	165

Chapter 27. Post-upgrade tasks for database applications and routines	167
--	------------

Chapter 28. Adopting new Version 10.5 functionality in database applications and routines	169
--	------------

Part 5. Appendixes 171

Appendix A. Important references	173
---	------------

Appendix B. Overview of the DB2 technical information	175
DB2 technical library in hardcopy or PDF format	175
Displaying SQL state help from the command line processor	178
Accessing different versions of the DB2 Information Center	178
Terms and conditions.	178

Appendix C. Notices	181
--------------------------------------	------------

Index	185
------------------------	------------

About this book

The Upgrading to DB2 Version 10.5 guide describes the upgrade process and concepts for each component of your DB2® database 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 database environment:

- Part 1, “Upgrading your DB2 database environment,” on page 1
- Part 2, “Upgrading DB2 servers,” on page 15
- Part 3, “Upgrading clients,” on page 115
- Part 4, “Upgrading applications and routines,” on page 137

Part 1. Upgrading your DB2 database environment

This part of the book contains the following chapters:

- Chapter 1, “Upgrade to DB2 Version 10.5,” on page 3
- Chapter 2, “Planning your DB2 environment upgrade,” on page 5

Chapter 1. Upgrade to DB2 Version 10.5

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 10.1, DB2 Version 9.8 or DB2 Version 9.7 copies and you want to upgrade them to DB2 Version 10.5, 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:

- Chapter 3, “DB2 servers upgrade,” on page 17 involves upgrading your existing instances and databases so that they can run in the new release.
- Chapter 13, “Clients upgrade,” on page 117 involves upgrading your client instances to keep the configuration of your existing clients.
- Chapter 21, “Database applications and routines upgrade,” on page 139 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 10.5:

- 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-DB2 Version 10.5 releases* refers to DB2 Version 10.1, DB2 Version 9.8 or DB2 Version 9.7.

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 10.5:

1. Review the new, deprecated, and discontinued functionality for DB2 Version 10.5 and for any releases between the release you are upgrading from and DB2 Version 10.5.
2. Plan how to and modify your database applications and routines. Ensure that they run successfully in DB2 Version 10.5.
3. Set up a DB2 Version 10.5 test server and create test databases.
4. Test your database applications and routines on a DB2 Version 10.5 test database to determine whether they run successfully. If your application requires a client, use a DB2 Version 10.5 client.
5. 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.
6. Upgrade your DB2 servers to DB2 Version 10.5 in your production environment. Ensure that they operate as expected.
7. Upgrade your clients to DB2 Version 10.5 in your production environment. Ensure that your clients operate as expected.
8. Test your database applications and routines in the DB2 Version 10.5 upgraded environment to determine whether they run as expected.
9. Make your upgraded environment available to users.

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 10.5. 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 at www.ibm.com/support (formerly known as DB2 migration portal) that provides access to additional resources and up-to-date

information about the upgrade process as they become available. These resources include educational material, white papers, and webcasts for upgrade.

Finally, plan to remove the use of deprecated functionality and incorporate new functionality from DB2 Version 10.5. 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.

Understanding upgrade paths

You must understand the supported upgrade paths before planning the upgrade of DB2 servers.

If you are upgrading from DB2 Version 9.7 or DB2 Version 10.1, follow the upgrade plan detailed in “Planning your DB2 servers upgrade.”

If you are upgrading from DB2 Version 9.8 follow the upgrade steps detailed in “Upgrading a DB2 pureScale server” on page 83.

Table 1. Upgrade paths

	Version 10.5 single partition Enterprise Server Edition	Version 10.5 multiple partition	Version 10.5 with DB2 pureScale® Feature
Version 9.7 or Version 10.1 single partition Enterprise Server Edition	Yes	Yes	Yes
Version 9.7 or Version 10.1 multiple partition	Yes. Drop all but one partition before or after upgrading the instance to Version 10.5.	Yes	Yes. Instance upgrade from Version 10.5 multi-partition Enterprise Server Edition to a DB2 pureScale instance will be blocked. Consolidate data on a single partition before or after upgrading the instance and database to Version 10.5 and then convert the single partition Enterprise Server Edition instance to DB2 pureScale instance.
Version 9.8 with DB2 pureScale Feature	No	No	Yes

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.

About this task

This task helps you create a an upgrade plan for DB2 servers in DB2 environments other than DB2 pureScale environments.

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 2. Upgrade plan details for DB2 servers.

Upgrade plan	Details
Prerequisites	<p>Ensure that you:</p> <ul style="list-style-type: none">• Ensure you meet the installation requirements for DB2 database products described in <i>Installing DB2 Servers</i>.• Review the information in “Understanding upgrade paths” on page 6• Meet all prerequisites for the upgrade task and subtasks, especially obtaining root or Local Administrator access and required DB2 authorization.• Review the information in the Chapter 4, “Upgrade essentials for DB2 servers,” on page 19 topic. It includes the following:<ul style="list-style-type: none">– “DB2 command actions to upgrade instances and databases” on page 19– “Upgrade restrictions for DB2 servers” on page 21– “DB2 server behavior changes” on page 23– “Deprecated or discontinued functionality that affects DB2 server upgrades” on page 27– “Disk space requirements for DB2 server upgrades” on page 27– “Support changes for 32-bit and 64-bit DB2 servers” on page 29– “Best practices for upgrading DB2 servers” on page 30
Pre-upgrade tasks	<p>Review the list of tasks in the Chapter 5, “Pre-upgrade tasks for DB2 servers,” on page 35 topic. It includes the following:</p> <ul style="list-style-type: none">• “Verifying that your databases are ready for upgrade” on page 36• “Backing up databases before or after upgrade” on page 38• “Backing up DB2 server configuration and diagnostic information” on page 40• “Increasing table space and log file sizes before upgrade” on page 41• “Changing raw devices to block devices (Linux)” on page 44• “Gathering pre-upgrade diagnostic information” on page 45• “Upgrading DB2 servers in a test environment” on page 46• “Taking a DB2 server offline for upgrade or for converting to a DB2 pureScale environment” on page 48

Table 2. 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 10.5 • “Upgrading DB2 Version 10.1 or DB2 Version 9.7 instances” on page 50 (for both Windows and Linux/UNIX) • “Upgrading the DB2 Administration Server (DAS)” on page 53 • “Upgrading databases” on page 54 <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"> • Chapter 6, “Upgrading a DB2 server (Windows),” on page 49 • Chapter 7, “Upgrading a DB2 server (Linux and UNIX),” on page 61 • Chapter 8, “Upgrading DB2 servers with specific characteristics,” on page 73 <p>Take note of the time required to upgrade your databases.</p>

Table 2. Upgrade plan details for DB2 servers. (continued)

Upgrade plan	Details
Post-upgrade tasks	<p>Review the list of tasks in the Chapter 9, “Post-upgrade tasks for DB2 servers,” on page 97 topic. It includes the following:</p> <ul style="list-style-type: none"> • 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. • “Adjusting adaptive compression settings” on page 99 • “Adjusting the log space size in upgraded databases” on page 100 • “Backing up DB2 server configuration and diagnostic information” on page 40 • “Activating a database after upgrade” on page 101 • Modify storage group attributes. For details, see “Storage group attributes.” in <i>Database Administration Concepts and Configuration Reference</i>. • “Managing DB2 server behavior changes” on page 101 • If the automatic collection of statistics failed on certain system catalog tables during database upgrade, see “Collecting catalog statistics” in <i>Troubleshooting and Tuning Database Performance</i> • “Rebinding packages in upgraded databases” on page 103 • Refresh the data in existing materialized query tables • “Upgrading explain tables” on page 103 • “Verifying upgrade of DB2 servers” on page 104 was successful • “Backing up databases before or after upgrade” on page 38 • Migrate to SQL replication Version 10.1. <p>In addition, consider adding the following tasks to your upgrade plan:</p> <ul style="list-style-type: none"> • Database log directories will have been changed • If you upgrade a DB2 server running high availability disaster recovery (HADR) replication, you must initialize HDAR replication. For details, see “Initializing high availability disaster recovery (HADR)” in <i>Data Recovery and High Availability Guide and Reference</i>. • After updating statistics for your upgraded databases, determine if index or table reorganization is necessary by running the REORGCHK command. For details, see “Determining when to reorganize tables and indexes” in <i>Troubleshooting and Tuning Database Performance</i>. • 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 or discontinued functionality that affects DB2 server upgrades” on page 27 • Chapter 10, “Adopting new Version 10.5 functionality in upgraded databases,” on page 107, where appropriate, to improve performance at the DB2 server level. <p>Review manageability, performance, and scalability enhancements in What's New for DB2 Version 10.5 to determine what new functionality you might want to apply to your environment.</p>

2. If you must be able to reverse the upgrade, add details to the plan about the tasks required to Chapter 12, “Reversing DB2 server upgrade,” on page 113. 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 3. Upgrade plan details for clients.

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 in Chapter 14, “Upgrade essentials for clients,” on page 119 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	<p>Include the following tasks:</p> <ul style="list-style-type: none"> • Chapter 3, “DB2 servers upgrade,” on page 17 • “Backing up client configuration information” on page 123 <p>In addition, check the list of Chapter 15, “Pre-upgrade tasks for clients,” on page 123 for optional tasks that you might want to perform for your environment such as “Upgrading clients in a test environment” on page 124.</p>
Upgrade task	<p>You must include these steps:</p> <ul style="list-style-type: none"> • Install DB2 Version 10.5 client • Upgrade client instance <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"> • Chapter 16, “Upgrading to Data Server Client (Windows),” on page 127 • Chapter 17, “Upgrading to Data Server Runtime Client (Windows),” on page 129 • Chapter 18, “Upgrading clients (Linux and UNIX),” on page 131
Post-upgrade tasks	<p>Include the following tasks:</p> <ul style="list-style-type: none"> • Review “DB2 server behavior changes” on page 23 • “Verifying your client upgrade” on page 135 was successful • Bind the database utilities and the DB2 CLI bind files. For details, see “Binding bind files after installing fix packs”.

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 4. Upgrade plan details for database applications.

Upgrade plan	Details
Prerequisites	<p>Ensure that you:</p> <ul style="list-style-type: none">• meet the installation prerequisitesinstallation requirements for DB2 database products described in <i>Installing DB2 Servers</i>.• meet the development software requirements. For details, see “Support for elements of the database application development environment” in <i>Getting Started with Database Application Development</i>• resolve any support issues in Chapter 22, “Upgrade essentials for database applications,” on page 141 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">• Chapter 13, “Clients upgrade,” on page 117 or install the DB2 Version 10.5 application driver.• Test your database applications in a DB2 Version 10.5 testing environment. If your applications run successfully, the rest of the upgrade steps are not required. <p>In addition, check the list of Chapter 24, “Pre-upgrade tasks for database applications and routines,” on page 151 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

Table 4. Upgrade plan details for database applications. (continued)

Upgrade plan	Details
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 10.5 and to remove use of functionality that is discontinued in DB2 Version 10.5. • 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 10.5. <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 154 • “Upgrading CLI applications” on page 155 • “Upgrading Java applications that use IBM Data Server Driver for JDBC and SQLJ” on page 157 • “Upgrading ADO.NET applications” on page 158 • “Upgrading scripts” on page 159
Post-upgrade tasks	<p>Perform the recommended Chapter 27, “Post-upgrade tasks for database applications and routines,” on page 167, especially:</p> <ul style="list-style-type: none"> • Tune performance of your database applications. • Remove the use of “Deprecated or discontinued functionality that affects DB2 server upgrades” on page 27. • Chapter 28, “Adopting new Version 10.5 functionality in database applications and routines,” on page 169 where appropriate.

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

Table 5. 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. For details, see “Support for elements of the database application development environment” in <i>Getting Started with Database Application Development</i>. • resolve any support issues in Chapter 23, “Upgrade essentials for routines,” on page 149 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 10.5 testing environment. If your routines run successfully, the rest of the upgrade steps are not required. <p>In addition, check the list of Chapter 24, “Pre-upgrade tasks for database applications and routines,” on page 151 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>

Table 5. Upgrade plan details for routines. (continued)

Upgrade plan	Details
Upgrade task	<p>You must include these steps:</p> <ul style="list-style-type: none"> • Modify your routines to support changes in DB2 Version 10.5 and to remove use of functionality that is discontinued in DB2 Version 10.5. • 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 10.5. <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 162 • “Upgrading Java routines” on page 164 • “Upgrading .NET CLR routines” on page 165
Post-upgrade tasks	<p>Perform the recommended Chapter 27, “Post-upgrade tasks for database applications and routines,” on page 167, especially:</p> <ul style="list-style-type: none"> • Remove the use of “Deprecated or discontinued functionality that affects DB2 server upgrades” on page 27 • Chapter 28, “Adopting new Version 10.5 functionality in database applications and routines,” on page 169 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 17
- Chapter 4, “Upgrade essentials for DB2 servers,” on page 19
- Chapter 5, “Pre-upgrade tasks for DB2 servers,” on page 35
- Chapter 6, “Upgrading a DB2 server (Windows),” on page 49
- Chapter 7, “Upgrading a DB2 server (Linux and UNIX),” on page 61
- Chapter 8, “Upgrading DB2 servers with specific characteristics,” on page 73
- Chapter 9, “Post-upgrade tasks for DB2 servers,” on page 97
- Chapter 11, “Migrating DB2 functionality to DB2 database product features,” on page 109
- Chapter 10, “Adopting new Version 10.5 functionality in upgraded databases,” on page 107
- Chapter 12, “Reversing DB2 server upgrade,” on page 113

Chapter 3. DB2 servers upgrade

Upgrading to DB2 Version 10.5 requires that you upgrade your existing DB2 servers.

Upgrading your DB2 server requires that you install a DB2 Version 10.5 copy and then upgrade all the instances and databases to be able to run them under the DB2 Version 10.5 copy.

You can directly upgrade existing DB2 Version 9.7, DB2 Version 9.8, or DB2 Version 10.1 instances and databases to DB2 Version 10.5. Learn details, limitations about the upgrade process, and possible issues that you must be aware of in Chapter 4, “Upgrade essentials for DB2 servers,” on page 19. Refer to the upgrading DB2 server tasks for details on how to upgrade to DB2 Version 10.5. In the upgrading DB2 server topics, the term *pre-DB2 Version 10.5 copy* refers to DB2 Version 9.7, Version 9.8, or Version 10.1.

On Windows operating systems, you have an option to automatically upgrade an existing pre-DB2 Version 10.5 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 before DB2 Version 9.7, upgrade them first to DB2 Version 9.7 or DB2 Version 10.1, and then upgrade to DB2 Version 10.5. It is recommended that you upgrade to the latest fix pack of DB2 Version 9.7. To move to a DB2 pureScale environment, upgrade to DB2 Version 9.7 and then upgrade to DB2 Version 10.5. Upgrading to DB2 Version 9.8 as an intermediate release requires that you upgrade to DB2 Version 9.7 first.

Upgrade to DB2 Version 10.5 is supported for the following DB2 products:

Table 6. DB2 database products supported for upgrade

DB2 Version	DB2 product name
Version 10.1	<ul style="list-style-type: none"> • DB2 Advanced Enterprise Server Edition • 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 • IBM[®] DB2 Performance Optimization Feature for Enterprise Server Edition • DB2 Storage Optimization Feature • IBM DB2 Advanced Access Control Feature • IBM DB2 High Availability Feature for Express Edition • IBM Homogeneous Replication Feature for DB2 Enterprise Server Edition • IBM Data Server Client • IBM Data Server Runtime Client
Version 9.8	IBM DB2 pureScale Feature
Version 9.7	<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 • IBM DB2 Performance Optimization Feature for Enterprise Server Edition • DB2 Storage Optimization Feature • IBM DB2 Advanced Access Control Feature • IBM DB2 High Availability Feature for Express Edition • IBM Homogeneous Replication Feature for DB2 Enterprise Server Edition • IBM Data Server Client • IBM Data Server Runtime Client

For DB2 products not supported, refer to “Deprecated or discontinued functionality that affects DB2 server upgrades” on page 27.

Chapter 4. Upgrade essentials for DB2 servers

Upgrading DB2 servers to DB2 Version 10.5 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 10.5:

- “DB2 command actions to upgrade instances and databases”
- “Upgrade restrictions for DB2 servers” on page 21
- “Best practices for upgrading DB2 servers” on page 30
- “Disk space requirements for DB2 server upgrades” on page 27
- “Support changes for 32-bit and 64-bit DB2 servers” on page 29
- “DB2 server behavior changes” on page 23
- “Deprecated or discontinued functionality that affects DB2 server upgrades” on page 27
- “Migration from non-DB2 relational database management systems” on page 33

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 10.5 on Windows and select the **Work with Existing** option and then choose a pre-Version 10.5 copy with the **upgrade** action, the command does the following things:

- Calls the **db2ckupgrade** command.
- Upgrades an existing instance to a new instance under a DB2 Version 10.5 copy.
- Upgrades instance profile registry variables. The global profile registry variables set by the user are not upgraded.
- Upgrades the database manager configuration file.
- Sets the **jdk_path** database manager configuration parameter.
- 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**.

For a successful instance upgrade, all files must exist for all instances and all files must have write access granted.

Review the **db2iupgrade** command for more information about the command and the options that can be specified.

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 the following database entities might be converted during the database upgrade:

- Database configuration file
- Log file header
- Global log file header file
- Table root page for all tables
- Index root page for all tables
- Catalog tables
- Storage group files
- Buffer pool files
- Table space files
- History file

For recoverable databases, the **UPGRADE DATABASE** command renames all the 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. For details, see Chapter 9, “Post-upgrade tasks for DB2 servers,” on page 97.

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 7. **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.
Enabled	Does not exist	RUNSTATS command with default parameters
Disabled	N/A	RUNSTATS command from the most recent call to the RUNSTATS command. ¹

Note:

1. 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 10.5 is supported from DB2 Version 10.1, DB2 Version 9.8 and DB2 Version 9.7. If you have an earlier version of DB2, you must upgrade to DB2 Version 9.7 or DB2 Version 10.1 before upgrading to DB2 Version 10.5.
- Upgrading to a DB2 Version 10.5 non-root installation is supported from a DB2 Version 10.1 and DB2 Version 9.7 non-root installation. Upgrading to a DB2 Version 10.5 non-root installation from a pre-DB2 Version 10.5 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 10.5. This action automatically installs DB2 Version 10.5 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 10.5. 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 10.5 is installed, and support for 32-bit kernels and 64-bit kernels has changed. See Table 12 on page 29.
- Upgrading from a system with multiple copies of DB2 Version 9.7 and DB2 Version 10.1 of 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-DB2 Version 10.5 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 “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.
- 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.

What is unsupported?

DB2 Version 10.5 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 10.5 or upgrade to a new DB2 server that meets the operating system requirements. See “Upgrading to a new DB2 server” on page 78 and “Installation requirements for DB2 database products” in *Installing DB2 Servers*.
- A version of DB2 before Version 9.7 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 10.5 or later. Run the **db2iupdt** command to update to a different fix pack levels or copies of DB2 Version 10.5.
- You try to upgrade from DB2 Version 10.5 back to DB2 Version 10.1 or DB2 Version 9.7. Chapter 12, “Reversing DB2 server upgrade,” on page 113 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 10.5 copy is unsupported. The following table describes the upgrade support for each type of instance by DB2 database product:

Table 8. Instance upgrade support for DB2 Version 10.5 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	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.
ese - default type for DB2 Enterprise Server Edition, DB2 Advanced Enterprise Server Edition, DB2 Workgroup Server Edition, and DB2 Advanced Workgroup Server Edition.	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 8 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. The **db2iupdt** command does not support downgrading from a higher-level instance type to a lower-level instance type. You can downgrade the instance type manually but avoid doing so 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 10.5.

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.
- A database is enabled as a high availability disaster recovery (HADR) standby database.

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 10.5 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 registry 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 parameter
- Deprecated and discontinued database configuration parameters
- Changes to physical design characteristics of databases
- Changes to authorities and privileges

New registry variables

New registry variables introduced in DB2 Version 10.5 have no impact on DB2 upgrade.

For more information, see “Some registry and environment variables have changed” in *What’s New for DB2 Version 10.5*.

Changes to existing registry variables

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

Table 9. Changes to existing registry variables

Name	Upgrade impact
DB2DSDRIVER_CFG_PATH	This variable now specifies multiple configuration files at same or different locations with different names. If filename is not specified in a path and name pair, then the file name defaults to a value of db2dsdriver.cfg.

For more information, see “Some registry and environment variables have changed” in *What’s New for DB2 Version 10.5*.

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 10.5* to determine the upgrade impact of deprecated registry variables. See “Discontinued registry variables” in *What’s New for DB2 Version 10.5* to determine the upgrade impact of discontinued registry variables.

If you are upgrading from DB2 Version 10.1 or earlier, consider removing deprecated registry variables in pre-Version 10.5 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 10.5 releases as they do not have the intended effect. See “Changes that impact Version 10.1 or earlier releases” on page 26 for details.

New database manager configuration parameters

New database manager configuration parameters introduced in DB2 Version 10.5 have no impact on DB2 upgrade. If you are upgrading from DB2 Version 9.8 or DB2 Version 9.7, review new database manager parameter in Version 10.1 that have an impact when upgrading from those releases. See DB2 server behaviour changes for details.

Changes to existing database manager configuration parameters

Changes to database manager configuration parameters introduced in DB2

Version 10.5 have no impact on DB2 upgrade. If you are upgrading from DB2 Version 9.8 or DB2 Version 9.7, review the changes for database manager configuration parameters in Version 10.1 that have an impact when upgrading from those releases. See DB2 server behaviour changes 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 9.7 or Version 9.8, consider removing deprecated database manager configuration parameters in Version 10.1 releases because the functionality associated with the parameters is obsolete or has been replaced by new functionality. Also, remove the use of discontinued database manager configuration parameters in Version 10.1 releases as they do not have the intended effect. See “Changes that impact Version 10.1 or earlier releases” on page 26 for details.

New database configuration parameters

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

Table 10. New database configuration parameters

Name	Upgrade impact
dft_table_org	This parameter specifies whether a user table is created as a column-organized table (value COLUMN) or a row-organized table (value ROW) if neither the ORGANIZE BY COLUMN nor the ORGANIZE BY ROW clause is specified on the CREATE TABLE statement. The default value for this parameter is ROW which has no impact on upgrade. If you have DDL scripts and you plan to change the default orientation of tables, modify your exiting scripts to specify the ORGANIZE BY COLUMN or the ORGANIZE BY ROW clause for the CREATE TABLE statements to ensure that tables are created with the proper orientation regardless of the setting of this parameter.
nchar_codeset	This parameter determines which data type that national character strings are mapped to.
string_units	This parameter specifies the default string units that are used when you are defining character data types and graphic data types. For upgraded Unicode databases, this parameter is set to GRAPHIC_CU32. However, databases created in DB2 Version 10.5 have this parameters set to CHAR_CU32 by default. For consistent mapping in environments that have upgraded databases and newly created databases, set the configuration parameter to the same value in all databases. There is no upgrade impact for non-Unicode databases because this parameter is set to GRAPHIC_CU16 for both upgraded and newly created databases.

