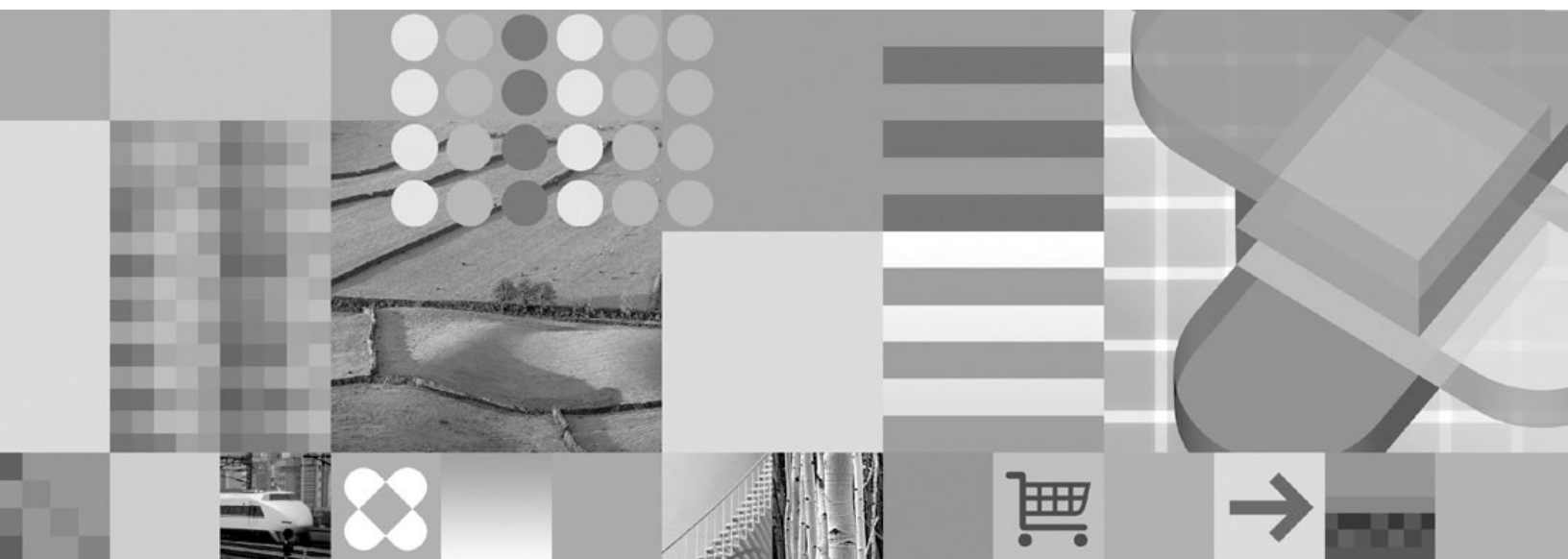


DB2 Version 9
for Linux, UNIX, and Windows



Migration Guide

DB2 Version 9
for Linux, UNIX, and Windows



Migration Guide

Before using this information and the product it supports, be sure to read the general information under *Notices*.

Edition Notice

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

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

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

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

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

© Copyright International Business Machines Corporation 2006. All rights reserved.

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

DB2 9 BETA

Contents

About this book	v
Who should use this book	v
How this book is structured	v

Part 1. Migration of your DB2 environment **1**

Chapter 1. Migration to DB2 Version 9. **3**

Chapter 2. Planning migration for your DB2 environment. **5**

Migration planning for your DB2 environment	5
Planning migration for your DB2 servers	6
Planning migration for your DB2 clients	9
Planning migration for your database applications and routines	11

Part 2. Migrating DB2 servers **15**

Chapter 3. Migration overview for DB2 servers **17**

Chapter 4. Migration essentials for DB2 servers **19**

Migration essentials for DB2 servers	19
What gets migrated.	20
Migration restrictions for DB2 servers	20
Migration recommendations for DB2 servers	23
Disk space requirements for DB2 server migration	26
Support changes for 32-bit and 64-bit DB2 servers	27
Functionality deprecated or discontinued in DB2 database products that impacts migration	29
DB2 client support for migration	31

Chapter 5. Pre-migration tasks **33**

Pre-migration tasks for DB2 servers	33
Verifying that your databases are ready for migration	34
Backing up databases before migration	36
Saving configuration information	37
Increasing table space and log file sizes before migration	39
Changing raw devices to block devices (Linux)	41
Migrating in a test environment	43
Capturing error and diagnostic information during migration	44
Taking a DB2 server offline before migration	46

Chapter 6. Migrating DB2 servers (Windows) **47**

Migrating a DB2 server (Windows)	47
Migrating instances.	48

Migrating the DB2 Administration Server (DAS)	50
Migrating databases	52

Chapter 7. Migrating DB2 servers (Linux and UNIX) **55**

Migrating a DB2 server (Linux and UNIX)	55
Migrating instances.	56
Migrating the DB2 Administration Server (DAS)	58
Migrating databases	60

Chapter 8. Migrating environments with specific characteristics. **63**

Migrating environments with specific characteristics	63
Migrating DB2 32-bit servers to 64-bit systems (Windows).	64
Migrating DB2 32-bit servers to 64-bit systems (Linux and UNIX)	65
Migrating to a new DB2 server	67
Migrating partitioned database environments	69
Migrating from a system with multiple DB2 copies (Linux and UNIX)	71
Migrating from DB2 UDB Version 7 servers (Windows).	73
Migrating from DB2 UDB Version 7 servers (Linux and UNIX)	73
Migrating DB2 servers in Microsoft Cluster Server environments.	74
Migrating DB2 Data Links Manager environments	76
Migrating from XML Extender	78
Migrating from non-DB2 relational database management systems	79

Chapter 9. Post-migration tasks **81**

Post-migration tasks for DB2 servers	81
Adjusting the log space size in migrated databases	83
Database activated after migration.	84
Changes in DB2 registry variables, configuration parameters, and database physical design characteristics	85
Conversion of type-1 indexes in migrated databases	93
Changes to the EXECUTE privilege on PUBLIC for migrated routines	94
Rebinding packages in migrated databases	95
Migrating explain tables	96
Ensuring system temporary table spaces page sizes meet requirements	97
Verifying migration of DB2 servers	98
Initialization of HADR replication in migrated databases	99

Chapter 10. Reversing DB2 server migration **101**

Part 3. Migrating DB2 clients **103**

Chapter 11. Migration overview for DB2 clients	105
Chapter 12. Migration essentials for DB2 clients	107
Chapter 13. Pre-migration tasks	111
Pre-migration tasks for DB2 clients	111
Saving DB2 client configuration information	111
Chapter 14. Migrating DB2 clients (Windows)	113
Migrating a DB2 Client (Windows)	113
Migrating a DB2 Runtime Client (Windows)	115
Migrating from DB2 clients Version 7 (Windows)	116
Chapter 15. Migrating DB2 clients (Linux and UNIX)	119
Migrating DB2 clients (Linux and UNIX)	119
Migrating from DB2 clients Version 7 (Linux and UNIX)	121
Chapter 16. Post-migration tasks	123
Post-migration tasks for DB2 clients	123
Recataloging nodes and databases that use NetBIOS and SNA protocols	123
Verifying migration of DB2 clients	125
<hr/>	
Part 4. Migrating database applications and routines	127
Chapter 17. Migration overview for database applications and routines	129
Chapter 18. Migration essentials for database applications	131
Chapter 19. Migration essentials for routines	137
Chapter 20. Pre-migration tasks for database applications and routines	139
Chapter 21. Migrating database applications	141
Migrating database applications	141
Migrating embedded SQL and CLI applications	143
Migrating Java applications that use IBM DB2 Driver for JDBC and SQLJ	145

Migrating Java applications that use DB2 JDBC Type 2 or 3 driver	146
Migrating ADO.NET applications	148
Migrating scripts	149
Migrating 32-bit database applications to run on 64-bit instances	151
Chapter 22. Migrating routines	153
Migrating routines	153
Migrating C, C++, and COBOL routines	155
Migrating Java routines	157
Migrating .NET CLR routines	159
Migrating SQL procedures	160
Migrating 32-bit external routines to run on 64-bit instances	162
Chapter 23. Post-migration tasks for database applications and routines	165
Appendixes	167
Appendix A. DB2 Information Center Version 9	169
Appendix B. Important references	171
Appendix C. DB2 Database technical information	173
Overview of the DB2 technical information	173
Documentation feedback	173
DB2 technical library in PDF format	174
Ordering printed DB2 books	176
Displaying SQL state help from the command line processor	177
Accessing different versions of the DB2 Information Center	177
Displaying topics in your preferred language in the DB2 Information Center	178
Updating the DB2 Information Center installed on your computer or intranet server	179
DB2 Visual Explain tutorial	180
DB2 troubleshooting information	181
Terms and Conditions	181
Appendix D. Notices	183
Trademarks	185
Index	187
Contacting IBM	191

About this book

The *Migration Guide* describes the migration process and concepts for each component of your DB2® environment. These components are DB2 servers, DB2 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 migrate DB2 servers and DB2 clients. It is also intended for programmers and other users who need to migrate database applications and routines.

How this book is structured

This book contains information on how to create a migration plan and how to migrate each component of your DB2 environment:

- Part 1, “Migration of your DB2 environment,” on page 1
- Part 2, “Migrating DB2 servers,” on page 15
- Part 3, “Migrating DB2 clients,” on page 103
- Part 4, “Migrating database applications and routines,” on page 127
- “Appendixes” on page 167

Part 1. Migration of your DB2 environment

This part of the book contains the following chapters:

Chapter 1, "Migration to DB2 Version 9," on page 3

Chapter 2, "Planning migration for your DB2 environment," on page 5

Chapter 1. Migration to DB2 Version 9

Upgrading to a new release of DB2 database product might require migration of your environment components. Migrating these components requires an understanding of DB2 database product migration concepts and a thorough understanding of DB2 database products. For example, if you have an existing environment using DB2 UDB Version 8 products and you want to install DB2 Version 9, then you must migrate your environment.

The migration process consists of all the tasks that you need to perform to have your DB2 environment running successfully on a new release. The migration of each of the components in your environment to DB2 Version 9 requires that you perform different tasks:

- Migrating DB2 servers involves migrating your existing instances and databases so that they can run in DB2 Version 9.
- Migrating DB2 clients involves migrating your client instances to keep the configuration of your existing DB2 clients.
- Migrating database applications and routines involves testing them in DB2 Version 9 and modifying them only when you need to support changes in DB2 Version 9.

The following information is provided to document the migration process for DB2 Version 9:

- Migration overviews define migration concepts and describe the migration process for a component.
- Migration essentials include the details about migration support, restrictions and recommendations that you need to know to plan your migration strategy.
- Pre-migration tasks describe all the preparation tasks that you need to perform before migration.
- Migration tasks describe step by step the basic migration process for a component and how to migrate environments with special characteristics.
- Post-migration tasks describe all the tasks that you need perform after migration to have your DB2 server running at the optimum level.

Related concepts:

- Chapter 18, "Migration essentials for database applications," on page 131
- Chapter 12, "Migration essentials for DB2 clients," on page 107
- "Migration essentials for DB2 servers" on page 19
- Chapter 19, "Migration essentials for routines," on page 137
- "Migration planning for your DB2 environment" on page 5

Chapter 2. Planning migration for your DB2 environment

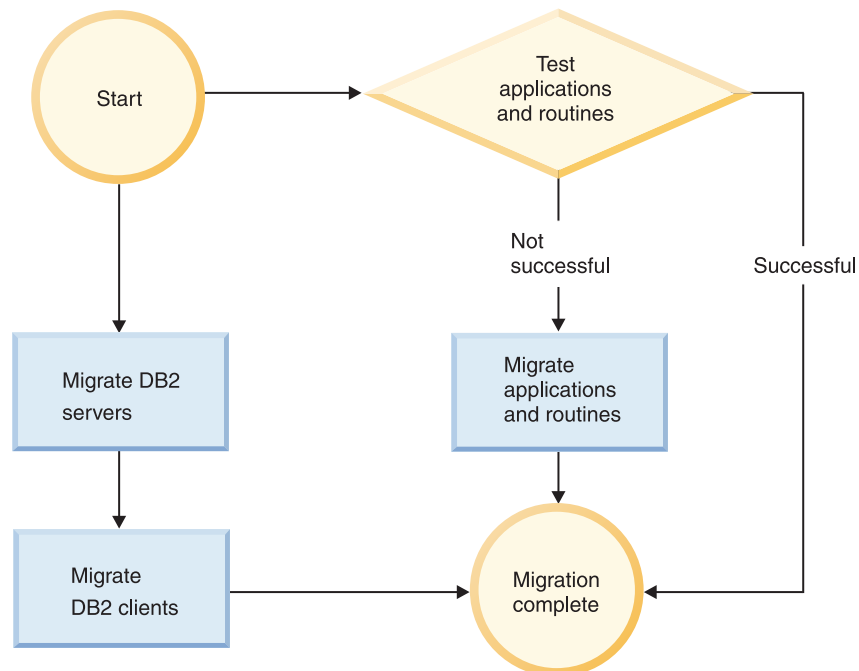
This chapter describes how to plan the migration for your environment. It contains the following sections:

- “Migration planning for your DB2 environment”
- “Planning migration for your DB2 servers” on page 6
- “Planning migration for your DB2 clients” on page 9
- “Planning migration for your database applications and routines” on page 11

Migration planning for your DB2 environment

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

First, you need to devise a strategy on how to approach the migration of your environment. You need to determine the order in which you are going to migrate each component. The characteristics of your environment and the information in migration essentials, especially the migration recommendations and restrictions, can help you determine your strategy. The following graphic depicts the recommended migration roadmap for the components in your environment:



The following is an example of a good **migration strategy** in which you test your database applications and routines and determine that they run successfully in DB2 Version 9:

1. Set up a DB2 Version 9 test server and create test databases.

2. Test your database applications and routines on a DB2 Version 9 test database to determine whether they run successfully.
3. Migrate in a test environment. Determine what the migration issues are and how to resolve them. Use this information to adjust your migration plan.
4. Migrate your DB2 servers DB2 Version 9 in your production environment. Ensure that they operate as expected.
5. Migrate your DB2 clients to DB2 Version 9 in your production environment. Ensure that your DB2 clients operate as expected.
6. Test your database applications and routines in the DB2 Version 9 migrated environment to determine whether they run as expected.
7. Make your migrated environment available to users.
8. Identify the use of deprecated features that will eventually become unsupported and new features that can improve the functionality and performance of your applications and routines. Plan how to modify your applications and routines.
9. Modify your database applications and routines as planned. Ensure that they run successfully in DB2 Version 9.

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

- Migration prerequisites
- Pre-migration tasks
- Migration tasks
- Post-migration tasks

Review previous migration plans and compare them with the migration plan for DB2 Version 9. Include in your new plan any steps related to internal procedures to request access, software installation or other system services within your organization.

Finally, plan to remove the use of deprecated features and incorporate new features from DB2 Version 9. Although you are only required to remove the use of unsupported features, you should also plan to remove the use of deprecated features after migration because they will become unsupported in a future release. Also, you should take advantage of new features for your database products, applications, and routines and to enhance functionality and improve performance.

Related tasks:

- “Planning migration for your database applications and routines” on page 11
- “Planning migration for your DB2 clients” on page 9
- “Planning migration for your DB2 servers” on page 6

Planning migration for your DB2 servers

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

Procedure:

To create a migration plan for your DB2 servers:

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

Table 1. Migration plan details for DB2 servers.

Migration plan	Details
Prerequisites	Ensure that you: <ul style="list-style-type: none"> • meet hardware and operating system requirements. • resolve any support issues in migration essentials for DB2 servers. • meet all prerequisites for the migration task and subtasks, especially obtaining root or Local Administrator access and required DB2 authorization.
Pre-migration tasks	Include the following tasks: <ul style="list-style-type: none"> • Migrate your DB2 server in a test environment to determine any migration issues • Verify that databases are ready for DB2 migration • Back up your databases • Save configuration information • Increase table space and log file sizes • Capture error and diagnostic information during migration • Take the DB2 server offline for DB2 migration In addition, check the list of pre-migration tasks for optional tasks that you might want to perform for your environment.
Migration task	You must include these steps: <ul style="list-style-type: none"> • Install DB2 Version 9 • Migrate instances • Migrate the DAS • Migrate databases Review the following migration tasks to determine the additional steps that are required to migrate your environment: <ul style="list-style-type: none"> • Migrating a DB2 server (Windows®) • Migrating a DB2 server (Linux® and UNIX®) • Migrating environments with specific characteristics Take note of the time required to migrate your databases.

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

Migration plan	Details
Post-migration tasks	<ul style="list-style-type: none"> • Reset the <i>diaglevel</i> database manager configuration parameter to the value set before the migration • Adjust the log space size • Review changes in DB2 registry variables values and configuration parameters settings • Convert type-1 indexes to type-2 indexes in migrated databases • Revoke EXECUTE privilege from PUBLIC on functions and procedures • Rebind packages in migrated databases • Migrate DB2 Explain tables • Checking your system temporary table spaces page size for the maximum row size in result sets • Activate your database after migration • Verify that your DB2 server migration was successful • Back up your databases after the migration is complete <p>In addition, check the list of post-migration tasks for optional tasks that you might want to perform for your environment. Consider adding the following tasks to your migration plan:</p> <ul style="list-style-type: none"> • Tune your DB2 server once migration is completed • Remove the use of deprecated features in DB2 Version 9 • Implement the use of new features, where appropriate, to improve performance at the DB2 server level. Review manageability, performance and scalability enhancements in What's New to determine what new features you might want to apply to your environment

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

Related concepts:

- “Migration essentials for DB2 servers” on page 19
- “Migrating environments with specific characteristics” on page 63
- “Functionality deprecated or discontinued in DB2 database products that impacts migration” on page 29
- “What's new for V9.1: Manageability enhancements summary” in *What's New*
- “What's new for V9.1: Performance enhancements summary” in *What's New*
- “What's new for V9.1: Scalability enhancements summary” in *What's New*
- “Migration planning for your DB2 environment” on page 5

Related tasks:

- “Migrating in a test environment” on page 43
- “Verifying that your databases are ready for migration” on page 34
- “Backing up databases before migration” on page 36
- “Saving configuration information” on page 37

- “Increasing table space and log file sizes before migration” on page 39
- “Capturing error and diagnostic information during migration” on page 44
- “Taking a DB2 server offline before migration” on page 46
- “Migrating instances” on page 48
- “Migrating the DB2 Administration Server (DAS)” on page 50
- “Migrating databases” on page 52
- “Migrating a DB2 server (Windows)” on page 47
- “Migrating a DB2 server (Linux and UNIX)” on page 55
- “Adjusting the log space size in migrated databases” on page 83
- “Rebinding packages in migrated databases” on page 95
- “Migrating explain tables” on page 96
- “Ensuring system temporary table spaces page sizes meet requirements” on page 97
- “Verifying migration of DB2 servers” on page 98
- “Developing a performance improvement process” in *Performance Guide*
- Chapter 10, “Reversing DB2 server migration,” on page 101
- “Planning migration for your database applications and routines” on page 11
- “Planning migration for your DB2 clients” on page 9

Related reference:

- “Pre-migration tasks for DB2 servers” on page 33
- “Changes in DB2 registry variables, configuration parameters, and database physical design characteristics” on page 85
- “Conversion of type-1 indexes in migrated databases” on page 93
- “Changes to the EXECUTE privilege on PUBLIC for migrated routines” on page 94
- “Database activated after migration” on page 84
- “Post-migration tasks for DB2 servers” on page 81

Planning migration for your DB2 clients

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

Procedure:

To create a migration plan for your DB2 clients:

1. Write the migration plan for DB2 clients, using all the details that apply to your environment:

Table 2. Migration plan details for DB2 clients.

Migration plan	Details
Prerequisites	<p>Ensure that you:</p> <ul style="list-style-type: none"> • meet hardware and operating system requirements. • resolve any support issues in migration essentials for DB2 clients including client and server connectivity. • meet all prerequisites for the migration task and subtasks, especially obtaining root or Local Administrator access and required DB2 authorization.
Pre-migration tasks	<ul style="list-style-type: none"> • Migrate your DB2 servers • Save your DB2 client configuration information
Migration task	<p>You must include these steps:</p> <ul style="list-style-type: none"> • Install V9 DB2 client • Migrate client instance <p>Review the following migration tasks to determine the additional steps that are required to migrate your environment:</p> <ul style="list-style-type: none"> • Migrating DB2 Client (Windows) • Migrating DB2 Runtime Client (Windows) • Migrating DB2 clients (Linux and UNIX) • Migrating from DB2 clients Version 7 (Windows) • Migrating from DB2 clients Version 7 (Linux and UNIX)
Post-migration tasks	<ul style="list-style-type: none"> • Recatalog nodes and databases using NetBIOS and SNA protocols • Review changes in DB2 registry variables values and configuration parameters settings • Verify that migration for DB2 clients was successful

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

Related concepts:

- Chapter 12, “Migration essentials for DB2 clients,” on page 107
- Chapter 3, “Migration overview for DB2 servers,” on page 17
- “Migration planning for your DB2 environment” on page 5

Related tasks:

- “Saving DB2 client configuration information” on page 111
- “Migrating a DB2 Client (Windows)” on page 113
- “Migrating a DB2 Runtime Client (Windows)” on page 115
- “Migrating DB2 clients (Linux and UNIX)” on page 119
- “Migrating from DB2 clients Version 7 (Windows)” on page 116
- “Migrating from DB2 clients Version 7 (Linux and UNIX)” on page 121
- “Recataloging nodes and databases that use NetBIOS and SNA protocols” on page 123
- “Verifying migration of DB2 clients” on page 125
- “Planning migration for your database applications and routines” on page 11
- “Planning migration for your DB2 servers” on page 6

Related reference:

- “Changes in DB2 registry variables, configuration parameters, and database physical design characteristics” on page 85

Planning migration for your database applications and routines

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

Procedure:

To create a migration plan for your database applications and routines:

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

Table 3. Migration plan details for database applications.

Migration plan	Details
Prerequisites	Ensure that you: <ul style="list-style-type: none">• meet hardware and operating system requirements.• meet the new development software requirements.• resolve any support issues in migration essentials for database applications during migration.• meet all prerequisites for the migration task and sub-tasks, especially obtaining required DB2 authorization.
Pre-migration tasks	Include the following tasks: <ul style="list-style-type: none">• Migrate your DB2 client or install V9 application driver• Test your database applications in a DB2 Version 9 testing environment. If your applications run successfully, the rest of migration steps are not required In addition, check the list of pre-migration tasks for optional tasks that you might want to perform for your environment. Even if your current operating system and development software are supported, consider including the following tasks to improve application performance: <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 3. Migration plan details for database applications. (continued)

Migration plan	Details
Migration task	<p>You must include these steps:</p> <ul style="list-style-type: none"> • Modify your application code to support changes in DB2 Version 9 and to remove use of features that are not supported in DB2 Version 9 • Modify your application to support changes specific to the development environment • Rebuild all database applications after completing your modifications. • Test your database applications using DB2 Version 9 <p>Review the following migration tasks to determine the additional steps that are required by your development environment to migrate database applications:</p> <ul style="list-style-type: none"> • Migrating embedded SQL and CLI applications • Migrating Java applications that use IBM DB2 Driver for JDBC and SQLJ • Migrating Java applications that use DB2 JDBC Type 2 or 3 driver • Migrating ADO.NET applications • Migrating scripts • Migrating 32-bit database applications to run on 64-bit instances
Post-migration tasks	<p>Perform the recommended post-migration tasks for database applications, especially:</p> <ul style="list-style-type: none"> • Tune performance of your database applications and routines • Remove the use of deprecated features in DB2 Version 9 • Implement the use of new features in DB2 Version 9 for application development where appropriate

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

Table 4. Migration plan details for routines.

Migration plan	Details
Prerequisites	<p>Ensure that you:</p> <ul style="list-style-type: none"> • meet hardware and operating system requirements. • meet the new development software requirements. • resolve any support issues in migration essentials for routines during migration. • meet all prerequisites for the migration task and sub-tasks, especially obtaining required DB2 authorization.

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

Migration plan	Details
Pre-migration tasks	<p>Include the following task:</p> <ul style="list-style-type: none"> • Test your routines in a DB2 Version 9 testing environment. If your routines run successfully, the rest of migration steps are not required <p>In addition, check the list of pre-migration tasks for optional tasks that you might want to perform for your environment. Even if your current operating system and development software are supported, consider including the following tasks to improve 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
Migration task	<p>You must include these steps:</p> <ul style="list-style-type: none"> • Modify your routines to support changes in DB2 Version 9 and to remove use of features that are not supported in DB2 Version 9 • Modify your routines to support changes specific to the development environment • Rebuild all external routines after completing your modifications • Retest your routines using DB2 Version 9 <p>Review the following migration tasks to determine the additional steps that are required by your development environment to migrate routines:</p> <ul style="list-style-type: none"> • Migrating C, C++, and COBOL routines • Migrating Java routines • Migrating .NET CLR routines • Migrating SQL procedures • Migrating 32-bit external routines to run on 64-bit instances
Post-migration tasks	<p>Perform the recommended post-migration tasks for routines, especially:</p> <ul style="list-style-type: none"> • Remove the use of deprecated features in DB2 Version 9 • Implement the use of new features in DB2 Version 9 for application development where appropriate

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

Related concepts:

- “What’s new for V9.1: New and changed development software support” in *What’s New*
- Chapter 18, “Migration essentials for database applications,” on page 131
- Chapter 11, “Migration overview for DB2 clients,” on page 105
- Chapter 20, “Pre-migration tasks for database applications and routines,” on page 139
- “Functionality deprecated or discontinued in DB2 database products that impacts migration” on page 29
- “What’s new for V9.1: Application development enhancements summary” in *What’s New*
- Chapter 19, “Migration essentials for routines,” on page 137

- “Migration planning for your DB2 environment” on page 5

Related tasks:

- “Migrating embedded SQL and CLI applications” on page 143
- “Migrating Java applications that use IBM DB2 Driver for JDBC and SQLJ” on page 145
- “Migrating Java applications that use DB2 JDBC Type 2 or 3 driver” on page 146
- “Migrating ADO.NET applications” on page 148
- “Migrating scripts” on page 149
- “Migrating 32-bit database applications to run on 64-bit instances” on page 151
- “Migrating C, C++, and COBOL routines” on page 155
- “Migrating Java routines” on page 157
- “Migrating .NET CLR routines” on page 159
- “Migrating SQL procedures” on page 160
- “Migrating 32-bit external routines to run on 64-bit instances” on page 162
- “Planning migration for your DB2 clients” on page 9
- “Planning migration for your DB2 servers” on page 6

Part 2. Migrating DB2 servers

This part of the book contains the following chapters:

Chapter 3, “Migration overview for DB2 servers,” on page 17

Chapter 4, “Migration essentials for DB2 servers,” on page 19

Chapter 5, “Pre-migration tasks,” on page 33

Chapter 6, “Migrating DB2 servers (Windows),” on page 47

Chapter 7, “Migrating DB2 servers (Linux and UNIX),” on page 55

Chapter 8, “Migrating environments with specific characteristics,” on page 63

Chapter 9, “Post-migration tasks,” on page 81

Chapter 3. Migration overview for DB2 servers

After you install DB2 Version 9 for Linux and UNIX on a DB2 server where DB2 UDB Version 8 is installed, you need to migrate all the instances and databases to be able to run them under this latest version. For Windows, you have an option to automatically migrate the instance during the installation. If you choose to migrate your instances during installation, you only need to migrate your databases.

The current level of DB2 that you have installed will determine the way to proceed with migration to DB2 Version 9:

- Migration from DB2 UDB Version 8
- Migration from DB2 UDB Version 7 or DataJoiner® Version 2

Migration from DB2 UDB Version 8

If you have DB2 UDB Version 8 installed, you can migrate directly to DB2 Version 9. Learn details, limitations about migration process, and possible issues that you need to be aware of in the “Migration essentials for DB2 servers” on page 19 section. Refer to Migrating a DB2 server for details on how to migrate to DB2 Version 9.

Migration from DB2 UDB Version 7 or DataJoiner Version 2

If you have DB2 UDB Version 6, DB2 UDB Version 7 or DataJoiner Version 2 installed, you need to migrate first to DB2 UDB Version 8, and then migrate to DB2 Version 9. Direct migration to DB2 Version 9 is not supported. It is recommended that you migrate to DB2 UDB Version 8.2. Refer to DB2 UDB Version 8.2 documentation for details on how to migrate to DB2 UDB Version 8.2.

Migration support for DB2 products

Migration to DB2 Version 9 is supported for the following DB2 products:

- DB2 UDB Enterprise Server Edition, Version 8
- DB2 UDB Workgroup Server Unlimited Edition, Version 8
- DB2 UDB Workgroup Server Edition, Version 8
- DB2 UDB Personal Edition, Version 8
- DB2 Universal Developer’s Edition, Version 8
- DB2 Personal Developer’s Edition, Version 8
- DB2 UDB Express Edition, Version 8
- DB2 Connect™ Unlimited Edition, Version 8
- DB2 Connect Enterprise Server Edition, Version 8
- DB2 Connect Application Server Edition, Version 8
- DB2 Connect Personal Edition, Version 8
- DB2 Administration Client, Version 8
- DB2 Application Development Client, Version 8
- DB2 Runtime Client, Version 8
- DB2 Query Patroller, Version 8

For DB2 products not supported, refer to Functionality not supported in DB2 database product that impacts migration.

DB2 database product migration portal

The purpose of the DB2 database product migration portal is to provide you with one place to access all additional resources and up-to-date information about the migration process as they become available. These resources include white papers and sample scripts for migration.

Related concepts:

- “What's new for V9.1: Changes in existing functionality summary” in *What's New*
- “What's new for V9.1: Discontinued functionality summary” in *What's New*
- “Migrating environments with specific characteristics” on page 63

Related tasks:

- “Migrating a DB2 server (Linux and UNIX)” on page 55
- “Migrating a DB2 server (Windows)” on page 47
- “Migrating from DB2 UDB Version 7 servers (Linux and UNIX)” on page 73
- “Migrating from DB2 UDB Version 7 servers (Windows)” on page 73
- “Planning migration for your DB2 servers” on page 6

Chapter 4. Migration essentials for DB2 servers

This chapter describes the migration essentials you need know for your DB2 servers. It contains the following sections:

- “Migration essentials for DB2 servers”
- “What gets migrated” on page 20
- “Migration restrictions for DB2 servers” on page 20
- “Migration recommendations for DB2 servers” on page 23
- “Disk space requirements for DB2 server migration” on page 26
- “Support changes for 32-bit and 64-bit DB2 servers” on page 27
- “Functionality deprecated or discontinued in DB2 database products that impacts migration” on page 29
- “DB2 client support for migration” on page 31

Migration essentials for DB2 servers

Migrating DB2 servers to DB2 Version 9 requires an understanding of migration concepts, migration restrictions, migration recommendations, and your DB2 server. Once you have a complete understanding of what migrating your DB2 server involves, you can create your own migration plan.

Consider the following factors to develop a complete understanding of migrating DB2 servers to DB2 Version 9:

- What gets migrated
- Migration recommendations for DB2 servers
- Disk space requirements for DB2 server migration
- Migration restrictions for DB2 servers
- Support changes for 32-bit and 64-bit DB2 servers
- Functionality deprecated or discontinued in DB2 database products that impacts migration
- DB2 client support for migration

Related concepts:

- “Migrating environments with specific characteristics” on page 63
- Chapter 18, “Migration essentials for database applications,” on page 131
- Chapter 12, “Migration essentials for DB2 clients,” on page 107
- Chapter 19, “Migration essentials for routines,” on page 137

Related tasks:

- “Migrating a DB2 server (Linux and UNIX)” on page 55
- “Migrating a DB2 server (Windows)” on page 47
- “Planning migration for your DB2 servers” on page 6

Related reference:

- “Pre-migration tasks for DB2 servers” on page 33
- “Post-migration tasks for DB2 servers” on page 81

What gets migrated

When the instance migration is called explicitly using the **db2imigr** command, or implicitly when you install DB2 Version 9 on Windows, the following actions occur:

- Migrates to a new instance under a DB2 Version 9 copy.
- Migrates instance profile registry variables. The global profile registry variables set by the user are not migrated.
- Migrates the database manager configuration (dbm cfg) file.
- Sets the *jdk_path* database manager (dbm cfg) parameter appropriately.
- Copies over other configuration files.
- In a Microsoft® Cluster Server (MSCS) environment, defines a new resource type, updates all DB2 MSCS resources to use the new resource type, removes the old resource type and brings all resources online.

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

When the database migration is called explicitly using the **MIGRATE DATABASE** command, or implicitly using the **RESTORE DATABASE** command from a DB2 UDB Version 8 backup, the following database entities are converted during database migration:

- Database configuration file
- Log file header
- Catalog tables
- Buffer pool files
- Index root page
- History file

Related concepts:

- “Migrating environments with specific characteristics” on page 63
- “Migration essentials for DB2 servers” on page 19
- Chapter 3, “Migration overview for DB2 servers,” on page 17
- “Migration recommendations for DB2 servers” on page 23
- “Migration restrictions for DB2 servers” on page 20

Related tasks:

- “Migrating a DB2 server (Linux and UNIX)” on page 55
- “Migrating a DB2 server (Windows)” on page 47

Migration restrictions for DB2 servers

Before you start the migration of your DB2 server, you need to understand what the support for migration is and what the restrictions are.

What is supported?

Consider the following restrictions before you migrate to DB2 Version 9:

- Migration is supported only from DB2 UDB Version 8. If you have DB2 UDB Version 7 or earlier, you need to migrate to DB2 UDB Version 8 first.
- On Windows operating system, you have a migrate option to automatically migrate an existing DB2 UDB version 8 copy during the DB2 Version 9 installation. This option automatically migrates all your instances and your DB2 administration server (DAS), and removes your DB2 UDB version 8 copy. If you do not choose the migrate option, you must manually migrate your instances by running the **db2imigr** command after installation and your DAS by running the **dasmigr** command.
- On Linux and UNIX operating system, you can only install a new copy of DB2 Version 9. You have to manually migrate your instances by running the **db2imigr** command after installation. You also have to manually migrate your existing DAS by running the **dasmigr** command.
- Instance bit size is determined by the platform where DB2 Version 9 is installed, and support for 32-bit kernels and 64-bit kernels has changed. The following table specifies what instance bit size is supported on what platform:

Table 5. DB2 32-bit and 64-bit instances support.

Instance bit size	OS Kernel	Operating systems supported
32-bit	32-bit	<ul style="list-style-type: none"> • Windows on x86 or X64 (Using DB2 for Windows on x86 Install image) • Linux on x86
64-bit	64-bit	Linux on Itanium Platform Family (IPF)
64-bit (with 32-bit support)	64-bit	<ul style="list-style-type: none"> • AIX®, HP-UX, or Solaris • Windows on x86-64 (Using DB2 for Windows for X64 Install Image) • Windows on IPF • Linux kernel on zSeries®, POWER™, or x86-64

- Migration is supported from a system with multiple DB2 copies. Each copy can be at the same level (only DB2 Version 9) or a different level of the DB2 database product. Each copy must have an installation name, and the option to specify the installation location. The name of the instance must be unique to all installation names.
- A partitioned database environment with multiple database partitions can be migrated. However, the migration requires previous installation of DB2 Version 9 on all participating database partition servers. The migration commands must be run from the database partition server that owns the instance.
- Restoring full database offline backups from DB2 UDB Version 8 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 for complete details on migration support using the RESTORE DATABASE command for DB2 UDB Version 8 backups.
- Migration from a DB2 server with Query Patroller installed is supported. The MIGRATE DATABASE command sets the *dyn_query_mgmt* database

configuration parameter to DISABLE. You need to set *dyn_query_mgmt* to ENABLE after you complete the database migration and install Query Patroller Version 9.

- Index Extensions are migrated as part of the database migration. However, you might need to re-create your indexes if you migrated from a DB2 UDB Version 8 32-bit instance to a DB2 Version 9 64-bit instance. See post-migration tasks for details.

What is not supported?

DB2 Version 9 installation will fail to run when:

- The operating system is not supported. Certain versions of UNIX, Linux and Windows operating systems are no longer supported such as AIX 4.3.3, Solaris 8, Windows 95, Windows 98, Windows NT and Windows Me. You need to upgrade to a supported version of the operating system before you migrate to DB2 Version 9. For a complete list of supported operating system, visit the System requirements Web page for DB2 database product installation.
- The operating system is running a 32-bit kernel on AIX, HP-UX, Solaris, or Linux (on zSeries, Power, or x86-64). A 64-bit kernel must be installed prior to installing DB2 Version 9. 32-bit instances are not supported on these platforms.

The **db2imigr** command will fail to run when:

- The instance is already a DB2 Version 9 instance or higher. Run the **db2iupdt** command to move among different fix pack levels or copies of DB2 Version 9.
- You try to migrate from DB2 Version 9 back to DB2 UDB Version 8.
- A connection to an HADR primary database cannot be established. The **db2imigr** command runs the **db2ckmig** command which requires a database connection to perform the required checks.
- DB2 Data Links Manager Version 8 is installed on the DB2 server. DB2 Data Links Manager is not supported in DB2 Version 9. However, when you run the **db2imigr** command, the error message that is generated includes instructions on how to migrate to a standard DB2 Version 9 instance without the DB2 Data Links Manager functionality.
- DB2 Data Warehouse Manager Version 8 and any extensions are installed on the DB2 server. DB2 Data Warehouse Manager is not supported in DB2 Version 9. However, when you run the **db2imigr** command, the error message that is generated includes instructions on how to migrate to a standard DB2 Version 9 instance without the DB2 Data Warehouse Manager functionality.
- The instance that you are trying to migrate is active. Run the **db2stop** command to stop the instance.

The MIGRATE DATABASE command will fail to run when:

- You encounter any of the problems described in the reason codes of error message SQL1704N.
- User-defined distinct types (UDTs) are encountered with the names XML, BINARY and VARBINARY. These UDTs must be dropped and re-created with a different name before database migration.
- User objects use the system-defined DATALINK data type. These objects must be dropped or altered before database migration. Also, if you

installed DB2 Net Search Extender (NSE) on your DB2 server, you must drop the UDFs created by NSE for Data Links support.

- A database that is in one of the following states:
 - Catalogued database does not exist
 - Backup pending
 - Roll-forward pending
 - One or more table spaces in an abnormal state
 - Transaction inconsistent
- A database that has been enabled as a high availability disaster recovery (HADR) standby database.
- You try to migrate a database from a DB2 server on Windows to a DB2 server on Linux or UNIX, or the reverse. Migration across platforms from Windows to Linux or UNIX is not supported.

Related concepts:

- “Functionality deprecated or discontinued in DB2 database products that impacts migration” on page 29
- “Migration essentials for DB2 servers” on page 19
- Chapter 17, “Migration overview for database applications and routines,” on page 129
- “Migration recommendations for DB2 servers” on page 23
- “Support changes for 32-bit and 64-bit DB2 servers” on page 27
- “Version 9 incompatibilities with previous releases and changed behaviors” in *Administration Guide: Planning*

Related tasks:

- “Migrating partitioned database environments” on page 69

Related reference:

- “Deprecated and discontinued features” in *Administration Guide: Planning*
- “Version 8 incompatibilities with previous releases” in *Administration Guide: Planning*

Migration recommendations for DB2 servers

Consider the following recommendations when planning your DB2 server migration:

Review changes in existing DB2 database product functionality

Changes in existing functionality introduced in DB2 Version 9 might potentially impact your applications, scripts, maintenance process, and any other aspects related your DB2 database product migration process. You need to review these changes and plan how to address these changes before the migration. Migrating in a test environment allows you to learn about possible issues, evaluate the impact on your environment and find a resolution.

Benchmark DB2 performance

Run a number of test queries before migrating your DB2 server. The **db2batch** benchmark tool can help you collect elapsed and CPU times for your test queries. Record the exact environment conditions where queries are run.

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

Migrating an SQL replication environment

Migrating an SQL replication environment from DB2 UDB Version 8 requires that you prepare for the migration before you migrate your DB2 server.

In addition to running pre-migration tasks for migration of an SQL replication environment, you must archive all DB2 log files used by the Capture program before migration. Then, after DB2 server migration, use the migration tools to convert your SQL replication environment to DB2 Version 9.

For complete information on migrating your SQL replication environment, see the Migrating to Replication Version 9 guide available at the Web page for DB2 Version 9 manuals.

Migrating DB2 Spatial Extender

If you have DB2 Spatial Extender installed and you migrated your spatially-enabled databases to DB2 Version 9, you must review the DB2 Spatial Extender and Geodetic Extender User's Guide and the migrating the Spatial Extender environment task for migration details.

Migrating a Microsoft Cluster Server environment

In a Microsoft Cluster Server (MSCS) environment, the recommendation is to install DB2 Version 9 as a new copy and then run **db2imigr** command to migrate the MSCS instance.

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

The support of certain versions of UNIX, Linux and Windows operating systems has been dropped in DB2 Version 9. If your operating system version is not supported, you need to upgrade your operating system before installing DB2 Version 9. Note that newer versions of operating systems can also bring new hardware requirements. For the most up-to-date operating system information, visit the DB2 system requirements page.

Even when you are not required but decide to upgrade, performing hardware and operating system upgrades separately from DB2 database product migration simplifies problem determination if you encounter migration difficulties. If you upgrade your software or hardware prior to a DB2 database product migration, ensure that your system is operating acceptably before attempting the migration process.

Migration of database applications and routines

If you migrate your DB2 server, you might also need to migrate your database applications and routines to support changes for DB2 Version 9 64-bit instances, SQL stored procedures, Java™ Virtual Machine (JVM), and development software.

Migration essentials for database applications and Migration essentials for routines include the factors that can impact your database application migration or routine migration. Review these factors and make any necessary changes to your database applications and routines to ensure that they run after the migration to DB2 Version 9.

In a migration testing environment, you can test and verify that your database applications and routines run successfully in DB2 Version 9 to find out if you need to migrate them. You can also migrate your database applications and routines before you migrate your production environment.

Devise a plan to reverse a migration

There is no utility to reverse a migration from DB2 Version 9 to DB2 UDB Version 8. To reverse a database migration, you must reinstall DB2 UDB Version 8, re-create instances under DB2 UDB Version 8, and restore your database backups. Review the Reversing DB2 databases system migration task to learn all the required steps.

Perform pre-migration tasks

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

Migrate DB2 servers first

As you upgrade your environment from DB2 UDB Version 8 to DB2 Version 9, if you migrate your DB2 clients to DB2 Version 9 before you migrate all of your DB2 servers to DB2 Version 9, there are some restrictions and limitations such as support of new DB2 database product features, network protocols, and connectivity.

To avoid these known restrictions and limitations, migrate all of your DB2 servers to DB2 Version 9 before you migrate any of your DB2 clients to DB2 Version 9. These restrictions and limitations are not associated with DB2 Connect.