For more information, see “Some database configuration parameters have been changed” in *What’s New for DB2 Version 10.5*.

Changes to existing database configuration parameters

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

Table 11. Changes to existing database configuration parameters

Name	Upgrade impact
hadr_syncmode	In Version 10.5, the default value for hadr_syncmode is changed from NEARSYNC to ASYNC for DB2 pureScale databases. For all other database types, the default for hadr_syncmode remains NEARSYNC.
hadr_target_list	In Version 10.5, initializing HADR without setting this parameter is deprecated. You should set this parameter regardless of the number of standby databases as part of the initialization process. For more details, see <i>Initializing HADR has changed</i> .

For more information, see “Some database configuration parameters have been changed” in *What’s New for DB2 Version 10.5*.

Deprecated and discontinued database configuration parameters

You should remove the use of database configuration parameters that are deprecated and discontinued because the functionality associated with the variable is obsolete or has been replaced by new functionality. See “Some database configuration parameters are deprecated or discontinued” in *What’s New for DB2 Version 10.5* to determine the upgrade impact of deprecated and discontinued database configuration parameters.

If you are upgrading from DB2 Version 9.7, consider removing deprecated database configuration parameters in pre-Version 10.1 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 Version 10.1 as they do not have the intended effect. See “Changes that impact Version 10.1 or earlier releases” for details.

Changes to physical design characteristics of databases

Review the What’s New and Changed documentation to determine if there are any changes to the physical design characteristics of databases that impact upgrade.

Changes to authorities and privileges

There are no changes to the authorities and privileges in this release.

See “Upgrade impact from DB2 command changes” on page 143 and “Upgrade impact from SQL statement changes” on page 145 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 10.1 or earlier releases

If you are upgrading from DB2 Version 10.1 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 10.5 releases that might also impact your upgrade:

- DB2 server behavior changes between DB2 Version 9.7 and DB2 Version 10.1
- DB2 server behavior changes between DB2 Version 9.5 and DB2 Version 9.7

Deprecated or discontinued functionality that affects DB2 server upgrades

You should be aware of functionality that is deprecated or discontinued in DB2 Version 10.5 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 10.5 is unsupported.

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

Deprecated or discontinued commands

The **db2IdentifyType1** command and the **STATISTICS YES** parameter of the **LOAD** command are discontinued. For more information, see DB2 command and SQL statement changes summary for details.

Review “Upgrade impact from DB2 command changes” on page 143 to learn what commands are deprecated and discontinued in DB2 Version 10.5 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.

Functionality that was deprecated or discontinued in DB2 pre-V10.5 releases

If you are upgrading from DB2 V10.5 versions, you must also review the changes made in DB2 Version 10.1 or Version 9.7 that might impact your environment after upgrading to DB2 Version 10.5. Review the following topic to learn about additional possible impacts on the upgrade of your DB2 server:

- Deprecated or discontinued functionality in DB2 Version 10.1 for upgrade from DB2 Version 10.1.
- Deprecated or discontinued functionality in DB2 Version 9.7 for upgrade from DB2 Version 9.7.

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 10.5.

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 automatic storage table spaces, you can increase the space on the current storage paths or add a new storage path.

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 10.5, 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 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 9.7 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 10.5 is installed. The following table summarizes the DB2 Version 10.5 bit size support that is available for each of the following operating systems:

Table 12. DB2 Version 10.5 32-bit and 64-bit support available per operating system

Operating systems	DB2 Version 10.5 support available
<ul style="list-style-type: none">• 32-bit Windows on x86 and x64 (Using DB2 Version 10.5 32-bit product)• 32-bit Linux on x86• 64-bit Windows on x64 (Using DB2 Version 10.5 32-bit product)	For DB2 Version 10.5 Developer Edition: <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 12. DB2 Version 10.5 32-bit and 64-bit support available per operating system (continued)

Operating systems	DB2 Version 10.5 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 10.5 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 10.5 Developer Edition on a 32-bit system. You can only upgrade to 32-bit instances on 32-bit Windows or 32-bit Linux on x86. For any other editions in DB2 Version 10.5, you must upgrade to 64-bit system.
- If you have 64-bit instances and you upgrade to DB2 Version 10.5 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 10.5 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 12 on page 29 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.

Best practices for upgrading DB2 servers

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

Review changes in existing DB2 database product functionality

Changes in existing functionality introduced in DB2 Version 10.5 can potentially impact your applications, scripts, maintenance processes, and any other aspects related your DB2 server upgrade process.

Changes in existing functionality introduced in pre-DB2 Version 10.5 releases can also have an impact. Review these changes and plan how to address these changes before the upgrade:

- Changed functionality in DB2 Version 9.7
- Changed functionality in DB2 Version 9.8
- Changed functionality in DB2 Version 10.1

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 before DB2 database product upgrade

The supported UNIX, Linux and Windows operating systems have changed in DB2 Version 10.5. Review “Installation requirements for DB2 servers and IBM data server clients” in *DB2 pureCluster Feature Installation and Upgrade Guide* to determine whether your operating system version is supported and if you need to upgrade your operating system before installing DB2 Version 10.5. Newer versions of operating systems can also bring new hardware requirements.

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

If you have a DB2 Version 9.7 copy on SUSE Linux Enterprise Server 10, first apply DB2 Version 9.7 Fix Pack 2 or later, before you upgrade the operating system to SUSE Linux Enterprise Server 11.

If you are upgrading a pre-DB2 Version 10.5 copy on POWER4 processor-based systems, first upgrade to POWER10 processor-based systems before upgrading to DB2 Version 10.5. POWER3 processor-based systems are not supported in DB2 Version 10.5.

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 **db2exp1n** 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 fall back from DB2 Version 10.5 to a pre-DB2 Version 10.5 release. See Chapter 12, “Reversing DB2 server upgrade,” on page 113 to learn all the required steps to reverse a database upgrade.

Perform pre-upgrade tasks

There are several pre-upgrade tasks outlined in the Chapter 5, “Pre-upgrade tasks for DB2 servers,” on page 35 topic that you should execute for a successful upgrade, such as backing up DB2 configuration parameters settings, ensure that you have enough disk free space for table spaces and log files, and verifying that databases are ready for upgrade.

Determine whether to upgrade DB2 servers or clients 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 manages any incompatibilities between releases. See Best practices for upgrading clients in the DB2 Version 10.5 documentation for details about incompatibilities. See “DB2 client considerations for the DB2 pureScale Feature” in *DB2 pureCluster Feature Installation and Upgrade Guide* for details about supported Version 9.8 functionality.

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.

Review the factors that can impact your database application upgrade or routine upgrade and make any necessary changes to your database applications and routines to ensure that they run after the upgrade. See Chapter 22, “Upgrade essentials for database applications,” on page 141 and Chapter 23, “Upgrade essentials for routines,” on page 149 for details about the factors that can impact your database application upgrade or routine upgrade.

In an upgrade testing environment, you can test and verify that your database applications and routines run successfully in DB2 Version 10.5 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 10.5 changes the database role from primary to standard. Upgrading standby databases to DB2 Version 10.5 is not supported because these databases are in rollforward pending state. Because of these restrictions, upgrading an HADR environment to DB2 Version 10.5 requires that you stop HADR, upgrade your DB2 server where the primary database resides, and then reinitialize HADR.

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

- Stop the HADR primary or standby databases as indicated in the Chapter 5, “Pre-upgrade tasks for DB2 servers,” on page 35.
- Upgrade the DB2 server where the primary database resides using one of the following tasks:
 - Chapter 6, “Upgrading a DB2 server (Windows),” on page 49
 - Chapter 7, “Upgrading a DB2 server (Linux and UNIX),” on page 61
- Reinitialize HADR as indicated in the Chapter 9, “Post-upgrade tasks for DB2 servers,” on page 97

Before you activate your standby database, you must move the log files from the previous version of DB2 out of the log path of the new version of DB2. If you do not move the log files, DB2 might try to use the old files and fail to initialize.

Migrating SQL replication environments

After upgrading your database servers, you can optionally migrate your SQL replication environment to DB2 Version 10.5. See “Migrating to SQL replication Version 10.5” 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 10.1, see *Upgrading to DB2 Spatial Extender Version 10.1* in *Spatial Extender 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, install DB2 Version 10.5 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 95 for details.

Upgrading from Query Patroller to Workload Manager

Query Patroller is discontinued. See *Migrating from Query Patroller to DB2 workload manager* for details on how to migrate.

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 Migration station Web page can help you to find the information that you need to port your application and its data from other database management systems. This Web page 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 Migration Factory 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 website offers technical resources for DB2 Information Management software. It features product information, downloads, learning resources, support, and communities. On this website 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 before upgrade in order to have a successful upgrade.

Procedure

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

1. Ensure that you have at least one free page of index space per object index to eliminate the overhead of a potential index rebuild. If an index root page does not have enough free space during upgrade, then the index will need to grow by one page. If a free page cannot be found in the index object, then a page will be requested from the tablespace. If the tablespace is full, then the entire index object will be marked invalid and will be rebuilt when the underlying table is accessed for the first time after upgrade.
2. 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.
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.

Refer to “Verifying that your databases are ready for upgrade” on page 36.

4. Upgrade from DB2 Query Patroller to Workload Manager. Query Patroller is discontinued. Perform the steps in “Migrating from Query Patroller to DB2 workload manager” in the DB2 Version 9.7 documentation.

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.

Refer to “Backing up databases before or after upgrade” on page 38.

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.

Refer to “Backing up DB2 server configuration and diagnostic information” on page 40.

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.

Refer to “Disk space requirements for DB2 server upgrades” on page 27 and “Increasing table space and log file sizes before upgrade” on page 41.

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-DB2 Version 10.5 copy.
You do not need to backup standard code page conversion tables. Upgrading your pre-DB2 Version 10.5 copy removes these tables because standard code page tables are contained in a DB2 Version 10.5 library.
10. Linux only: Change raw devices to block devices.
Refer to “Changing raw devices to block devices (Linux)” on page 44.
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.
Refer to “Upgrading DB2 servers in a test environment” on page 46.
12. If the diagnostic error capture level (set by the **diaglevel** parameter) is 2 or less, set this parameter to 3 or higher before upgrading. See “Setting the diagnostic log file error capture level” in *Troubleshooting and Tuning Database Performance*.
13. Take the DB2 server offline for upgrade.
Refer to “Taking a DB2 server offline for upgrade or for converting to a DB2 pureScale environment” on page 48.
14. Refresh the data in existing materialized query tables. All materialized query tables that depend on the system views are dropped during database upgrade. After upgrade you must refresh the data in existing materialized query tables by using the **REFRESH TABLE** statement.

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 is 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 affect 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.

Before you begin

- 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 10.5 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*.

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 10.5 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 must insert the DB2 Version 10.5 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 that is created in the current directory that includes details on errors and warnings.

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

In a partitioned database environment, the **db2ckupgrade** command must be issued only once. It checks all partitions.

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 fails.
6. If you created user-defined objects that are dependent on discontinued administrative routines, drop the dependent objects and re-create them using the routine or view that replaces the discontinued routine. The **db2ckupgrade** command returns the DBT5534W warning message when a user-defined object is dependent on discontinued administrative routines. If you upgrade a database that has dependent objects, the **UPGRADE DATABASE** command drops the discontinued administrative routines and marks the dependent objects inoperative or invalid.

For more details, see “Some administrative routines are discontinued” in *What’s New for DB2 Version 10.5*.

7. If you created workload management objects that have a conflict with system-reserved ID during database upgrade, drop these objects and re-create them after you upgrade the database. The **db2ckupgrade** command returns the DBT5512E error message when a workload management object cannot be upgraded because the ID of that object conflicts with a system-reserved ID. Perform the following actions:
 - a. Generate the DDL statements to re-create the workload management objects by issuing the **db2look** command with the **wlm** parameter.
 - b. Drop all of the workload management objects from the database.

After upgrading the database, re-create the workload management objects in the upgraded database by issuing the DDL statements that you generated with the **db2look** command.

8. 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 fails.

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 145 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. In some workload attributes, if the intention is to represent an actual asterisk, then you can use two asterisks (**). The **UPGRADE DATABASE** command replaces the single asterisk (*) with two asterisks (**) depending the type of connection attribute.

11. If you created global variables of XML data type or created compiled SQL functions with parameters of XML data type or XML data type in the RETURNS clause, you must upgrade to the Version 10.1 Fix Pack 1 software or later fix pack releases that support the XML data type in these database objects. If you decide to upgrade to the Version 10.1 software, you must drop these database objects and re-create them specifying a supported data type.

The **db2ckupgrade** command writes the ADM4004W warning message to the log file when a database has global variables of XML data type or compiled SQL functions with parameters of XML data type or XML data type in the RETURNS clause. The XML data type is not supported on these database objects. Therefore, these database objects are invalidated during the database upgrade.

12. Ensure that the log file for **db2ckupgrade** command contains the following text: Version of DB2CKUPGRADE being run: Version 10.5. This text confirms that you are running the correct level of the **db2ckupgrade** command.
13. Check and fix any invalid flavor fields on SQLSPCS files by using the **fixtbspflvr** tool. Details about this tool can be obtained from <http://www.ibm.com/support/>.

Backing up databases before or after upgrade

Before and after the upgrade process to DB2 Version 10.5, 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 10.5, you cannot backup databases until you upgrade them.

Before you begin

- 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:

```
db2 LIST DATABASE DIRECTORY
```

Procedure

To perform a full offline 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:

```
db2 LIST APPLICATIONS
```

If all applications are disconnected, this command returns the following message:

```
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 database_alias USER username USING password TO backup-dir
```

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

In partitioned database environments, back up all database partitions. For details, see “Backing up partitioned databases” in *Data Recovery and High Availability Guide and Reference*.

If you activated and configured DB2 Advanced Copy Services (ACS) on your databases in DB2 Version 9.7 or later, you can use the **USE SNAPSHOT** parameter to perform a snapshot backup. However, you can only restore a snapshot backup to an instance of the same version. You cannot use snapshot backup to upgrade to a new server. For details, see Performing a snapshot backup in *Data Recovery and High Availability Guide and Reference*.

If you performed a full online or offline database backup recently and you cannot perform another one before upgrading, you can perform an incremental offline database backup instead

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

```
cd backup-dir  
db2ckbkp SAMPLE.0.arada.NODE0000.CATN0000.20091014114322.001
```

```
[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, or conversion to DB2 pureScale, allows you to verify DB2 server behavior after upgrade, or converting to DB2 pureScale, and to re-create instances and databases.

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

Before you begin

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, or convert to DB2 pureScale, 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 in the `db2support.zip` compressed zip file under the output directory. A summary report in HTML format is included. In the `db2supp_opt.zip` file that is also included, you should check the `optimizer.log` file to verify that the collection of information was performed successfully.

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

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/sqllib/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.

- 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.

- 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.

- 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
```

- 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.

Before you begin

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

About this task

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, MEMBER, 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   TABLE(MON_GET_TABLESPACE(NULL,-2)) AS T
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.

3. 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"
```

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.

4. 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 **logfilesiz**, **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 roll back 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 .

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.

Before you begin

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

Restrictions

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.

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.

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
```

Gathering pre-upgrade diagnostic information

Before creating or upgrading an instance and before updating to the next fix pack, you might need to gather diagnostic information to help troubleshoot any problem that might come up after the upgrade or update.

Before you begin

Some of the collections that are performed will take a long time to complete. Please have a sufficient amount of time before your scheduled upgrade or update to complete the collection of the diagnostic information.

About this task

If you plan to create or upgrade an instance, or update to the next available fix pack, it is helpful to gather performance, configuration, and environment information to help diagnose any future problems that might arise after you perform the upgrade or update. The gathering of this diagnostic information is done through the **db2fodc -preupgrade** and **db2support -preupgrade** commands.

Restrictions

You must be using Version 9.7 Fixpack 5 or later to use the **db2fodc -preupgrade** and **db2support -preupgrade** commands.

Procedure

To gather a sufficient amount of information to diagnose any future problems that might arise when performing an upgrade or update, you need to perform the following steps:

1. Issue the **db2fodc -preupgrade -db database_name** command at high usage and idle times.

This command collects performance related information that might be needed for future problems. After collection is completed the information is stored in a newly created directory named FODC_Preupgrade_<timestamp>_<member>.

Note: To gather better performance information, issue the **db2fodc -preupgrade** command multiple times at different usage levels. This gives IBM support a more complete picture of the performance of DB2.

2. Issue the **db2support -preupgrade -d database_name** command.

This command collects configuration and environment information, and information from the FODC preupgrade directories created previously.

Results

After collection is completed a db2support_preupgrade.zip file which contains all the collected information is created in the current directory.

What to do next

If any problems arise after the upgrade or update you might be required to send the db2support_preupgrade.zip file to IBM support for analysis. The db2support_preupgrade.zip file must be kept until it is determined that the upgrade or update is functioning normally.

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 10.5.

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.

Before you begin

You must have root user 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 10.1, DB2 Version 9.8 , or DB2 Version 9.7. 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. Perform the steps in “Creating database duplicates” on page 47 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 10.5.
6. Perform the steps in “Upgrading DB2 Version 10.1 or DB2 Version 9.7 instances” on page 50.
7. Perform the steps in “Upgrading databases” on page 54. 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, MEMBER, 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 TABLE(MON_GET_TABLESPACE(NULL,-2)) AS T
      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 the steps in Chapter 9, “Post-upgrade tasks for DB2 servers,” on page 97 that apply to your DB2 server.
10. Perform the steps in “Verifying upgrade of DB2 servers” on page 104 to ensure 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 10.5 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.

Before you begin

Ensure that you have SYSCTRL or SYSADM authority.

About this task

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 to re-create 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. Refer to “Increasing table space and log file sizes before upgrade” on page 41.
4. Export data subsets of your production databases and import these data subsets into your test databases. For details, see “Exporting Data” and “Importing Data” in *Data Movement Utilities Guide and Reference*. 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 for upgrade or for converting to a DB2 pureScale environment

Before you can continue with the upgrade process, or the conversion of your environment for DB2 pureScale, 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.

Before you begin

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 10.5 requires that you install a new DB2 Version 10.5 copy and then upgrade your existing instances and databases to this new copy.

If you choose to automatically upgrade your existing pre-DB2 Version 10.5 copy during the DB2 Version 10.5 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 10.5 copy, you must manually upgrade your instances, your DAS, and databases.

This upgrade task describes the steps for direct upgrade to DB2 Version 10.5 from DB2 Version 10.1 or DB2 Version 9.7. Review the steps in upgrading environments with specific characteristics and determine which task applies better to your environment.

### Before you begin

- 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. Refer to “Best practices for upgrading DB2 servers” on page 30 and “Disk space requirements for DB2 server upgrades” on page 27.
- Perform pre-upgrade tasks. Refer to Chapter 5, “Pre-upgrade tasks for DB2 servers,” on page 35.

### Restrictions

- This procedure applies only to upgrade from DB2 32-bit servers when you install the DB2 Version 10.5 32-bit database product or from DB2 64-bit servers when you install the DB2 Version 10.5 64-bit database product. The instance bit size is determined by the operating system and the DB2 Version 10.5 database product that you install, see “Support changes for 32-bit and 64-bit DB2 servers” on page 29 for details.
- Additional upgrade restrictions apply. Refer to “Upgrade restrictions for DB2 servers” on page 21. Review the complete list.

### Procedure

To upgrade a DB2 server to DB2 Version 10.5:

1. Log on to the DB2 server as a user with Local Administrator authority.
2. Install DB2 Version 10.5 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. Refer to “Verifying that your databases are ready for upgrade” on page 36.

- To create a new copy of DB2 Version 10.5, 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 10.5 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 10.5, upgrade your DB2 Version 10.1 or DB2 Version 9.7 instances to this new copy. Refer to “Upgrading DB2 Version 10.1 or DB2 Version 9.7 instances.”
5. 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 10.5. Refer to “Upgrading the DB2 Administration Server (DAS)” on page 53.
6. Upgrade your databases. Refer to “Upgrading databases” on page 54.

## 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. Refer to Chapter 9, “Post-upgrade tasks for DB2 servers,” on page 97 and “Verifying upgrade of DB2 servers” on page 104.

---

## Upgrading DB2 Version 10.1 or DB2 Version 9.7 instances

As part of the overall process of upgrading your DB2 database server to DB2 Version 10.5, you must upgrade your instances.

### Before you begin

- You must have root user authority on Linux and UNIX operating systems or Local Administrator authority on Windows.
- You must install any DB2 database 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. This step is important in partitioned database environments because the **db2ckupgrade** command might

return an error in one database partition and cause the instance upgrade to fail. Refer to “Verifying that your databases are ready for upgrade” on page 36.

- On Linux and UNIX operating systems, ensure that there is 5GB of free space in the /tmp directory. The instance upgrade trace file is written to /tmp.
- Gather pre-upgrade diagnostic information to help diagnose any problem that might occur after the upgrade.

## About this task

On Linux and UNIX operating systems, you must manually upgrade your instances. On Windows operating systems, you must manually upgrade them if you did not choose to automatically upgrade your existing DB2 copy during the DB2 Version 10.5 installation.

If you are upgrading from DB2 Version 9.8, follow the steps in “Upgrading a DB2 pureScale server” on page 83.

### 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.
- For additional restrictions on instance upgrade, review “Upgrade restrictions for DB2 servers” on page 21.
- You must be upgrading from DB2 Version 10.1 or DB2 Version 9.7.

## Procedure

To manually upgrade your existing instances to DB2 Version 10.5 using the **db2iupgrade** command:

1. Determine if you can upgrade your existing instances to a DB2 Version 10.5 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 | Examples                                                                                                    |
|------------------|-------------------------------------------------------------------------------------------------------------|
| Linux and UNIX   | db2 GET DBM CFG   grep 'Node type'<br>Node type = Partitioned database server with local and remote clients |
| Windows          | db2 GET DBM CFG   find "Node type"<br>Node type = Partitioned database server with local and remote clients |

- Review Table 8 on page 22 to determine the instance type by 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 10.5 copy of DB2 Enterprise Server Edition. On Linux and UNIX operating systems, you can upgrade to a DB2 Version 10.5 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 10.5 copy that you installed, you must install a copy of the DB2 Version 10.5 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 database server with root user 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 10.5 copy location. The **db2iupgrade** command only needs to be run on the instance owning node. 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 fencedID ] InstName<sup>a</sup></code> |
| Windows          | <code>"%DB2PATH%" \bin\db2iupgrade InstName /u:user,password<sup>b</sup></code> |

**Note:**

- a. Where *DB2DIR* is set to the location you specified during DB2 Version 10.5 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 10.5 installation, *user* and *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 database 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 calls the **db2ckupgrade** command to verify that the local databases are ready for grade. The `update.log` is specified as the log file for **db2ckupgrade**, and the default log file created for **db2iupgrade** is `/tmp/db2ckupgrade.log.processID`. 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 **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 database 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 10.5 by running the **db2level** command:  
 db2level

The Informational tokens should include a string like "DB2 Version 10.5.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.