Enable autonomic computing features

DB2 Version 9 enables several autonomic computing features when you create a database:

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

However, when you migrate your database to DB2 Version 9, *auto_runstats* retains its previous value and *self_tuning_mem* is set to off.

You should consider enabling these features to gain performance and manageability improvements. Also, enabling the self tuning of memory feature allows you to manage the new memory requirements.

Related concepts:

- “Benchmark testing” in *Performance Guide*
- “Explain tools” in *Performance Guide*

- “Version 9 incompatibilities with previous releases and changed behaviors” in *Administration Guide: Planning*
- “What’s new for V9.1: Changes in existing functionality summary” in *What’s New*
- “What’s new for V9.1: Discontinued functionality summary” in *What’s New*
- “Migrating environments with specific characteristics” on page 63
- Chapter 12, “Migration essentials for DB2 clients,” on page 107
- Chapter 17, “Migration overview for database applications and routines,” on page 129
- “Support changes for 32-bit and 64-bit DB2 servers” on page 27

Related tasks:

- “Migrating a DB2 server (Linux and UNIX)” on page 55
- “Migrating a DB2 server (Windows)” on page 47
- “Migrating in a test environment” on page 43

Related reference:

- “Deprecated and discontinued features” in *Administration Guide: Planning*
- “Pre-migration tasks for DB2 servers” on page 33

Disk space requirements for DB2 server migration

You need to be aware that the migration 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 migrating to DB2 Version 9.

Database directory files

The files SQLSPCS.1 and SQLSPCS.2 contain table space information. During migration to DB2 Version 9, these files grow to four times their previous size but the total data size on disk does not exceed the new size of the SQLSPCS.1 and SQLSPCS.2 files. For example, if you have two files whose size totals 512 KB before migration, you will need at least 2 MB of free disk space.

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 migrating. System catalog table space is required for both old and new database catalogs during migration. 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 table spaces, you can increase the size of existing containers. For DMS (Database Managed Space) table spaces, you can also add additional containers although this might trigger data rebalancing. You can reduce the size of the containers after migration.

Log file space

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

To ensure sufficient log file space is available, you can set the database configuration parameters *logfilesiz*, *logprimary* and *logsecond* to twice their current values. If you already have available a large log file space, it might not be necessary to increase these parameters. Also on partitioned database environments, you only need to increase the log space in the catalog database partition server.

You must update these database configuration parameters values before you migrate the instance to DB2 Version 9, because you will not be able to update these database configuration parameters until you issue the `MIGRATE 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 `MIGRATE DATABASE` command.

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

Related tasks:

- “Modifying containers in a DMS table space” in *Administration Guide: Implementation*
- “Increasing table space and log file sizes before migration” on page 39
- “Migrating a DB2 server (Linux and UNIX)” on page 55
- “Migrating a DB2 server (Windows)” on page 47

Related reference:

- “ALTER TABLESPACE statement” in *SQL Reference, Volume 2*
- “GET DATABASE CONFIGURATION command” in *Command Reference*
- “UPDATE DATABASE CONFIGURATION command” in *Command Reference*

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

DB2 Version 9 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 details about supported architectures on each operating system.

The bit size for the instance can no longer be specified when you create or migrate an instance. The bit size for new instances is determined by the operating system where DB2 Version 9 is installed. The following table summarizes the DB2 Version 9 bit size support that is available for each of the following operating systems:

Table 6. DB2 Version 9 support available per operating system.

Operating systems	DB2 Version 9 support available
<ul style="list-style-type: none"> • 32-bit Windows on x86 and X64 (Using x86 DB2 Version 9 32-bit product) • 32-bit Linux on x86 	<ul style="list-style-type: none"> • 32-bit instances only • 32-bit DB2 server, DB2 client, and GUI tools packages • 32-bit IBM® Software Development Kit (SDK) for Java
<ul style="list-style-type: none"> • 64-bit Linux on Itanium Platform Family (IPF) 	<ul style="list-style-type: none"> • 64-bit instances • 64-bit DB2 server and DB2 client • 64-bit applications (Java and non-Java), and 64-bit server routines • 64-bit IBM SDK for Java
<ul style="list-style-type: none"> • 64-bit kernels of AIX, HP-UX, or Solaris • 64-bit Windows on X64 and IPF • 64-bit Linux kernel on x86-64, 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 DB2 client • 64-bit applications and server routines • 32-bit DB2 client side application support • 32-bit fenced stored procedures/UDFs only (non-Java) • Java fenced Stored Procedures/UDFs • 64-bit IBM SDK for Java

When you migrate from a 32-bit instance to DB2 Version 9 on a 32-bit system there are no issues to consider. You can only migrate to a 32-bit instance on 32-bit Windows or 32-bit Linux on x86.

When you migrate from a 64-bit instance to DB2 Version 9 on a 64-bit system, there might be incompatibilities due to the shared library path specification. For example, if the path specified to link the DB2 libraries to your application is the DB2 product directory, the application fails to run because DB2 Version 9 has a different path. However, if you linked the libraries using the library path under the instance home directory (`$INSTHOME/sqlib/lib`), your application will run successfully. You can only migrate to a 64-bit instance on a 64-bit system.

Only when you migrate from a 32-bit instance to DB2 Version 9 on a 64-bit system will you need to deal with incompatibilities due to the shared library path specification and unsupported features (see the details on available support in Table 6). For example, 32-bit unfenced stored procedures in any supported language except Java are not supported. By simply dropping and recreating these stored procedures as fenced you can resolve this issue quickly.

Some of these migration issues are due to the migration from 32-bit to 64-bit support and are not inherent to DB2 Version 9.

Related concepts:

- Chapter 18, “Migration essentials for database applications,” on page 131
- “Migration essentials for DB2 servers” on page 19
- “Version 9 incompatibilities with previous releases and changed behaviors” in *Administration Guide: Planning*
- “32-bit and 64-bit support for embedded SQL applications” in *Developing Embedded SQL Applications*

Related tasks:

- “Migrating 32-bit database applications to run on 64-bit instances” on page 151
- “Migrating 32-bit external routines to run on 64-bit instances” on page 162
- “Migrating DB2 32-bit servers to 64-bit systems (Linux and UNIX)” on page 65
- “Migrating DB2 32-bit servers to 64-bit systems (Windows)” on page 64

Related reference:

- “Deprecated and discontinued features” in *Administration Guide: Planning*

Functionality deprecated or discontinued in DB2 database products that impacts migration

The use of functionality that was available in DB2 UDB Version 8 but that is no longer supported or deprecated in DB2 Version 9 can impact your migration. Also, you should be aware of the DB2 products that are no longer supported in DB2 Version 9 because migration from these products is not supported.

You need to be aware of these functionality changes and make plans for additional tasks before or after migration to deal with these changes. Some of these tasks are mentioned below.

DB2 administration tools

In Version 9, the DB2 administration tools such as, the Control Center, the Command Editor, and the Health Center are supported only on Windows on x86, Windows on X64 (AMD64/Intel® EM64T), Linux on x86 (IA32), and Linux on AMD64 /Intel EM64T. You need to invoke these tools only from a DB2 Client on any of the supported platforms.

DB2 JDBC type 3 driver

The DB2 JDBC type 3 driver was deprecated for DB2 UDB Version 8, and is no longer supported in DB2 Version 9. The driver archive file is not installed in DB2 Version 9. You must modify your Java applications and applets to use the IBM DB2 Driver for JDBC and SQLJ with type 4 connections. You don't have to install a DB2 client, you only need to copy the db2jcc.jar file.

Extended storage

Extended storage is no longer supported in DB2 Version 9. There is no impact on Linux because extended storage was not supported. There is also no impact on 64-bit operating systems because extended storage is not needed. If you want to allocate more memory than the virtual addressable memory limit on 32-bit Windows operating systems, consider using Address Windowing Extensions (AWE) buffer pools using the DB2_AWE registry variable as a workaround.

The SYSCAT.BUFFERPOOLS view structure is the same, the ESTORE column value is set to 'N' during migration because extended storage for buffer pools is no longer supported.

NetBIOS and SNA protocol

The NetBIOS protocol support is no longer available in DB2 Version 9. You need to remove the NetBIOS keyword in the DB2COMM registry variable otherwise the database manager returns an error when you start your instance. You cannot catalog nodes using this protocol; the **CATALOG**

NETBIOS NODE command returns an error because you are specifying an invalid protocol. You need to uncatalog any nodes that you cataloged specifying the NetBIOS protocol, and databases that you cataloged on these nodes. If you try to connect to any databases cataloged on a node using NetBIOS protocol, your connection request returns also an error because you are using an invalid protocol.

Although support for the SNA protocol was discontinued in DB2 UDB Version 8, this change in support was not enforced. In DB2 Version 9, you need to remove the SNA keyword in the DB2COMM registry variable, and uncatalog nodes and databases using SNA protocol. The **CATALOG APPC NODE** or **CATALOG APPN NODE** commands return an error because you are specifying an invalid protocol.

Raw logs

The use of raw devices for database logging is deprecated and will be removed in a future release. You need to change the *newlogpath* database configuration parameter setting to a disk device instead of a raw device. The following example illustrates how to change the *newlogpath* parameter setting:

```
$ 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. You can stop and restart the database manager to make this new setting effective.

Snapshot monitor SQL table functions

All the SQL table functions with names that start with **SNAPSHOT_** are deprecated in DB2 version 9. These functions are supported for compatibility only and will be removed in a future release.

New equivalent functions with similar names that start with **SNAP_GET_** are available. These new functions might have different parameters and additional columns therefore you need to review them before using the new function names on your applications.

DB2 products

DB2 UDB Version 8 installation has the option to install additional products and functionality. Support for some of these products has been dropped in DB2 Version 9. DB2 UDB Version 8 is also a prerequisite to installations of other DB2 products. A DB2 Version 9 installation does not support any of the following products as installation options or as a prerequisite component:

- DB2 Data Warehouse Center
- DB2 Data Warehouse Manager
- DB2 Information Catalog Center
- DB2 Data Links Manager
- DB2 DataJoiner

These DB2 products do not work on DB2 Version 9. If you installed any of these products on your DB2 server, you could uninstall them and then migrate your DB2 server to DB2 Version 9 without the DB2 product functionality. The **db2imigr** command fails if any of these products is installed but provides instructions on how to migrate to a DB2 Version 9

instance without the DB2 product functionality. Migrating to a DB2 Version 9 instance allows you to preserve your existing databases.

Also, if any of these products created database objects such as user-defined types, user-defined functions and stored procedures, these database objects remain unaltered in the database after you uninstall any of these products. Before you migrate the database, you might need to drop these objects from your database because they could cause the database migration to fail.

Related concepts:

- “Migration restrictions for DB2 servers” on page 20
- “Migration essentials for DB2 servers” on page 19
- “What’s new for V9.1: Deprecated functionality summary” in *What’s New*
- “What’s new for V9.1: Discontinued functionality summary” in *What’s New*
- “Version 9 incompatibilities with previous releases and changed behaviors” in *Administration Guide: Planning*
- “Supported drivers for JDBC and SQLJ” in *Developing Java Applications*
- “Node directory” in *Administration Guide: Implementation*

Related tasks:

- “Migrating DB2 Data Links Manager environments” on page 76

Related reference:

- “UNCATALOG DATABASE command” in *Command Reference*
- “UNCATALOG NODE command” in *Command Reference*
- “Snapshot monitor SQL Administrative Views” in *System Monitor Guide and Reference*

DB2 client support for migration

Migration to DB2 Version 9 client is supported only from DB2 UDB Version 8 clients, excluding Run-Time Client Lite and Runtime Client.

Because DB2 UDB Version 8 clients can connect to DB2 Version 9 servers, the recommendation is to migrate the DB2 server first and then to migrate the DB2 clients. This migration sequence ensures that your applications still work and that you avoid any known limitations when you connect from DB2 Version 9 clients to DB2 UDB Version 8 servers.

DB2 clients running any of the following products can connect to a server running DB2 Version 9 for Linux, UNIX, and Windows:

- DB2 UDB for Linux, UNIX, and Windows, Version 8
- DB2 Version 9 for Linux, UNIX, and Windows
- DB2 for z/OS® Version 7 and Version 8
- DB2 for iSeries™ Version 5
- DB2 for VM and VSE Version 7

DB2 client running DB2 Version 9 for Linux, UNIX, and Windows can connect to any of the following DB2 servers running:

- DB2 UDB for Linux, UNIX, and Windows, Version 8

- DB2 Version 9 for Linux, UNIX, and Windows
- DB2 Connect Version 8 and Version 9

Additionally, DB2 Connect Personal Edition can connect to the following database servers:

- DB2 for z/OS Version 7 and Version 8
- DB2 for iSeries Version 5
- DB2 for VM and VSE Version 7

New functionality is only available to DB2 clients running DB2 Version 9 for Linux, UNIX, and Windows that are connecting to DB2 servers running DB2 Version 9 for Linux, UNIX, and Windows or DB2 Connect Version 9.

Related concepts:

- Chapter 12, “Migration essentials for DB2 clients,” on page 107
- “Migration essentials for DB2 servers” on page 19
- Chapter 11, “Migration overview for DB2 clients,” on page 105

Related reference:

- “Supported combinations of client and server versions” in *Quick Beginnings for DB2 Clients*

Chapter 5. Pre-migration tasks

This chapter describes the pre-migration tasks for DB2 servers. It contains the following sections:

- “Pre-migration tasks for DB2 servers”
- “Verifying that your databases are ready for migration” on page 34
- “Backing up databases before migration” on page 36
- “Saving configuration information” on page 37
- “Increasing table space and log file sizes before migration” on page 39
- “Changing raw devices to block devices (Linux)” on page 41
- “Migrating in a test environment” on page 43
- “Capturing error and diagnostic information during migration” on page 44
- “Taking a DB2 server offline before migration” on page 46

Pre-migration tasks for DB2 servers

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

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

1. If you installed DB2 Net Search Extender (NSE) on your DB2 server, drop specific UDFs by using the following commands:

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

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

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

2. Verify that databases are ready for DB2 migration to identify any problems before the actual migration. You must resolve them before you proceed with the migration.
3. Back up your databases to be able to migrate them to a new migrated system or restore them in the original pre-migration system.
4. Save configuration information to have a record of your current configuration that you can compare with the configuration after the migration. You can also use this information to create new instances or databases using the same configuration that you had before migration.
5. Archive all of the DB2 log files, either for SQL replication if the log files are needed by the Capture program or for HADR replication if the log files are needed to create a standby database.

6. Review the disk space requirements to ensure that you have enough free disk space, temporary table space and log space for the migration 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 migration.
7. Linux only: Change raw devices to block devices.
8. Optional: Stop HADR on the primary and standby databases. You can migrate only the primary database.
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 DB2 UDB Version 8 copy. You do not need to backup standard code page conversion tables. Migrating your DB2 UDB Version 8 copy removes these tables because standard code page tables are contained in a DB2Version 9 library.
10. Optional: Migrate your DB2 server in a test environment to identify migration issues and to verify that applications, scripts, tools and routines work as expected before migrating your production environment.
11. Optional: Apply the DB2 UDB Version 8.2 latest FixPak. This allows you to take advantage of the fixes for migration commands and eliminate potential issues. Consider taking another backup of your databases after applying the FixPak.
12. Capture error and diagnostic information during migration. Additional diagnostic information is useful in problem determination when the normal migration logging information in the db2diag.log is not sufficient.
13. Take the DB2 server offline for migration.

Related concepts:

- “Migrating environments with specific characteristics” on page 63

Related tasks:

- “Migrating a DB2 server (Linux and UNIX)” on page 55
- “Migrating a DB2 server (Windows)” on page 47

Related reference:

- “ARCHIVE LOG command” in *Command Reference*
- “BACKUP DATABASE command” in *Command Reference*
- “db2ckmig - Database pre-migration tool command” in *Command Reference*
- “db2stop - Stop DB2 command” in *Command Reference*
- “GET DATABASE CONFIGURATION command” in *Command Reference*
- “GET DATABASE MANAGER CONFIGURATION command” in *Command Reference*

Verifying that your databases are ready for migration

Before you migrate your databases, it is important to use the **db2ckmig** command to verify that your databases are ready for migration.

This command verifies that all the following conditions are true:

- A catalogued database actually exists.
- A database is not in an inconsistent state.
- A database is not in a backup pending state.

- A database is not in rollforward pending state.
- Table spaces are in a normal state.
- A database does not contain user-defined types (UDTs) with the name XML, DATALINK, BINARY or VARBINARY.
- A database does not have orphan rows in system catalog tables.
- A database enabled as an HADR primary database allows successful connections.
- A database role is not standby for an HADR primary database.
- If SYSCATSPACE is a DMS table space and AUTORESIZE is not enabled, SYSCATSPACE has at least 50% free pages of total pages.

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

The **db2imigr** calls the **db2ckmig** command. The **db2imigr** fails if the **db2ckmig** command finds any of the conditions listed above are not true, and returns the DBI1205E error code.

Prerequisite:

Ensure that you have SYSADM authority.

Restriction:

In a partitioned database environment to verify that your databases are ready for migration, you must run the **db2ckmig** command on each database partition.

Procedure:

1. Log on as the DB2 instance owner that you want to migrate.
2. Stop the instance by running the **db2stop** command.
3. From a DB2 command line prompt move to the appropriate directory:
 - On UNIX or Linux, the path for this command is \$DB2DIR/bin/db2ckmig (DB2DIR is set to the location that you specified during the DB2 Version 9 installation).
 - On Windows, you need to insert the DB2 Version 9 product CD in the drive and change to the \db2\Windows\utilities directory.
4. Run the **db2ckmig** command to verify that the databases that are owned by the current instance are ready to be migrated and to generate a log file.

```
db2ckmig sample -l db2ckmig.log -u adminuser -p password
db2ckmig was successful. Database(s) can be migrated.
```

where *sample* is the database name, and db2ckmig.log is the log file that includes details on errors and warnings.

Each time you issue this command, it overwrites the existing log file. You can rename the log file to avoid losing the error details. You must correct these errors before you migrate.

When the **db2imigr** command runs the **db2ckmig** command, the log file specified is the migration.log file in the instance home directory for Linux and UNIX or in the current directory for Windows.

5. Ensure that the log file for **db2ckmig** command contains the following text: Version of DB2CKMIG being run: VERSION 9. This text confirms that you are running the correct level of the **db2ckmig** command.
6. Start the instance by running the **db2start** command.

Related concepts:

- “Migrating environments with specific characteristics” on page 63

Related tasks:

- “Migrating a DB2 server (Linux and UNIX)” on page 55

Related reference:

- “Pre-migration tasks for DB2 servers” on page 33
- “db2ckmig - Database pre-migration tool command” in *Command Reference*

Backing up databases before migration

Before you start the migration process to DB2 Version 9, it is strongly recommended that you perform a full *offline* backup of your databases. If an error occurs during the migration process, you need full database backups to recover and migrate your databases.

Prerequisites:

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

```
db2 LIST DATABASE DIRECTORY
```

Procedure:

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

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

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

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

```
db2 force application all
```

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

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

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

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

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

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

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

Related concepts:

- “Migrating environments with specific characteristics” on page 63
- “System administration authority (SYSADM)” in *Administration Guide: Implementation*

Related tasks:

- “Migrating a DB2 server (Linux and UNIX)” on page 55
- “Migrating a DB2 server (Windows)” on page 47

Related reference:

- “BACKUP DATABASE command” in *Command Reference*
- “db2ckbkp - Check backup command” in *Command Reference*
- “FORCE APPLICATION command” in *Command Reference*
- “LIST APPLICATIONS command” in *Command Reference*
- “Pre-migration tasks for DB2 servers” on page 33

Saving configuration information

It is strongly advised that you save your settings for database and database manager configuration parameters before DB2 server migration. You can use configuration parameter settings to verify that migration is complete, 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-migration differences in the database manager behavior or performance.

After you migrate to DB2 Version 9, you should compare the configuration parameter settings that you saved with the new post-migration settings. New parameters introduced in DB2 Version 9 have values that could impact the behavior or performance of your application. You should evaluate which values are more suitable for your application.

Prerequisites:

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 save your DB2 server configuration and diagnostic information:

1. Run the **db2support** command to **collect information from your DB2 servers**. This command allows you to collect information on 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` option collects the database system catalog, database and database manager configuration parameters settings, DB2 registry variables settings. The information collected is stored on a compressed zip file in output directory. A summary report in HTML format is included. You need to run this command for all your databases.

It is important that you keep this zip file after you complete the migration for several months. The information in the zip file can help in quickly resolving any performance issues with the new release.

2. **Save 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 > /migration/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 FOR ALL clause.

3. 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 **save 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.

4. 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 **save your settings for database configuration parameters** and redirect the command output to a file to save these settings for each database:

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

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

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

5. 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 -l -o sample_tbs.db2
```

6. 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 **save 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.

After migration, you should review your DB2 profile registry variable settings using the information that you saved before migration. In DB2 Version 9, there are new registry variables and some existing variables have new default values that can influence the database manager behavior and performance. Although these changes can improve your application, you must be aware of the new registry variables and evaluate which values best suit your application.