Otherwise, you can drop your existing DAS and create a new DAS in DB2 Version 10.5. See “Creating a DB2 administration server (DAS) ” in Installing DB2 Servers.

Start using IBM Data Studio and IBM Optim<sup>™</sup> tools. For a mapping between these recommended tools and Control Center tools, see “Table of recommended tools versus Control Center tools” in the *What’s New for DB2 Version 10.5* book.

**Important:** The DB2 Administration Server (DAS) has been deprecated in Version 9.7 and might be removed in a future release. The DAS is not supported in DB2 pureScale environments. Use software programs that use the Secure Shell protocol for remote administration. For more information, see “ DB2 administration server (DAS) has been deprecated” at .

### Before you begin

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

#### Restrictions

- 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 10.5 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 in DB2 Version 10.5, perform the following steps:
  - Upgrade the instance that owns the tools catalog database. For details, see “Upgrading DB2 Version 10.1 or DB2 Version 9.7 instances” on page 50.
  - Upgrade the tools catalog database. For details, see “Upgrading databases” on page 54
  - 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 must 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 DB2 Version 10.5 instance to use the task scheduling capability. See “CREATE TOOLS CATALOG command ” in *Command Reference*.

## Results

You can now use the DAS to administer DB2 Version 10.5 instances, as well as pre-DB2 Version 10.5 instances.

---

## Upgrading databases

After you upgraded your instances to DB2 Version 10.5, you must upgrade each database under each instance.

### Before you begin

- 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 Chapter 5, “Pre-upgrade tasks for DB2 servers,” on page 35.
- Ensure that you installed DB2 Version 10.5 and upgraded the instance to DB2 Version 10.5.

### Restrictions

- Review the steps in “Upgrade restrictions for DB2 servers” on page 21 for database upgrade.

### Procedure

To upgrade a DB2 database to DB2 Version 10.5:

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. Recatalog the database using the **CATALOG DATABASE** command:  

```
db2 CATALOG DB database_name as database_alias
```
4. Optional: Issue the **db2 LIST DATABASE DIRECTORY** command to ensure that the database is in the list of all catalogued databases in the current instance.



5. 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.

Also, consider using the **REBINDALL** parameter, which specifies that a **REBIND** of all packages is performed during upgrade

6. 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. For details, see “Increasing table space and log file sizes before upgrade” on page 41. 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 that are not supported by database upgrade. These cases are described in “Upgrade restrictions for DB2 servers” on page 21.

7. If the **UPGRADE DATABASE** command returns the SQL1243W warning message, you must 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
```

8. If the **UPGRADE DATABASE** command returns the SQL1499W warning message and writes the ADM7535W warning message with all the details to the administration notification log, then the command failed to refresh the table space attributes in the catalog table. However, the database was upgraded successfully. However, the database was upgraded successfully.
9. If the **UPGRADE DATABASE** command returns the SQL1499W warning message and writes the ADM4003E warning message with all the details to the administration notification log, then the command failed to upgrade the DB2 Text Search catalogs or indexes due to an error in a stored procedure.
10. If the **UPGRADE DATABASE** command returns the SQL1499W warning message and writes the ADM7534W warning message with all the details to the administration notification log, then the command failed to refresh the table space attributes in the catalog table. However, the database was upgraded successfully. However, the database was upgraded successfully.
11. 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.

12. 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 141 for details.

13. 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.

14. If the **UPGRADE DATABASE** command returns the SQL0473N error message, you must 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. See Chapter 12, "Reversing DB2 server upgrade," on page 113.

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 36.

15. If the **UPGRADE DATABASE** command returns the DBT5512 error message, the command failed to upgrade the database because the ID of a workload management object conflicts with a system-reserved ID. To upgrade the database, perform the following actions:
  - a. Generate the DDL statements to re-create the workload management objects by issuing the **db2look** command with the **-wlm** parameter.
  - b. Drop all of the workload management objects from the database.
  - c. Resolve all issues that are reported by the **db2ckupgrade** command and block the database from being upgraded.
  - d. Upgrade the database.
  - e. Re-create the workload management object in the upgraded database by issuing the DDL statements that you generated with the **db2look** command.

16. If the **UPGRADE DATABASE** command returns the SQL1700N error message, you must reverse the database migration and re-create database objects that use restricted schema names with a schema name that is not restricted. See Chapter 12, "Reversing DB2 server upgrade," on page 113.

To avoid the **UPGRADE DATABASE** command failure, re-create these database objects during "Verifying that your databases are ready for upgrade" on page 36.

17. If the **UPGRADE DATABASE** command returns the ADM4003E error message, then upgrade the DB2 Text Search catalog and indexes manually. For details, see **SYSTS\_UPGRADE\_CATALOG** and **SYSTS\_UPGRADE\_INDEX**.



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

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

19. Verify that 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 10.1.0
SQL authorization ID = TESTDB2
Local database alias = SAMPLE

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

Alternatively, if you have sample files that are 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 that is specified during DB2 Version 10.5 installation, and *sample* is the database name.

## What to do next

After upgrading a DB2 database, perform the recommended post-upgrade tasks to ensure a successful database upgrade. See Chapter 9, “Post-upgrade tasks for DB2 servers,” on page 97.

---

## Upgrading a server to DB2 Version 10.5 pureScale

Upgrading a DB2 server to DB2 Version 10.5 pureScale requires that you first upgrade to DB2 Version 10.5 and then convert to DB2 Version 10.5 pureScale.

### Before you begin

- 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. Refer to “Best practices for upgrading DB2 servers” on page 30 and “Disk space requirements for DB2 server upgrades” on page 27.
- Perform pre-upgrade tasks. Refer to Chapter 5, “Pre-upgrade tasks for DB2 servers,” on page 35.

### Restrictions

- Only automatic storage table spaces are supported in a pureScale environment and all the table spaces must be on a GPFS™ storage path.

- Additional upgrade restrictions apply. Refer to “Upgrade restrictions for DB2 servers” on page 21. Review the complete list.

## About this task

The following tasks describe the steps for direct upgrade from DB2 Version 9.7 or Version 10.1 to DB2 Version 10.5 pureScale.

## Scenario 1

### About this task

In this scenario, the catalog table space is Database Managed Space (DMS) and the user table spaces are mostly System Managed Space (SMS). The database is not enabled for automatic storage.

### Procedure

To upgrade a server to DB2 Version 10.5 pureScale environment:

1. Log on to the DB2 server as a user with Local Administrator authority.
2. Install DB2 Version 10.5 using the DB2 Setup wizard. Do not create an instance.
3. Run the **db2ckupgrade** command.
4. Run the **db2checkSD** command.
5. Upgrade your DB2 Version 9.7 or Version 10.1 instances. Refer to “Upgrading DB2 Version 10.1 or DB2 Version 9.7 instances” on page 50.
6. Upgrade your databases. Refer to “Upgrading databases” on page 54.
7. Run the **db2cluster -prepare** command to set up GPFS storage path. Refer to . Setting up a GPFS file system for a DB2 pureScale environment.
8. Create a new automatic storage enabled DB2 Version 10.5 database on the GPFS storage path.
9. Load the data into the new database using the **db2move** with the **copy** option.
10. Run the **db2checkSD** command.
11. Convert the instance to a DB2 Version 10.5 pureScale environment. Refer to . Converting your existing DB2 instances to a DB2 Version 10.5 pureScale environment.

## Scenario 2

### About this task

In this scenario, the catalog table space is System Managed Space (SMS) and the user table spaces are Database Managed Space (DMS). The database is not managed by automatic storage.

### Procedure

To upgrade a server to DB2 Version 10.5 pureScale environment:

1. Log on to the DB2 server as a user with Local Administrator authority.
2. Install DB2 Version 10.5 using the DB2 Setup wizard. Create an ESE instance.
3. Run the **db2cluster -prepare** command to set up GPFS storage path. Refer to . Setting up a GPFS file system for a DB2 pureScale environment.

4. Create a new automatic storage enabled DB2 Version 10.5 database on the GPFS storage path.
5. Convert the table spaces to use automatic storage. Use a redirected restore operation with the **TRANSPORT** and **SET TABLESPACE CONTAINERS** command and specify the **USING AUTOMATIC STORAGE** parameter, to move all the schemas to the new database. Refer to . Converting table spaces to use automatic storage.
6. Run the **db2checkSD** command.
7. Convert the instance to a DB2 Version 10.5 pureScale environment. Refer to . Converting your existing DB2 instances to a DB2 Version 10.5 pureScale environment.

## Scenario 3

### About this task

In this scenario, the catalog table space is Database Managed Space (DMS) and the user table spaces are mostly Database Managed Space (DMS).

### Procedure

To upgrade a server to DB2 Version 10.5 pureScale environment:

1. Log on to the DB2 server as a user with Local Administrator authority.
2. Run the **ALTER DATABASE** command with the **ADD STORAGE ON** storage-path option to enable your database for automatic storage, if required.
3. Install DB2 Version 10.5 using the DB2 Setup wizard. Create a DB2 Enterprise Server Edition instance.
4. Run the **db2cluster -prepare** command to set up GPFS storage path. Refer to . Setting up a GPFS file system for a DB2 pureScale environment.
5. Run the **db2ckupgrade** command.
6. Take a full offline backup of your database.
7. Do a redirected restore of the backup database into the DB2 Version 10.5 instance. Convert the table spaces to automatic storage and move the table spaces to GPFS storage path, if required. Refer to . Converting table spaces to use automatic storage.
8. Run the **db2checkSD** command.
9. Convert the instance to a DB2 Version 10.5 pureScale environment. Refer to . Converting your existing DB2 instances to a DB2 Version 10.5 pureScale environment.



---

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

Upgrading a DB2 server to DB2 Version 10.5 on Linux and UNIX requires that you install a new DB2 Version 10.5 copy and then manually upgrade your existing instances and databases to this new copy.

### Before you begin

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. Refer to “Best practices for upgrading DB2 servers” on page 30 and “Disk space requirements for DB2 server upgrades” on page 27.
- Perform pre-upgrade tasks. Refer to Chapter 5, “Pre-upgrade tasks for DB2 servers,” on page 35.

### About this task

This upgrade task describes the steps for direct upgrade to DB2 Version 10.5 from DB2 Version 9.7 or DB2 Version 10.1 regardless of the instance bit size. Review Chapter 8, “Upgrading DB2 servers with specific characteristics,” on page 73 and determine which task applies better to your environment.

#### 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 10.5 64-bit instances. The operating system and DB2 Version 10.5 database product that you installed determines the instance bit size, see “Support changes for 32-bit and 64-bit DB2 servers” on page 29 for details.
- Additional upgrade restrictions apply. Refer to “Upgrade restrictions for DB2 servers” on page 21. Review the complete list.

### Procedure

To upgrade a DB2 server to DB2 Version 10.5:

1. Log on to the DB2 server as root.
2. Install DB2 Version 10.5. 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 10.5.
3. Install all DB2 add-on products that were installed in the DB2 copy from which you are upgrading.
4. Upgrade DB2 Version 9.7 or DB2 Version 10.1 instances from the same installation path that you indicated during the DB2 Version 10.5 installation. Refer to “Upgrading DB2 Version 10.1 or DB2 Version 9.7 instances” on page 50.

5. Optional: Upgrade your DAS if you want to keep your existing DAS configuration and use new functionality available in DB2 Version 10.5. Refer to “Upgrading the DB2 Administration Server (DAS)” on page 53.
6. Upgrade databases. Refer to “Upgrading databases” on page 54.

## What to do next

After upgrading the DB2 server, perform the recommended Chapter 9, “Post-upgrade tasks for DB2 servers,” on page 97 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 Version 10.1 or DB2 Version 9.7 instances

As part of the overall process of upgrading your DB2 database server to DB2 Version 10.5, you must upgrade your instances.

### Before you begin

- You must have root user authority on Linux and UNIX operating systems or Local Administrator authority on Windows.
- You must install any DB2 database 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. This step is important in partitioned database environments because the **db2ckupgrade** command might return an error in one database partition and cause the instance upgrade to fail. Refer to “Verifying that your databases are ready for upgrade” on page 36.
  - On Linux and UNIX operating systems, ensure that there is 5GB of free space in the /tmp directory. The instance upgrade trace file is written to /tmp.
  - Gather pre-upgrade diagnostic information to help diagnose any problem that might occur after the upgrade.

### About this task

On Linux and UNIX operating systems, you must manually upgrade your instances. On Windows operating systems, you must manually upgrade them if you did not choose to automatically upgrade your existing DB2 copy during the DB2 Version 10.5 installation.

If you are upgrading from DB2 Version 9.8, follow the steps in “Upgrading a DB2 pureScale server” on page 83.

#### 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.
- For additional restrictions on instance upgrade, review “Upgrade restrictions for DB2 servers” on page 21.
- You must be upgrading from DB2 Version 10.1 or DB2 Version 9.7.

## Procedure

To manually upgrade your existing instances to DB2 Version 10.5 using the **db2iupgrade** command:

1. Determine if you can upgrade your existing instances to a DB2 Version 10.5 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 | Examples                                                                                                    |
|------------------|-------------------------------------------------------------------------------------------------------------|
| Linux and UNIX   | db2 GET DBM CFG   grep 'Node type'<br>Node type = Partitioned database server with local and remote clients |
| Windows          | db2 GET DBM CFG   find "Node type"<br>Node type = Partitioned database server with local and remote clients |

- Review Table 8 on page 22 to determine the instance type by 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 10.5 copy of DB2 Enterprise Server Edition. On Linux and UNIX operating systems, you can upgrade to a DB2 Version 10.5 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 10.5 copy that you installed, you must install a copy of the DB2 Version 10.5 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 database server with root user 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 10.5 copy location. The **db2iupgrade** command only needs to be run on the instance owning node. 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 fencedID ] InstName<sup>a</sup></code> |
| Windows          | <code>"%DB2PATH%" \bin\db2iupgrade InstName /u:user,password<sup>b</sup></code> |

### Note:

- a. Where *DB2DIR* is set to the location you specified during DB2 Version 10.5 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 10.5 installation, *user* and *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 database 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 calls the **db2ckupgrade** command to verify that the local databases are ready for grade. The `update.log` is specified as the log file for **db2ckupgrade**, and the default log file created for **db2iupgrade** is `/tmp/db2ckupgrade.log.processID`. 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 **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 database 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 10.5 by running the **db2level** command:  
`db2level`

The Informational tokens should include a string like "DB2 Version 10.5.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.

Otherwise, you can drop your existing DAS and create a new DAS in DB2 Version 10.5. See "Creating a DB2 administration server (DAS) " in *Installing DB2 Servers*.

Start using IBM Data Studio and IBM Optim tools. For a mapping between these recommended tools and Control Center tools, see "Table of recommended tools versus Control Center tools" in the *What's New for DB2 Version 10.5* book.

**Important:** The DB2 Administration Server (DAS) has been deprecated in Version 9.7 and might be removed in a future release. The DAS is not supported in DB2 pureScale environments. Use software programs that use the Secure Shell protocol for remote administration. For more information, see " DB2 administration server (DAS) has been deprecated" at .

### Before you begin

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

### Restrictions

- 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 10.5 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 in DB2 Version 10.5, perform the following steps:
  - Upgrade the instance that owns the tools catalog database. For details, see “Upgrading DB2 Version 10.1 or DB2 Version 9.7 instances” on page 50.
  - Upgrade the tools catalog database. For details, see “Upgrading databases” on page 54
  - 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 must 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 DB2 Version 10.5 instance to use the task scheduling capability. See “CREATE TOOLS CATALOG command ” in *Command Reference*.

## Results

You can now use the DAS to administer DB2 Version 10.5 instances, as well as pre-DB2 Version 10.5 instances.

---

## Upgrading databases

After you upgraded your instances to DB2 Version 10.5, you must upgrade each database under each instance.

### Before you begin

- 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 Chapter 5, “Pre-upgrade tasks for DB2 servers,” on page 35.
- Ensure that you installed DB2 Version 10.5 and upgraded the instance to DB2 Version 10.5.

#### Restrictions

- Review the steps in “Upgrade restrictions for DB2 servers” on page 21 for database upgrade.

## Procedure

To upgrade a DB2 database to DB2 Version 10.5:

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. Recatalog the database using the **CATALOG DATABASE** command:  

```
db2 CATALOG DB database_name AS database_alias
```
4. Optional: Issue the **db2 LIST DATABASE DIRECTORY** command to ensure that the database is in the list of all catalogued databases in the current instance.
5. 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.

Also, consider using the **REBINDALL** parameter, which specifies that a **REBIND** of all packages is performed during upgrade

6. 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. For details, see “Increasing table space and log file sizes before upgrade” on page 41. After the database upgrade is complete reset the value of **logfilsiz**, **logprimary** and **logsecond** database configuration parameters.

There are additional error codes that are returned by the **UPGRADE DATABASE** command for specific cases that are not supported by database upgrade. These cases are described in “Upgrade restrictions for DB2 servers” on page 21.

7. If the **UPGRADE DATABASE** command returns the SQL1243W warning message, you must 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
```

8. If the **UPGRADE DATABASE** command returns the SQL1499W warning message and writes the ADM7535W warning message with all the details to the administration notification log, then the command failed to refresh the table space attributes in the catalog table. However, the database was upgraded successfully. However, the database was upgraded successfully.
9. If the **UPGRADE DATABASE** command returns the SQL1499W warning message and writes the ADM4003E warning message with all the details to the administration notification log, then the command failed to upgrade the DB2 Text Search catalogs or indexes due to an error in a stored procedure.
10. If the **UPGRADE DATABASE** command returns the SQL1499W warning message and writes the ADM7534W warning message with all the details to the administration notification log, then the command failed to refresh the table space attributes in the catalog table. However, the database was upgraded successfully. However, the database was upgraded successfully.
11. 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.
12. 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 141 for details.
13. 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.
14. If the **UPGRADE DATABASE** command returns the SQL0473N error message, you must 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. See Chapter 12, "Reversing DB2 server upgrade," on page 113.  
  
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 36.
15. If the **UPGRADE DATABASE** command returns the DBT5512 error message, the command failed to upgrade the database because the ID of a workload

management object conflicts with a system-reserved ID. To upgrade the database, perform the following actions:

- a. Generate the DDL statements to re-create the workload management objects by issuing the **db2look** command with the **-wlm** parameter.
  - b. Drop all of the workload management objects from the database.
  - c. Resolve all issues that are reported by the **db2ckupgrade** command and block the database from being upgraded.
  - d. Upgrade the database.
  - e. Re-create the workload management object in the upgraded database by issuing the DDL statements that you generated with the **db2look** command.
16. If the **UPGRADE DATABASE** command returns the SQL1700N error message, you must reverse the database migration and re-create database objects that use restricted schema names with a schema name that is not restricted. See Chapter 12, “Reversing DB2 server upgrade,” on page 113.
- To avoid the **UPGRADE DATABASE** command failure, re-create these database objects during “Verifying that your databases are ready for upgrade” on page 36.
17. If the **UPGRADE DATABASE** command returns the ADM4003E error message, then upgrade the DB2 Text Search catalog and indexes manually. For details, see **SYSTS\_UPGRADE\_CATALOG** and **SYSTS\_UPGRADE\_INDEX**.
18. Compare your database configuration settings after upgrade with the configuration settings you had before you upgraded your database. Verify that the following settings and database information are the same:
- Database configuration parameter settings
  - Table spaces information
  - Packages information for your applications only