Related concepts:

- “Migration recommendations for DB2 servers” on page 23
- “Configuration parameters” in *Performance Guide*

Related tasks:

- “Migrating a DB2 server (Linux and UNIX)” on page 55
- “Migrating a DB2 server (Windows)” on page 47

Related reference:

- “LIST PACKAGES/TABLES command” in *Command Reference*
- “LIST TABLESPACES command” in *Command Reference*
- “Pre-migration tasks for DB2 servers” on page 33
- “Changes in DB2 registry variables, configuration parameters, and database physical design characteristics” on page 85
- “db2look - DB2 statistics and DDL extraction tool command” in *Command Reference*
- “db2set - DB2 profile registry command” in *Command Reference*
- “db2support - Problem analysis and environment collection tool command” in *Command Reference*
- “GET DATABASE CONFIGURATION command” in *Command Reference*
- “GET DATABASE MANAGER CONFIGURATION command” in *Command Reference*
- “Configuration parameters summary” in *Performance Guide*

Increasing table space and log file sizes before migration

Before you start migrating 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 migrate your databases.

Prerequisite:

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

Restriction:

Additional considerations are required in partitioned database environments to increase table space sizes, since 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 migrate:

```
db2 CONNECT TO sample
```

- Determine your table space disk usage using the following command:

```
db2 LIST TABLESPACES SHOW DETAIL
```

Collect the number of total pages, used pages, free pages and page size. Refer to the following table for a summary of the information obtained from the previous command:

Table 7. Table space information for sample database

Table space	Type	Total pages	Used pages	Free pages	Page size
SYSCATSPACE	SMS	8172	8172	N/A	4086
TEMPSPACE1	SMS	10	10	N/A	4086

- Increase the size of the system catalog table spaces. If you have an SMS table space, ensure that you have at least the same amount of used pages in free disk space; in this example, about 32 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:

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

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)"
```

- Increase the size of the temporary table spaces. 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 64 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 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 total pages for the system catalog table space in free disk space. If MAXSIZE is set to an integer value, ensure that this value is at least twice the amount of total pages. The following query returns the current size (amount of total pages in bytes) and MAXSIZE of the table space TEMPSpace1 in the SAMPLE database:

```
db2 SELECT TBSP_CURRENT_SIZE, TBSP_MAX_SIZE
      FROM table(SNAP_GET_TBSP_PART('SAMPLE', -1)) T
      WHERE TBSP_NAME = 'TEMPSPACE1'
```

If TBSP_MAX_SIZE is less than twice the value of TBSP_CURRENT_SIZE, then you need to increase MAXSIZE using the ALTER TABLESPACE statement:

```
db2 ALTER TABLESPACE TEMPSpace1 MAXSIZE (<TBSP_CURRENT_SIZE*2/1024>) K
```

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

- Determine the current log space size using the GET DATABASE CONFIGURATION command. Record the values for *logfilesiz*, *logprimary*, and *logsecond* database configuration parameters:


```

db2 GET DB CFG FOR sample |grep '(LOG[FPS]
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 LOGPRIMARY current value * 2
db2 UPDATE DB CFG FOR sample using LOGSECOND current value * 2

```

If you already have a large log space, you might not need to increase it. Perhaps consider just increasing the number of secondary log files as a safety measure only if you have created a very large number of database objects on your database.

7. Optional: Enable infinite active log 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 migration fails, the time to rollback the transactions will depend on how many archived logs need to be retrieved. The following command shows an example on how to enable archive logging to disk and infinite logging:

```

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

```

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

Although these are dynamic parameters, all applications must disconnect from this database before the new values become effective.

Related concepts:

- “Space requirements for log files” in *Administration Guide: Planning*
- “Table spaces in database partition groups” in *Administration Guide: Implementation*

Related tasks:

- “Migrating a DB2 server (Linux and UNIX)” on page 55
- “Migrating a DB2 server (Windows)” on page 47

Related reference:

- “Configuration parameters for database logging” in *Data Recovery and High Availability Guide and Reference*
- “Pre-migration tasks for DB2 servers” on page 33
- “GET DATABASE CONFIGURATION command” in *Command Reference*
- “LIST TABLESPACES command” in *Command Reference*
- “UPDATE DATABASE CONFIGURATION command” in *Command Reference*
- “ALTER TABLESPACE statement” in *SQL Reference, Volume 2*

Changing raw devices to block devices (Linux)

If you are still using raw (character) devices as containers for table spaces or log files, you must change your raw (character) devices to raw block devices before you migrate to DB2 Version 9. The previous raw I/O method that required binding the block device to a raw (character) device using the raw utility is deprecated in

DB2 Version 9, 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 the using the raw (character) device method.

Prerequisite:

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

Restriction:

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

Procedure:

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

```
db2 CONNECT TO sample
db2 QUIESCE DATABASE DEFER FORCE CONNECTIONS
db2 DEACTIVATE DATABASE database-alias
```
3. Use the raw -a system command to see which raw bindings you defined. This information will help you determine the block device you should use to replace a raw device for each container on your table spaces.
4. Create a configuration file for the **db2relocatedb** command. Use the clauses CONT_PATH and LOG_DIR to specify the old value with the new value. For example, you can create the moveraw.cfg file with the following content:

```
DB_NAME=SAMPLE
DB_PATH=/databases/SAMPLE
INSTANCE=db2inst1
NODENUM=0
LOG_DIR=/dev/raw/lograw,/dev/sda5
CONT_PATH=/dev/raw/raw1,/dev/sda1
CONT_PATH=/dev/raw/raw2,/dev/sda2
```
5. Execute the **db2relocatedb** command to change the configuration of the database files:

```
db2relocatedb -f moveraw.cfg
```
6. Activate your database:

```
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:

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

If you are restoring from a DB2 UDB Version 8 backup in DB2 Version 9, you must do a redirected restore to indicate block devices instead of raw character devices for your containers and log path.

Related concepts:

- “Redefining table space containers during a restore operation (redirected restore)” in *Data Recovery and High Availability Guide and Reference*

Related tasks:

- “Setting up a direct disk access device on Linux” in *Administration Guide: Implementation*
- “Attaching a direct disk access device” in *Administration Guide: Implementation*
- “Backing up databases before migration” on page 36
- “Migrating a DB2 server (Linux and UNIX)” on page 55

Related reference:

- “Pre-migration tasks for DB2 servers” on page 33
- “ACTIVATE DATABASE command” in *Command Reference*
- “db2relocatedb - Relocate database command” in *Command Reference*
- “QUIESCE command” in *Command Reference*

Migrating in a test environment

If you migrate to DB2 Version 9 in a test environment before you migrate your production environment, you are able to address problems during the migration process more effectively. You are also able to evaluate the impact of changes introduced in DB2 Version 9. This method allows you to verify that applications, scripts, tools and maintenance procedures work properly before migrating your production environment. In addition, you can assess how long the migration process takes and solidify your migration plan.

DB2 Version 9 and DB2 UDB Version 8 can coexist on the same system. You can install DB2 Version 9 as a separate test environment, while your applications are still up and running under DB2 UDB Version 8 in a production environment on the same system.

In this parallel test environment, you can create your DB2 Version 9 test instances and databases to test your applications. You can also migrate your current databases to this test environment by restoring them from a DB2 UDB Version 8 offline full backup, and identify any issues with the database migration process.

Prerequisites:

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

Procedure:

To mimic your production environment in a test environment, you need to do the following tasks:

1. Install DB2 UDB Version 8.
2. Re-create your instances.
3. Re-create your databases.
4. Perform pre-migration tasks.
5. Install DB2 Version 9.
6. Migrate your instances.

7. Migrate your databases.
8. Verify the migration.
9. Test your applications, scripts, tools and maintenance procedures.

Related concepts:

- “What’s new for V9.1: Changes in existing functionality summary” in *What’s New*
- “What’s new for V9.1: Discontinued functionality summary” in *What’s New*
- “Migrating environments with specific characteristics” on page 63
- “Migration essentials for DB2 servers” on page 19

Related tasks:

- “Migrating instances” on page 48
- “Migrating databases” on page 52
- “Verifying migration of DB2 servers” on page 98
- “An overview of installing your DB2 product (Linux and UNIX)” in *Quick Beginnings for DB2 Servers*
- “An overview of installing your DB2 product (Windows)” in *Quick Beginnings for DB2 Servers*
- “Migrating a DB2 server (Linux and UNIX)” on page 55
- “Migrating a DB2 server (Windows)” on page 47
- “Creating a database” in *Administration Guide: Implementation*
- “Creating additional instances” in *Administration Guide: Implementation*

Related reference:

- “Pre-migration tasks for DB2 servers” on page 33

Capturing error and diagnostic information during migration

In DB2 Version 9, all significant migration events are logged in the db2diag.log file when the diagnostic error level is set to 3 (default value). Migration failure messages are also logged in the db2diag.log file when the diagnostic error level is set to 1 or higher. You need to set the *diaglevel* database manager configuration parameter to 3 or higher in order to capture information about migration events.

You can change the diagnostic error level to log additional information during the migration process, this additional information can be helpful in problem determination. Before migration to DB2 Version 9, you should set the *diaglevel* database manager configuration parameter to 4 in order to capture additional information in the db2diag.log file. A value of 4 logs all errors, warnings, and informational messages.

Prerequisite:

Ensure that you have SYSADM authority.

Procedure:

To change the diagnostic level to log additional information about the migration process in the db2diag.log file:

1. Log on as the instance owner.

2. Take note of the current setting for the *diaglevel* database manager configuration parameter:

```
db2 GET DBM CONFIGURATION
...
Diagnostic error capture level          (DIAGLEVEL) = 3
...
```

You need to be aware of this value to restore this diagnostic level after migration is completed. The default value for the *diaglevel* parameter is 3.

3. Set the *diaglevel* database manager configuration parameter to 4:

```
db2 UPDATE DBM CONFIGURATION USING diaglevel 4
```

- 4.

Restart the instance so that the change takes effect:

```
db2stop
db2start
```

5. Migrate database.

6. Restore the diagnostic error level to its original setting. The following example restores the diagnostic error level to the default value:

```
db2 UPDATE DBM CONFIGURATION USING diaglevel 3
```

The *diagpath* database manager parameter allows you to specify the fully qualified path for DB2 diagnostic information. This directory contains the administration notification log file and the *db2diag.log* file, it could possibly contain dump files, trap files, and an alert log files, depending on your operating system. It is recommended that you keep the default value. If you need to change the *diagpath* parameter, use the following command:

```
db2 UPDATE DBM CONFIGURATION USING diagpath directory
```

Where *directory* represents the location to store all error and diagnostic log files. Use a centralized location, especially if you have multiple database instances.

Related concepts:

- “First failure data capture information” in *Troubleshooting Guide*

Related tasks:

- “Migrating a DB2 server (Linux and UNIX)” on page 55
- “Migrating a DB2 server (Windows)” on page 47

Related reference:

- “diaglevel - Diagnostic error capture level configuration parameter” in *Performance Guide*
- “diagpath - Diagnostic data directory path configuration parameter” in *Performance Guide*
- “Pre-migration tasks for DB2 servers” on page 33
- “GET DATABASE MANAGER CONFIGURATION command” in *Command Reference*
- “UPDATE DATABASE CONFIGURATION command” in *Command Reference*

Taking a DB2 server offline before migration

This task describes how to take your DB2 UDB Version 8 server offline for DB2 Version 9 migration. Before you can continue with the migration process, you must stop the DB2 license service, stop all command line processor sessions, disconnect applications and users, and stop the database manager.

Prerequisites:

- Your system must meet the installation requirements for DB2 Version 9 before starting the migration process.
- You must have SYSADM authority.

Procedure:

To take your server offline:

1. Stop the DB2 license service:

```
db2licd -end
```
2. On Windows 2000, the properties of a service can be set so that it restarts if the service fails. If the *restart on failure* option is set for any DB2 services, it must be disabled before proceeding.
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. 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
```
5. When all applications and users are disconnected, stop each database manager instance:  

```
db2stop
```

### Related tasks:

- “Migrating a DB2 server (Linux and UNIX)” on page 55
- “Migrating a DB2 server (Windows)” on page 47

### Related reference:

- “Pre-migration tasks for DB2 servers” on page 33
- “db2licm - License management tool command” in *Command Reference*
- “db2stop - Stop DB2 command” in *Command Reference*
- “FORCE APPLICATION command” in *Command Reference*
- “LIST APPLICATIONS command” in *Command Reference*

---

## Chapter 6. Migrating DB2 servers (Windows)

This chapter describes how to migrate your DB2 servers on Windows. It contains the following sections:

- “Migrating a DB2 server (Windows)”
- “Migrating instances” on page 48
- “Migrating the DB2 Administration Server (DAS)” on page 50
- “Migrating databases” on page 52

---

### Migrating a DB2 server (Windows)

Migration is required if you have instances and databases running on DB2 UDB Version 8 that you want to run on DB2 Version 9. If you choose to automatically migrate your existing DB2 UDB Version 8 copy during the DB2 Version 9 installation, your instances and DB2 administration server (DAS) are migrated but you still need to migrate your databases after installation. If you choose to install a new DB2 Version 9 copy, you must manually migrate your instances, your DAS and databases.

This migration task describes the steps for direct migration from DB2 UDB Version 8 to DB2 Version 9. Review “Migrating environments with specific characteristics” on page 63 and determine which task applies better to your environment.

#### Prerequisites:

- Ensure that you have Local Administrator authority.
- Review migration recommendations and disk space requirements.
- Perform pre-migration tasks.

#### Restrictions:

- This procedure applies only to DB2 server migration from 32-bit instances to DB2 Version 9 32-bit instances, or from 64-bit instances to DB2 Version 9 64-bit instances. In DB2 Version 9, the instance is 32-bit only in 32-bit Windows on x86 or X64 (if you Install DB2 Version 9 32-bit database product). The instance is 64-bit only in 64-bit Windows on X64.
- Migration is supported only from DB2 UDB Version 8.
- Direct migration is not supported from DB2 UDB Version 7 or earlier. You must migrate first to DB2 UDB Version 8.
- Additional migration restrictions apply. Review the complete list.

#### Procedure:

To migrate to DB2 Version 9:

1. Log on as a Local Administrator.
2. Install DB2 Version 9 by running the setup.exe command to launch the DB2 Setup wizard. You have two options:
  - Select the option Migrate on the Install a Product panel to migrate an existing DB2 UDB Version 8 copy. Your DB2 UDB Version 8 copy is removed, and all your instances and your DAS are automatically migrated.

You will get a warning that recommends that you to run the **db2ckmig** command if you have local databases. If you completed the pre-migration tasks, ignore this warning and continue the migration. Otherwise, verify that your databases are ready for DB2 migration before continuing with the installation.

- Select the option Install New on the Install a Product panel. This option creates a new copy of DB2 Version 9, and you must migrate your instances after installation.
3. Optional: Migrate the DAS if you want to keep your existing DAS configuration and to administer your DB2 Version 9 instances using the Control Center.
  4. Migrate your databases.

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

**Related concepts:**

- “Migrating environments with specific characteristics” on page 63
- “Migration recommendations for DB2 servers” on page 23
- “Migration restrictions for DB2 servers” on page 20
- “Migration essentials for DB2 servers” on page 19
- “System administration authority (SYSADM)” in *Administration Guide: Implementation*

**Related tasks:**

- “Installing DB2 servers (Windows)” in *Quick Beginnings for DB2 Servers*
- “Verifying that your databases are ready for migration” on page 34
- “Migrating instances” on page 48
- “Migrating the DB2 Administration Server (DAS)” on page 50
- “Migrating databases” on page 52
- “Verifying migration of DB2 servers” on page 98
- “Migrating DB2 32-bit servers to 64-bit systems (Windows)” on page 64
- “An overview of installing your DB2 product (Windows)” in *Quick Beginnings for DB2 Servers*

**Related reference:**

- “Disk space requirements for DB2 server migration” on page 26
- “Pre-migration tasks for DB2 servers” on page 33
- “Post-migration tasks for DB2 servers” on page 81
- “db2ckmig - Database pre-migration tool command” in *Command Reference*
- “Multiple DB2 copies roadmap” in *Administration Guide: Implementation*

---

## Migrating instances

As part of the overall process of migrating your DB2 UDB Version 8 server to DB2 Version 9, you must migrate your instances. On Linux and UNIX, you must manually migrate them. On Windows, you must manually migrate them if you did not choose to automatically migrate your existing DB2 UDB Version 8 copy during the DB2 Version 9 installation.



To manually migrate your DB2 UDB Version 8 instances, use the **db2imigr** command.

**Prerequisites:**

- You must have root authority on Linux and UNIX operating systems or Local Administrator authority on Windows.
- Before running the **db2imigr** command, it is recommended:
  - On UNIX, ensure that there is 20 MB of free space in the /tmp directory. The instance migration trace file is written to /tmp.
  - Verify that databases are ready for DB2 migration.

**Restrictions:**

- Direct migration is not supported from DB2 UDB Version 7 or earlier. You must migrate first to DB2 UDB Version 8.
- Additional migration restrictions apply. Review the complete list for instance migration.

**Procedure:**

To migrate an instance:

1. Stop your DB2 UDB Version 8 instances by running the **db2stop** command:  
db2stop
2. Log on as root authority on Linux and UNIX operating systems or Local Administrator authority on Windows.
3. Migrate instances by running the **db2imigr** command.

**On Linux and UNIX operating systems:**

```
$DB2DIR/instance/db2imigr -u fencedID InstName
```

where

**DB2DIR**

is set to the location you specified during DB2 Version 9 installation. The default installation path for UNIX is /opt/IBM/db2/V9.1, and for Linux is /opt/ibm/db2/V9.1.

**-u fencedID**

Is the user under which the fenced user-defined functions (UDFs) and stored procedures will run.

*InstName*

Is the login name of the instance owner.

**On Windows operating systems:**

```
"%DB2PATH%"bin\db2imigr InstName /u:user,password
```

where

**DB2PATH**

is set to the location you specified during DB2 Version 9 installation. The default installation path for Windows is Program Files\IBM\SQLLIB.

**/u: user,password**

are the user name and password under which the DB2 service will run.

*InstName*

Is the login name of the instance owner.

The **db2imigr** command calls implicitly the **db2ckmig** command indicating migration.log as the log file in the instance home directory for Linux and UNIX or in the current directory, where you are running the **db2imigr** command, for Windows. The **db2imigr** does not run as long as the **db2ckmig** command reports errors. Check the log file if you encounter any errors.

4. Log on as the instance owner.
5. Restart your instance by running the **db2start** command:  

```
db2start
```
6. Verify that your instance is running on to DB2 Version 9 by running the **db2level** command:  

```
db2level
```

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

**Related concepts:**

- "Migration restrictions for DB2 servers" on page 20
- "Migration essentials for DB2 servers" on page 19

**Related tasks:**

- "Verifying that your databases are ready for migration" on page 34

**Related reference:**

- "db2ckmig - Database pre-migration tool command" in *Command Reference*
- "db2icrt - Create instance command" in *Command Reference*
- "db2imigr - Migrate instance command" in *Command Reference*

---

## Migrating the DB2 Administration Server (DAS)

As part of the overall migration process to DB2 Version 9, you can migrate your DB2 Administration Server (DAS) to keep your existing DAS configuration. Otherwise, you can drop your existing DAS and create a new DAS in DB2 Version 9. You only need a DAS running on DB2 Version 9 to use the Control Center for remote administration of DB2 Version 9 instances, job management and job scheduling.

On Windows, if you chose to automatically migrate your DB2 UDB Version 8 installation, the DAS is also migrated along with your instances.

After installing DB2 Version 9, you can manually migrate the DAS by running the **dasmigr** command.

**Prerequisite:**

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

**Restrictions:**

- Migration is supported only from DB2 UDB Version 8.
- You can have only one DAS per DB2 server.

## Procedure:

To migrate the DAS:

1. On Linux and UNIX operating systems, log on as the DAS owner and stop the DAS using the **db2admin** command as follows:

```
db2admin stop
```

On Windows operating systems the **dasmigr** command stops and starts the DAS .

2. Log on as root on Linux and UNIX operating systems or Local Administrator authority on Windows.
3. Migrate the DAS under DB2 UDB Version 8 by running the **dasmigr** command:

### On Linux and UNIX operating systems:

```
$DB2DIR/instance/dasmigr
```

where DB2DIR is the location you specified during DB2 Version 9 installation.

### On Windows operating systems:

```
%DB2PATH%\bin\dasmigr
```

where DB2PATH is the location you specified during DB2 Version 9 installation.

You can now use the Control Center for remote administration of DB2 Version 9 instances, as well as DB2 UDB Version 8 instances.

4. On Linux and UNIX operating systems, log on as the DAS owner and start the DAS using the **db2admin** command as follows:

```
db2admin start
```

You can verify that the DAS was migrated to DB2 Version 9 by running the **db2daslevel** command.

5. If you created a tools catalog database on your DB2 UDB Version 8 system and want to use your existing scripts and schedules on the DB2 Version 9 Control Center, you have two options:
  - If you migrated to DB2 Version 9 the instance that owns the Version 8 tools catalog database, you must migrate this database and verify that the DAS is configured to access it.
  - If you did not migrate to DB2 Version 9 the instance that owns the Version 8 tools catalog database, simply verify that the DAS is configured to access this database.

Run the GET ADMIN CFG command to display the current configuration settings for the tools database catalog:

```
db2 get admin cfg
```

```
Admin Server Configuration
...
Tools Catalog Database (TOOLS CAT_DB) = toolsdb
Tools Catalog Database Instance (TOOLS CAT_INST) = db2inst1
Tools Catalog Database Schema (TOOLS CAT_SCHEMA) = cc
Scheduler User ID =
```

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

**Related concepts:**

- “Migration essentials for DB2 servers” on page 19

**Related tasks:**

- “Creating a DB2 administration server (DAS)” in *Administration Guide: Implementation*

**Related reference:**

- “dasmigr - Migrate the DB2 administration server command” in *Command Reference*

---

## Migrating databases

After you migrated your instances to DB2 Version 9, you need to migrate each database under each instance.

**Prerequisites:**

- Ensure that you have SYSADM authority.
- Backup your databases using the BACKUP DATABASE command before you migrate the instance.
- You must have DB2 Version 9 installed, and migrated an instance to DB2 Version 9.

**Restrictions:**

- Migration is supported only from DB2 UDB Version 8.
- Additional migration restrictions apply. Review the complete list for database migration.

**Procedure:**

To migrate a DB2 database:

1. Log on as the instance owner or a user with SYSADM authority.
2. Migrate the database using the MIGRATE DATABASE command:  

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

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

You can also migrate your databases using the RESTORE DATABASE command instead of the MIGRATE DATABASE command.

3.

If the database migration fails and returns the error message SQL1704N with a reason code that describes the cause of the failure. Find this SQL error code in the Message Reference to read the list of the possible solutions for each reason code. One of the most common causes of migration failure is that the log file space is not large enough, in which case the following error is returned:

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

You must and execute the MIGRATE DATABASE command again. Once the database migration is complete reset the value of *logfilesiz*, *logprimary*, and *logsecond* database configuration parameters.

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

4. If the database migration returns the warning message SQL1243W, you need to drop or rename the SYSTOOLS.DB2LOOK\_INFO table. Otherwise, the ALTER TABLE and COPY SCHEMA statements will fail to run.

Check if the SYSTOOLS.DB2LOOK\_INFO table exists by running the following command:

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

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

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

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

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

5. Compare your database configuration settings after migration with the configuration settings you had before you migrated your database. Verify the following settings and database information are the same:

- database configuration parameter settings
- table spaces information
- packages information for your applications only

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

6. Verify your database migration is successful. Connect to the migrated databases and issue a small query:

```
db2 connect to sample
```

```
Database Connection Information
```

```
Database server = DB2/AIX64 9.1.0
SQL authorization ID = TESTDB2
Local database alias = SAMPLE
```

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

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

```
cd samplefile-dir-clp
db2 connect to sample
db2 -tvf testdata.db2
```

where *samplefile-dir-clp* is DB2DIR/samples/clp on Linux and UNIX and DB2DIR\samples\clp on Windows, DB2DIR represents the location specified during DB2 Version 9 installation, and sample is the database name.

After migrating a DB2 database, performing the recommended post-migration tasks ensures a successful database migration.

#### Related concepts:

- “Migration restrictions for DB2 servers” on page 20
- “Migration essentials for DB2 servers” on page 19

#### Related tasks:

- “Increasing table space and log file sizes before migration” on page 39
- “Saving configuration information” on page 37
- “Verifying migration of DB2 servers” on page 98

**Related reference:**

- “Post-migration tasks for DB2 servers” on page 81
- “Changes in DB2 registry variables, configuration parameters, and database physical design characteristics” on page 85
- “Pre-migration tasks for DB2 servers” on page 33
- “MIGRATE DATABASE command” in *Command Reference*
- “RESTORE DATABASE command” in *Command Reference*

---

## Chapter 7. Migrating DB2 servers (Linux and UNIX)

This chapter describes how to migrate your DB2 servers on Linux and UNIX. It contains the following sections:

- “Migrating a DB2 server (Linux and UNIX)”
- “Migrating instances” on page 56
- “Migrating the DB2 Administration Server (DAS)” on page 58
- “Migrating databases” on page 60

---

### Migrating a DB2 server (Linux and UNIX)

Migration is required if you have instances and databases running on DB2 UDB Version 8 that you want to run on DB2 Version 9. You must manually migrate your instances, your DB2 administration server (DAS), and your databases after you install DB2 Version 9.

This migration task describes the steps for direct migration from DB2 UDB Version 8 to DB2 Version 9. Review “Migrating environments with specific characteristics” on page 63 and determine which task applies better to your environment.

#### Prerequisites:

Before migrating the DB2 server:

- Ensure that you have root access.
- Review the System requirements Web page for DB2 database product installation. The prerequisites for Linux and UNIX operating systems have changed.
- Review migration recommendations and disk space requirements.
- Perform pre-migration tasks.

#### Restrictions:

- This procedure applies only to DB2 server migration from 32-bit instances to DB2 Version 9 32-bit instances, or from 64-bit instances to DB2 Version 9 64-bit instances. In DB2 Version 9, the instance is 32-bit only in Linux on x86-32 and the instance is 64-bit on all the other supported Linux and UNIX operating systems.
- Migration is supported only from DB2 UDB Version 8.
- Direct migration is not supported from DB2 UDB Version 7 or earlier. You must migrate first to DB2 UDB Version 8.
- Additional migration restrictions apply. Review the complete list.

#### Procedure:

To migrate to DB2 Version 9:

1. Log on as root.
2. Install DB2 Version 9. Run the **db2setup** command and select Install New on the Install a Product panel to install a new copy of DB2 Version 9.

3. Migrate instances from the same installation path that you indicated during DB2 Version 9 installation.
4. Optional: Migrate the DB2 Administration Server if you want to keep your existing configuration and to administer your DB2 Version 9 instances using the Control Center.
5. Migrate databases.

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

**Related concepts:**

- “Migrating environments with specific characteristics” on page 63
- “Migration recommendations for DB2 servers” on page 23
- “Migration restrictions for DB2 servers” on page 20
- “Migration essentials for DB2 servers” on page 19
- “System administration authority (SYSADM)” in *Administration Guide: Implementation*

**Related tasks:**

- “Installing DB2 servers (Linux and UNIX)” in *Quick Beginnings for DB2 Servers*
- “Migrating instances” on page 48
- “Migrating the DB2 Administration Server (DAS)” on page 50
- “Migrating databases” on page 52
- “Verifying migration of DB2 servers” on page 98
- “Migrating DB2 32-bit servers to 64-bit systems (Linux and UNIX)” on page 65
- “An overview of installing your DB2 product (Linux and UNIX)” in *Quick Beginnings for DB2 Servers*

**Related reference:**

- “Disk space requirements for DB2 server migration” on page 26
- “Pre-migration tasks for DB2 servers” on page 33
- “Post-migration tasks for DB2 servers” on page 81

---

## Migrating instances

As part of the overall process of migrating your DB2 UDB Version 8 server to DB2 Version 9, you must migrate your instances. On Linux and UNIX, you must manually migrate them. On Windows, you must manually migrate them if you did not choose to automatically migrate your existing DB2 UDB Version 8 copy during the DB2 Version 9 installation.

To manually migrate your DB2 UDB Version 8 instances, use the **db2imigr** command.

**Prerequisites:**

- You must have root authority on Linux and UNIX operating systems or Local Administrator authority on Windows.
- Before running the **db2imigr** command, it is recommended:



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

**Restrictions:**

- Direct migration is not supported from DB2 UDB Version 7 or earlier. You must migrate first to DB2 UDB Version 8.
- Additional migration restrictions apply. Review the complete list for instance migration.

**Procedure:**

To migrate an instance:

1. Stop your DB2 UDB Version 8 instances by running the **db2stop** command:  
db2stop
2. Log on as root authority on Linux and UNIX operating systems or Local Administrator authority on Windows.
3. Migrate instances by running the **db2imigr** command.

**On Linux and UNIX operating systems:**

```
$DB2DIR/instance/db2imigr -u fencedID InstName
```

where

**DB2DIR**

is set to the location you specified during DB2 Version 9 installation. The default installation path for UNIX is /opt/IBM/db2/V9.1, and for Linux is /opt/ibm/db2/V9.1.

**-u fencedID**

Is the user under which the fenced user-defined functions (UDFs) and stored procedures will run.

*InstName*

Is the login name of the instance owner.

**On Windows operating systems:**

```
"%DB2PATH%"bin\db2imigr InstName /u:user,password
```

where

**DB2PATH**

is set to the location you specified during DB2 Version 9 installation. The default installation path for Windows is Program Files\IBM\SQLLIB.

**/u: user,password**

are the user name and password under which the DB2 service will run.

*InstName*

Is the login name of the instance owner.

The **db2imigr** command calls implicitly the **db2ckmig** command indicating migration.log as the log file in the instance home directory for Linux and UNIX or in the current directory, where you are running the **db2imigr** command, for Windows. The **db2imigr** does not run as long as the **db2ckmig** command reports errors. Check the log file if you encounter any errors.

4. Log on as the instance owner.
5. Restart your instance by running the **db2start** command:  
`db2start`
6. Verify that your instance is running on to DB2 Version 9 by running the **db2level** command:  
`db2level`

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

**Related concepts:**

- "Migration restrictions for DB2 servers" on page 20
- "Migration essentials for DB2 servers" on page 19

**Related tasks:**

- "Verifying that your databases are ready for migration" on page 34

**Related reference:**

- "db2ckmig - Database pre-migration tool command" in *Command Reference*
- "db2icrt - Create instance command" in *Command Reference*
- "db2imigr - Migrate instance command" in *Command Reference*

---

## Migrating the DB2 Administration Server (DAS)

As part of the overall migration process to DB2 Version 9, you can migrate your DB2 Administration Server (DAS) to keep your existing DAS configuration. Otherwise, you can drop your existing DAS and create a new DAS in DB2 Version 9. You only need a DAS running on DB2 Version 9 to use the Control Center for remote administration of DB2 Version 9 instances, job management and job scheduling.

On Windows, if you chose to automatically migrate your DB2 UDB Version 8 installation, the DAS is also migrated along with your instances.

After installing DB2 Version 9, you can manually migrate the DAS by running the **dasmigr** command.

**Prerequisite:**

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

**Restrictions:**

- Migration is supported only from DB2 UDB Version 8.
- You can have only one DAS per DB2 server.

**Procedure:**

To migrate the DAS:

1. On Linux and UNIX operating systems, log on as the DAS owner and stop the DAS using the **db2admin** command as follows:  
`db2admin stop`

On Windows operating systems the **dasmigr** command stops and starts the DAS .

2. Log on as root on Linux and UNIX operating systems or Local Administrator authority on Windows.
3. Migrate the DAS under DB2 UDB Version 8 by running the **dasmigr** command:

**On Linux and UNIX operating systems:**

```
$DB2DIR/instance/dasmigr
```

where DB2DIR is the location you specified during DB2 Version 9 installation.

**On Windows operating systems:**

```
%DB2PATH%\bin\dasmigr
```

where DB2PATH is the location you specified during DB2 Version 9 installation.

You can now use the Control Center for remote administration of DB2 Version 9 instances, as well as DB2 UDB Version 8 instances.

4. On Linux and UNIX operating systems, log on as the DAS owner and start the DAS using the **db2admin** command as follows:

```
db2admin start
```

You can verify that the DAS was migrated to DB2 Version 9 by running the **db2daslevel** command.

5. If you created a tools catalog database on your DB2 UDB Version 8 system and want to use your existing scripts and schedules on the DB2 Version 9 Control Center, you have two options:
  - If you migrated to DB2 Version 9 the instance that owns the Version 8 tools catalog database, you must migrate this database and verify that the DAS is configured to access it.
  - If you did not migrate to DB2 Version 9 the instance that owns the Version 8 tools catalog database, simply verify that the DAS is configured to access this database.

Run the GET ADMIN CFG command to display the current configuration settings for the tools database catalog:

```
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 =
```

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

**Related concepts:**

- “Migration essentials for DB2 servers” on page 19

**Related tasks:**

- “Creating a DB2 administration server (DAS)” in *Administration Guide: Implementation*

**Related reference:**

- “dasmigr - Migrate the DB2 administration server command” in *Command Reference*

---

## Migrating databases

After you migrated your instances to DB2 Version 9, you need to migrate each database under each instance.

**Prerequisites:**

- Ensure that you have SYSADM authority.
- Backup your databases using the BACKUP DATABASE command before you migrate the instance.
- You must have DB2 Version 9 installed, and migrated an instance to DB2 Version 9.

**Restrictions:**

- Migration is supported only from DB2 UDB Version 8.
- Additional migration restrictions apply. Review the complete list for database migration.

**Procedure:**

To migrate a DB2 database:

1. Log on as the instance owner or a user with SYSADM authority.
2. Migrate the database using the MIGRATE DATABASE command:

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

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

You can also migrate your databases using the RESTORE DATABASE command instead of the MIGRATE DATABASE command.

- 3.

If the database migration fails and returns the error message SQL1704N with a reason code that describes the cause of the failure. Find this SQL error code in the Message Reference to read the list of the possible solutions for each reason code. One of the most common causes of migration failure is that the log file space is not large enough, in which case the following error is returned:

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

You must and execute the MIGRATE DATABASE command again. Once the database migration is complete reset the value of *logfilesiz*, *logprimary*, and *logsecond* database configuration parameters.

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

4. If the database migration returns the warning message SQL1243W, you need to drop or rename the SYSTOOLS.DB2LOOK\_INFO table. Otherwise, the ALTER TABLE and COPY SCHEMA statements will fail to run.

Check if the SYSTOOLS.DB2LOOK\_INFO table exists by running the following command:

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

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

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

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

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

5. Compare your database configuration settings after migration with the configuration settings you had before you migrated your database. Verify the following settings and database information are the same:
  - database configuration parameter settings
  - table spaces information
  - packages information for your applications only

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

6. Verify your database migration is successful. Connect to the migrated databases and issue a small query:

```
db2 connect to sample
```

```
Database Connection Information
```

```
Database server = DB2/AIX64 9.1.0
SQL authorization ID = TESTDB2
Local database alias = SAMPLE
```

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

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

```
cd samplefile-dir-clp
db2 connect to sample
db2 -tvf testdata.db2
```

where *samplefile-dir-clp* is DB2DIR/samples/clp on Linux and UNIX and DB2DIR\samples\clp on Windows, DB2DIR represents the location specified during DB2 Version 9 installation, and sample is the database name.

After migrating a DB2 database, performing the recommended post-migration tasks ensures a successful database migration.

#### Related concepts:

- “Migration restrictions for DB2 servers” on page 20
- “Migration essentials for DB2 servers” on page 19

#### Related tasks:

- “Increasing table space and log file sizes before migration” on page 39
- “Saving configuration information” on page 37
- “Verifying migration of DB2 servers” on page 98

#### Related reference:

- “Post-migration tasks for DB2 servers” on page 81

- “Changes in DB2 registry variables, configuration parameters, and database physical design characteristics” on page 85
- “Pre-migration tasks for DB2 servers” on page 33
- “MIGRATE DATABASE command” in *Command Reference*
- “RESTORE DATABASE command” in *Command Reference*

---

## Chapter 8. Migrating environments with specific characteristics

This chapter describes how to migrate environments with specific characteristics. It contains the following sections:

- “Migrating environments with specific characteristics”
- “Migrating DB2 32-bit servers to 64-bit systems (Windows)” on page 64
- “Migrating DB2 32-bit servers to 64-bit systems (Linux and UNIX)” on page 65
- “Migrating to a new DB2 server” on page 67
- “Migrating partitioned database environments” on page 69
- “Migrating from a system with multiple DB2 copies (Linux and UNIX)” on page 71
- “Migrating from DB2 UDB Version 7 servers (Windows)” on page 73
- “Migrating from DB2 UDB Version 7 servers (Linux and UNIX)” on page 73
- “Migrating DB2 servers in Microsoft Cluster Server environments” on page 74
- “Migrating DB2 Data Links Manager environments” on page 76
- “Migrating from XML Extender” on page 78
- “Migrating from non-DB2 relational database management systems” on page 79

---

### Migrating environments with specific characteristics

There are many factors that can impact the overall migration process, and the complexity of your environment is one of these factors. If you installed multiple DB2 product components, if you are migrating from a 32-bit operating system to a 64-bit operating system, or if you are migrating from a release of the product earlier than V8, you must perform migration tasks that include steps specific to that environment instead of the basic DB2 server migration task.

Determine which of the following migration tasks applies to your environment, and perform this migration task:

- Migrating DB2 32-bit servers to 64-bit systems (Windows)
- Migrating DB2 32-bit servers to 64-bit systems (Linux and UNIX)
- Migrating to a new DB2 server
- Migrating partitioned database environments
- Migrating from a system with multiple DB2 copies (Linux and UNIX)
- Migrating from DB2 UDB Version 7 servers (Windows)
- Migrating from DB2 UDB Version 7 servers (Linux and UNIX)
- Migrating DB2 servers in Microsoft Cluster Server environments
- Migrating DB2 Data Links Manager environments
- Migrating from XML Extender
- Migrating from non-DB2 relational database management systems

#### Related concepts:

- “Migration essentials for DB2 servers” on page 19

#### Related tasks:

- “Migrating a DB2 server (Linux and UNIX)” on page 55
- “Migrating a DB2 server (Windows)” on page 47

**Related reference:**

- “Pre-migration tasks for DB2 servers” on page 33
- “Post-migration tasks for DB2 servers” on page 81

---

## Migrating DB2 32-bit servers to 64-bit systems (Windows)

On the Windows operating systems, there are two ways to migrate your DB2 UDB Version 8 32-bit server to a DB2 Version 9 64-bit server. One way is to migrate your existing DB2 UDB Version 8 32-bit server to DB2 Version 9 32-bit server, and then install DB2 Version 9 64-bit server. This procedure is covered by this task and only applies to Windows on X64.

The other way is to migrate to a new DB2 server where DB2 Version 9 64-bit database product is installed.

**Prerequisites:**

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

**Restrictions:**

- Migration is supported only from DB2 UDB Version 8.
- Direct migration is not supported from DB2 UDB Version 7 or earlier. You must migrate first to DB2 UDB Version 8.
- Additional migration restrictions apply. Review the complete list.

**Procedure:**

To migrate a DB2 UDB Version 8 32-bit server to a DB2 Version 9 64-bit server:

1. Log on as a Local Administrator.
2. Install DB2 Version 9 32-bit database product and choose to migrate your existing DB2 UDB Version 8 copy. All your DB2 UDB Version 8 instances and your DB2 Administration Server (DAS) are automatically migrated. Do not install additional copies of 32-bit DB2 Version 9.

You will get a warning that recommends that you to run the **db2ckmig** command if you have local databases. Ignore this warning and continue the migration if you completed the pre-migration tasks. Otherwise, verify that your databases are ready for DB2 migration before you continue with the installation.

3. Install DB2 Version 9 64-bit database product and choose to migrate. This procedure removes DB2 Version 9 32-bit database product, and updates your existing instances to 64-bit instances.
4. Migrate your databases.

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



**Related concepts:**

- “Migration recommendations for DB2 servers” on page 23
- “Migration restrictions for DB2 servers” on page 20
- “Migrating environments with specific characteristics” on page 63
- “Migration essentials for DB2 servers” on page 19
- “Support changes for 32-bit and 64-bit DB2 servers” on page 27

**Related tasks:**

- “Migrating to a new DB2 server” on page 67
- “Installing DB2 servers (Windows)” in *Quick Beginnings for DB2 Servers*
- “Verifying that your databases are ready for migration” on page 34
- “Migrating databases” on page 52
- “Verifying migration of DB2 servers” on page 98
- “An overview of installing your DB2 product (Windows)” in *Quick Beginnings for DB2 Servers*
- “Migrating a DB2 server (Windows)” on page 47

**Related reference:**

- “Disk space requirements for DB2 server migration” on page 26
- “Pre-migration tasks for DB2 servers” on page 33
- “Post-migration tasks for DB2 servers” on page 81
- “Changes in DB2 registry variables, configuration parameters, and database physical design characteristics” on page 85
- “db2ckmig - Database pre-migration tool command” in *Command Reference*
- “Multiple DB2 copies roadmap” in *Administration Guide: Implementation*

---

## Migrating DB2 32-bit servers to 64-bit systems (Linux and UNIX)

There are some additional steps to migrate your database to DB2 Version 9 on 64-bit operating systems, if you are migrating from a DB2 UDB Version 8 32-bit instance on AIX, HP-UX, Solaris, Linux on zSeries, Linux on POWER, and Linux on x86-64. You need to install a 64-bit kernel before proceeding with the DB2 Version 9 installation, and subsequent migration.

If you are migrating to DB2 Version 9 on a new 64-bit system, follow the same procedure to migrate to a new DB2 server. Use this same procedure when you want to migrate to DB2 Version 9 on 64-bit Linux on Itanium Platform Family (IPF).

**Prerequisites:**

- Ensure that you have root access.
- Ensure that you have SYSADM, SYSCTRL, or SYSMANT authority.
- Review the System requirements Web page for DB2 database product installation. The prerequisites for Linux and UNIX operating systems have changed.
- Review migration recommendations and disk space requirements.
- Perform pre-migration tasks.

**Restrictions:**

- Migration is supported only from DB2 UDB Version 8.
- Direct migration is not supported from DB2 UDB Version 7 or earlier. You must migrate first to DB2 UDB Version 8.
- Additional migration restrictions apply. Review the complete list.

**Procedure:**

To migrate from a DB2 UDB Version 8 32-bit server to a DB2 Version 9 64-bit server:

1. Log on as root.
2. Optional: Update any 32-bit instances to 64-bit instances on systems with 64-bit kernel (except Linux on x86). Use the **db2iupdt** command:

```
db2stop
$DB2DIR/instance/db2iupdt -w 64 instance_name
db2 start
```

where DB2DIR is set to the DB2 UDB Version 8 installation path.

This step is only recommended if you are also migrating to 64-bit applications. After migration to DB2 Version 9, all your instances are migrated to 64-bit instances, and \$INSTHOME/sql1lib/lib points to 64-bit shared libraries. You should test your applications to ensure the run successfully. If you do not update to a 64-bit instance, \$INSTHOME/sql1lib/lib points to 32-bit shared libraries.

The platforms that provide 64-bit instances and include 32-bit shared libraries are AIX, HP-UX, Solaris, Linux on zSeries, Linux on Power, and Linux on x86-64.

3. Install DB2 Version 9 on your 64-bit system. Run the **db2setup** command and select Install New on the Install a Product panel to install a new copy of DB2 Version 9.
4. Migrate instances from the same installation path that you indicated during DB2 Version 9 installation. In the migrated instance the \$INSTHOME/sql1lib/lib is a link to 32-bit shared libraries.
5. Optional: Migrate the DB2 Administration Server if you want to keep your existing configuration and to administer your DB2 Version 9 instances using the Control Center.
6. Migrate databases.

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

If you have 32-bit database applications or routines that will access databases migrated to DB2 Version 9 64-bit instances, you need to ensure that they run successfully after migration. See the following tasks for details:

- Migrating database applications
- Migrating routines

**Related concepts:**

- “Migration recommendations for DB2 servers” on page 23
- “Migration restrictions for DB2 servers” on page 20
- “Support changes for 32-bit and 64-bit DB2 servers” on page 27
- “Migrating environments with specific characteristics” on page 63

- “Migration essentials for DB2 servers” on page 19

**Related tasks:**

- “Migrating to a new DB2 server” on page 67
- “Installing DB2 servers (Linux and UNIX)” in *Quick Beginnings for DB2 Servers*
- “Migrating instances” on page 48
- “Migrating the DB2 Administration Server (DAS)” on page 50
- “Migrating databases” on page 52
- “Verifying migration of DB2 servers” on page 98
- “Migrating database applications” on page 141
- “Migrating routines” on page 153
- “Migrating a DB2 server (Linux and UNIX)” on page 55
- “An overview of installing your DB2 product (Linux and UNIX)” in *Quick Beginnings for DB2 Servers*

**Related reference:**

- “Disk space requirements for DB2 server migration” on page 26
- “Pre-migration tasks for DB2 servers” on page 33
- “Post-migration tasks for DB2 servers” on page 81
- “db2iupdt - Update instances command” in *Command Reference*
- “db2setup - Install DB2 command” in *Command Reference*

---

## Migrating to a new DB2 server

If you want to migrate to a new DB2 Version 9 server, you need to recreate your instances and then restore your DB2 UDB Version 8 databases. After restoring the database, the RESTORE DATABASE command automatically runs the MIGRATE DATABASE command.

**Prerequisites:**

- Ensure that you have root access on Linux and UNIX operating systems or Local Administrator on Windows.
- Ensure that you have SYSADM authority.
- Review the System requirements Web page for DB2 database product installation. The prerequisites for operating systems have changed.
- Review migration recommendations and disk space requirements.
- Perform pre-migration tasks.

**Restrictions:**

- Migration is supported only from DB2 UDB Version 8.
- Direct migration is not supported from DB2 UDB Version 7 or earlier. You must migrate first to DB2 UDB Version 8.
- Additional migration restrictions apply. Review the complete list.

**Procedure:**

To migrate to a new DB2 Version 9 server:

1. Backup your DB2 UDB Version 8 databases using the BACKUP DATABASE command.

2. Log on as root on Linux and UNIX operating systems or Local Administrator on Windows on the new DB2 server.
3. Install DB2 Version 9 on the new DB2 server.
4. by running the `db2icrt` from the DB2 Version 9 copy that you installed in the previous step. Then restore the database manager configuration parameter values for each instance using the `UPDATE DATABASE MANAGER CONFIGURATION` command and the values that you saved in the pre-migration tasks.
5. Optional: Create a new DB2 Administration Server (DAS) on DB2 Version 9. You need a DAS if you want to use the Control Center to administer your DB2 Version 9 instances, to manage jobs, and to schedule jobs.
6. Transfer DB2 UDB Version 8 backup files for all the databases that you want to migrate to the new DB2 server.
7. Log on as the instance owner.
8. Migrate the database using the `RESTORE DATABASE` command.

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

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

In a partitioned database environment, you must execute the `RESTORE DATABASE` command in all database partitions.

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

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

The error message `SQL1704N` indicates the database migration failed. Find this SQL error code in the Message Reference 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 migration failed because the database logs are full. If this error occurs, complete the following steps to migrate the database:

- a. Increase the total size for all log files.
- b. Migrate the database using the `MIGRATE DATABASE` command.
- c. If the log file size is still not large enough, the following error is returned:

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

You must increase log file size and attempt to migrate the database again.

- d. Once migration is completed reset the value of `logfilesiz`, `logprimary`, and `logsecond` database configuration parameters.
10. Compare your database configuration settings after migration with the configuration settings you had before you migrated your database. Verify the following settings and database information are the same:
    - database configuration parameter settings
    - table spaces information
    - packages information
  11. Verify your database migration is successful. Connect to the migrated databases and issue a small query:

```
db2 connect to sample
```

Database Connection Information

```
Database server = DB2/AIX64 9.1.0
SQL authorization ID = TESTDB2
Local database alias = SAMPLE
```

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

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

**Related concepts:**

- “Migration recommendations for DB2 servers” on page 23
- “Migration restrictions for DB2 servers” on page 20
- “Backup and restore operations between different operating systems and hardware platforms” in *Data Recovery and High Availability Guide and Reference*
- “Migrating environments with specific characteristics” on page 63

**Related tasks:**

- “Creating an instance using db2icrt” in *Installation and Configuration Supplement*
- “Creating a DB2 administration server (DAS)” in *Administration Guide: Implementation*
- “Increasing table space and log file sizes before migration” on page 39
- “Migrating databases” on page 52
- “Verifying migration of DB2 servers” on page 98
- “An overview of installing your DB2 product (Linux and UNIX)” in *Quick Beginnings for DB2 Servers*
- “An overview of installing your DB2 product (Windows)” in *Quick Beginnings for DB2 Servers*
- “Migrating a DB2 server (Linux and UNIX)” on page 55
- “Migrating a DB2 server (Windows)” on page 47

**Related reference:**

- “Disk space requirements for DB2 server migration” on page 26
- “Pre-migration tasks for DB2 servers” on page 33
- “Post-migration tasks for DB2 servers” on page 81
- “Changes in DB2 registry variables, configuration parameters, and database physical design characteristics” on page 85
- “MIGRATE DATABASE command” in *Command Reference*

---

## Migrating partitioned database environments

DB2 Version 9 partitioned databases can be migrated from the catalog database partition server or any other database partition server. Should the migration process fail, you can retry migration from the catalog database partition server or any other database partition server again.

**Prerequisites:**

- Ensure that you have root access on Linux and UNIX operating systems or Local Administrator on Windows.
- Ensure that you have SYSADM authority.
- Review the System requirements Web page for DB2 database product installation. The prerequisites for operating systems have changed.
- Review migration recommendations and disk space requirements.
- Perform pre-migration tasks.

**Restrictions:**

- Migration is supported only from DB2 UDB Version 8.
- Direct migration is not supported from DB2 UDB Version 7 or earlier. You must migrate first to DB2 UDB Version 8.
- Additional migration restrictions apply. Review the complete list.
- The catalog database partition server must be up and running.

**Procedure:**

To migrate DB2 servers in a partitioned database environment:

1. Perform a full offline backup for all databases, verify that your databases are ready for migration, and any other pre-migration tasks that apply.
2. Install DB2 Version 9 on each participant database partition server and setup your partitioned database environment.
3. Migrate each instance on the database partition server that owns the instance. You can skip this step if you selected to automatically migrate the instances during DB2 Version 9 installation on Windows.
4. Migrate each database from the catalog database partition server. If one of the non-catalog database partition server is not available during migration, all database partitions on this server are not migrated. However, you can run the MIGRATE DATABASE command again to process that particular database partition server afterwards when it is up and running.
5. Create a new DB2 Administration Server (DAS) on each database partition server. If you need to keep your existing DAS settings, you can migrate the DAS on each participating database partition server instead of creating a new DAS.

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

**Related concepts:**

- “Migration recommendations for DB2 servers” on page 23
- “Migration restrictions for DB2 servers” on page 20
- “Migrating environments with specific characteristics” on page 63

**Related tasks:**

- “Migrating instances” on page 48
- “Migrating databases” on page 52
- “Creating a DB2 administration server (DAS)” in *Administration Guide: Implementation*
- “Migrating the DB2 Administration Server (DAS)” on page 50
- “Verifying migration of DB2 servers” on page 98

- “Setting up a partitioned database environment” in *Quick Beginnings for DB2 Servers*
- “Migrating a DB2 server (Linux and UNIX)” on page 55
- “Migrating a DB2 server (Windows)” on page 47

**Related reference:**

- “Disk space requirements for DB2 server migration” on page 26
- “Pre-migration tasks for DB2 servers” on page 33
- “Post-migration tasks for DB2 servers” on page 81
- “MIGRATE DATABASE command” in *Command Reference*
- “dasmigr - Migrate the DB2 administration server command” in *Command Reference*
- “db2imigr - Migrate instance command” in *Command Reference*

---

## Migrating from a system with multiple DB2 copies (Linux and UNIX)

On Linux and UNIX, you can migrate to DB2 Version 9 from a DB2 server that has multiple copies of DB2 Enterprise Server Edition (ESE) Version 8. If have installed several alternate fix packs as a completely new copy of DB2 ESE, you could have multiples copies of DB2 ESE on the same DB2 server. You need to manually migrate any DB2 UDB Version 8 instances to a DB2 Version 9 copy of your choice.

You can manually migrate a DB2 UDB Version 8 instance at any fix pack level by executing the **db2imigr** command from the target DB2 Version 9 copy of your choice. Once an instance is migrated to a DB2 Version 9 copy, you cannot migrate to another DB2 Version 9 copy. Also you cannot migrate to DB2 UDB Version 8. However, you can upgrade an instance between different DB2 copies of DB2 Version 9 using the **db2iupdt** command.

**Prerequisites:**

- Ensure that you have root access.
- Review the System requirements Web page for DB2 database product installation. The prerequisites for Linux and UNIX operating systems have changed.
- Review migration recommendations and disk space requirements.
- Perform pre-migration tasks.

**Restrictions:**

- Migration is supported only from DB2 UDB Version 8.
- Direct migration is not supported from DB2 UDB Version 7 or earlier. You must migrate first to DB2 UDB Version 8.
- Additional migration restrictions apply. Review the complete list.

**Procedure:**

To migrate the DB2 server:

1. Log on as root.
2. Install DB2 Version 9. Run the **db2setup** command and select Install New on the Install a Product panel to install a new copy of DB2 Version 9.

You can install multiple DB2 Version 9 copies, if you want to migrate your DB2 UDB Version 8 instances at different levels to different DB2 Version 9 copies.

- Migrate each instances using the **db2imigr** command from the installation path of the DB2 Version 9 copy of your choice.

Assume that you have the following DB2 copies and instances on an AIX server:

Table 8. Directory examples for DB2 copies.

| DB2 level installed | Instance name        | Installation directory for each DB2 copy   |
|---------------------|----------------------|--------------------------------------------|
| DB2 UDB Version 8   | db2inst1<br>db2inst2 | /usr/opt/db2_08_01<br>/usr/opt/db2_08_FP7/ |
| DB2 Version 9       | None                 | /opt/IBM/db2/V9.1<br>/home/db2/myV9.1      |

You can then run the following commands to successfully migrate your instances to DB2 Version 9:

Table 9. Instance migration command examples.

| Migrate Instance | Commands                                                       |
|------------------|----------------------------------------------------------------|
| db2inst1         | cd /opt/IBM/db2/V9.1/instance<br>db2imigr -u db2fenc1 db2inst1 |
| db2inst2         | cd /home/db2/myV9.1/instance<br>db2imigr -u db2fenc2 db2inst2  |

- Optional: Migrate the DB2 Administration Server if you want to keep your existing configuration and to administer your DB2 Version 9 instances using the Control Center.
- Log on as the instance owner of the migrated instance.
- Migrate databases.

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

#### Related concepts:

- “Migration recommendations for DB2 servers” on page 23
- “Migration restrictions for DB2 servers” on page 20
- “Migrating environments with specific characteristics” on page 63
- “Migration essentials for DB2 servers” on page 19

#### Related tasks:

- “Installing DB2 servers (Linux and UNIX)” in *Quick Beginnings for DB2 Servers*
- “Migrating instances” on page 48
- “Migrating the DB2 Administration Server (DAS)” on page 50
- “Migrating databases” on page 52
- “Verifying migration of DB2 servers” on page 98
- “Migrating a DB2 server (Linux and UNIX)” on page 55
- “Migrating a DB2 server (Windows)” on page 47
- “Working with existing DB2 copies” in *Quick Beginnings for DB2 Servers*

#### Related reference:

- “Disk space requirements for DB2 server migration” on page 26
- “Pre-migration tasks for DB2 servers” on page 33



- “Post-migration tasks for DB2 servers” on page 81
- “db2imigr - Migrate instance command” in *Command Reference*
- “db2setup - Install DB2 command” in *Command Reference*
- “Multiple DB2 copies roadmap” in *Administration Guide: Implementation*

---

## Migrating from DB2 UDB Version 7 servers (Windows)

There is no direct migration to DB2 Version 9 from DB2 UDB Version 7. You must migrate first to DB2 UDB Version 8 and then migrate to DB2 Version 9. You should migrate to the latest DB2 UDB Version 8.2 FixPak to take advantage of all fixes that could affect migration.

### Prerequisites:

- Ensure that you have Local Administrator access.
- Ensure that you have SYSADM authority.

### Restrictions:

- Migration to DB2 Version 9 is supported only from DB2 UDB Version 8.
- Migration to DB2 UDB Version 8 is supported from DB2 UDB Version 7.
- Changes in operating system support might require you to upgrade your hardware and operating system level before you migrate to DB2 Version 9.

### Procedure:

To migrate a DB2 UDB Version 7 server to DB2 Version 9:

1. Migration to DB2 UDB Version 8 from DB2 UDB Version 7.
2. Rebind packages in migrated databases.
3. Verify that the migration to DB2 UDB Version 8 was successful.
4. Migrate to DB2 Version 9 (Windows).

### Related concepts:

- “Migrating environments with specific characteristics” on page 63
- “Migration essentials for DB2 servers” on page 19
- Chapter 3, “Migration overview for DB2 servers,” on page 17

### Related tasks:

- “Migrating a DB2 server (Windows)” on page 47
- “An overview of installing your DB2 product (Windows)” in *Quick Beginnings for DB2 Servers*

### Related reference:

- “MIGRATE DATABASE command” in *Command Reference*

---

## Migrating from DB2 UDB Version 7 servers (Linux and UNIX)

There is no direct migration to DB2 Version 9 from DB2 UDB Version 7. You must migrate first to DB2 UDB Version 8 and then migrate to DB2 Version 9. You should migrate to the latest DB2 UDB Version 8.2 FixPak to take advantage of all fixes that could affect migration.

**Prerequisites:**

- Ensure that you have root access.
- SYSADM authority is required.

**Restrictions:**

- Migration to DB2 Version 9 is supported only from DB2 UDB Version 8.
- Migration to DB2 UDB Version 8 is supported from DB2 UDB Version 7.
- Changes in operating system support might also require that you upgrade hardware and operating system level before you migrate to DB2 Version 9.

**Procedure:**

To migrate a DB2 UDB Version 7 server to DB2 Version 9:

1. Migration to DB2 UDB Version 8 from DB2 UDB Version 7.
2. Rebind packages in migrated databases.
3. Verify that the migration to DB2 UDB Version 8 was successful.
4. Migrate to DB2 Version 9 (Linux and UNIX).

**Related concepts:**

- “Migrating environments with specific characteristics” on page 63
- “Migration essentials for DB2 servers” on page 19
- Chapter 3, “Migration overview for DB2 servers,” on page 17

**Related tasks:**

- “Migrating a DB2 server (Linux and UNIX)” on page 55

**Related reference:**

- “db2imigr - Migrate instance command” in *Command Reference*
- “MIGRATE DATABASE command” in *Command Reference*

---

## Migrating DB2 servers in Microsoft Cluster Server environments

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. When you migrate to DB2 Version 9, you need to migrate your MSCS instance.

You can run the **db2imigr** command to migrate your MSCS instance and to migrate existing DB2 Version 8 MSCS resources to DB2 Version 9 DB2 MSCS resources.

**Prerequisites:**

- Ensure that you have Local Administrator access.
- SYSADM authority is required.
- Review migration recommendations and disk space requirements.
- Perform pre-migration tasks.

**Restrictions:**

- Migration is supported only from DB2 UDB Version 8.

- This procedure refers to DB2 server migration from 32-bit instances to 32-bit instances, or from 64-bit instances to 64-bit instances.
- Additional migration restrictions apply. Review the complete list.

**Procedure:**

To migrate an MSCS instance:

1. Log on as a Local Administrator.
2. Back up your databases.
3. Install DB2 Version 9 in all nodes in the MSCS cluster. Run the setup.exe command to launch the DB2 Setup wizard and select the option Install New on the Install a Product panel. Do not choose the option migrate.
4. 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.
5. Migrate your MSCS instances by running the **db2imigr** 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 migration eliminates conflict with existing DB2 UDB Version 8 MSCS resources.

```
$DB2DIR\bin\db2imigr /u:user,password MSCS-InstName
```

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

6. Bring online the group of resources containing the migrated instance using the Cluster Administrator. For more information on using the Cluster Administrator refer to MSCS documentation.
7. Optional: If you want to administer your DB2 Version 9 instances using the Control Center and keep your existing configuration for your MSCS environment, migrate the DB2 Administration Server (DAS). If you choose to create a new DAS, you have to re-configure the DAS settings for your MSCS environment.
8. Migrate your databases.

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

**Related concepts:**

- "Migration recommendations for DB2 servers" on page 23
- "Migration restrictions for DB2 servers" on page 20
- "Migration essentials for DB2 servers" on page 19
- "Migrating environments with specific characteristics" on page 63
- "Environment variables and the profile registry" in *Administration Guide: Implementation*
- "Microsoft Cluster Server support" in *Data Recovery and High Availability Guide and Reference*

**Related tasks:**

- "Backing up databases before migration" on page 36
- "Migrating the DB2 Administration Server (DAS)" on page 50

- “Creating a DB2 administration server (DAS)” in *Administration Guide: Implementation*
- “Migrating databases” on page 52
- “Verifying migration of DB2 servers” on page 98
- “Migrating instances” on page 48
- “Migrating a DB2 server (Windows)” on page 47
- “Migrating DB2 32-bit servers to 64-bit systems (Windows)” on page 64

**Related reference:**

- “Disk space requirements for DB2 server migration” on page 26
- “Pre-migration tasks for DB2 servers” on page 33
- “Post-migration tasks for DB2 servers” on page 81
- “db2imigr - Migrate instance command” in *Command Reference*

---

## Migrating DB2 Data Links Manager environments

Migration from DB2 UDB Version 8 to DB2 Version 9 is not supported on a DB2 server where Data Links Manager is installed or where Data Links functionality is enabled. However, you can migrate to DB2 Version 9 if you remove the Data Links Manager functionality.

**Prerequisites:**

- Ensure that you have root access on Linux and UNIX operating systems or Local Administrator on Windows.
- Ensure that you have SYSADM authority .
- Review the System requirements Web page for DB2 database product installation. The prerequisites for Linux and UNIX operating systems have changed.
- Review migration recommendations and disk space requirements.
- Perform pre-migration tasks.

**Restrictions:**

- Migration is supported only from DB2 UDB Version 8.
- Direct migration is not supported from DB2 UDB Version 7 or earlier. You must migrate first to DB2 UDB Version 8.
- Additional migration restrictions apply. Review the complete list.

**Procedure:**

To migrate a DB2 server in the Data Links environment to DB2 Version 9:

1. Remove Data Links Manager from your databases.
2. Drop all references to the DATALINK data type from tables, distinct types, structured types, user-defined functions (UDFs), methods, and dependent objects.
3. If you installed DB2 Net Search Extender (NSE), you need to drop the following UDFs :

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

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

4. Uninstall Data Links Manager on the DB2 server that you want to migrate.
5. Update your instances to eliminate the Data Links Manager software and run as a DB2 server only by running the **db2iupdt** command:

```
db2iupdt instance-name
```

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

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

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

7. Install DB2 Version 9 on your DB2 server. Proceed to step 9, if you are installing DB2 Version 9 on Windows and selected to migrate your existing DB2 UDB Version 8 copy.
8. Migrate instances from the same installation path you indicated in step 7.
9. Optional: Migrate the DB2 Administration Server if you want to keep your existing configuration and to administer your DB2 Version 9 instances using the Control Center.
10. Migrate databases.

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

**Related concepts:**

- “Migration recommendations for DB2 servers” on page 23
- “What’s new for V9.1: Data Links Manager no longer supported” in *What’s New*
- “Functionality deprecated or discontinued in DB2 database products that impacts migration” on page 29
- “Migrating environments with specific characteristics” on page 63
- “Migration essentials for DB2 servers” on page 19
- “Migration restrictions for DB2 servers” on page 20

**Related tasks:**

- “Migrating instances” on page 48
- “Migrating the DB2 Administration Server (DAS)” on page 50
- “Migrating databases” on page 52
- “Verifying migration of DB2 servers” on page 98

**Related reference:**

- “Disk space requirements for DB2 server migration” on page 26
- “Pre-migration tasks for DB2 servers” on page 33
- “Post-migration tasks for DB2 servers” on page 81

---

## Migrating from XML Extender

DB2 Version 9 supports native XML storage in an annotated tree form similar to that of the XML Document Object Model (DOM). This support includes a new XML type, XML indexes and a series of SQL/XML functions. You can migrate your database application from XML Extender to use the native XML storage in DB2 Version 9.

### Prerequisite:

A DB2 UDB Version 8 server where XML Extender is installed.

### Restriction:

The native XML functionality is only supported on Unicode databases. If you created your databases in DB2 UDB Version 8 as a Unicode database, you can migrate to DB2 Version 9 and start using the XML functionality. Otherwise, you must first convert your database to a Unicode database.

### Procedure:

To migrate from the XML Extender to the new native XML storage support:

1. Migrate to DB2 Version 9 (Windows) or Migrate to DB2 Version 9 (Linux and UNIX).
2. Convert your databases to Unicode databases. If you created your databases in DB2 UDB Version 8 as a Unicode database, you can start using the XML functionality with your migrated database. Otherwise, you must export your database, create your database again by running the CREATE DATABASE with the clause USING CODESET utf-8 TERRITORY *territory*, and then load your data.
3. Add XML type columns to your tables. Use the ALTER TABLE command:

```
db2 ALTER TABLE table_name
 ADD column_name XML [NOT NULL]
```

You only need to perform this step if you store intact your XML documents in a column of data type CLOB, VARCHAR, XMLCLOB, XMLVARCHAR, or XMLFILE.

4. Register your XML schemas in the XML Schema repository (XSR). If you have document type definitions (DTDs), you must convert them to XML schemas and then register them in the XSR. You only need to perform this step only if you want to validate your XML documents.
5. Import XML documents into the table with the new XML data type column.
6. Convert your application to use annotated XML schema decomposition to store content from XML documents in table columns, and the new SQL/XML functions to construct or publish XML using the new XML data type.

Details on all these migration steps and examples of application migration are available in a white paper series published at developerWorks® Information Management. You can download this white paper series from the DB2 database product migration portal as it becomes available.

### Related concepts:

- “What’s new for V9.1: XML support in SQL statements and SQL/XML functions” in *What’s New*

- “Migrating environments with specific characteristics” on page 63
- “Migration essentials for DB2 servers” on page 19
- Chapter 3, “Migration overview for DB2 servers,” on page 17
- “Annotated XML schema decomposition” in *XML Guide*
- “Application programming language support for XML” in *XML Guide*
- “Native XML data store overview” in *XML Guide*

**Related tasks:**

- “Migrating a DB2 server (Windows)” on page 47
- “Migrating a DB2 server (Linux and UNIX)” on page 55
- “Converting non-Unicode databases to Unicode” in *Administration Guide: Planning*
- “Registering and enabling XML schemas for decomposition” in *XML Guide*
- “Decomposing XML documents with annotated XML schemas” in *XML Guide*

**Related reference:**

- “Restrictions on native XML data store” in *XML Guide*

---

## Migrating 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 a database application 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<sup>®</sup> Dynamic Server, Informix Extended Parallel Server (XPS), Microsoft SQL Server, Oracle, and Sybase Enterprise to DB2 database products. MTK runs on AIX, Linux, Solaris, and Windows operating systems. The only language supported is English. MTK is available as a complementary download from the IBM Migration Toolkit Web page.

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

- The IBM DB2 porting Web site can help you to find the information that you need to port your application and its data from other database management systems. This Web site describes the common migration steps and provides

resources including tools and education. Additional resources are provided for IBM customers and IBM Business Partners.

- The DB2 enablement and porting workshop is designed for ISVs and IBM Business Partners who plan to sell or port applications to DB2 database products. You can bring your existing database application to the workshop. Visit the workshop Web page for details and schedule.
- The IBM Virtual Innovation Center (VIC) is an online knowledge and enablement center that provides educational courses, live mentoring, online technical support, solution roadmaps, client simulations, answers to FAQs, case studies, and discussion forums.
- The DB2 Migrate Now! end-to-end offering for strategic IBM Business Partners that includes migration tool kits, complementary online education, information, sales teams and other resources to assist you in planning and implementing your migration to DB2 products from Oracle, Sybase, and Microsoft SQL server.
- The developerWorks Information Management Web site offers technical resources for DB2 Information Management software. It features product information, downloads, learning resources, support, and communities. On this Web site you can find many articles and tutorials that can help you to learn about the features of DB2 database products and how to use them in your applications.

**Related concepts:**

- “Relational databases” in *SQL Reference, Volume 1*
- “Migrating environments with specific characteristics” on page 63



---

## Chapter 9. Post-migration tasks

This chapter describes the post-migration tasks for DB2 servers. It contains the following sections:

- “Post-migration tasks for DB2 servers”
- “Adjusting the log space size in migrated databases” on page 83
- “Database activated after migration” on page 84
- “Changes in DB2 registry variables, configuration parameters, and database physical design characteristics” on page 85
- “Conversion of type-1 indexes in migrated databases” on page 93
- “Changes to the EXECUTE privilege on PUBLIC for migrated routines” on page 94
- “Rebinding packages in migrated databases” on page 95
- “Migrating explain tables” on page 96
- “Ensuring system temporary table spaces page sizes meet requirements” on page 97
- “Verifying migration of DB2 servers” on page 98
- “Initialization of HADR replication in migrated databases” on page 99

---

### Post-migration tasks for DB2 servers

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

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

1. If you set the *diaglevel* database manager configuration parameter to 4 as recommended in the pre-migration tasks for DB2 servers, reset this parameter to the value set before the migration.
2. Adjust the log space size. If you changed your log space setting as recommended in the pre-migration tasks for DB2 servers, reset the *logfilsiz*, *logprimary*, and *logsecond* database configuration parameters to the values they had before migration. Ensure that the amount of log space that you allocate is adequate for your DB2 server.
3. Activate your database after migration to start up your database and all necessary database services.
4. Review changes in DB2 registry variables and configuration parameters. There are new registry variables, new configuration parameters, and new default values for registry variables and configuration parameters introduced in DB2 Version 9 that can impact the behavior of your application. There are also changes in physical design characteristics of databases.
5. . The variables that you set at the global profile level, using the **db2set -g** command, are not migrated. The global profile variables apply to all instances pertaining to a specific DB2 Version 9 copy. Therefore after migration, use the configuration information that you saved in the pre-migration tasks to restore the values of your global profile registry variables for every DB2 Version 9 copy.

6. Convert type-1 indexes to type-2 indexes in migrated databases to take advantage of type-2 index benefits. Also, type-1 indexes are deprecated in DB2 Version 9, you should convert them before they are no longer supported.
7. If you are using index extensions or spatial indexes and you migrated from a DB2 UDB Version 8 32-bit instance to a DB2 Version 9 64-bit instance, you need to re-create your index extensions or spatial indexes. If you are a Spatial Extender user, review the migrating the Spatial Extender environment task for details on how to re-create your spatial indexes. The DB2 Spatial Extender and Geodetic Extender User's Guide and Reference is available by clicking at the library link on the DB2 Spatial Extender Web page.
8. Revoke the EXECUTE privilege from PUBLIC on functions and procedures to maintain secure database access.
9. Rebind packages in migrated databases to validate packages and to use updated statistics or new index information.
10. Migrate DB2 explain tables, if you need to retain explain table information that you previously gathered.
11. Ensure that you meet system temporary table spaces page sizes requirements to accommodate the largest row size in your result sets from queries or positioned updates, and create a system temporary table space with at larger page size if necessary.
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 UDB Version 8 copy and DB2DIR is the location of your DB2 Version 9 copy. You do not need to copy standard code page conversion tables.  

If you migrated your existing DB2 UDB Version 8 copy on Windows operating systems, you can restore the customized code page conversion tables that you backed up as part of the pre-migration tasks for DB2 servers to the DB2PATH\conv directory, where DB2PATH is the location of your DB2 Version 9 copy.
13. Verify that your DB2 server migration was successful. Test your applications and tools to ensure that the DB2 server is working as expected.
14. Back up your databases after the migration is complete.
15. If you migrate a DB2 server running high availability disaster recovery (HADR) replication, initialize HADR replication.
16. If you have Query Patroller installed, set the *dyn\_query\_mgmt* database parameter to ENABLE after you complete the migration to have Query Patroller intercept and capture information about your queries. Enabling this parameter is specified as a post-installation step in the Verifying the installation of Query Patroller server task.

Once your DB2 server performance is stable, take advantage of optimizer improvements and collect statistics for new features by update statistics for your migrated databases. During database migration to DB2 Version 9, the statistics collected from your existing database catalog 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 migrated databases, determine whether 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 UDB Version 8 copies that you no longer need.

**Related concepts:**

- “Guidelines for collecting and updating statistics” in *Performance Guide*
- “Migrating environments with specific characteristics” on page 63
- “Migration essentials for DB2 servers” on page 19

**Related tasks:**

- “Migrating a DB2 server (Linux and UNIX)” on page 55
- “Migrating a DB2 server (Windows)” on page 47

---

## Adjusting the log space size in migrated 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-migration task, you can restore additional free space to your DB2 server.

**Prerequisite:**

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

**Restrictions:**

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

**Procedure:**

1. Connect to the database that you migrated:

```
db2 CONNECT TO sample
```

where *sample* is the database name.

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

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

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

3. Optional: Increase your log file size settings. The RID for log records has increased in the amount of 2 bytes, depending on the type of log record this could represent less than 2% increase in the log record size.

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

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

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

**Related concepts:**

- “What’s new for V9.1: Increased log, table space, and memory requirements due to larger record identifiers (RIDs)” in *What’s New*
- “Space requirements for log files” in *Administration Guide: Planning*

**Related reference:**

- “DB2 log records” in *Administrative API Reference*
- “UPDATE DATABASE CONFIGURATION command” in *Command Reference*
- “Configuration parameters for database logging” in *Data Recovery and High Availability Guide and Reference*
- “Post-migration tasks for DB2 servers” on page 81

---

## Database activated after migration

Start up your database and all necessary database services with the **ACTIVATE DATABASE** command. After this command is executed successfully your database is available for connections. In DB2 Version 9, the database directory files were modified to include additional information for new features such as self tuning memory. Because of these modifications, ensure that your buffer pools are activated without any problems by activating your database.

You can verify that all database services are running properly, all buffer pools are activated, and 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.

The following example illustrates the use of this command to activate the sample database:

```
db2 ACTIVATE DATABASE sample
```

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.

**Related concepts:**

- “Quick-start tips for performance tuning” in *Performance Guide*

**Related reference:**

- “ACTIVATE DATABASE command” in *Command Reference*
- “Post-migration tasks for DB2 servers” on page 81

---

## Changes in DB2 registry variables, configuration parameters, and database physical design characteristics

In DB2 Version 9, there are new registry variables, new configuration parameters, new default values for registry variables and configuration parameters, deprecated and discontinued variables, deprecated database manager configuration parameters, and discontinued database configuration parameters. There are also physical design characteristics of databases that have changed and that might impact migration.

As a general rule, instance profile variables that you set in your DB2 profile registry or your system environment retain their values after instance migration. Some global profile registry variables are set by the DB2 installation procedure such as, DB2SYSTEM and DB2PATH. However, the global profile registry variables that you set by running the **db2set -g** command are not migrated. You need to define them after migration.

Existing database and database manager configuration parameters also retain their values after migration. However, for new configuration parameters, the default value assigned could impact your application behavior or performance.

After migrating your DB2 server, compare your registry variable and configuration parameter values to their values before migration. If you find any differences, take the time to review these changes because they could alter the behavior or performance of your application. However, consider carefully whether to disable any new features because they provide changes for new resources needed by the database manager. You should disable new features only if you experience negative performance.

The following sections described in detail all of the changes for variables, database and database manager configuration parameters, and physical design characteristics of databases:

- “New registry variables”
- “Changes to existing registry variables” on page 86
- “Deprecated and discontinued variables” on page 87
- “New database manager configuration parameters” on page 88
- “Changes to existing database manager configuration parameters” on page 88
- “Deprecated database manager configuration parameters” on page 89
- “New database configuration parameters” on page 89
- “Changes to existing database configuration parameters” on page 89
- “Discontinued database configuration parameters” on page 91
- “Changes to physical design characteristics of databases” on page 91

### New registry variables

#### DB2\_COPY\_NAME (Windows)

This variable stores the name of the DB2 copy currently in use. The default value is the name of the default DB2 copy on your machine. If you need to switch to a different copy of DB2, run the `INSTALLPATH\bin\db2envvars.bat` command to change the copy currently in use.

#### DB2\_ENABLE\_AUTOCONFIG\_DEFAULT

If you set the value of this dynamic variable to YES, the Configuration Advisor runs automatically when you create a database. By default, this variable is not set which has the same effect as setting the variable to YES. Set this variable to NO if you do not want to run the Configuration Advisor at database creation, same behavior as in DB2 UDB Version 8:

```
db2set DB2_ENABLE_AUTOCONFIG_DEFAULT = NO
```

Alternatively, you can explicitly choose not to run the Configuration Advisor when using the CREATE DATABASE command as follows:

```
db2 CREATE DB AUTOCONFIGURE APPLY NONE
```

This autonomic computing feature significantly improves the performance of newly created databases while incurring in little system overhead. Therefore, carefully consider the impact before you decide to disable this feature. Review also the *self\_tuning\_mem* and *auto\_runstats* database configuration parameter references because their default values have changed.

### **DB2\_MAX\_LOB\_BLOCK\_SIZE**

This registry variable sets the maximum amount of LOB or XML data to be returned in a block. This is not a hard maximum; if this maximum is reached on the server during data retrieval, the server finishes writing out the current row before generating a reply for the command, such as FETCH, to the client. The default value is 0 (no limit).

### **DB2\_OPT\_MAX\_TEMP\_SIZE**

Use this variable to limit the amount of space that queries can use in the temporary table spaces. When you migrate a database, this variable is set to the default value of null, which indicates no limit. If you set this variable, the optimizer can choose a plan that is more expensive in order to use less space in the temporary table spaces, as shown in the following example:

```
db2set DB2_OPT_MAX_TEMP_SIZE=10240 (size in MB)
```

### **DB2RCMD\_LEGACY\_MODE (Windows)**

When this variable is set to NO, OFF, FALSE, 0, or null (default), it enables the DB2 Remote Command Service to run in a secure mode, which is only available if your domain controller is running Windows 2000 or later. To run without enhanced security, set DB2RCMD\_LEGACY\_MODE to YES, ON, TRUE, or 1.

```
db2set DB2RCMD_LEGACY_MODE=YES
```

## **Changes to existing registry variables**

### **DB2\_LARGE\_PAGE\_MEM**

Use this variable to enable large page support, thus increasing DB2 memory usage. Enable large page support when your database shared memory region is extremely large, when you have a fixed workload, or when you are setting up benchmark configurations.

On DB2 Version 9 on a 64-bit AIX operating system, if you set this variable to DB, you cannot enable self tuning of database shared memory by setting the *database\_memory* configuration parameter to AUTOMATIC.

#### **DB2\_MEM\_TUNING\_RANGE (AIX and Windows)**

This registry variable indicates a sequence of percentages to indicate the minimum (minfree) and maximum (maxfree) amount of physical memory that the DB2 instance leaves free, depending on the need for memory by the instance if self tuning of database shared memory is enabled. By default, this variable is not set, the database manager will calculate values for minfree and maxfree based on the amount of memory on the server.

You should not set this variable, unless you are running the self-tuning memory manager (STMM), have *database\_memory* set to AUTOMATIC, and are experiencing problems related to an insufficient amount of free physical memory.

#### **DB2\_PINNED\_BP (AIX, HP-UX, Linux)**

This variable is used to specify the database global memory (including buffer pools) associated with the database in the main memory. This setting limits the ability to dynamically increase the overall database shared memory configuration.

On DB2 Version 9 on AIX operating system, if you set this variable to "YES", self tuning for database shared memory cannot be enabled by setting the *database\_memory* configuration parameter to AUTOMATIC. On DB2 Version 9, this variable also applies to Linux operating systems. In addition to setting this registry variable to YES, the library libcap.so.1 is also required.

### **Deprecated and discontinued variables**

#### **DB2\_FORCE\_FCM\_BP (AIX)**

The DB2\_FORCE\_FCM\_BP variable is deprecated in DB2 Version 9 because only 64-bit kernels of AIX operating systems are supported and they do not have shared memory segment size restrictions. The default is to enable shared memory communications between logical nodes to improve performance as well as to provide consistency with other platforms.

After migration to DB2 Version 9, if this variable is set to NO and you do not need to use domain socket communications instead of shared memory, you should set it to YES by running the following command:

```
db2set DB2_FORCE_FCM_BP=YES
```

#### **DB2\_LGPAGE\_BP**

This registry variable has been deprecated and could be removed in a future release. It has been replaced by the DB2\_LARGE\_PAGE\_MEM registry variable. The equivalent to setting DB2\_LGPAGE\_BP to ON is setting DB2\_LARGE\_PAGE\_MEM to DB:

```
db2set DB2_LARGE_PAGE_MEM=DB
```

#### **DB2LINUXAIO**

This variable has been deprecated and could be removed in a future release. This variable allows you to enable asynchronous I/O (AIO) support on Linux. In future releases, you will not need to enable this feature using this variable.

## **DB2\_SCATTERED\_IO**

This variable is no longer supported. Prior to DB2 Version 9, if you were running on a system that contained the vectored raw I/O performance improvement Linux kernel patch, setting the value of this variable to ON allowed you to increase performance by enabling scattered I/O. In DB2 Version 9, the variable is no longer required because scattered I/O is always enabled.

## **New database manager configuration parameters**

### **fcnumchannels**

A fast communication manager (FCM) channel is used to transfer data between partitions. This parameter specifies the number of FCM channels for each database partition. The behavior of the *fcnumchannels* parameter is modeled after the deprecated *fcnumrqb* parameter and represents a combination of that parameter and the *fcnumanchors* and *fcnumconnect* parameters, which are also deprecated.

During migration, the value of the *fcnumchannels* parameter is calculated as the maximum of the *fcnumrqb*, *fcnumanchors*, and *fcnumconnect* parameter values to get the closest approximation to the DB2 UDB Version 8 configuration.

## **Changes to existing database manager configuration parameters**

### **fcnumbuffers**

You can now set the *fcnumbuffers* parameter to AUTOMATIC, which is the default value in DB2 Version 9. If you do, FCM monitors resource usage and incrementally releases resources if they are not used within 30 minutes. If the database manager cannot allocate the number of resources specified when an instance is started, it scales back the configuration values incrementally until it can start the instance.

Because the *fcnumbuffers* parameter represents a value for each partition in DB2 Version 9 but represented a value shared among all logical partitions in DB2 UDB Version 8, during migration a new value is calculated by dividing the existing value by the number of partitions to leave the actual number of resources allocated unchanged. If the *DB2\_FORCE\_FCM\_BP* variable is not set or is set to NO on AIX operating systems, the value of the *fcnumbuffers* parameter remains unchanged regardless of the number of database partitions, because in this case it represents a value for each partition.

Consider setting this parameter to AUTOMATIC to take advantage of a more optimal tuning by leveraging autonomic configurations that adapt to changing workloads. Also, this setting allows you to free resources and increase the memory available for database processing.

```
db2 UPDATE DATABASE MANAGER CONFIGURATION
 USING fcnumbuffers AUTOMATIC
```



## Deprecated database manager configuration parameters

### **fcm\_num\_anchors, fcm\_num\_connect, and fcm\_num\_rqb**

These parameters are deprecated in DB2 Version 9. For compatibility purposes, you can still update these parameters using the UPDATE DATABASE CONFIGURATION command therefore any existing DB2 CLP command scripts continue to run.

### **min\_priv\_mem and priv\_mem\_thresh**

These parameters are deprecated in DB2 Version 9. If you set a value for each of these configuration parameters, the values are ignored. Use the DB2MEMMAXFREE registry variable instead to specify the maximum number of bytes of unused private memory that is retained by DB2 database system processes before unused memory is returned to the operating system.

## New database configuration parameters

### **self\_tuning\_mem**

This parameter is a master switch for enabling or disabling self tuning of memory in a database. When the switch is set to OFF, no self tuning occurs. If the switch is set to ON, self tuning could potentially occur if you set to AUTOMATIC any of the corresponding database configuration parameters such as *database\_memory*, *sheapthres\_shr*, *sorheap*, *pckcachesz*, and *locklist*.

By default, the *self\_tuning\_mem* parameter is set to ON for newly created databases. When you migrate a database to DB2 Version 9, this parameter is set to OFF. You should consider enabling this autonomic feature to gain the performance and manageability benefits that it provides. To enable self tuning in your migrated databases, manually set this parameter to ON:

```
db2 UPDATE DB CFG FOR database-name USING self_tuning_mem ON
```

If you enable self tuning in your migrated databases, you should be aware that the memory tuner must consume additional resources to run automatically. You might need to increase the value of the *maxappls* and *max\_connections* parameters by 2 to include the memory tuner processes if your maximum number of applications or your number of active applications is close to the limit.

## Changes to existing database configuration parameters

### **auto\_runstats**

This parameter enables automated statistics collection that provides the optimizer with up-to-date information to determine the most efficient plan for accessing your data. This autonomic computing feature is now enabled by default in DB2 Version 9 on newly created databases.

When you migrate your database, this parameter retains its previous value. If this value is set to OFF, you should consider enabling this feature in order to improve performance. To enable automated statistics collection, set this parameter to ON by using the following command:

```
db2 UPDATE DB CFG FOR database-name
 USING auto_runstats ON auto_tbl_maint ON auto_maint ON
```

You can disable this feature after you create a database by using the following command:

```
db2 UPDATE DATABASE CONFIGURATION USING auto_runstats OFF
```

### **avg\_appls**

This parameter is used by the SQL optimizer to help estimate how much buffer pool will be available at run time for the access plan chosen. By default, this parameter is set to AUTOMATIC for databases that created in IBM DB2 Version 9, which means that the optimizer determines the value of the *avg\_appls* parameter.

When you migrate your database, this parameter retains its previous value. In a multi-user environment, it is very important to set this parameter to the estimated number of complex query applications that typically run against the database.

### **database\_memory**

The value COMPUTED is new for this parameter in DB2 Version 9, and indicates that the database manager calculates a value for this parameter based on other database memory configuration parameters. This is the same behavior achieved by setting *database\_memory* to AUTOMATIC in DB2 UDB Version 8. The value AUTOMATIC enables the new self-tuning memory management feature on AIX and Windows operating systems only.

When you migrate your database with the *database\_memory* parameter set to AUTOMATIC, during the migration this parameter is changed to COMPUTED. You should reset this parameter to AUTOMATIC on AIX and Windows operating systems if you want to enable self-tuning of memory.

### **dyn\_query\_mgmt**

If the value of this parameter is ENABLE, the MIGRATE DATABASE command sets it to DISABLE. After you migrate your database and install Query Patroller Version 9, you need to reset the *dyn\_query\_mgmt* parameter to ENABLE to have Query Patroller capture information about your queries.