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

19. Verify that 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 10.1.0
SQL authorization ID = TESTDB2
Local database alias = SAMPLE
```

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

Alternatively, if you have sample files that are 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 that is specified during DB2 Version 10.5 installation, and *sample* is the database name.

## What to do next

After upgrading a DB2 database, perform the recommended post-upgrade tasks to ensure a successful database upgrade. See Chapter 9, “Post-upgrade tasks for DB2 servers,” on page 97.

---

## Upgrading a server to DB2 Version 10.5 pureScale

Upgrading a DB2 server to DB2 Version 10.5 pureScale on Linux and UNIX requires that you first upgrade to DB2 Version 10.5 and then convert to DB2 Version 10.5 pureScale.

### Before you begin

- 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. Refer to “Best practices for upgrading DB2 servers” on page 30 and “Disk space requirements for DB2 server upgrades” on page 27.
- Perform pre-upgrade tasks. Refer to Chapter 5, “Pre-upgrade tasks for DB2 servers,” on page 35.

### Restrictions

- Only automatic storage table spaces are supported in a pureScale environment and all the table spaces must be on a GPFS storage path.
- Additional upgrade restrictions apply. Refer to “Upgrade restrictions for DB2 servers” on page 21. Review the complete list.

### About this task

The following tasks describe the steps for direct upgrade from DB2 Version 9.7 or Version 10.1 to DB2 Version 10.5 pureScale.

## Scenario 1

### About this task

In this scenario, the catalog table space is Database Managed Space (DMS) and the user table spaces are mostly System Managed Space (SMS). The database is not enabled for automatic storage.

### Procedure

To upgrade a server to DB2 Version 10.5 pureScale environment:

1. Log on to the DB2 server as root.
2. Install DB2 Version 10.5. 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 10.5. Do not create an instance.
3. Run the **db2ckupgrade** command.
4. Run the **db2checkSD** command.
5. Upgrade your DB2 Version 9.7 or Version 10.1 instances. Refer to “Upgrading DB2 Version 10.1 or DB2 Version 9.7 instances” on page 50.

6. Upgrade your databases. Refer to “Upgrading databases” on page 54.
7. Run the **db2cluster -prepare** command to set up GPFS storage path. Refer to . Setting up a GPFS file system for a DB2 pureScale environment.
8. Create a new automatic storage enabled DB2 Version 10.5 database on the GPFS storage path.
9. Load the data into the new database using the **db2move** with the **copy** option.
10. Run the **db2checkSD** command.
11. Convert the instance to a DB2 Version 10.5 pureScale environment. Refer to . Converting your existing DB2 instances to a DB2 Version 10.5 pureScale environment.

## Scenario 2

### About this task

In this scenario, the catalog table space is System Managed Space (SMS) and the user table spaces are Database Managed Space (DMS). The database is not managed by automatic storage.

### Procedure

To upgrade a server to DB2 Version 10.5 pureScale environment:

1. Log on to the DB2 server as a user with Local Administrator authority.
2. Install DB2 Version 10.5. 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 10.5. Create an ESE instance.
3. Run the **db2cluster -prepare** command to set up GPFS storage path. Refer to . Setting up a GPFS file system for a DB2 pureScale environment.
4. Create a new automatic storage enabled DB2 Version 10.5 database on the GPFS storage path.
5. Convert the table spaces to use automatic storage. Use a redirected restore operation with the **TRANSPORT** and **SET TABLESPACE CONTAINERS** command and specify the **USING AUTOMATIC STORAGE** parameter, to move all the schemas to the new database. Refer to . Converting table spaces to use automatic storage.
6. Run the **db2checkSD** command.
7. Convert the instance to a DB2 Version 10.5 pureScale environment. Refer to . Converting your existing DB2 instances to a DB2 Version 10.5 pureScale environment.

## Scenario 3

### About this task

In this scenario, the catalog table space is Database Managed Space (DMS) and the user table spaces are mostly Database Managed Space (DMS).

### Procedure

To upgrade a server to DB2 Version 10.5 pureScale environment:

1. Log on to the DB2 server as a user with Local Administrator authority.
2. Run the **ALTER DATABASE** command with the **ADD STORAGE ON** storage-path option to enable your database for automatic storage, if required.

3. Install DB2 Version 10.5. 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 10.5. Create a DB2 Enterprise Server Edition instance.
4. Run the **db2cluster -prepare** command to set up GPFS storage path. Refer to . Setting up a GPFS file system for a DB2 pureScale environment.
5. Run the **db2ckupgrade** command.
6. Take a full offline backup of your database.
7. Do a redirected restore of the backup database into the DB2 Version 10.5 instance. Convert the table spaces to automatic storage and move the table spaces to GPFS storage path, if required. Refer to . Converting table spaces to use automatic storage.
8. Run the **db2checkSD** command.
9. Convert the instance to a DB2 Version 10.5 pureScale environment. Refer to . Converting your existing DB2 instances to a DB2 Version 10.5 pureScale environment.





---

## 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 those tasks:

- “Upgrading DB2 32-bit servers to 64-bit systems (Windows)”
- “Upgrading non-root installations” on page 74
- “Upgrading a DB2 server with multiple DB2 copies” on page 76
- “Upgrading to a new DB2 server” on page 78
- “Upgrading a DB2 server by using online backups from a previous release” on page 81
- “Upgrading partitioned database environments” on page 82
- “Upgrading a DB2 pureScale server” on page 83
- Upgrading DB2 Text Search for administrator or root install
- Upgrading DB2 Text Search for non-root install (Linux and UNIX)
- Upgrading a multi-partition instance without DB2 Text Search
- “Upgrading DB2 servers in Microsoft Cluster Server environments” on page 95
- Upgrading DB2 Spatial Extender Version 10.5

---

### 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 10.5 64-bit server. One way is to upgrade your existing DB2 32-bit server to DB2 Version 10.5 32-bit server, and then upgrade to DB2 Version 10.5 64-bit server.

The other way is to upgrade to a new computer where DB2 Version 10.5 64-bit database product is installed.

#### Before you begin

- Ensure that you have Local Administrator authority.
- Ensure that the DB2 server is running 64-bit windows operating system.
- Review “Best practices for upgrading DB2 servers” on page 30 and “Disk space requirements for DB2 server upgrades” on page 27.
- Perform pre-upgrade tasks. See Chapter 5, “Pre-upgrade tasks for DB2 servers,” on page 35.

#### Restrictions

- This procedure is covered by this task and only applies to Windows on x64.

- Additional upgrade restrictions apply. See “Upgrade restrictions for DB2 servers” on page 21. Review the complete list.

## Procedure

To upgrade a pre-DB2 Version 10.5 32-bit server to a DB2 Version 10.5 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 Version 10.1, or DB2 Version 9.7 or 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 10.1 or DB2 Version 9.7 32-bit server copy. You can only update instances of the same version.
  - If you have instances running on multiple pre-DB2 Version 10.5 copies of different version, upgrade all instances to the highest release of pre-DB2 Version 10.5 copies. For example, if you have a Version 10.1 and a Version 9.7 instance, upgrade your Version 9.7 instance to the DB2 Version 10.1 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 Version 10.1 32-bit server copy, or DB2 Version 9.7 32-bit server copy
3. Install DB2 Version 10.5 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.
4. If you want your applications to access DB2 Version 10.5 copy through the default interface, set the DB2 Version 10.5 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*.
5. Upgrade your databases.
6. If you want to have your instances running on multiples copies of DB2 Version 10.5, install additional DB2 Version 10.5 copies and issue the **db2iupdt** command to run an instance under a different DB2 Version 10.5 copy.

## 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. See Chapter 9, “Post-upgrade tasks for DB2 servers,” on page 97 and “Verifying upgrade of DB2 servers” on page 104.

---

## Upgrading non-root installations

Upgrading DB2 Version 10.1, or DB2 Version 9.7 non-root installations to DB2 Version 10.5 on Linux and UNIX requires that you install DB2 Version 10.5 as a non-root user and then upgrade your databases to the DB2 Version 10.5 non-root installation.

### Before you begin

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. See “Best practices for upgrading DB2 servers” on page 30 and “Disk space requirements for DB2 server upgrades” on page 27.
- Perform pre-upgrade tasks that apply, especially verifying that your databases are ready for upgrade. 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 database product is not installed. See Chapter 5, “Pre-upgrade tasks for DB2 servers,” on page 35 and “Verifying that your databases are ready for upgrade” on page 36.

#### Restrictions

- You cannot upgrade a DB2 Version 9.7 root installation to a DB2 Version 10.5 non-root installation. You can upgrade databases from a DB2 Version 9.7 root installation to a DB2 Version 10.5 non-root installation by restoring database backups taken in the DB2 Version 9.7 root installation. Use the same process described in “Upgrading to a new DB2 server” on page 78.
- On Linux and UNIX operating systems except for Linux on x86, your existing 32-bit or 64-bit instances are upgraded to DB2 Version 10.5 64-bit instances. The operating system and DB2 Version 10.5 database product that you installed determines the instance bit size, see “Support changes for 32-bit and 64-bit DB2 servers” on page 29 for details.
- Additional upgrade restrictions apply. Review the complete list in “Upgrade restrictions for DB2 servers” on page 21.

### Procedure

To upgrade a non-root installation to DB2 Version 10.5:

1. Log on to the DB2 server as the non-root user for the DB2 Version 10.1 or DB2 Version 9.7 non-root installation.
2. Review Table 8 on page 22 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 10.5 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 10.1 or DB2 Version 9.7 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 DB2 Version 10.1 or DB2 Version 9.7 copy is removed.

The DB2 product installation also verifies the following conditions:

- The directory `INSTHOME/sql1ib_v101` 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.

**Important:** The command **db2\_install** is deprecated and might be removed in a future release. Use the **db2setup** command with a response file instead.

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 **db2nrupgrade** command to upgrade the non-root instance as follows:

```
cd $HOME/sqllib/instance
db2nrupgrade -b BackupDir
```

Where *BackupDir* is the backup directory for the configuration files of the non-root installation before upgrade. The backup directory is in the **db2setup** log in the format of *sqllib\_vVR* where *V* is the version number and *R* is the release number of the old copy. For example, if you have Version 9.7 installed and then install Version 10.5 using the **db2setup** command, you can find the name of the backup directory as *sqllib\_v101* in the **db2setup** log file.

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. See “Upgrading databases” on page 54.
8. Enable root-based features by running the **db2rfe** command.
9. If you had additional DB2 products installed in your DB2 Version 10.1 or DB2 Version 9.7 non-root copy, install one DB2 product at a time.

## What to do next

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. See Chapter 9, “Post-upgrade tasks for DB2 servers,” on page 97 and “Verifying upgrade of DB2 servers” on page 104.

---

## Upgrading a DB2 server with multiple DB2 copies

Upgrading a DB2 server with multiple pre-DB2 Version 10.5 DB2 copies, requires that you install DB2 Version 10.5 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 10.1 and Version 9.7 installed.

You can manually upgrade a pre-DB2 Version 10.5 instance at any fix pack level by executing the **db2iupgrade** command from the target DB2 Version 10.5 copy of your choice. After an instance is upgraded to a DB2 Version 10.5 copy, you cannot upgrade it to another DB2 Version 10.5 copy. However, you can update an instance between different DB2 Version 10.5 copies using the **db2iupdt** command.

## Before you begin

- 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. See “Best practices for upgrading DB2 servers” on page 30 and “Disk space requirements for DB2 server upgrades” on page 27.
- Perform pre-upgrade tasks. See Chapter 5, “Pre-upgrade tasks for DB2 servers,” on page 35.

### 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 73 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. See “Upgrade restrictions for DB2 servers” on page 21.

## 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 10.5 as a new copy of DB2 Version 10.5 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 10.5 copies, if you want to upgrade your existing instances to different DB2 Version 10.5 copies.
3. Upgrade instances using the **db2iupgrade** command from the installation path of the DB2 Version 10.5 copy of your choice. See “Upgrading DB2 Version 10.1 or DB2 Version 9.7 instances” on page 50. For example, assume that you have the following DB2 copies and instances on an AIX server and a Windows server:

Table 13. Directory examples for DB2 copies.

| Instance name        | OS      | DB2 copy directory                      |
|----------------------|---------|-----------------------------------------|
| db2inst1             | AIX     | /opt/IBM/db2/V9.7                       |
| db2inst2             | AIX     | /opt/IBM/db2/V10.1                      |
| db2inst3             | AIX     | /home/db2/myV10.1                       |
| No instances created | AIX     | /opt/IBM/db2/V10.5<br>/home/db2/myV10.5 |
| DB2_97               | Windows | C:\Program Files\IBM\SQLLIB_97\         |
| No instances created | Windows | C:\Program Files\IBM\SQLLIB_10.5\       |

You can then run the following commands to successfully upgrade your instances to DB2 Version 10.5:

Table 14. Instance upgrade command examples.

| Upgrade Instance | Commands                                                                          |
|------------------|-----------------------------------------------------------------------------------|
| db2inst1         | cd /opt/IBM/db2/V10.5/instance<br>./db2iupgrade -u db2fenc1 db2inst1              |
| db2inst2         | cd /opt/IBM/db2/V10.5/instance<br>./db2iupgrade db2inst2                          |
| db2inst3         | cd /home/db2/myV10.5/instance<br>./db2iupgrade db2inst3                           |
| DB2_97           | cd C:\Program Files\IBM\SQLLIB_10.5\<br>db2iupgrade DB2_97 /u:db2admin1,password1 |

- Optional: Upgrade the DB2 Administration Server if you want to keep your existing configuration to administer your DB2 Version 10.5 instances. See “Upgrading the DB2 Administration Server (DAS)” on page 53.
- Log on to the DB2 server as a user with SYSADM authority.
- Upgrade databases. See “Upgrading databases” on page 54.

## 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. See Chapter 9, “Post-upgrade tasks for DB2 servers,” on page 97 and “Verifying upgrade of DB2 servers” on page 104.

## Upgrading to a new DB2 server

If you want to upgrade to a new DB2 Version 10.5 server, re-create your instances and then upgrade your databases by restoring a pre-DB2 Version 10.5 database backup. After restoring the database backup, the **RESTORE DATABASE** command automatically runs the **UPGRADE DATABASE** command.

### Before you begin

- 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. See “Best practices for upgrading DB2 servers” on page 30 and “Disk space requirements for DB2 server upgrades” on page 27.
- Perform pre-upgrade tasks. See Chapter 5, “Pre-upgrade tasks for DB2 servers,” on page 35.

### Restrictions

- Review the upgrade restrictions for DB2 servers. See “Upgrade restrictions for DB2 servers” on page 21.

### About this task

Use this procedure to upgrade your databases to a new server that has the same operating system as the old server. You can also use this procedure to upgrade



your databases when the backup and restore operations are supported between the operating systems. For more information about this support, see Backup and restore operations between different operating systems and hardware platforms.

## Procedure

To upgrade to a new DB2 Version 10.5 server:

1. Perform a full offline database backup of your existing databases and any other pre-upgrade tasks that apply. See “Backing up databases before or after upgrade” on page 38. 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 10.5 on the new DB2 server.
4. Create your instances on the new DB2 server by running the **db2icrt** command from the DB2 Version 10.5 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 10.5. You need a DAS if you want to keep your existing DAS configuration and use new functionality available in DB2 Version 10.5.
6. Transfer pre-DB2 Version 10.5 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. If sqlcode 7535 is returned as follows:

```
SQL2517W The database was restored and then upgraded to the current release.
The database upgrade returned sqlcode "7535" and tokens "*N".
```

then you can run the **UPGRADE DATABASE** command again.

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. See "Increasing table space and log file sizes before upgrade" on page 41.
- b. Upgrade the database using the **UPGRADE DATABASE** command. See "Upgrading databases" on page 54.
- 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 10
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 10.5 installation, and `sample` is the database name.

## 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. See Chapter 9, “Post-upgrade tasks for DB2 servers,” on page 97 and “Verifying upgrade of DB2 servers” on page 104.

---

## Upgrading a DB2 server by using online backups from a previous release

You can rebuild your database on a previous release by using online database backups from the same release and then upgrade to DB2 Version 10.5.

### Before you begin

Before upgrading your DB2 server:

- Ensure that you have root access on Linux and UNIX operating systems or Local Administrator authority on Windows.
- All necessary full or incremental online pre-DB2 Version 10.5 database backups of your databases so that you can rebuild your databases by 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 Chapter 5, “Pre-upgrade tasks for DB2 servers,” on page 35.

### Procedure

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

1. Transfer pre-DB2 Version 10.5 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 10.1 copy, you must have a DB2 Version 10.1 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 by using the **RESTORE DATABASE** command with the **REBUILD WITH ALL TABLESPACES IN DATABASE** parameter followed by the **ROLLFORWARD DATABASE** command. For 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 by using one of the following tasks:
  - Chapter 6, “Upgrading a DB2 server (Windows),” on page 49
  - Chapter 7, “Upgrading a DB2 server (Linux and UNIX),” on page 61

---

## Upgrading partitioned database environments

Upgrading partitioned database environments requires that you install DB2 Version 10.5 as a new copy in all database partition servers, upgrade the instances and then upgrade the databases.

### Before you begin

- 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 “Best practices for upgrading DB2 servers” on page 30 and “Disk space requirements for DB2 server upgrades” on page 27.
- Perform pre-upgrade tasks. Refer to Chapter 5, “Pre-upgrade tasks for DB2 servers,” on page 35.

### 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 10.5. If you choose the **upgrade** action when you select the **Work with Existing** option on the **Install a Product** panel, the installation process fails.
- Additional upgrade restrictions apply. Refer to “Upgrade restrictions for DB2 servers” on page 21. Review the complete list.

### Procedure

To upgrade DB2 servers in a partitioned database environment:

1. Perform a full offline backup for all database partitions. 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. Refer to Chapter 5, “Pre-upgrade tasks for DB2 servers,” on page 35.
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 10.5 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. Refer to “Upgrading DB2 Version 10.1 or DB2 Version 9.7 instances” on page 50. 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. Refer to “Upgrading databases” on page 54. The catalog partition must be available when you issue the **UPGRADE DATABASE** regardless on what database partition you issue this command from.  
  
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.
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. Refer to “Upgrading the DB2 Administration Server (DAS)” on page 53.

## 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 Chapter 9, “Post-upgrade tasks for DB2 servers,” on page 97 and “Verifying upgrade of DB2 servers” on page 104.

---

## Upgrading a DB2 pureScale server

Upgrading a DB2 pureScale server to DB2 Version 10.5 on Linux and UNIX requires that you install a new DB2 Version 10.5 copy and then manually upgrade your existing instances and databases to this new copy.

### Before you begin

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. Refer to “Best practices for upgrading DB2 servers” on page 30 and “Disk space requirements for DB2 server upgrades” on page 27.

- Perform pre-upgrade tasks such as verifying that your databases are ready for upgrade and backing up database before upgrade. For more details, see Chapter 5, “Pre-upgrade tasks for DB2 servers,” on page 35.

## About this task

This upgrade task describes the steps for direct upgrade to DB2 Version 10.5 from DB2 Version 9.8.

### Restrictions

- Review the complete list of upgrade restrictions at “Upgrade restrictions for DB2 servers” on page 21.

## Procedure

To upgrade a DB2 server to DB2 Version 10.5:

1. Log on to the DB2 server as the instance owner.
2. Stop the database manager by issuing the **db2stop** command as follows:  
 db2stop force (Disconnects all users and stops the instance)  
 db2 terminate (Terminates back-end process)
3. Stop all instance processes in other members by issuing the **db2stop instance on <hostname>** command where *hostname* is the name of each member in the cluster.
4. Install DB2 Version 10.5 by performing the following steps:
  - a. Log on to the DB2 server with root user authority.
  - b. Put the cluster management software into maintenance mode on all members and cluster caching facilities (CFs) by issuing the **db2cluster -cm -enter -maintenance -all** command. This command stops the peer domain services on all hosts and prevents it from restarting during system maintenance.
  - c. Put the cluster file system into maintenance mode on all members and CFs by issuing the **db2cluster -cfs -enter -maintenance -all** command. This command stops all hosts from accessing the cluster file system (GPFS) during system maintenance.
  - d. Install DB2 Version 10.5 by using the **db2setup** command in all members and CFs. The DB2Setup wizard provides a clear flow through which you can launch a DB2 pureScale Feature installation from one member and successfully setup a DB2 pureScale environment across multiple members. The cluster management software and the cluster file system software are also upgraded during the installation to meet the V10.5 requirements.
  - e. Take the cluster management software out of maintenance mode by issuing the **db2cluster -cm -exit -maintenance -all** command.
  - f. Take the cluster file system software out of maintenance mode by issuing the **db2cluster -cfs -exit -maintenance -all** command.
  - g. Commit changes to the cluster file system by issuing the **db2cluster -cfs -commit** command.
  - h. Restart the DB2 instance processes on all members with updated resources for the cluster management software and the cluster file system software by issuing the **db2start instance on <hostname>** command.
5. Install all DB2 add-on products that were installed in the DB2 copy from which you are upgrading.

6. Upgrade DB2 Version 9.8 instances. Refer to “Upgrading DB2 Version 9.8 instances.”
7. Upgrade databases. Refer to “Upgrading databases” on page 54.

## What to do next

After upgrading the DB2 server, perform the recommended Chapter 9, “Post-upgrade tasks for DB2 servers,” on page 97 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 Version 9.8 instances

As part of the overall process of upgrading your DB2 database server to DB2 Version 10.5, you must upgrade your Version 9.8 instances.

### Before you begin

- Your DB2 Version 9.8 instance must be a DB2 pureScale instance.
- You must have root user authority on Linux and UNIX operating systems.
- You must install any DB2 database 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. This step is important in DB2 pureScale environments because the **db2ckupgrade** command might return an error in one member and cause the instance upgrade to fail. Refer to “Verifying that your databases are ready for upgrade” on page 36.
  - On Linux and UNIX operating systems, ensure that there is 5GB of free space in the /tmp directory. The instance upgrade trace file is written to /tmp.
  - Gather pre-upgrade diagnostic information to help diagnose any problem that might occur after the upgrade. For details, see “Gathering pre-upgrade diagnostic information” on page 45.

### About this task

On Linux and UNIX operating systems, you must manually upgrade your DB2 pureScale instances from Version 9.8.

#### 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.
- For additional restrictions on instance upgrade, review “Upgrade restrictions for DB2 servers” on page 21.

### Procedure

To manually upgrade your existing Version 9.8 instances to DB2 Version 10.5 using the **db2iupgrade** command:

1. Log on to the DB2 server with root user authority.
2. Upgrade your existing Version 9.8 instances by issuing the **db2iupgrade** command from the target DB2 Version 10.5 copy location. You should issue the

**db2iupgrade** command from the Version 10.5 installation path from all the members first and then from the CFs. The following example shows how to use this command:

```
$DB2DIR/instance/db2iupgrade [-u fencedID] InstName
```

Where *DB2DIR* is set to the location that you specified during DB2 Version 10.5 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.

If you did not install all DB2 database 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.

**Note:** You must stop the Version 9.8 instance using the **db2stop** command before issuing the **db2iupgrade** command. If you do not stop Version 9.8 instance before using the **db2iupgrade** command, your instance upgrade might fail.

3. Log on to the DB2 database server as a user with sufficient authority to start your instance.
4. Restart the DB2 instance on all members and CFs with updated resources for the cluster management software and the cluster file system software by issuing the **db2start instance on <hostname>** command, and then issue the **db2start** command. If you find inconsistencies between the cluster manager resource model and the db2nodes.cfg repair the cluster manager resources by using the **db2cluster -cm -repair -resources** command.
5. Verify that your instances are running on to DB2 Version 10.5 by running the **db2level** command: The Informational tokens should include a string like "DB2 Version 10.5.X.X" where X is a digit number.
6. Rebuild the contents of the network resiliency condition and response resources in the cluster by issuing the **db2cluster -cfs -repair -network\_resiliency -all** command.

## What to do next

After upgrading your Version 9.8 DB2 pureScale instance, you must upgrade your database. For more details, see "Upgrading databases" on page 54.

## Upgrading databases

After you upgraded your instances to DB2 Version 10.5, you must upgrade each database under each instance.

### Before you begin

- 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 Chapter 5, "Pre-upgrade tasks for DB2 servers," on page 35.
- Ensure that you installed DB2 Version 10.5 and upgraded the instance to DB2 Version 10.5.

Restrictions



- Review the steps in “Upgrade restrictions for DB2 servers” on page 21 for database upgrade.

## Procedure

To upgrade a DB2 database to DB2 Version 10.5:

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. Recatalog the database using the **CATALOG DATABASE** command:  
`db2 CATALOG DB database_name AS database_alias`
4. Optional: Issue the **db2 LIST DATABASE DIRECTORY** command to ensure that the database is in the list of all catalogued databases in the current instance.
5. 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.

Also, consider using the **REBINDALL** parameter, which specifies that a **REBIND** of all packages is performed during upgrade

6. 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. For details, see “Increasing table space and log file sizes before upgrade” on page 41. After the database upgrade is complete reset the value of **logfilsiz**, **logprimary** and **logsecond** database configuration parameters.

There are additional error codes that are returned by the **UPGRADE DATABASE** command for specific cases that are not supported by database upgrade. These cases are described in “Upgrade restrictions for DB2 servers” on page 21.

7. If the **UPGRADE DATABASE** command returns the SQL1243W warning message, you must 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
```

8. If the **UPGRADE DATABASE** command returns the SQL1499W warning message and writes the ADM7535W warning message with all the details to the administration notification log, then the command failed to refresh the table space attributes in the catalog table. However, the database was upgraded successfully. However, the database was upgraded successfully.
9. If the **UPGRADE DATABASE** command returns the SQL1499W warning message and writes the ADM4003E warning message with all the details to the administration notification log, then the command failed to upgrade the DB2 Text Search catalogs or indexes due to an error in a stored procedure.
10. If the **UPGRADE DATABASE** command returns the SQL1499W warning message and writes the ADM7534W warning message with all the details to the administration notification log, then the command failed to refresh the table space attributes in the catalog table. However, the database was upgraded successfully. However, the database was upgraded successfully.
11. 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.
12. 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 141 for details.
13. 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.
14. If the **UPGRADE DATABASE** command returns the SQL0473N error message, you must 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. See Chapter 12, “Reversing DB2 server upgrade,” on page 113.

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 36.
15. If the **UPGRADE DATABASE** command returns the DBT5512 error message, the command failed to upgrade the database because the ID of a workload management object conflicts with a system-reserved ID. To upgrade the database, perform the following actions:
  - a. Generate the DDL statements to re-create the workload management objects by issuing the **db2look** command with the **-wlm** parameter.
  - b. Drop all of the workload management objects from the database.



- c. Resolve all issues that are reported by the **db2ckupgrade** command and block the database from being upgraded.
  - d. Upgrade the database.
  - e. Re-create the workload management object in the upgraded database by issuing the DDL statements that you generated with the **db2look** command.
16. If the **UPGRADE DATABASE** command returns the SQL1700N error message, you must reverse the database migration and re-create database objects that use restricted schema names with a schema name that is not restricted. See Chapter 12, “Reversing DB2 server upgrade,” on page 113.
- To avoid the **UPGRADE DATABASE** command failure, re-create these database objects during “Verifying that your databases are ready for upgrade” on page 36.
17. If the **UPGRADE DATABASE** command returns the ADM4003E error message, then upgrade the DB2 Text Search catalog and indexes manually. For details, see **SYSTS\_UPGRADE\_CATALOG** and **SYSTS\_UPGRADE\_INDEX**.
18. Compare your database configuration settings after upgrade with the configuration settings you had before you upgraded your database. Verify that the following settings and database information are the same:
- Database configuration parameter settings
  - Table spaces information
  - Packages information for your applications only
- You need not check package information for system generated packages. The information about system generated packages can change after upgrade.
19. Verify that 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 10.1.0
SQL authorization ID = TESTDB2
Local database alias = SAMPLE
```

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

Alternatively, if you have sample files that are 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 that is specified during DB2 Version 10.5 installation, and *sample* is the database name.

## What to do next

After upgrading a DB2 database, perform the recommended post-upgrade tasks to ensure a successful database upgrade. See Chapter 9, “Post-upgrade tasks for DB2 servers,” on page 97.

---

## Upgrading DB2 Text Search environments

Upgrading DB2 Text Search environments requires that you upgrade the DB2 server, instance, and all databases in the instance. Follow the steps to upgrade root installations or non-root installations that apply to your environment.

### Upgrading DB2 Text Search for administrator or root installation

To obtain the latest functionality upgrade your DB2 Text Search instance. You must upgrade the DB2 server, instance, and all databases when you are upgrading the text search instance.

#### Before you begin

Before you begin to upgrade DB2 Text Search as administrator or root, complete the following steps:

1. Log in as the instance owner or a user with SYSADM authority.
2. Stop the DB2 database instance and the DB2 Text Search instance service.
3. Back up the DB2 Text Search configuration directory:
  - For Linux and UNIX operating systems, it is located under:  
`$INSTHOME/sqllib/db2tss/config`  
  
where *INSTHOME* represents the instance home path.
  - For Windows systems, it is located under:  
`<INSTPROF>\<INSTNAME>\db2tss\config`  
  
where *<INSTPROF>* represents the instance profile directory and *<INSTNAME>* indicates the name of the instance to be upgraded.
4. If you enabled DB2 Text Search for rich text document support, disable rich text document support. For more information about how to disable rich text document support, see the topic about disabling DB2 Text Search for rich text document support.

#### About this task

The following steps describe the process to upgrade DB2 Text Search Version 9.7 or Version 10.1 root installations on Linux or UNIX operating system, or for administrators on the Windows platform.

#### Procedure

1. Log on to the DB2 server as root on Linux and UNIX operating systems or user with Local Administrator authority on Windows operating systems. If you are upgrading a multipartitioned instance, you must perform instance upgrade from the instance-owning partition.
2. Install a new copy of V10.5 with a custom installation and make sure that DB2 Text Search is selected. DB2 Text Search is an optional component that is available only when you select a custom installation. You also can choose to install a new V10.5 copy over an earlier DB2 version by selecting Work-With-Existing mode and selecting DB2 Text Search as the component to be upgraded. You do not have to upgrade the DB2 instances after the installation with this approach.

3. Upgrade the DB2 Text Search server for your DB2 instances by issuing the **configTool upgradeInstance** command.

- For Linux and UNIX operating systems:

```
$DB2DIR/db2tss/bin/configTool upgradeConfigFolder
-sourceConfigFolder $DB2DIR/cfg/db2tss/config
-targetConfigFolder $INSTHOME/sql11ib/db2tss/config
```

where, *INSTHOME* is the instance home directory and *DB2DIR* is the location of the newly installed V10.5 copy.

- For Windows operating systems:

```
<DB2PATH>\db2tss\bin\configTool upgradeConfigFolder
-sourceConfigFolder "<DB2PATH>\CFG\DB2TSS\CONFIG"
-targetConfigFolder "<INSTPROFDIR>\<INSTANCENAME>\DB2TSS\CONFIG"
```

where, *<DB2PATH>* is the location of the newly installed V10.5 copy and *<INSTPROFDIR>* is the instance profile directory.

**Note:** For Windows systems, if the DB2 instance was not configured previously for DB2 Text Search and the DB2 version to be upgraded is Version 9.7 Fix Pack 1 or later, you can skip this step.

The **configTool upgradeInstance** command replaces, modifies, and merges text search configuration and data files and directories.

#### The config directory

The command copies the following files into the *<ECMTS\_HOME>\config* directory if the files do not already exist in this directory:

- constructors.xml
- ecmts\_logging.properties
- ecmts\_config\_logging.properties

The following files are copied and any existing files are overwritten:

- build\_info.properties
- constructors.xsd
- ecmts\_config\_logging.properties
- mimetypes.xml
- monitoredEventsConfig.xml

The configuration settings from the following files are merged to the *configuration.xml* file. Values are added for new settings, and values are maintained for existing settings.

- config.xml
- jetty.xml

The following files are not modified:

- authentication.xml
- key.txt
- All files in the collections subdirectory

#### The log directory

The command does not change the contents of the existing log directory. However, when new log files are generated, those new files might replace existing log files.

The **configTool upgradeInstance** command does not upgrade text search filters for an integrated text search server.

4. Upgrade the current DB2 instance by issuing the **db2iupgrade** command.
  - For Linux and UNIX operating systems, the command is located under the `$DB2DIR/instance` directory, where `DB2DIR` is the location of the newly installed DB2 database server V10.5 copy.

```
db2iupgrade -j "TEXT_SEARCH [[,service-name] | [,port-number]]" DB2INST
```
  - For Windows operating systems, the property file is located in `<DB2PATH>\bin` directory, where `<DB2PATH>` is the location of the newly installed DB2 V10.5 copy.

```
db2iupgrade DB2INST /j "TEXT_SEARCH [[,service-name] | [,port-number]]"
```

For more information, see the topic about **db2iupgrade** command.

**Note:** If you installed a new V10.5 copy with the upgrade option, and selected DB2 Text Search as a feature to be upgraded, then you can skip this step.

5. Back up the values for all configurable properties of DB2 Text Search that were used in the previous release by running the following script:
  - For Linux and UNIX operating systems:

```
$DB2DIR/db2tss/bin/bkuptscfg.sh $INSTNAME
```

where, `DB2DIR` represents the location of the newly installed V10.5 copy, and `INSTNAME` represents the name of the instance to be upgraded.

- For Windows operating systems:

```
<DB2PATH>\db2tss\bin\bkuptscfg.bat <INSTANCENAME> <DB2PATH>
```

where, `<DB2PATH>` represents the location of the newly installed V10.5 copy, `<INSTANCENAME>` represents the name of the instance to be upgraded.

The backed-up configurable properties are redirected into one property file:

- For Linux and UNIX operating systems, the property file is located in the `$INSTHOME/sqllib/db2tss/config/db2tssrvupg.cfg` directory, where `INSTHOME` represents the instance home directory.
- For Windows operating systems, the property file is located in the `<INSTPROFDIR>\<INSTANCENAME>\db2tss\config\db2tssrvupg.cfg` directory, where `<INSTPROFDIR>` represents the instance profile directory. You can obtain the instance profile directory by issuing the **db2set DB2INSTPROF** command, and `<INSTANCENAME>` represents the name of the instance to be upgraded.

**Note:** If the DB2 instance was not configured with DB2 Text Search in an earlier copy of a DB2 release, you can skip this step.

6. Set the `DB2INSTANCE` environment variable to the current upgraded instance.
7. Upgrade the databases by issuing the **DB2 UPGRADE DATABASE** command. If the **DB2 UPGRADE DATABASE** command returns the ADM4003E error message, upgrade the DB2 Text Search catalog and indexes manually by using the `SYSTS_UPGRADE_CATALOG` and `SYSTS_UPGRADE_INDEX` stored procedures.
8. For each upgraded database, verify whether the text search server properties information in the text search `SYSIBMTS.TSSERVERS` catalog table is correct by comparing the property values backed up in step 7. If the value of the token or port number in the catalog table is empty or incorrect, you must

update the text server information manually. For details about how to update, see the topic about updating DB2 Text Search server information.

9. Review the values for all DB2 Text Search configurable properties. Compare with the values that you backed up to ensure that they have correct values. Issue the following command to check the configuration values:  
`configTool printAll -configPath <configuration-directory>`
10. If you disabled DB2 Text Search for rich text document support, you have to install DB2 V10.5 Accessories Suite. For more information, see the topic about installing DB2 Accessories Suite.
11. Then enable rich text document support. For more information, see the topic about enabling DB2 Text Search for rich text and proprietary format support.
12. Verify that the upgrade was successful by starting the DB2 Text Search instance service. If you disabled rich text document support, verify that rich text document support is enabled by issuing text search queries and compare with pre-upgrade results.

## Upgrading DB2 Text Search for non-root installation (Linux and UNIX)

If you are upgrading DB2 Text Search Version 10.5, you must upgrade the DB2 server, instance, and all databases.

### Before you begin

Complete the following tasks before you begin to upgrade your text search server:

1. Enable the root-based features for your user ID. You might have to ask a system administrator with root access to issue the **db2rfe** command.
2. Log in as the instance owner or as a user with SYSADM authority. Then stop the DB2 instance and the DB2 Text Search instance service.
3. Back up the old DB2 copy into a *<backup-dir>* directory.
4. If you enabled DB2 Text Search for rich text document support, disable rich text document support. For more information about how to disable rich text document support, see disabling DB2 Text Search for rich text document support.
5. Log on to the DB2 server as a non-root user. Review the database instance type to ensure it can be upgraded as a non-root installation.

### Procedure

To upgrade DB2 Text Search:

1. Install a new DB2 Version 10.5 copy with the **db2nrupgrade** upgrade command. Select the DB2 Text Search component that you want to upgrade. If you specified the **-f nobackup** parameter and the DB2 database product installation failed, you must manually install the DB2 database product by selecting the DB2 Text search component from the feature tree and then upgrade the non-root instance by issuing the following command:

```
db2nrupgrade -b <backup-dir> -j "TEXT_SEARCH"
```

*<backup-dir>* specifies the directory where the configuration files from the old DB2 version are stored. For details about the upgrade non-root instance command, see **db2nrupgrade** command.

2. Back up values for all configurable properties of DB2 Text Search that is used in the previous release before the database upgrade by running the following script:

```
$INSTHOME/sql1lib/db2tss/bin/bkuptscfg.sh
```

The backed-up configurable properties are redirected into the `$INSTHOME/sql1lib/db2tss/config/db2tssrvupg.cfg` property file.

3. Upgrade the existing databases by issuing the **UPGRADE DATABASE** command.
4. For each upgraded database, verify whether the text search properties information in the text search catalog table `SYSIBMTS.SYSTSSERVERS` is correct by comparing the information with the property values from step 6. If the value of token or port number in the catalog table is empty or incorrect, you must update the text server information manually. For more information about the upgrading non-root instance, see updating DB2 Text Search server information.
5. Upgrade the DB2 Text Search server for your instances by issuing the **configTool upgradeInstance** command.

- For Linux and UNIX operating systems:

```
$DB2DIR/db2tss/bin/configTool upgradeConfigFolder
-sourceConfigFolder $DB2DIR/cfg/db2tss/config
-targetConfigFolder $INSTHOME/sql1lib/db2tss/config
```

*INSTHOME* is the instance home directory and *DB2DIR* is the location of the newly installed V10.5 copy.

6. Compare the values that you backed up in step 6 with the values for all the DB2 Text search configurable properties to ensure that all the values are correct. Issue the following command to check the configuration values:  
`configTool printAll -configPath configuration-directory`
7. If you disabled DB2 Text Search for rich text document support, you must install the DB2 V10.5 Accessories Suite. For information about the Accessories Suite, see installing DB2 Accessories Suite for DB2 Text Search.
8. Then enable rich text document support. For more information about enabling support, see enabling DB2 Text Search for rich text and proprietary format support.
9. Verify that the upgrade was successful by starting the DB2 Text Search instance service. If you disabled rich text document support, verify that rich text document support is enabled by issuing text search queries and compare with pre-upgrade results.

## Upgrading a multi-partition instance without DB2 Text Search

To obtain the latest functionality upgrade your DB2 Text Search instance. You need to upgrade the DB2 server, instance, and all databases when upgrading the text search instance.

### About this task

Starting in DB2 Version 10.1, text search supports indexes in a partitioned database environment. The following steps describe the process to upgrade a DB2 Version 10.1 or Version 9.7 multi-partition instance for root install. DB2 Text Search should not be installed on the instances.

### Procedure

1. Log in as the instance owner or a user with SYSADM authority.



2. Install a new copy of the DB2 Text Search version you are upgrading to, and perform a custom installation. DB2 Text Search is an optional component that is only available when you select a custom installation.
3. Upgrade your instances by issuing the **db2iupgrade** command:  
`db2iupgrade /j "text_search [[,service-name]] [[,port-number]]"`
4. Upgrade the existing databases by issuing **DB2 UPGRADE DATABASE** command.
5. For each upgraded database, update the text server information manually. For more information, see the topic about updating DB2 Text Search server information.

---

## Upgrading DB2 servers in Microsoft Cluster Server environments

Upgrading DB2 servers in Microsoft Cluster Server (MSCS) environments to DB2 Version 10.5 requires that you install DB2 Version 10.5 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-DB2 Version 10.5 MSCS resources to DB2 Version 10.5 DB2 MSCS resources.

### Before you begin

- Ensure that you have Local Administrator access.
- SYSADM authority is required.
- Review upgrade recommendations and disk space requirements. Refer to “Best practices for upgrading DB2 servers” on page 30 and “Disk space requirements for DB2 server upgrades” on page 27.
- Perform pre-upgrade tasks, especially back up your databases. Refer to Chapter 5, “Pre-upgrade tasks for DB2 servers,” on page 35 and “Backing up databases before or after upgrade” on page 38.

### Restrictions

- This procedure applies only to upgrade from DB2 32-bit servers when you install the DB2 Version 10.5 32-bit database product, or from DB2 64-bit servers when you install the DB2 Version 10.5 64-bit database product. The instance bit size is determined by the operating system and the DB2 Version 10.5 database product that you install, see “Support changes for 32-bit and 64-bit DB2 servers” on page 29 for details.
- Use only the **Install New** option in the **Install a Product** panel to install DB2 Version 10.5. If you choose the **upgrade** action when you select the **Work with Existing** option on the **Install a Product** panel, the installation process fails.
- Additional upgrade restrictions apply. Refer to “Upgrade restrictions for DB2 servers” on page 21. Review the complete list.

### Procedure

To upgrade a DB2 server in an MSCS environment to DB2 Version 10.5:

1. Log on to the DB2 server as a user with Local Administrator authority.
2. Install DB2 Version 10.5 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-DB2 Version 10.5 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 by using the Cluster Administrator.
6. Bring online the group of resources containing the upgraded instance by 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 10.5.. Refer to "Upgrading the DB2 Administration Server (DAS)" on page 53.  
If you choose to create a new DAS, you have to re-configure the DAS settings for your MSCS environment.
8. Upgrade your databases. Refer to "Upgrading databases" on page 54.

## 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 Chapter 9, "Post-upgrade tasks for DB2 servers," on page 97 and "Verifying upgrade of DB2 servers" on page 104.



---

## 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. Existing tables that have row compression enabled from a pre-DB2 Version 10.5 database will have classic row compression enabled. If you want to use adaptive compression, it must be enabled after the upgrade is performed. For details, see *Adjusting adaptive compression settings*.
3. Adjust the log space size. If you changed your log space setting as recommended in the pre-upgrade tasks for DB2 servers, reset the **logfilsiz**, **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 100 for details.
4. Ensure that existing libraries for your external routines remain on the original location before the upgrade, if necessary, restore these libraries from the backup that you perform in “Backing up DB2 server configuration and diagnostic information” on page 40.
5. Activate your database after upgrade to start up your database and all necessary database services. See “Activating a database after upgrade” on page 101 for details.
6. Automatic storage table spaces inherit media attribute values, including overhead, device read rate and data tag attributes, from the storage group it is using by default. After upgrading to DB2 Version 10.5, the existing table spaces retain their settings and the OVERHEAD and DEVICE READ RATE attributes for the storage group are set to undefined. You can set media attributes with the ALTER STOGROUP statement. For details, see *Storage group attributes*.
7. 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 10.5 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 101 for details.
8. 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*.
9. . If you did not use the **REBINDALL** option on the **UPGRADE DATABASE** command then rebind packages in upgraded databases. 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 103 for details.

10. Refresh the data in existing materialized query tables by using the **REFRESH TABLE** statement. Materialized query tables (MQT) on unicode databases using language aware collation, where the MQT definition involves a LIKE predicate or substring function involved in a basic predicate, need to be refreshed.
11. Migrate DB2 explain tables to retain explain table information that you previously gathered. See “Upgrading explain tables” 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 10.1, or Version 9.7 copy and *DB2DIR* is the location of your DB2 Version 10.5 copy. You do not have to copy standard code page conversion tables.  
  
If you upgraded your existing DB2 Version 10.1, or Version 9.7 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 10.5 copy.
13. Upgrade existing target tables for event monitors that write to tables and to unformatted event (UE) tables by using the new **EVMON\_UPGRADE\_TABLES** procedure. For details, see Event monitor data retention from release to release.
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 104 for details.
15. Back up your databases after the DB2 server upgrade is complete. See “Backing up databases before or after upgrade” on page 38 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.
17. If you have not already done so, you must migrate your SQL Replication in order to support new LSN formats. For details, see Migrating to SQL replication Version 10.1

## What to do next

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

- If you upgraded your existing DB2 Version 10.1, or Version 9.7 copy, the database log directories will have been changed. Review the *db2diag.log* file which will have entries detailing the new log directories. If a user defined log directory is used, for example */usr/logpath*, after upgrade the location of the log files will be */usr/logpath/NODE0000/LOGSTREAM0000*. The old log directory will only contain renamed log files. If the default database directory is being used, for example */home/db2user/db2inst/NODE0000/SQL00001/SQLLOGDIR*, after upgrade the location of the log files will be */home/db2user/db2inst/NODE0000/SQL00001/LOGSTREAM0000*. The old log directory will only contain renamed log files.
- 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 10.5 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.

- 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 10.5, 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 10.1, Version 9.7 or DB2 Version 9.8 copies that you no longer need.

---

## Adjusting adaptive compression settings

Existing tables that have row compression enabled from a pre-DB2 Version 10.5 database will be upgraded to have classic row compression enabled. If you want to use adaptive compression you must enable it after the upgrade is performed.

### Before you begin

The default behaviour for compression has changed in DB2 Version 10.1, and has the syntax for enabling compression. For details, see “ALTER TABLE and CREATE TABLE statement have been changed.”

### About this task

Existing tables that have row compression enabled from a pre-DB2 Version 10.5 database will be upgraded to have classic row compression enabled. If you want to use adaptive compression you must enable it after the upgrade is performed.

### Procedure

To take advantage of adaptive compression the following steps must be performed.

1. Estimate storage space savings by executing the administrative function **ADMIN\_GET\_TAB\_COMPRESS\_INFO**. Compare the generated estimate with the current or actual compression table savings. If the estimated compression savings that can be achieved using adaptive compression meet your requirements, proceed with enabling adaptive compression.
2. Perform **ALTER TABLE** with **COMPRESS YES ADAPTIVE** clause to enable adaptive compression. Modification of existing data rows and population of new rows will then be automatically subject to adaptive compression. Existing table rows are not immediately subject to adaptive compression as a result of issuing this ALTER statement. Any subsequent modification of existing rows or input of new rows into the table will lead to the application of adaptive compression.
3. If you want to compress all existing rows, you can perform a classic table reorganization to immediately have all existing rows compressed, in a table that has been enabled for adaptive compression. The classic table reorganization should ideally be performed with the **RESETDICTIONARY** parameter to achieve the maximum compression possible. Subsequent reorganization for the purposes of

better compressing data rows may no longer be required. If desired, use the **ADMIN\_MOVE\_TABLE** procedure instead of performing a classic table reorganization.

---

## 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.

### Before you begin

To increase the size of table spaces and log space, you must have SYSCTRL or SYSADM authority.

#### Restrictions

On a partitioned database environment, you must 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. To support larger log record headers, increase the log space setting, by approximately 10% - 15% over what you used for DB2 Version 9.7.
4. To support larger log record headers, increase the **softmax** parameter by 10% - 15% over what you used for DB2 Version 9.7.

```
db2 UPDATE DB CFG FOR sample using SOFTMAX $1.15 * \textit{previous-value}$
```

**Important:** The **softmax** database configuration parameter is deprecated in Version 10.5 and might be removed in a future release. For more information, see *Some database configuration parameters are deprecated in What's New for DB2 Version 10.5*.

5. Double the value for the **logbufsz** parameter:

```
db2 UPDATE DB CFG FOR sample using LOGBUFSZ $2 * \textit{previous-value}$
```

6. Disconnect from the database that you upgraded:

```
db2 CONNECT RESET
```

**logfilsiz** changes take effect only 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.

### Before you begin

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

### Procedure

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.

### Results

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.

### About this task

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.

## Procedure

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:
  - “DB2 server behavior changes” on page 23
  - Consider removing registry variables that have been deprecated or discontinued in DB2 Version 10.5 or earlier releases that can impact the upgrade of your DB2 server:
    - 
    - Deprecated registry variables in DB2 Version 10.1
    - Discontinued registry variables in DB2 Version 10.1
    - Deprecated registry variables in DB2 Version 9.7
    - Discontinued registry variables in DB2 Version 9.7
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 10.5 copy.
3. Review the information about new, changed, and deprecated database manager configuration parameters, and based on the upgrade impact, choose the appropriate settings:
  - “DB2 server behavior changes” on page 23
  - There are no database manager configuration parameters that have been deprecated or discontinued in this release. However, if you are upgrading from DB2 Version 10.1 or earlier, consider removing database manager configuration parameters that have been deprecated in pre-DB2 Version 10.5 releases:
    - Deprecated database manager configuration parameters in DB2 Version 10.1
    - Deprecated database manager configuration parameters in DB2 Version 9.7
4. Review the information about new, changed, deprecated, and discontinued database configuration parameters, and based on the upgrade impact, choose the appropriate settings:
  - “DB2 server behavior changes” on page 23
  - Review the topic for further details on functionality that has been deprecated or discontinued in this release. If you are upgrading from DB2 Version 10.1 or earlier, consider removing database manager configuration parameters that have been deprecated or discontinued in pre-DB2 Version 10.5 releases:
    - Deprecated and discontinued database configuration parameters in DB2 Version 10.1
    - Deprecated and discontinued database configuration parameters in DB2 Version 9.7
5. Review the changes in database physical design characteristics and security, and based on the upgrade impact, modify database objects accordingly:
  - “DB2 server behavior changes” on page 23



## What to do next

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.

---

## 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.

### Before you begin

Ensure that you have DBADM authority.

### About this task

Packages will be implicitly rebound the first time that an application uses them after you upgrade your database. To eliminate this overhead, you can explicitly rebind invalid packages. You must explicitly rebind inoperative packages. Alternatively, you can specify the **REBINDALL** option on the **UPGRADE DATABASE** command in “Upgrading databases” on page 54.

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

### Procedure

To rebind packages in upgraded databases:

1. Log on as a user with DBADM authority.
2. Rebind all invalid packages in each database:
  - From the CLP, run the **db2rbind** command, as follows:

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

The all clause rebinds valid and invalid packages. Review the log file specified by *logfile*, and address any issues.
  - From IBM Data Studio, open the task assistant to rebind packages.
3. Verify that your DB2 server upgrade was successful. For details, see Verify your DB2 server upgrade. Test your applications and tools to ensure that the server is working as expected. For details, see “Verifying upgrade of DB2 servers” on page 104.

### Results

After you have rebound all your database packages, you can automatically take advantage of optimizer improvements. Refer to Chapter 22, “Upgrade essentials for database applications,” on page 141 for details on the optimizer improvements available in this release.

---

## Upgrading explain tables

If you must maintain explain table information that you gathered in your DB2 copies from previous releases, upgrade your explain tables to DB2 Version 10.5.

## Before you begin

Ensure that you have DBADM authority. For additional authorization details, see *Command Reference*.

## About this task

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

## Procedure

To upgrade the explain tables, run the **db2exmig** command, as follows:

```
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 that you are migrating. This parameter is required.
- *userid* and *password* represent the current user's ID and password. These parameters are optional.

## Results

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

## What to do next

Use the **db2expln** command to see the access plan information in the upgraded explain tables.

---

## Verifying upgrade of DB2 servers

When you upgrade your DB2 server, it is a good measure to run some tests on the new 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.

## Before you begin

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 10.5 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 10.5 functionality in upgraded databases

After you upgrade your DB2 server, enhance the functionality and improve the performance of your upgraded databases by adopting new Version 10.5 functionality.

### Before you begin

You must upgrade your DB2 server to Version 10.5.

### Procedure

Perform any of the following steps to adopt the specified Version 10.5 functionality in your upgraded DB2 environment:

- Use index that includes an expression in its key definition to enhance the performance of queries that contain expressions. For more information, Expression-based Indexes.
- Use DB2 column-organized tables to add columnar capabilities, improve the storage and query performance of DB2 databases. For more information, see Column-organized tables.

### What to do next

If you upgraded your DB2 server from DB2 Version 9.7, adopt functionality that is introduced in Version 10.1 releases in your upgraded DB2 environment. See the following topics for details:

- Adopting new DB2 Version 10.1 functionality in upgraded databases in the *Upgrading to DB2 Version 10.5* guide.



---

## Chapter 11. Migrating DB2 functionality to DB2 database product features

Migrating DB2 functionality to specific DB2 database 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 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.

#### Before you begin

- 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. 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. Query Patroller has been discontinued in Version 10.1.

#### 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 workload management features introduced in DB2 Version 9.5, the DB2 governor utility was deprecated in Version 9.7 and might be removed in a future release. It is not supported in DB2 pureScale environments. For more information, see “DB2 Governor and Query Patroller have been deprecated” at <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0054901.html>.

## Procedure

To migrate from DB2 Governor to DB2 WLM:

1. Upgrade the data server where the Governor is installed to DB2 Version 10.5 so that you have an environment where DB2 WLM and the Governor can coexist. Use one of the following tasks:

- Chapter 6, “Upgrading a DB2 server (Windows),” on page 49
- Chapter 7, “Upgrading a DB2 server (Linux and UNIX),” on page 61

After the upgrade, there is a default workload created to identify all the user database activities and the workload is mapped to the default user service class which defines an execution environment. The Governor **ACTION NICE** rule clause is managed in only the default user service class. You cannot use the Governor to alter the priority of agents in user-defined service superclasses and subclasses. However, all other governor rules are enforced for all user-defined service classes.

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 rowssel 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.





---

## 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.

### Before you begin

- 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. See “Best practices for upgrading DB2 servers” on page 30 and “Disk space requirements for DB2 server upgrades” on page 27.
  - Take an offline full backup of all databases that you are going to upgrade. See “Backing up databases before or after upgrade” on page 38.
  - Back up all database manager configuration parameter values for each instance and all database configuration parameter values for each database. See “Backing up DB2 server configuration and diagnostic information” on page 40.
  - Perform other pre-upgrade tasks that apply to your environment. See Chapter 5, “Pre-upgrade tasks for DB2 servers,” on page 35.
- Keep your existing pre-DB2 Version 10.5 copy during upgrade of your DB2 server. To do this, select the **Install New** option to create a new copy when installing DB2 Version 10.5. Do not select the **Work with an existing** option and then choose a pre-DB2 Version 10.5 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. See “Upgrade restrictions for DB2 servers” on page 21. 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 10.5 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 10.5 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-DB2 Version 10.5 instances to DB2 Version 10.5, re-create your instances in the pre-DB2 Version 10.5 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-DB2 Version 10.5 instance, log on to the DB2 server as the instance owner and restore your upgraded databases from a pre-DB2 Version 10.5 offline full backup by running the **RESTORE DATABASE** command. You cannot upgrade your databases from DB2 Version 10.5 to pre-DB2 Version 10.5 release.

If you recreated the instances using the same instance owner they had before upgrade and you did not upgrade a database to a DB2 Version 10.5 instance, the database is still in pre-DB2 Version 10.5 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\LOGSTREAM0000
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 117
- Chapter 14, “Upgrade essentials for clients,” on page 119
- Chapter 15, “Pre-upgrade tasks for clients,” on page 123
- Chapter 16, “Upgrading to Data Server Client (Windows),” on page 127
- Chapter 17, “Upgrading to Data Server Runtime Client (Windows),” on page 129
- Chapter 18, “Upgrading clients (Linux and UNIX),” on page 131
- Chapter 20, “Post-upgrade tasks for clients,” on page 135



---

## Chapter 13. Clients upgrade

Upgrading to DB2 Version 10.5 might require the upgrade of your clients.

Upgrading a client involves installing a DB2 Version 10.5 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 10.5. You can directly upgrade to DB2 Version 10.5 clients from Version 10.1, or Version 9.7. If you have Version 9.5 or earlier clients, upgrade to any Version 9.7 or Version 10.1 client first.

Review Chapter 14, “Upgrade essentials for clients,” on page 119 for details about upgrade support and options available for clients.



---

## Chapter 14. Upgrade essentials for clients

Upgrading clients to DB2 Version 10.5 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 10.5.

In the upgrading client topics, the term *pre-DB2 Version 10.5 clients* refers to Version 10.1, and Version 9.7 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 DB2 Version 10.5 client:

*Table 15. Upgrade options for DB2 Version 10.5 clients*

| Upgrading from                                                                                                                                   | Upgrading to                                         | Upgrade support details                                                                                                                                                                                                                                                                                                                                                                                                                     |
|--------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"><li>Version 10.1 Data Server Client</li><li>Version 9.7 Data Server Client</li></ul> (Windows)                 | DB2 Version 10.5 Data Server Client(Windows)         | <p>You have two options:</p> <ul style="list-style-type: none"><li>Install the DB2 Version 10.5 Data Server Client, and choose a pre-DB2 Version 10.5 client copy with the <b>upgrade</b> action in the <b>Work with Existing</b> window. The client instance is then automatically upgraded for you.</li><li>Install a new copy of the DB2 Version 10.5 Data Server Client, and then manually upgrade existing client instances.</li></ul> |
| <ul style="list-style-type: none"><li>Version 10.1 Data Server Runtime Client</li><li>Version 9.7 Data Server Runtime Client</li></ul> (Windows) | DB2 Version 10.5 Data Server Runtime Client(Windows) | <ul style="list-style-type: none"><li>Install the DB2 Version 10.5 Data Server Runtime Client as a new copy, and then manually upgrade your existing client instance.</li></ul>                                                                                                                                                                                                                                                             |
| All Version 10.1, or Version 9.7 clients (Linux or UNIX)                                                                                         | All DB2 Version 10.5 clients (Linux or UNIX)         | <ul style="list-style-type: none"><li>Install a new copy of any DB2 Version 10.5 client, and then manually upgrade your existing client instance.</li></ul>                                                                                                                                                                                                                                                                                 |

When you upgrade a client instance, the bit size is determined by the operating systems where you installed the DB2 Version 10.5 client. Refer to Table 12 on page 29 for details.

### Upgrade restrictions for clients

Review “Upgrade restrictions for DB2 servers” on page 21 for information regarding instance upgrade and operating system support. These restrictions also apply to clients and can impact their upgrade.

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 10.5, the following support for connectivity between clients and DB2 servers is available:

*Table 16. DB2 Version 10.5 connectivity support*

| Client                                    | DB2 server                               | Client connectivity support                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------------------------------------|------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 32-bit or 64-bit DB2 Version 10.5 clients | 32-bit or 64-bit DB2 Version 10.5 server | Version 10.5 clients other than the IBM Data Server Driver for JDBC and SQLJ can establish 32-bit or 64-bit connections. For the IBM Data Server Driver for JDBC and SQLJ: <ul style="list-style-type: none"> <li>• With type 4 connectivity, a 32-bit or 64-bit Java application can connect to a 32-bit or 64-bit server.</li> <li>• With type 2 connectivity <ul style="list-style-type: none"> <li>– A 32-bit or 64-bit Java application can make a remote connection to a 32-bit or 64-bit server.</li> <li>– A 64-bit Java application can make a local connection to a 32-bit or 64-bit server.</li> <li>– A 32-bit Java application can make a local connection only to a 32-bit server.</li> </ul> </li> </ul> |
| 32-bit or 64-bit DB2 Version 9.7 clients  | 32-bit or 64-bit DB2 Version 10.5 server | Only DB2 Version 9.7 or earlier functionality is available.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 32-bit or 64-bit DB2 Version 10.1 clients | 32-bit or 64-bit DB2 Version 10.5 server | Only DB2 Version 10.1 or earlier functionality is available.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

Connections to DB2 Version 9.1 servers from a Version 10.5 client is supported. However, DB2 Version 9.1 reached end of support on April 30, 2012. For more support lifecycle information, see <http://www-01.ibm.com/software/data/support/lifecycle/>. For continued Version 9.1 support, a service extension is required.

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 DB2 Version 10.5 client, a pre-DB2 Version 10.5 server lists only relational indexes while a DB2 Version 10.5 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 143 and “Upgrade impact from SQL statement changes” on page 145 for details.



---

## Best practices for upgrading clients

Consider the following best practices when planning your client upgrade.

### Determine whether to upgrade clients or DB2 servers first

In general, upgrade clients after you upgrade your DB2 servers is the traditional approach. Supported pre-DB2 Version 10.5 clients can connect to DB2 Version 10.5 servers. However, the functionality introduced in releases after the pre-DB2 Version 10.5 client release is not available. If you plan to use this functionality in your applications, upgrade your clients to DB2 Version 10.5 or install new DB2 Version 10.5 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 141 for changes to DB2 APIs, DB2 commands, and SQL statements
- “DB2 server behavior changes” on page 23 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 27 for discontinued functionality not supported by DB2 Version 10.5 clients
- “Changed functionality” in DB2 Version 10.5 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 occurred during the upgrade process. You can also test your database applications and determine if you must upgrade them to run successfully in DB2 Version 10.5.

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-DB2 Version 10.5 servers using DB2 Version 10.5 clients

### Install a new client copy instead of upgrading existing client

If you have software that requires a pre-DB2 Version 10.5 client, install the DB2 Version 10.5 client as a new copy and keep your existing client copy to satisfy the software requirement. Then create a DB2 Version 10.5 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.

### Procedure

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.  
For more details, see Chapter 14, “Upgrade essentials for clients,” on page 119.
2. Review the supported and non-supported client configurations.
3. Plan your upgrade strategy.  
For more details, see Chapter 2, “Planning your DB2 environment upgrade,” on page 5. For example, you might need to upgrade your DB2 server first, then your clients.
4. Optional: Upgrade your DB2 servers.  
For more details, see Chapter 3, “DB2 servers upgrade,” on page 17.
5. Back up your client configuration information.  
For more details, see “Backing up 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.  
For more details, see “Upgrading clients in a test environment” on page 124.

---

## 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.

### Before you begin

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 must 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. Use 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 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 10.5.

### Before you begin

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

#### 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.

### 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          | "%DB2PATH%\bin\db2icrt -s client<br><i>InstName</i>           |
| Linux and UNIX   | <i>\$DB2DIR/instance/db2icrt -s client</i><br><i>InstName</i> |

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 DB2 Version 10.5 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 15 on page 119 to determine what client product to install.
5. Upgrade your client instance by running the **db2iupgrade** command:

| Operating system | DB2 command                                   |
|------------------|-----------------------------------------------|
| Windows          | "%DB2PATH%\bin\db2iupgrade <i>InstName</i>    |
| Linux and UNIX   | <i>\$DB2DIR/instance/db2iupgrade InstName</i> |

where **DB2PATH** and *DB2DIR* are set to the location of the DB2 Version 10.5 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 that the client upgrade was successful.
9. Test your applications, scripts, tools, and maintenance procedures by using the DB2 Version 10.5 client.



---

## Chapter 16. Upgrading to Data Server Client (Windows)

Upgrading an existing client copy to DB2 Version 10.5 requires that you install a DB2 Version 10.5 Data Server Client copy and then upgrade your client instance to retain your client configuration and to connect to all your previously cataloged databases.

### Before you begin

- 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.

Refer to Chapter 15, “Pre-upgrade tasks for clients,” on page 123.

### About this task

When you install a DB2 Version 10.5 Data Server Client, you can choose to automatically upgrade an existing pre-DB2 Version 10.5 client copy. Your existing client instances are upgraded to a new DB2 Version 10.5 Data Server Client copy and the existing pre-DB2 Version 10.5 client copy is removed. You can also choose to install a new copy of DB2 Version 10.5 Data Server Client and then manually upgrade your existing client instance after installation.

#### Restrictions

- The bit size of the client instance is determined by the operating system where you install a DB2 Version 10.5 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 12 on page 29 for details.

### Procedure

To upgrade from an existing client copy to a DB2 Version 10.5 Data Server Client on Windows:

1. Install DB2 Version 10.5 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 9.5 Data Server Client or Version 9.7 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 DB2 Version 10.5 Data Server Client and keep your existing client copy. After installation, you must manually upgrade the client instance to run on the DB2 Version 10.5 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 DB2 Version 10.5 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 DB2 Version 10.5 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 DB2 Version 10.5 Data Server Client copy through the default interface set the DB2 Version 10.5 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 DB2 Version 10.5 client instance instead of upgrading the existing client instance. You only need to create a new DB2 Version 10.5 client instance when you want to keep multiple client copies running on the same machine, or create a testing environment. To create a new DB2 Version 10.5 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.

## What to do next

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. Refer to Chapter 20, “Post-upgrade tasks for clients,” on page 135 and “Verifying your client upgrade” on page 135.



---

## Chapter 17. Upgrading to Data Server Runtime Client (Windows)

Upgrading an existing Runtime Client copy to DB2 Version 10.5 requires that you install a DB2 Version 10.5 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 DB2 Version 10.5 Data Server Runtime Client copy, you can manually upgrade your existing client instance from a Version 10.1, or Version 9.7 Data Server Runtime Client.

### Before you begin

- Ensure that you have SYSADM, SYSCTRL, or SYSMAINT authority and Local Administrator authority to run the **db2iupgrade** and the **db2icrt** commands.
- Review supported connectivity between clients and DB2 servers in Chapter 14, “Upgrade essentials for clients,” on page 119.
- Perform pre-upgrade tasks for clients.  
Refer to Chapter 15, “Pre-upgrade tasks for clients,” on page 123.

### Restrictions

- The bit size of the client instance is determined by the operating systems where you install DB2 Version 10.5 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 12 on page 29 for details.

### Procedure

To upgrade from a Version 10.1, or Version 9.7 DB2 Runtime Client copy to DB2 Version 10.5 Data Server Runtime Client on Windows:

1. Install DB2 Version 10.5 Data Server Runtime Client. See “Installing IBM data server clients (Windows)” in *Installing IBM Data Server Clients*. Run the DB2 Setup wizard for all languages.
2. If you want your applications to use the DB2 Version 10.5 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 DB2 Version 10.5 Data Server Runtime Client installation and *InstName* is the name of the instance.

5. Optional: You can create a new DB2 Version 10.5 client instance instead of upgrading an existing client instance. You only need to create a new DB2 Version 10.5 client instance when you want to keep multiple client copies running on the same machine. To create a new DB2 Version 10.5 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.

## What to do next

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. Refer to Chapter 20, “Post-upgrade tasks for clients,” on page 135 and “Verifying your client upgrade” on page 135.

---

## Chapter 18. Upgrading clients (Linux and UNIX)

Upgrading existing clients to DB2 Version 10.5 requires that you install a DB2 Version 10.5 client copy and then upgrade your existing client instances to retain your client configuration and to connect to all your previously cataloged databases.

### Before you begin

- Ensure that you have root user authority.
- Ensure that you have SYSADM, SYSCTRL, or SYSMAINT 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 database servers in Chapter 14, “Upgrade essentials for clients,” on page 119.
- Perform pre-upgrade tasks for clients. Refer to Chapter 15, “Pre-upgrade tasks for clients,” on page 123.

### Restrictions

- You can only upgrade from a DB2 Version 10.1, or DB2 Version 9.7 Data Server Client to a DB2 Version 10.5 Data Server Client.
- You can only upgrade from a DB2 Version 10.1, or DB2 Version 9.7 Data Server Runtime Client to a DB2 Version 10.5 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 DB2 Version 10.5 64-bit client instances. The bit size of the client instance is determined by the operating system where you install the DB2 Version 10.5 client. Refer to Table 12 on page 29 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 DB2 Version 10.5 clients:

1. Install the appropriate DB2 Version 10.5 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 DB2 Version 10.1, or DB2 Version 9.7 Data Server Client, install a new DB2 Version 10.5 Data Server Client.
  - If you are upgrading from a DB2 Version 10.1, or DB2 Version 9.7 Data Server Runtime Client, install a new DB2 Version 10.5 Data Server Runtime Client copy.
2. Log on to the system with root user authority.
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 DB2 Version 10.5 client installation. The default installation path for UNIX is `/opt/IBM/db2/V10.5` and for Linux is `/opt/ibm/db2/V10.5`.
- *InstName* is the login name of the client instance owner.

4. Optional: You can also create a new DB2 Version 10.5 client instance instead of upgrading the existing client instance. You only need to create a new DB2 Version 10.5 client instance when you want to keep multiple client copies running on the same machine. To create a new DB2 Version 10.5 client instance, run the **db2icrt** command with the option **-s**:

```
$DB2DIR/instance/db2icrt -s client InstName
```

where

- *DB2DIR* is set to the location that you specified during the DB2 Version 10.5 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 that the changed values are compatible with your database applications.

## What to do next

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. Refer to Chapter 20, “Post-upgrade tasks for clients,” on page 135 and “Verifying your client upgrade” on page 135.

---

## Chapter 19. Upgrading to IBM Data Server Driver Package

Upgrading to IBM Data Server Driver Package (DSDRIVER) requires that you install a DB2 Version 10.5 DSDRIVER and optionally set the default client interface.

### Before you begin

- Review supported connectivity between DB2 clients and DB2 servers in Chapter 14, “Upgrade essentials for clients,” on page 119.

### Procedure

1. Install a DB2 Version 10.5 DSDRIVER copy. See “Installation methods for IBM data server clients” in *Installing IBM Data Server Clients* for details.
  - If there is no existing DSDRIVER installed, then install the latest version of the DSDRIVER. The new DSDRIVER will be installed to a new copy.
  - If there is one existing copy of the DSDRIVER:
    - If there is an existing DSDRIVER and a copy name is not provided for the new install, the default behavior is to install the DSDRIVER on top of that copy and upgrade it to the current level.
    - If there is an existing DSDRIVER and a copy name is provided in the install command line or in the response file (for the silent install) the DSDRIVER will be installed to that copy, whether it is a new copy, or an existing DSDRIVER copy.
  - If there are 2 or more existing DSDRIVER copies:
    - If one of the existing DSDRIVER copies is set as the default DB2 client interface copy:
      - If no copy name is provided during install, the DSDRIVER will be installed on top of the default client interface copy.
      - If a copy name is provided during install, the DSDRIVER will be installed to that copy, whether it is an existing copy or a new one.
    - If none of the existing DSDRIVER copies is set as the default DB2 client interface copy:
      - If no copy name is provided during install, the DSDRIVER install will be stopped with message DBI20006E Installing the IBM Data Server Driver Package failed because the installer could not determine whether to install a new copy or to upgrade an existing copy because no copy name was specified.
      - If a copy name is provided during install the DSDRIVER will be installed to that copy, whether it is an existing copy or a new one.

### Note:

- The installer will handle the case where the release level of the existing copy is higher than the current.
2. Optional: If you have installed a IBM Version 10.1, or IBM Version 9.7 Data Server Client copy, you can use this existing Data Server Client copy to configure the DB2 Version 10.5 DSDRIVER copy by issuing the following command:  

```
db2dsdcfgfill [-i instance-name | -p instance-directory] [-o output-dir]
```
  3. If you want your applications to use the DB2 Version 10.5 DSDRIVER copy through the default interface, set the DB2 Version 10.5 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*.

### **What to do next**

After upgrading your IBM Data Server Driver Package, perform only the post-upgrade tasks for DB2 clients that apply. Refer to Chapter 20, “Post-upgrade tasks for clients,” on page 135.

---

## 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.

### Procedure

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

1. 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 10.5 that can impact the behavior of your application.

For more details, see “Managing DB2 server behavior changes” on page 101.

2. Verify that upgrading your clients was successful.

For more details, see “Verifying your client upgrade.”

---

## 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.

### Before you begin

- 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 10.5
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 before 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 139
- Chapter 22, "Upgrade essentials for database applications," on page 141
- Chapter 23, "Upgrade essentials for routines," on page 149
- Chapter 24, "Pre-upgrade tasks for database applications and routines," on page 151
- Chapter 25, "Upgrading database applications," on page 153
- Chapter 26, "Upgrading routines," on page 161
- Chapter 27, "Post-upgrade tasks for database applications and routines," on page 167
- Chapter 28, "Adopting new Version 10.5 functionality in database applications and routines," on page 169



---

## Chapter 21. Database applications and routines upgrade

Upgrading to DB2 Version 10.5 involves the upgrade of your database applications and routines if changes in DB2 Version 10.5 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 10.5 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 10.5, you should:
  - Review upgrade essentials for database applications to identify any changes in DB2 Version 10.5 that can impact your applications.
  - Review upgrade essentials for routines to identify any changes in DB2 Version 10.5 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 10.5 testing environment.
- Verify that your applications and routines perform as expected in your DB2 Version 10.5 production environment before deploying them.

If your applications and routines use any functionality that is deprecated in DB2 Version 10.5, 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 10.5 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 Version 10.5.

### 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 Version 10.5.

In UNIX operating systems, only 64-bit kernels are supported. Your 32-bit instances are upgraded to Version 10.5 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 Version 10.5 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 or later methods or JDBC 3.0 or earlier methods. To manage the behavioral differences between the driver that supports JDBC 4.0 or later 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 157 for details.

The DB2 JDBC Type 2 driver is discontinued in Version 10.1. 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 version of IBM Data Server Driver for JDBC and SQLJ that support JDBC 3.0 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* for details.

See “DB2 for Linux, UNIX, and Windows and IBM Data Server Driver for JDBC and SQLJ levels” 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. See “JDBC differences between versions of the IBM Data Server Driver for JDBC and SQLJ” for details about the differences between those drivers.

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.

### **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 Version 10.5:

- DB2 API functions
- DB2 command line processor (CLP) and system commands

### **SQL statements**

Review the changes to SQL statements in Version 10.5 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 built-in administrative routines and views**

After database upgrade to Version 10.5, the system catalog views under the SYSCAT schema remain compatible with catalog views that you defined in previous releases. 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 built-in administrative routines and views. In addition, all of the built-in table functions with names that start with SNAPSHOT\_ are discontinued.

Review the following topics to determine if you have applications and scripts that are impacted by changes to system catalog views and built-in administrative routines and views:

- Some administrative routines are discontinued
- System catalog
- “Deprecated built-in administrative routines and their replacement routines or views” in *Administrative Routines and Views*

### **Optimizer and query execution plans**

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.

### 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 23 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 10.5 clients to access databases in Version 10.5 servers. However, your applications are restricted to the functionality available for that client. Review Chapter 14, “Upgrade essentials for clients,” on page 119 to learn details about client connectivity and to identify changes in support that can impact your DB2 clients.

### Upgrade of applications from DB2 Version 9.7

If you are upgrading from DB2 Version 9.7, review changes in application driver support, 32-bit and 64-bit DB2 server support, and discontinued functionality between pre-Version 10.5 releases that might also impact your applications and scripts:

- Changes between DB2 Version 10.1 and DB2 Version 9.7 that impact applications.

---

## Upgrade impact from DB2 API changes

The changes in Version 10.5 to DB2 APIs can impact your existing applications after you upgrade to Version 10.5.

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 17. Changes to DB2 APIs*

| DB2 API            | Summary of changes with upgrade impact                                                                                                                                                                                    |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| db2DatabaseUpgrade | <p>The COBOL and FORTRAN language support for the db2DatabaseUpgrade API is deprecated.</p> <p>For more information, see COBOL and FORTRAN language support for the db2DatabaseUpgrade API is deprecated for details.</p> |

---

## Upgrade impact from DB2 command changes

The changes in Version 10.5 to DB2 command line processor (CLP) and system commands can impact your existing applications and scripts after you upgrade to Version 10.5.

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 18. Changes to DB2 CLP and system commands*

| Command                                                    | Summary of changes with upgrade impact                                                                                                                                                                                                                                                                                |
|------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>db2cat</b> ,<br><b>db2exfmt</b> ,<br><b>db2expln</b>    | The output for the <b>db2cat</b> , <b>db2exfmt</b> <b>db2expln</b> commands now display information for random ordering of index keys.<br><br>For more information, see DB2 command and SQL statement changes summary for details.                                                                                    |
| <b>db2pd</b>                                               | The <b>-apinfo</b> parameter now displays additional information about current and past activities of the UOW.<br><br>For more information, see DB2 command and SQL statement changes summary for details.                                                                                                            |
| <b>db2look</b>                                             | The <b>db2look</b> command now generates DDL statements to create column-organized tables in addition to row-organized tables.<br><br>For more information, see DB2 command and SQL statement changes summary for details.                                                                                            |
| <b>db2support</b>                                          | The <b>-d</b> parameter now supports collection of information from multiple databases. To specify multiple databases, separate the database names with a comma.<br><br>For more information, see DB2 command and SQL statement changes summary for details.                                                          |
| <b>db2IdentifyType1</b>                                    | The <b>db2IdentifyType1</b> command is discontinued.<br><br>For more information, see DB2 command and SQL statement changes summary for details.                                                                                                                                                                      |
| <b>LOAD</b>                                                | For column-organized tables, automatic statistics collection occurs by default during the <b>LOAD REPLACE</b> command. To explicitly disable automatic statistics collection, specify the <b>STATISTICS NO</b> parameter.<br><br>For more information, see DB2 command and SQL statement changes summary for details. |
| <b>STATISTICS YES</b> parameter of the <b>LOAD</b> command | The <b>STATISTICS YES</b> parameter of the <b>LOAD</b> command is discontinued.<br><br>For more information, see DB2 command and SQL statement changes summary for details.                                                                                                                                           |



---

## Upgrade impact from SQL statement changes

The changes to SQL statements in Version 10.5 can impact your existing applications and scripts after you upgrade to Version 10.5.

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

*Table 19. Changes to SQL statements*

| SQL statement          | Summary of changes with upgrade impact                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CREATE TABLE statement | <p>Because the default table organization that is specified by the <b>dft_table_org</b> database configuration parameter is ROW, there is no impact to the upgrade. After the upgrade, if you change the default organization to COLUMN or set the <b>DB2_WORKLOAD</b> registry variable to ANALYTICS before creating the database, scripts or applications that use the CREATE TABLE statement without the ORGANIZE BY COLUMN and ORGANIZE BY ROW clauses create tables with column table organization. Ensure to include the explicit clause to indicate the table organization as a good practice.</p> <p>For more information, see DB2 command and SQL statement changes summary for details.</p> |

Refer to the *SQL Reference Volume 2* guide for details about any of the statements.

---

## Upgrade impact from system catalog changes

In Version 10.5, system catalog objects are modified to support new functionality. These changes can impact your existing applications and scripts after you upgrade to Version 10.5.

### System catalog views

The following system catalog views have changed in Version 10.5. Most modifications to catalog views consist of new columns, changed descriptions, changed column data types, and increased column lengths.

- SYSCAT.ATTRIBUTES catalog view
- SYSCAT.CHECKS catalog view
- SYSCAT.COLUMNS catalog view
- SYSCAT.CONTROLS catalog view
- SYSCAT.DATATYPES catalog view
- SYSCAT.INDEXCOLUSE catalog view
- SYSCAT.INDEXES catalog view
- SYSCAT.PACKAGES catalog view
- SYSCAT.ROUTINEPARMS catalog view
- SYSCAT.ROUTINES catalog view
- SYSCAT.ROWFIELDS catalog view
- SYSCAT.SERVICECLASSES catalog view
- SYSCAT.STOGROUPS catalog view
- SYSCAT.TABDEP catalog view

- SYSCAT.TABLES has a new column called TABLEORG to indicate the table organization
- SYSCAT.TABLESPACES catalog view
- SYSCAT.TRIGGERS catalog view
- SYSCAT.VARIABLES catalog view
- SYSCAT.VIEWS catalog view
- SYSSTAT.COLUMNS catalog view
- SYSSTAT.INDEXES catalog view
- SYSSTAT.TABLES catalog view

For more information, see .Some system catalog views, built-in functions and global variables, built-in administrative routines and views have been added and changed for details.

## **Built-in administrative views and routines**

The following administrative views and routines have changed in Version 10.5. Most modifications consist of new columns, new values, changed column data types, and increased column lengths:

- ADMINTABINFO administrative view and ADMIN\_GET\_TAB\_INFO table function
- ENV\_SYS\_INFO administrative view
- MON\_BP\_UTILIZATION administrative view
- MON\_CONNECTION\_SUMMARY administrative view
- MON\_CURRENT\_SQL administrative view
- MON\_DB\_SUMMARY administrative view
- MON\_FORMAT\_XML\_COMPONENT\_TIMES\_BY\_ROW table function
- MON\_FORMAT\_XML\_METRICS\_BY\_ROW table function
- MON\_FORMAT\_XML\_TIMES\_BY\_ROW table function
- MON\_GET\_APPL\_LOCKWAIT table function
- MON\_GET\_BUFFERPOOL table function
- MON\_GET\_CONNECTION table function
- MON\_GET\_CONNECTION\_DETAILS table function
- MON\_GET\_PKG\_CACHE\_STMT table function
- MON\_GET\_SERVICE\_SUBCLASS table function
- MON\_GET\_SERVERLIST table function
- MON\_GET\_TABLE table function
- MON\_GET\_TABLESPACE table function
- MON\_GET\_TABLE\_USAGE\_LIST table function
- MON\_TBSP\_UTILIZATION administrative view
- MON\_GET\_UNIT\_OF\_WORK
- MON\_GET\_UNIT\_OF\_WORK\_DETAILS
- MON\_GET\_WORKLOAD table function
- MON\_GET\_WORKLOAD\_DETAILS table function
- MON\_WORKLOAD\_SUMMARY administrative view

For more information, see .Some system catalog views, built-in functions and global variables, built-in administrative routines and views have been added and changed for details.

In addition, all of the administrative routines with names that start with SNAPSHOT have been deprecated since DB2 Version 9.1. For more information, see . Some system catalog views, built-in functions and global variables, built-in administrative routines and views have been added and changed for details.

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 9.7, the following additional system catalog changes between pre-Version 10.5 releases can also impact your applications and scripts:

- System catalog changes between DB2 Version 10.1 and DB2 Version 9.7.



---

## 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 141 could also impact your routines.

### Development software support

The information about development software support in Chapter 22, “Upgrade essentials for database applications,” on page 141 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.

### Unfenced external routines

During database upgrade to DB2 Version 10.5 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 162 for details about how to manage this change.

### 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

The IBM Software Developer's Kit (SDK) for Java 1.4.2 is deprecated and might be discontinued in a future release.

For DB2 Version 9.7 or later, the default JDBC driver to run JDBC routines is the IBM Data Server Driver for JDBC and SQLJ. See “Upgrading Java routines” on page 164 for details on how to manage this change.

---

## 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.

### Procedure

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.  
For more details, see Chapter 22, “Upgrade essentials for database applications,” on page 141.
2. Review upgrade essentials for routines to determine which changes might impact your routines.  
For more details, see Chapter 23, “Upgrade essentials for routines,” on page 149.
3. Plan your upgrade strategy.  
For more details, see Chapter 2, “Planning your DB2 environment upgrade,” on page 5.
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 DB2 Version 10.5 application driver if your application requires one.  
For more details, see Chapter 13, “Clients upgrade,” on page 117.  
Although DB2 Version 10.5 server provides connectivity support for earlier clients, using a DB2 Version 10.5 client eliminates any limitations and incompatibilities between releases.
8. Test your database applications in a DB2 Version 10.5 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.  
For more details, see “Upgrading DB2 servers in a test environment” on page 46 and Chapter 25, “Upgrading database applications,” on page 153.
9. Test your routines in a DB2 Version 10.5 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.  
For more details, see “Upgrading DB2 servers in a test environment” on page 46 and Chapter 26, “Upgrading routines,” on page 161.





---

## Chapter 25. Upgrading database applications

Upgrading your existing database applications to DB2 Version 10.5 involves managing the changes between DB2 Version 10.5 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 10.5 that impact your applications, to remove the use of deprecated or discontinued functionality in DB2 Version 10.5, or to use new functionality.

### Before you begin

- Ensure that you have access to a DB2 Version 10.5 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. See Chapter 24, “Pre-upgrade tasks for database applications and routines,” on page 151.

### Restrictions

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 10.5:

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.

See “Upgrade impact from DB2 command changes” on page 143, “Upgrade impact from SQL statement changes” on page 145, and “Upgrade impact from system catalog changes” on page 145.

2. If you identified changes specific to the development environment that impact your applications, modify them to support these changes. See Chapter 22, “Upgrade essentials for database applications,” on page 141. Upgrade your:

- Embedded SQL applications. See “Upgrading embedded SQL applications.”
  - CLI applications. See “Upgrading CLI applications” on page 155.
  - Java applications that use the IBM Data Server Driver for JDBC and SQLJ. See “Upgrading Java applications that use IBM Data Server Driver for JDBC and SQLJ” on page 157.
  - ADO and .NET applications. See “Upgrading ADO.NET applications” on page 158.
  - Scripts that use DB2 CLP commands and SQL statements. See “Upgrading scripts” on page 159.
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.
  4. Test your database applications to verify your changes and to ensure that they run as expected using DB2 Version 10.5.

## What to do next

After upgrading your database applications, perform the recommended post-upgrade tasks for database applications to ensure that your upgrade was successful. See Chapter 27, “Post-upgrade tasks for database applications and routines,” on page 167.

---

## Upgrading embedded SQL applications

Upgrading your existing embedded SQL applications to DB2 Version 10.5 involves managing the changes between DB2 Version 10.5 and previous releases that impact these applications and verifying that these applications function as expected.

### Before you begin

- Ensure that you have access to a DB2 Version 10.5 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. See Chapter 25, “Upgrading database applications,” on page 153.

### Restrictions

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 10.5:

1. If you modified the library path environment variables, ensure that those variables include the correct DB2 shared library path for your applications . 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 10.5 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. 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.
    - 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.

## What to do next

After upgrading your embedded SQL applications, perform the remaining steps in the upgrading database applications task. See Chapter 25, “Upgrading database applications,” on page 153.

---

## Upgrading CLI applications

Upgrading your existing CLI applications to DB2 Version 10.5 involves managing the changes between DB2 Version 10.5 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.

### Before you begin

- Ensure that you have access to a DB2 Version 10.5 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. For details, see “C and C++ development software”.
- Perform previous steps in the Chapter 25, “Upgrading database applications,” on page 153 task.

### Restrictions

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 10.5:

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 Chapter 22, “Upgrade essentials for database applications,” on page 141. 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 10.5 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 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 Chapter 22, “Upgrade essentials for database applications,” on page 141. 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 Chapter 22, “Upgrade essentials for database applications,” on page 141.
  - If you do not have the original source code available, run the **db2chglibpath** 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.

## What to do next

After upgrading your CLI applications, perform the remaining steps in the Chapter 25, “Upgrading database applications,” on page 153 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 10.5 that can impact these applications.

### Before you begin

- Review the upgrade essentials for applications to identify key changes that might impact your Java database applications. See Chapter 22, “Upgrade essentials for database applications,” on page 141.
- Ensure that you have access to a DB2 Version 10.5 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. See Chapter 25, “Upgrading database applications,” on page 153.

### Restrictions

- 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 10.5:

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 4.13 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 3.63 or later
2. Adjust your applications to manage the differences between the current version of the IBM Data Server Driver for JDBC and SQLJ and the previous versions.
3. 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

## Results

Upon completion of this task, your Java application should perform successfully using DB2 Version 10.5.

## What to do next

After upgrading your Java applications, perform the remaining steps in the upgrading database applications task. See Chapter 25, “Upgrading database applications,” on page 153.

---

## Upgrading ADO.NET applications

Upgrading your existing ADO.NET applications to DB2 Version 10.5 involves managing the changes between DB2 Version 10.5 and previous releases that impact these applications and verifying that these applications function as expected.

### Before you begin

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 10.5. 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.
- Ensure that you have access to a DB2 Version 10.5 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 Chapter 25, “Upgrading database applications,” on page 153 task.

### Procedure

To upgrade your ADO.NET applications to DB2 Version 10.5:

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.

## What to do next

After upgrading your ADO.NET applications, perform the remaining steps in the Chapter 25, “Upgrading database applications,” on page 153 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 10.5 and previous releases related to SQL statements, DB2 CLP and system commands, SQL Administrative views and routines, built-in functions, and catalog views.

### Before you begin

- Ensure that you have access to a DB2 Version 10.5 server, including instances and databases.
- Ensure that a DB2 Version 10.5 client is installed.
- Perform the previous steps in the upgrading database applications task.

### Restrictions

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 10.5:

1. Run your scripts to detect any incompatibilities with DB2 Version 10.5. 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 10.5 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 in 24
  - Deprecated and discontinued database manager configuration parameters in 25
  - Deprecated and discontinued database configuration parameters in 26
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. See “Upgrade impact from system catalog changes” on page 145. 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 10.5.

### **What to do next**

After upgrading your scripts, perform the remaining steps in the upgrading database applications task. See Chapter 25, “Upgrading database applications,” on page 153.



---

## Chapter 26. Upgrading routines

Upgrading your existing routines to DB2 Version 10.5 involves managing the changes between DB2 Version 10.5 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, re-create your external routines in the database, and re-create SQL routines.

Test your routines in a DB2 Version 10.5 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 10.5, or to use new functionality.

### Before you begin

- Review upgrade essentials for routines to identify any changes that apply to your routines. See Chapter 23, “Upgrade essentials for routines,” on page 149.
- Ensure that you have access to upgraded DB2 Version 10.5 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. See Chapter 24, “Pre-upgrade tasks for database applications and routines,” on page 151.
- 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*.

### Restrictions

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 10.5 databases:

1. If you identified changes in DB2 Version 10.5 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. See “Upgrading C, C++, and COBOL routines” on page 162.
  - Java routines. See “Upgrading Java routines” on page 164.

- .NET CLR routines. See “Upgrading .NET CLR routines” on page 165.
- 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 10.5.

## What to do next

After upgrading your routines, perform the recommended post-upgrade tasks for routines. See Chapter 27, “Post-upgrade tasks for database applications and routines,” on page 167.

---

## Upgrading C, C++, and COBOL routines

Upgrading your existing C, C++, or COBOL routines to DB2 Version 10.5 involves managing the changes between DB2 Version 10.5 and previous releases that impact these routines and verifying that they function as expected.

### Before you begin

- Ensure that you have access to a DB2 Version 10.5 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. See Chapter 26, “Upgrading routines,” on page 161.

### Restrictions

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 10.5, do the following:

1. If you upgraded to a DB2 Version 10.5 64-bit instance, change your routine libraries or routine definitions according to the following table:

Table 20. Upgrading C, C++, and COBOL routines to a DB2 Version 10.5 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 10.5 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"> <li>Rebuild the routine source code into a 64-bit library using the DB2 Version 10.5 bldrtn scripts and redeploy the library to the DB2 server.</li> <li>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.</li> </ul>                                                                                                                                                                                                                                                                                                                          |

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 154 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. See “Rebinding packages in upgraded databases” on page 103.
- 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 .

## What to do next

After upgrading your C, C++, or COBOL routines, perform the remaining steps in the upgrading routines task. See Chapter 26, “Upgrading routines,” on page 161.

---

## Upgrading Java routines

Upgrading your existing Java routines to DB2 Version 10.5 involves managing the changes between DB2 Version 10.5 and previous releases that impact these routines and ensure that these routines function as expected.

### Before you begin

The following prerequisites must be met to perform this task:

- Ensure that you have access to a DB2 Version 10.5 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 Chapter 23, “Upgrade essentials for routines,” on page 149 which are the installation path of SDK for Java 7.

If you must use an SDK for Java other than the one installed in your DB2 Version 10.5 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. Test your Java routines in your DB2 Version 10.5 database. If testing is successful and your Java routine perform as expected, you do not need to perform any additional steps.
3. If you 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 157 to learn how to manage those differences.
4. If the pre-upgrade value of the **jdk\_path** parameter was the installation path of SDK for Java 6 or Java 1.4.2, manage any differences in behavior between the SDK specified by the **jdk\_path** parameter and the SDK for Java 7.
5. 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.

6. 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.
7. If you created projects in the Development Center to develop your Java routines, upgrade any existing projects to the IBM Data Studio using the upgrade wizard.

## What to do next

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 10.5 and previous releases that impact these routines and verifying that they function as expected.

### Before you begin

- Review the Chapter 23, “Upgrade essentials for routines,” on page 149 to identify key changes that might apply to your .NET CLR routines.
- Ensure that you have access to a DB2 Version 10.5 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 Chapter 26, “Upgrading routines,” on page 161 task.

### Procedure

To upgrade your .NET CLR routines to DB2 Version 10.5:

1. Connect to the DB2 Version 10.5 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-DB2 Version 10.5 32-bit instance to DB2 Version 10.5 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 10.5.

## **What to do next**

After upgrading your .NET CLR routines, perform the remaining steps in the Chapter 26, “Upgrading routines,” on page 161 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.

### Procedure

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 10.5 in your database applications and routines before that functionality becomes discontinued.

For more details, see “Deprecated or discontinued functionality that affects DB2 server upgrades” on page 27.

5. Adopt new DB2 Version 10.5 functionality in database applications, where appropriate, to improve performance or add new functionality. Check the Sample files to understand how the new functionality works.

For more details, see Chapter 28, “Adopting new Version 10.5 functionality in database applications and routines,” on page 169.





---

## Chapter 28. Adopting new Version 10.5 functionality in database applications and routines

After upgrading to Version 10.5, enhance the functionality and improve the performance of your database applications by adopting new Version 10.5 functionality.

### Before you begin

You must upgrade your DB2 server to Version 10.5.

### Procedure

For applications that access upgraded databases, perform any of the following steps to adopt the specified Version 10.5 functionality:

- Enable the batch CALL statement support in CLI applications to optimize network flow by specifying an array size with the `SQL_ATTR_PARAMSET_SIZE` statement attribute and provide argument data in form of an array. For more details, see *Calling stored procedures from CLI applications*.
- Use NOT ENFORCED primary key and unique constraints to avoid performance costs during insert, update, and delete operations on a table and space requirements that are associated with enforcing a unique constraint when it is known that the data already conforms to the unique constraint. It also provides the same potential performance benefits for queries. For more details, see *Informational constraints*.

### What to do next

If you upgraded from DB2 Version 9.7, adopt functionality introduced in DB2 Version 9.7 in your database applications and routines. See *Adopting new DB2 Version 9.7 functionality in database applications and routines* in the *Upgrading to DB2 Version 9.7* guide for details.



---

## **Part 5. Appendixes**



---

## Appendix A. Important references

The following list of references can help you with the upgrade of your DB2 database environment.

### DB2 operating system requirements Web page

You can find the operating system and hardware requirements for DB2 Version 10.5 installation in “Installation requirements for DB2 database products” in *Installing DB2 Servers*.

### DB2 Information Center

You can find the information in this book in the online *DB2 Information Center* at . Refer to the “Upgrading” topic under the “Database fundamentals” section. The title for the most high level topic is “Upgrading to DB2 Version 10.5”. 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 10.5 manuals in PDF format

DB2 DB2 Version 10.5 manuals in PDF format are available for complimentary download at [www.ibm.com/support/docview.wss?rs=71&uid=swg27009474](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 [www.ibm.com/software/data/db2/upgrade/portal](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 [www.ibm.com/software/data/education/](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 [www.ibm.com/software/data/education/selfstudy.html](http://www.ibm.com/software/data/education/selfstudy.html).

### developerWorks Information Management Web site

The developerWorks Information Management Web site at [www.ibm.com/developerworks/data](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 database forums

The DB2 database forums are places to exchange ideas and share solutions with your peers in the IBM DB2 database product community. In addition, DB2 database forums include forums that are mirrors to DB2 database newsgroups, such as the `ibm.software.db2.udb` and `ibm.software.db2.udb.beta` newsgroups. The DB2 database forums are hosted by developerWorks at [www.ibm.com/developerworks/forums/db2\\_forums.jsp](http://www.ibm.com/developerworks/forums/db2_forums.jsp).

---

## Appendix B. Overview of the DB2 technical information

DB2 technical information is available in multiple formats that can be accessed in multiple ways.

DB2 technical information is available through the following tools and methods:

- DB2 Information Center
  - Topics (Task, concept and reference topics)
  - 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](http://ibm.com).

You can access additional DB2 technical information such as technotes, white papers, and IBM Redbooks® publications online at [ibm.com](http://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 email to [db2docs@ca.ibm.com](mailto: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 email 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/e-business/linkweb/publications/servlet/pbi.wss](http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss). English and translated DB2 Version 10.1 manuals in PDF format can be downloaded from [www.ibm.com/support/docview.wss?rs=71&uid=swg27009474](http://www.ibm.com/support/docview.wss?rs=71&uid=swg27009474).

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 21. DB2 technical information*

| <b>Name</b>                                                         | <b>Form Number</b> | <b>Available in print</b> | <b>Availability date</b> |
|---------------------------------------------------------------------|--------------------|---------------------------|--------------------------|
| <i>Administrative API Reference</i>                                 | SC27-5506-00       | Yes                       | July 28, 2013            |
| <i>Administrative Routines and Views</i>                            | SC27-5507-00       | No                        | July 28, 2013            |
| <i>Call Level Interface Guide and Reference Volume 1</i>            | SC27-5511-00       | Yes                       | July 28, 2013            |
| <i>Call Level Interface Guide and Reference Volume 2</i>            | SC27-5512-00       | Yes                       | July 28, 2013            |
| <i>Command Reference</i>                                            | SC27-5508-00       | Yes                       | July 28, 2013            |
| <i>Database Administration Concepts and Configuration Reference</i> | SC27-4546-00       | Yes                       | July 28, 2013            |
| <i>Data Movement Utilities Guide and Reference</i>                  | SC27-5528-00       | Yes                       | July 28, 2013            |
| <i>Database Monitoring Guide and Reference</i>                      | SC27-4547-00       | Yes                       | July 28, 2013            |
| <i>Data Recovery and High Availability Guide and Reference</i>      | SC27-5529-00       | Yes                       | July 28, 2013            |
| <i>Database Security Guide</i>                                      | SC27-5530-00       | Yes                       | July 28, 2013            |
| <i>DB2 Workload Management Guide and Reference</i>                  | SC27-5520-00       | Yes                       | July 28, 2013            |
| <i>Developing ADO.NET and OLE DB Applications</i>                   | SC27-4549-00       | Yes                       | July 28, 2013            |
| <i>Developing Embedded SQL Applications</i>                         | SC27-4550-00       | Yes                       | July 28, 2013            |
| <i>Developing Java Applications</i>                                 | SC27-5503-00       | Yes                       | July 28, 2013            |
| <i>Developing Perl, PHP, Python, and Ruby on Rails Applications</i> | SC27-5504-00       | No                        | July 28, 2013            |
| <i>Developing RDF Applications for IBM Data Servers</i>             | SC27-5505-00       | Yes                       | July 28, 2013            |
| <i>Developing User-defined Routines (SQL and External)</i>          | SC27-5501-00       | Yes                       | July 28, 2013            |
| <i>Getting Started with Database Application Development</i>        | GI13-2084-00       | Yes                       | July 28, 2013            |



*Table 21. DB2 technical information (continued)*

| <b>Name</b>                                                                          | <b>Form Number</b> | <b>Available in print</b> | <b>Availability date</b> |
|--------------------------------------------------------------------------------------|--------------------|---------------------------|--------------------------|
| <i>Getting Started with DB2 Installation and Administration on Linux and Windows</i> | GI13-2085-00       | Yes                       | July 28, 2013            |
| <i>Globalization Guide</i>                                                           | SC27-5531-00       | Yes                       | July 28, 2013            |
| <i>Installing DB2 Servers</i>                                                        | GC27-5514-00       | Yes                       | July 28, 2013            |
| <i>Installing IBM Data Server Clients</i>                                            | GC27-5515-00       | No                        | July 28, 2013            |
| <i>Message Reference Volume 1</i>                                                    | SC27-5523-00       | No                        | July 28, 2013            |
| <i>Message Reference Volume 2</i>                                                    | SC27-5524-00       | No                        | July 28, 2013            |
| <i>Net Search Extender Administration and User's Guide</i>                           | SC27-5526-00       | No                        | July 28, 2013            |
| <i>Partitioning and Clustering Guide</i>                                             | SC27-5532-00       | Yes                       | July 28, 2013            |
| <i>pureXML Guide</i>                                                                 | SC27-5521-00       | Yes                       | July 28, 2013            |
| <i>Spatial Extender User's Guide and Reference</i>                                   | SC27-5525-00       | No                        | July 28, 2013            |
| <i>SQL Procedural Languages: Application Enablement and Support</i>                  | SC27-5502-00       | Yes                       | July 28, 2013            |
| <i>SQL Reference Volume 1</i>                                                        | SC27-5509-00       | Yes                       | July 28, 2013            |
| <i>SQL Reference Volume 2</i>                                                        | SC27-5510-00       | Yes                       | July 28, 2013            |
| <i>Text Search Guide</i>                                                             | SC27-5527-00       | Yes                       | July 28, 2013            |
| <i>Troubleshooting and Tuning Database Performance</i>                               | SC27-4548-00       | Yes                       | July 28, 2013            |
| <i>Upgrading to DB2 Version 10.5</i>                                                 | SC27-5513-00       | Yes                       | July 28, 2013            |
| <i>What's New for DB2 Version 10.5</i>                                               | SC27-5519-00       | Yes                       | July 28, 2013            |
| <i>XQuery Reference</i>                                                              | SC27-5522-00       | No                        | July 28, 2013            |

*Table 22. DB2 Connect-specific technical information*

| <b>Name</b>                                                                | <b>Form Number</b> | <b>Available in print</b> | <b>Availability date</b> |
|----------------------------------------------------------------------------|--------------------|---------------------------|--------------------------|
| <i>DB2 Connect Installing and Configuring DB2 Connect Personal Edition</i> | SC27-5516-00       | Yes                       | July 28, 2013            |
| <i>DB2 Connect Installing and Configuring DB2 Connect Servers</i>          | SC27-5517-00       | Yes                       | July 28, 2013            |
| <i>DB2 Connect User's Guide</i>                                            | SC27-5518-00       | Yes                       | July 28, 2013            |

---

## 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.

### Procedure

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

Documentation for other versions of DB2 products is found in separate information centers on [ibm.com](http://ibm.com)<sup>®</sup>.

### About this task

For DB2 Version 10.1 topics, the *DB2 Information Center* URL is <http://pic.dhe.ibm.com/infocenter/db2luw/v10r1>.

For DB2 Version 9.8 topics, the *DB2 Information Center* URL is <http://pic.dhe.ibm.com/infocenter/db2luw/v9r8/>.

For DB2 Version 9.7 topics, the *DB2 Information Center* URL is <http://pic.dhe.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>.

---

## Terms and conditions

Permissions for the use of these publications are granted subject to the following terms and conditions.

**Applicability:** These terms and conditions are in addition to any terms of use for the IBM website.

**Personal use:** You may reproduce these publications for your personal, noncommercial 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.

**Rights:** 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 previous 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.

**IBM 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 [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml)



---

## 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 about 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.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, 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, changes, or both in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to websites not owned by IBM are provided for convenience only and do not in any manner serve as an endorsement of those

websites. The materials at those websites are not part of the materials for this IBM product and use of those websites 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  
U59/3600  
3600 Steeles Avenue East  
Markham, Ontario L3R 9Z7  
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](http://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](http://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 or registered trademarks of Oracle, its affiliates, 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, Celeron, 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 165

## Numerics

32-bit servers  
upgrading to 64-bit systems 73

## A

ACTIVATE DATABASE command  
post-upgrade tasks for DB2 servers 101  
ADO.NET applications  
upgrading 158  
applications  
post-upgrade tasks  
new functionality adoption 169  
overview 167  
removing deprecated functionality 167  
tuning 167  
pre-upgrade tasks 151  
upgrade impact  
built-in administrative routine and view changes 145  
built-in routine changes 145  
catalog view changes 145  
DB2 API changes 143  
DB2 command changes 144  
SQL statement changes 145  
upgrading  
planning 11, 141  
process 139, 153  
automatic storage databases  
upgraded databases 107

## B

BACKUP DATABASE command  
upgrade tasks for DB2 servers 39  
backups  
client configuration 123  
databases  
upgrade tasks for DB2 servers 39  
DB2 server configuration 40  
built-in administrative routines  
upgrade impact 145  
built-in administrative views  
upgrade impact 145  
built-in routines  
upgrade impact 145  
built-in views  
upgrade impact 145

## C

catalog views  
upgrade impact 145

CLI  
applications  
upgrading 155  
clients  
post-upgrade tasks  
managing server changes 135  
overview 135  
verifying upgrade 135  
pre-upgrade tasks  
backing up configuration 123  
overview 123  
reviewing upgrade essentials 123  
upgrading DB2 servers 123  
upgrading in test environments 124  
upgrading  
best practices 121  
Data Server Client (Windows) 127  
Data Server Runtime Client (Windows) 129  
Linux 131  
overview 117, 119  
planning 10  
UNIX 131  
command line processor (CLP)  
scripts  
upgrade impact 144  
upgrading 159  
commands  
dasmigr  
upgrading DAS 53, 64  
db2ckupgrade  
pre-upgrade tasks for DB2 servers 36  
db2exmig  
post-upgrade tasks for DB2 servers 104  
db2iupgrade  
failure causes 21  
overview 19  
upgrading instances 50, 62  
upgrading pureScale instances 85  
db2tdbmgr  
upgrading DAS 53, 64  
deprecated  
upgrade impact 27  
discontinued  
upgrade impact 27  
UPGRADE DATABASE  
upgraded database entities 19  
upgrading databases 54, 65, 86  
configuration  
backups  
clients 123  
pre-upgrade tasks for DB2 servers 40  
configuration parameters  
saving settings before upgrading DB2 servers 40  
upgrade impact 23, 101  
Control Center  
discontinued tools 27

## D

dasmigr command  
upgrading DAS 53, 64

- database applications
  - adopting new functionality 169
  - upgrading
    - impact of release changes 141
    - process 139, 153
- databases
  - duplicating to test DB2 server upgrade 47
  - impact of physical design characteristic changes on upgrade 23
  - new functionality adoption after upgrade 107
  - pre-upgrade tasks 36
  - upgrading
    - procedure 54, 65, 86
- DB2 administration server (DAS)
  - upgrading 53, 64
- DB2 Governor
  - migrating to DB2 workload manager 109
- DB2 Information Center
  - versions 178
- DB2 pureScale environments
  - upgrading
    - DB2 servers 83
- DB2 pureScale instances
  - upgrading 85
- DB2 servers
  - changes
    - post-upgrade tasks for clients 135
    - summary 23
  - falling back to a previous release 113
  - post-upgrade tasks
    - activating databases 101
    - activating services 101
    - adjusting log space 100
    - managing server changes 101
    - overview 97
    - rebinding packages 103
    - upgrading explain tables 104
    - verifying upgrade 104
  - pre-upgrade tasks
    - backing up configuration 40
    - backing up databases 39
    - changing raw devices to block devices (Linux) 44
    - gathering diagnostic information 45
    - increasing log space 42
    - increasing table space sizes 42
    - overview 35
    - taking servers offline 48
    - upgrading test environments 46
    - verifying databases 36
  - pureScale 57, 69
  - reversing upgrade 113
  - upgrade impact
    - behavior changes 23
    - deprecated functionality 27
    - discontinued functionality 27
    - registry variables 23
  - upgrade path
    - planning 6
  - upgrading 57, 69
    - 32-bit to 64-bit 73
    - best practices 30
    - databases 54, 65, 86
    - DB2 administration server (DAS) 53, 64
    - instances 50, 62
    - Linux 61
    - multiple DB2 copies 77
    - new server 78

- DB2 servers (*continued*)
  - upgrading (*continued*)
    - partitioned database environments 82
    - planning 7
    - process 17
    - pureScale 83
    - pureScale instances 85
    - support 19
    - UNIX 61
    - using online database backups 81
    - Windows 49
- DB2 Text Search
  - non-root upgrade 93
  - upgrading 90, 93, 94
- DB2 workload management
  - DB2 Governor
    - migrating 109
- DB2\_USE\_DB2JCCT2\_JROUTINE variable
  - upgrading Java routines 164
- db2batch command
  - verifying upgrade 104
- db2ckupgrade command
  - pre-upgrade tasks for DB2 servers 36
- db2exmig command
  - post-upgrade tasks for DB2 servers 104
- db2fodc command
  - pre-upgrade tasks for DB2 servers 45
- db2iupgrade command
  - failures 21
  - upgrading instances 19, 50, 62
  - upgrading pureScale instances 85
- db2rbind command
  - post-upgrade tasks for DB2 servers 103
- db2support command
  - diagnostic data collection 45
  - pre-upgrade tasks for DB2 servers 40, 45
- db2tdbmgr command
  - upgrading DAS 53, 64
- deprecated functionality
  - removing 167
  - upgrade impact 27
- Direct I/O (DIO)
  - changing raw devices to block devices (Linux) 44
- discontinued functionality
  - upgrade impact 27
- disk space
  - requirements 27
- documentation
  - overview 175
  - PDF files 175
  - printed 175
  - terms and conditions of use 178

## E

- embedded SQL applications
  - upgrading 154
- explain tables
  - upgrading 104

## F

- FORTTRAN language
  - applications
    - upgrading 154

## H

- help
  - SQL statements 178

## I

- IBM data server clients
  - IBM Data Server Client 127
- IBM Data Server Driver for JDBC and SQLJ
  - upgrading Java applications 157
- IBM Data Server Driver Package
  - upgrading 133
- IBM Data Server Runtime Client
  - upgrading (Windows) 129
- instances
  - 32-bit and 64-bit upgrade support 29
  - upgrading 21, 50, 62

## J

- Java
  - applications
    - upgrading (IBM Data Server Driver for JDBC and SQLJ) 157
  - routines
    - upgrading 164
- jdk\_path configuration parameter
  - routines
    - upgrading 164

## L

- Linux
  - changing raw devices to block devices 44
  - upgrading
    - clients 131
    - DB2 servers 61
    - non-root installations 74
- logs
  - space requirements
    - adjusting 100
    - increasing 42
    - upgrading DB2 servers 27

## M

- Microsoft Cluster Server (MSCS)
  - upgrading 95
- Microsoft SQL Server
  - migrating 33
- migration
  - applications
    - overview 139
  - clients 117
  - DB2 Governor to DB2 workload manager 109
  - DB2 servers 17
  - Microsoft SQL Server 33
  - non-DB2 relational databases 33
  - Oracle 33
  - overview 3
  - routines 139
  - Sybase 33
  - XML Extender to XML data store 109
- multiple DB2 copies
  - upgrading DB2 servers 77

## N

- non-root installations
  - upgrading 74
- notices 181

## O

- O\_DIRECT 44
- online backups
  - upgrading DB2 servers 81
- Oracle
  - migrating 33

## P

- partitioned database environments
  - upgrading 82
- partitioned indexes
  - upgraded databases 107
- partitioned tables
  - XML data
    - upgraded databases 107
- post-upgrade tasks
  - applications
    - adopting new functionality 169
    - removing deprecated functionality 167
    - tuning 167
  - clients
    - managing server changes 135
    - overview 135
    - verifying upgrade 135
  - databases
    - activating 101
    - activating services 101
    - log space adjustments 100
    - new functionality adoption 107
    - rebinding packages 103
  - DB2 servers
    - managing behavior changes 101
    - overview 97
    - upgrading explain tables 104
    - verifying upgrade 104
  - routines
    - new functionality adoption 169
    - removing deprecated functionality 167
    - tuning 167
- pre-upgrade tasks
  - applications 151
  - clients
    - backing up configuration 123
    - overview 123
    - upgrading in test environments 124
  - databases
    - backing up 39
    - verifying that databases are ready to upgrade 36
  - DB2 servers
    - backing up configuration 40
    - changing raw devices to block devices (Linux) 44
    - gathering diagnostic information 45
    - overview 35
    - taking servers offline 48
    - upgrading in test environments 46
  - log file sizes
    - increasing 42
  - table space sizes
    - increasing 42

## R

- raw devices
  - changing to block devices 44
- raw I/O
  - changing raw devices to block devices (Linux) 44
- raw logs
  - deprecated
  - upgrade impact 27
- REBIND command
  - post-upgrade tasks for DB2 servers 103
- rebinding
  - post-upgrade tasks for DB2 servers 103
- references
  - upgrades 173
- registry variables
  - saving settings before upgrading DB2 servers 40
  - upgrade impact 23
  - upgrading 101
- restore utility
  - upgrading DB2 servers 78
- REXX language
  - applications
    - embedded SQL (upgrading) 154
  - embedded SQL statements 154
- routines
  - planning upgrade 11
  - post-upgrade tasks
    - new functionality adoption 169
    - overview 167
    - removing deprecated functionality 167
    - tuning 167
  - pre-upgrade tasks 151
  - upgrading
    - .NET 165
    - C 162
    - C++ 162
    - COBOL 162
    - Java 164
    - overview 139, 161
    - support 149

## S

- scenarios
  - upgrading DB2 servers 73
- scripts
  - upgrade impact 141
  - upgrading 159
- SQL
  - administrative routines
    - upgrading 159
  - administrative views
    - upgrading 159
- SQL statements
  - help
    - displaying 178
  - upgrade impact 145
  - upgrading scripts 159
- statistical views
  - upgraded databases 107
- stored procedures
  - upgrade support 149
  - upgrading 161
- Sybase
  - migrating 33

- system catalogs
  - views
    - upgrade impact 145
- system commands
  - scripts
    - upgrade impact 144
    - upgrading 159

## T

- table spaces
  - requirements
    - upgrading DB2 servers 27
- terms and conditions
  - publications 178
- test environments
  - upgrading clients 124
  - upgrading DB2 servers
    - creating database duplicates 47
    - procedure 46
- tools catalog database
  - upgrading 53, 64
- tuning
  - applications 167
  - routines 167
- type-1 indexes
  - discontinued
    - upgrade impact 27

## U

- UNIX
  - upgrading
    - clients 131
    - DB2 servers 61
    - non-root installations 74
- UPGRADE DATABASE command
  - failures 21
  - upgraded database entities 19
  - upgrading databases 54, 65, 86
- upgrade paths
  - DB2 servers 6
- upgraded databases
  - new functionality adoption 107
- upgrades
  - .NET CLR routines 165
  - 32-bit servers 29
  - 64-bit servers 29
  - applications
    - ADO .NET 158
    - built-in administrative routine and view changes 145
    - built-in routine changes 145
    - C 154
    - catalog view changes 145
    - CLI 155
    - COBOL 154
    - DB2 API changes 143
    - DB2 command changes 144
    - embedded SQL 154
    - FORTTRAN 154
    - Java using IBM Data Server Driver for JDBC and SQLJ 157
    - overview 3, 139, 141
    - planning 11
    - post-upgrade tasks 167
    - pre-upgrade tasks 151

- upgrades (*continued*)
  - applications (*continued*)
    - procedure 153
    - REXX 154
    - SQL statement changes 145
  - best practices
    - clients 121
    - DB2 servers 30
  - C applications 154
  - C routines 162
  - C++ routines 162
  - clients
    - Linux 131
    - overview 3, 117, 119
    - planning 10
    - post-upgrade tasks 135
    - pre-upgrade tasks 123
    - test environments 124
    - UNIX 131
  - COBOL applications 154
  - COBOL routines 162
  - database applications 153
  - databases
    - duplicate databases for test environments 47
    - procedure 54, 65, 86
  - DB2 Administration Server (DAS) 53, 64
  - DB2 environments 3
  - DB2 pureScale environments
    - instances 85
  - DB2 pureScale instances 85
  - DB2 servers 57, 69
    - 32-bit to 64-bit servers 73
    - adjusting log space 100
    - best practices 30
    - complex environments 73
    - configuration parameter changes 23
    - configuration parameters 101
    - database physical characteristic changes 23
    - DB2 pureScale server 83
    - discontinued functionality 21
    - Linux 61
    - log space requirements 27
    - multiple DB2 copies 77
    - new 78
    - overview 3, 17, 19
    - partitioned database environments 82
    - performance 30
    - physical characteristics 101
    - planning 7
    - post-upgrade tasks 97
    - pre-upgrade tasks 35
    - registry variable changes 23
    - registry variables 101
    - restrictions 21
    - reversing 113
    - table space requirements 27
    - taking servers offline 48
    - test environments 46
    - UNIX 61
    - using online database backups 81
    - Windows 49
  - development software 151
  - explain tables 104
  - HADR 21
  - IBM Data Server Driver Package 133
  - instance type 21

- upgrades (*continued*)
  - instances
    - 32-bit upgrade support 29
    - 64-bit upgrade support 29
    - procedure 50, 62
  - Microsoft Cluster Server (MSCS) 95
  - non-root installations 74
  - operating systems 151
  - overview 3
  - planning
    - applications 11
    - clients 10
    - DB2 environments 5
    - DB2 servers 7
    - DB2 upgrade portal 5
    - routines 11
  - pureScale 57, 69
  - references 173
  - routines
    - C 162
    - C++ 162
    - COBOL 162
    - Java 164
    - overview 3, 139, 149
    - planning 11
    - post-upgrade tasks 167
    - pre-upgrade tasks 151
    - procedure 161
  - scripts
    - overview 141
    - procedure 159
  - SQL replication environments 30
  - tools catalog database 53, 64
  - Windows
    - IBM Data Server Client 127
    - IBM Data Server Runtime Client 129
  - upgrading to DB2 Version 10.1
    - details v
    - upgrading applications and routines 137
    - upgrading clients 115
    - upgrading DB2 environments 1
    - upgrading DB2 servers 15
  - user-defined routines
    - upgrading 149, 161

## W

- web sites
  - DB2 Migrate Now! 33
  - developerWorks - Information Management 33
  - IBM Virtual Innovation Center 33
- Windows
  - upgrading
    - DB2 servers 49
    - IBM Data Server Client 127
    - IBM Data Server Runtime Client 129

## X

- XML data
  - partitioned database environments 107
  - partitioned tables 107







Printed in USA

SC27-5513-00





Spine information:

IBM DB2 10.5 for Linux, UNIX, and Windows

Upgrading to DB2 Version 10.5