```
db2 UPDATE DB CFG FOR database-name USING DYN_QUERY_MGMT ENABLE
```

Details about when to enable this parameter are available in the Verifying installation of Query Patroller task.

### **num\_iocleaners and num\_ioservers**

You should always tune these parameters. However, you can specify the new default value AUTOMATIC to calculate initial values for these parameters based on the system current settings.

When you migrate your database, these parameters retain the same value they had before the database migration. If you want that the database manager calculates the values for these parameters, you need to manually set these parameters to AUTOMATIC by using the following command:

```
db2 UPDATE DB CFG FOR database-name
USING NUM_IOCLEANERS AUTOMATIC
```

### **sheapthres\_shr, sortheap, pckcachesz, locklist and maxlocks**

The new value AUTOMATIC for these parameters enable self tuning of memory, provided that the *self\_tuning\_mem* parameter value is ON. Other conditions apply depending on the configuration parameter. You can find more details in the Enabling self tuning memory task.

When you migrate your database, these parameters retain the same value they had before the database migration. If you want to enable self tuning for these parameters, you need to manually set them to AUTOMATIC, as shown in the following example:

```
db2 UPDATE DB CFG FOR database-name USING SORTHEAP AUTOMATIC
db2 UPDATE DB CFG FOR database-name USING SELF_TUNING_MEMORY ON
```

You should consider setting all of these parameters to AUTOMATIC because requirements for memory have increased in DB2 Version 9.

### **applheapsz and stmtheap**

As of DB2 UDB Version 8.1 FixPak 9, these parameters have different default values for databases created under 64-bit instances. If you are migrating to a DB2 Version 9 64-bit instance from a 32-bit instance on DB2 UDB Version 8.2 FixPak 8 or earlier, you might need to increase the values of these parameters when the values are less than the defaults. The following example shows how to set these parameters to the default values for a single-partition database environment:

```
db2 UPDATE DB CFG FOR database-name USING STMTHEAP 4096
db2 UPDATE DB CFG FOR database-name USING APPLHEAPSZ 256
```

### **Discontinued database configuration parameters**

#### **estore\_seg\_sz and num\_estore\_segs**

These parameters are no longer available because extended storage is no longer supported. This change only impacts 32-bit Windows operating systems, where you can allocate Address Windowing Extensions (AWE) buffer pools using the DB2\_AWE registry variable instead of extended storage. On 64-bit operating systems, extended storage is not needed.

### **Changes to physical design characteristics of databases**

#### **OVERHEAD and TRANSFERRATE settings for table spaces**

The default value for OVERHEAD changed from 12.67 ms to 7.5 ms, and the default value for TRANSFERRATE changed from 0.18 ms to 0.06 ms. These new values were obtained by averaging values calculated for new disks in the market using the formulas described in the Table space impact on query optimization concepts. The new default values apply only to databases created in DB2 Version 9.

When you migrate your database, OVERHEAD and TRANSFERRATE retain their previous value. If you decide to use the new default values for your migrated database, remember to change the values for these parameters in all table spaces:

```
db2 ALTER TABLESPACE tablespace-name OVERHEAD 7.5
TRANSFERRATE 0.06
```

You should run the RUNSTATS utility after changing these values to update the query execution plans.

### Buffer pool self-tuning memory

You can enable self tuning of memory on new or existing buffer pools by setting the buffer pool size to AUTOMATIC using the CREATE BUFFERPOOL or ALTER BUFFERPOOL statements.

```
db2 CREATE BUFFERPOOL bp1 SIZE AUTOMATIC PAGESIZE 8K
db2 ALTER BUFFERPOOL bp2 SIZE AUTOMATIC
```

Remember that you must set the *self\_tuning\_mem* parameter to ON and that you must enable at least one more memory consumer to activate the memory tuner, in addition to the buffer pool self tuning.

### Automatic storage enabled as default in CREATE DATABASE

When you create a database in DB2 Version 9, automatic storage is enabled by default. Databases that are enabled for automatic storage have a set of one or more storage paths, and the database manager assigns containers on these storage paths to table spaces that are defined to use automatic storage.

When you migrate your database, the automatic storage enablement remains unchanged. You can only enable automatic storage when you first create a database, and you can not disable it after you created an automatic storage database. You must specify AUTOMATIC STORAGE NO in the CREATE DATABASE command to have the same default as in DB2 UDB Version 8:

```
db2 CREATE DATABASE database-name AUTOMATIC STORAGE NO
```

The purpose of this change is to provide a performance advantage to newly created database by simplifying storage management.

#### Related concepts:

- “DB2 registry and environment variables” in *Performance Guide*
- “About automatic maintenance” in *Administration Guide: Planning*
- “Automatic statistics collection” in *Performance Guide*

#### Related tasks:

- “Saving configuration information” on page 37
- “Configuring DB2 with configuration parameters” in *Performance Guide*
- “Declaring, showing, changing, resetting, and deleting registry and environment variables” in *Administration Guide: Implementation*

#### Related reference:

- “Post-migration tasks for DB2 servers” on page 81
- “auto\_maint - Automatic maintenance configuration parameter” in *Performance Guide*
- “AUTOCONFIGURE command” in *Command Reference*
- “Configuration parameters summary” in *Performance Guide*
- “UPDATE DATABASE CONFIGURATION command” in *Command Reference*
- “self\_tuning\_mem- Self tuning memory configuration parameter” in *Performance Guide*

- “UPDATE DATABASE MANAGER CONFIGURATION command” in *Command Reference*
- “CREATE BUFFERPOOL statement” in *SQL Reference, Volume 2*
- “ALTER BUFFERPOOL statement” in *SQL Reference, Volume 2*
- “ALTER TABLESPACE statement” in *SQL Reference, Volume 2*

---

## Conversion of type-1 indexes in migrated databases

You must consider converting any existing type-1 indexes to type-2 indexes after migration in order to improve performance and use automatic maintenance features.

All new indexes created in DB2 Version 9 are type-2 indexes, except when you create an index on a table that already had type-1 indexes, in which case the new index is also type-1. You can only have type-1 indexes on databases that you migrated to DB2 Version 9 that were created on DB2 UDB Version 7 or earlier.

The advantages of type-2 indexes are to improve concurrency because the use of next-key locking is reduced to a minimum, and use columns with length greater than 255 bytes as part of the index key. A table must have only type-2 indexes before online table REORG and online table LOAD commands can be used against the table. This restriction also applies to DB2 Version 9, in addition to other restrictions such as not being able to add XML type columns to a table with type-1 indexes.

You can easily convert your indexes using the REORG INDEXES/TABLE command:

```
db2 REORG INDEXES ALL FOR TABLE employee CONVERT
```

If you plan to reorganize your indexes, it is an excellent opportunity to add the convert option because this option converts your type-1 indexes and has no effect on your type-2 indexes.

If you want to verify if you have type-1 indexes, you can use the inspect tool:

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

The formatted output from the **db2inspf** command in the sample.out file shows the type of index for each table:

```
...
Table phase start (ID Signed: 83, Unsigned: 83; Tablespace ID: 0) :

Data phase start. Object: 83 Tablespace: 0
The index type is 2 for this table.
DAT Object Summary: Total Pages 1 - Used Pages 0 - Free Space 70 %
Data phase end.

Index phase start. Object: 83 Tablespace: 0
INX Object Summary: Total Pages 3 - Used Pages 3
Index phase end.
Table phase end.
...
```

Another advantage of converting to type-2 indexes using the REORG INDEXES/TABLE command is that you will also convert unique indexes created on your database before DB2 UDB Version 5. Alternatively, if you are not

converting your type-1 indexes with this command, you must run the **db2uiddl** command to generate the CREATE UNIQUE INDEX statements in a script. At your convenience you can run this script to convert these unique indexes to DB2 Version 9 semantics.

**Related concepts:**

- “Index structure” in *Performance Guide*

**Related reference:**

- “db2uiddl - Prepare unique index conversion to V5 semantics command” in *Command Reference*
- “INSPECT command” in *Command Reference*
- “REORG INDEXES/TABLE command” in *Command Reference*
- “Post-migration tasks for DB2 servers” on page 81

---

## Changes to the EXECUTE privilege on PUBLIC for migrated routines

During the database migration to DB2 UDB Version 8, the EXECUTE privilege was granted to PUBLIC for all existing functions, methods, and external stored procedures. If you want to revoke this privilege from PUBLIC for all these routines, you can run the **db2undgp** command to revoke the EXECUTE privilege on all these routines:

```
db2undgp -d sample -o revoke.db2
```

In this example, the option `-o` creates a file that contains all the REVOKE statements needed to remove the EXECUTE privilege from PUBLIC. You can review or edit this file to remove any specific statements when you want to keep the EXECUTE privilege granted to PUBLIC for any routine. Alternatively, after running the **db2undgp** command you can grant the EXECUTE privilege to a specific user or PUBLIC on specific database objects using the following statements:

```
db2 GRANT EXECUTE ON FUNCTION schema-name.* to PUBLIC or
db2 GRANT EXECUTE ON FUNCTION schema-name.* to USERID
```

If you ran the **db2undgp** command after you migrated to DB2 UDB Version 8, you do not have to run this command again after your database is migrated to DB2 Version 9. However, if you did not run this command after you migrated to DB2 UDB Version 8, you should run this command after your database is migrated to DB2 Version 9. The support for the **db2undgp** command will be removed in a future release.

**Related concepts:**

- “Routine privileges” in *Administration Guide: Implementation*

**Related tasks:**

- “Revoking privileges” in *Administration Guide: Implementation*

**Related reference:**

- “db2undgp - Revoke execute privilege command” in *Command Reference*
- “GRANT (Routine Privileges) statement” in *SQL Reference, Volume 2*
- “Post-migration tasks for DB2 servers” on page 81
- “REVOKE (Routine Privileges) statement” in *SQL Reference, Volume 2*

---

## Rebinding packages in migrated databases

During database migration, all packages for user applications and routines are marked as invalid. You need to rebind invalidated packages to take advantage of changes in the DB2 server and new statistics. Packages are also placed into an invalid state if they are dependent on a database object 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.

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

### Prerequisite:

Ensure that you have SYSADM authority.

### Restrictions:

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

### Procedure:

To rebind packages in migrated databases:

1. Log on as the instance owner.
2. Rebind all invalid packages in each database by running the **db2rbind** command:

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

The all clause rebinds valid and invalid packages.

3. Test your applications

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

The README file, which is in the installation files, contains details on rebinding specific packages for a specific DB2 Version 9 level.

### Related concepts:

- “Rebinding existing packages with the REBIND command” in *Developing Embedded SQL Applications*
- “Binding” in *Administration Guide: Planning*

### Related tasks:

- “Verifying migration of DB2 servers” on page 98
- “Migrating a DB2 server (Linux and UNIX)” on page 55
- “Migrating a DB2 server (Windows)” on page 47

### Related reference:

- “db2rbind - Rebind all packages command” in *Command Reference*
- “Post-migration tasks for DB2 servers” on page 81

- “REBIND command” in *Command Reference*

---

## Migrating explain tables

The MIGRATE DATABASE command does not migrate explain tables. If you need to maintain explain table information that you previously gathered in DB2 UDB Version 8, you need to migrate your explain tables to DB2 Version 9.

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

### Prerequisite:

Ensure that you have SYSADM or DBADM authority.

### Procedure:

To migrate the explain tables, use the **db2exmig** command:

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

where:

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

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

### Related concepts:

- “Explain tools” in *Performance Guide*
- “The explain tables and organization of explain information” in *Performance Guide*

### Related tasks:

- “Migrating a DB2 server (Linux and UNIX)” on page 55
- “Migrating a DB2 server (Windows)” on page 47
- “Migrating databases” on page 52

### Related reference:

- “db2exmig - Migrate explain tables command” in *Command Reference*
- “Post-migration tasks for DB2 servers” on page 81



---

## Ensuring system temporary table spaces page sizes meet requirements

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

### Prerequisites:

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

### Procedure:

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

1. Determine the maximum row size in your result sets from queries or positioned updates. Monitor your queries or calculate the maximum row size using the DDL statement that you used to create your tables.
2. List your table spaces using the LIST TABLESPACES command, as shown in the following example:

```
db2 LIST TABLESPACES SHOW DETAIL
...
Tablespace ID = 1
Name = TEMPSPACE1
Type = System managed space
Contents = System Temporary data
State = 0x0000
 Detailed explanation:
 Normal
Total pages = 10
Useable pages = 10
Used pages = 10
Free pages = Not applicable
High water mark (pages) = Not applicable
Page size (bytes) = 4096
Extent size (pages) = 32
Prefetch size (pages) = 320
Number of containers = 10
...
```

You can identify the system temporary table spaces in the output by looking for table spaces whose Contents fields have a value of System Temporary data. Take note of the page size for each of your system temporary table spaces and the page size of the table spaces where the tables referenced in the queries or updates were created.

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

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

where `maximum_row_size` is the maximum row size for your result sets, and `maximum_row_length` is the maximum length allowed based on the largest

page size of all of your system temporary table spaces. Review the Database Manager Page Size-specific Limits to determine the maximum row length per table space page size.

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

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

```
db2 CREATE SYSTEM TEMPORARY TABLESPACE tmp_tbsp
 PAGESIZE 8K
 MANAGED BY SYSTEM
 USING ('d:\tmp_tbsp','e:\tmp_tbsp')
```

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

**Related reference:**

- “SQL and XQuery limits” in *SQL Reference, Volume 1*
- “LIST TABLESPACES command” in *Command Reference*
- “CREATE TABLESPACE statement” in *SQL Reference, Volume 2*
- “Post-migration tasks for DB2 servers” on page 81

---

## Verifying migration of DB2 servers

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

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

**Prerequisite:**

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

**Procedure:**

To verify that your DB2 server migration was successful:

1. Log on as a user with the same authority level that is required to run the SQL statements in the script.

- 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.
- Run your script using the **db2batch** command. The following example shows you how to run this tool with the testdata.db2 sample script:

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

where *samplefile-dir-clp* is DB2DIR/samples/clp on Linux and UNIX and DB2DIR\samples\clp on Windows, DB2DIR represents the location for your DB2 Version 9 copy, sample is the database name, and the option -o r 0 p3 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
```

#### Related concepts:

- Chapter 3, "Migration overview for DB2 servers," on page 17
- "Benchmark testing" in *Performance Guide*
- "Examples of db2batch tests" in *Performance Guide*

#### Related tasks:

- "Migrating a DB2 server (Linux and UNIX)" on page 55

#### Related reference:

- "Post-migration tasks for DB2 servers" on page 81
- "db2batch - Benchmark tool command" in *Command Reference*

---

## Initialization of HADR replication in migrated databases

During migration to DB2 Version 9 in a high availability disaster recovery (HADR) replication environment, a database role is changed from primary to standard. Migration of standby databases is not supported because these databases are in roll forward pending state.

Before migration, you should stop HADR on the primary and standby databases because you can only migrate the primary database. After migration, re-create your standby databases and initialize HADR on the primary and standby databases again.

**Related concepts:**

- “Database configuration for high availability disaster recovery (HADR)” in *Data Recovery and High Availability Guide and Reference*
- “High availability disaster recovery overview” in *Data Recovery and High Availability Guide and Reference*

**Related tasks:**

- “Initializing high availability disaster recovery (HADR)” in *Data Recovery and High Availability Guide and Reference*
- “Stopping high availability disaster recovery (HADR)” in *Data Recovery and High Availability Guide and Reference*

**Related reference:**

- “Post-migration tasks for DB2 servers” on page 81
- “START HADR command” in *Command Reference*
- “STOP HADR command” in *Command Reference*

---

## Chapter 10. Reversing DB2 server migration

There is no utility to reverse the migration of a DB2 server. If you must reverse a migration, it is essential to create a plan using the steps on this procedure. Performing a migration in a test environment will help you identify any issues with the process and avoid having to reverse the migration.

This procedure includes the steps that you need to perform to reverse the migration.

### Prerequisites:

- Ensure that you have *SYSADM* authority, as well as root on Linux and UNIX operating systems or Local Administrator authority on Windows operating systems.
- Perform the following steps before migrating your DB2 server:
  - Review migration recommendations and disk space requirements.
  - Take an offline full backup of all databases that you are going to migrate.
  - Save all database manager configuration parameter values for each instance, and all database configuration parameter values for each database.
  - Perform other pre-migration tasks that apply to your environment.
- Keep your existing DB2 UDB Version 8 copy during migration of your DB2 server. To do this, select the option Install New to create a new copy when installing DB2 Version 9. Do not select the option Migrate on Windows operating systems.

### Restrictions:

- This procedure applies only to DB2 server migration. 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 migration restrictions apply. Review the complete list.

### Procedure:

To reverse a migration you need to perform the following steps:

1. Log on as the DB2 Version 9 instance owner.
2. Drop all databases in DB2 Version 9 by running the `DROP DATABASE` command.
3. Log on as root on Linux and UNIX operating systems or Local Administrator on Windows operating systems.
4. Drop your DB2 Version 9 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 migrated your DB2 UDB Version 8 instances to DB2 Version 9, re-create your instances in DB2 UDB Version 8 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 DB2 UDB Version 8 instance, log on as the instance owner and re-create your databases by running the RESTORE DATABASE command into your DB2 UDB Version 8 instance. You cannot migrate your databases from DB2 Version 9 to DB2 UDB Version 8.

**Related concepts:**

- “Migration recommendations for DB2 servers” on page 23
- “Migration restrictions for DB2 servers” on page 20

**Related tasks:**

- “Backing up databases before migration” on page 36
- “Saving configuration information” on page 37

**Related reference:**

- “Disk space requirements for DB2 server migration” on page 26
- “Pre-migration tasks for DB2 servers” on page 33
- “db2cfexp - Connectivity configuration export tool command” in *Command Reference*
- “db2cfimp - Connectivity configuration import tool command” in *Command Reference*
- “db2icrt - Create instance command” in *Command Reference*
- “db2idrop - Remove instance command” in *Command Reference*
- “DROP DATABASE command” in *Command Reference*
- “RESTORE DATABASE command” in *Command Reference*
- “UPDATE DATABASE MANAGER CONFIGURATION command” in *Command Reference*

---

## Part 3. Migrating DB2 clients

This part of the book contains the following chapters:

Chapter 11, "Migration overview for DB2 clients," on page 105

Chapter 12, "Migration essentials for DB2 clients," on page 107

Chapter 13, "Pre-migration tasks," on page 111

Chapter 14, "Migrating DB2 clients (Windows)," on page 113

Chapter 15, "Migrating DB2 clients (Linux and UNIX)," on page 119

Chapter 16, "Post-migration tasks," on page 123





---

## Chapter 11. Migration overview for DB2 clients

Migrating a DB2 client involves migrating the client instance and making sure that you can connect to all your cataloged databases. A client instance allows you to connect your application to a DB2 server and keeps the information about your cataloged nodes and databases.

The current level of DB2 client that you have installed determines the way to proceed with migration to DB2 Version 9. You can directly migrate to V9 DB2 clients from V8 DB2 clients. If you have V7 or earlier DB2 clients, you need to migrate to any V8 DB2 client first.

In DB2 Version 9, the DB2 clients are DB2 Client and DB2 Runtime Client. The DB2 Client merges the functionality of the previous DB2 Application Development Client and DB2 Administration Client. The DB2 Runtime Client is a small footprint client that only supports TCP/IP and Named Pipes protocols.

Migration to V9 DB2 Client is supported for the following DB2 clients:

- V8 DB2 Administration Client
- V8 DB2 Application Development Client

After you install V9 DB2 Runtime Client as a new DB2 client copy, migration of the client instance is supported for the following DB2 clients:

- V8 DB2 Run-Time
- V8 DB2 Run-Time Client Lite

Details on how to migrate the client instance after installation are provided in the Migration essentials for DB2 clients.

### **Related concepts:**

- “DB2 client setup overview” in *Quick Beginnings for DB2 Clients*
- Chapter 12, “Migration essentials for DB2 clients,” on page 107

### **Related tasks:**

- “Migrating a DB2 Client (Windows)” on page 113
- “Migrating a DB2 Runtime Client (Windows)” on page 115
- “Migrating DB2 clients (Linux and UNIX)” on page 119
- “Migrating from DB2 clients Version 7 (Linux and UNIX)” on page 121
- “Migrating from DB2 clients Version 7 (Windows)” on page 116
- “Planning migration for your DB2 clients” on page 9



---

## Chapter 12. Migration essentials for DB2 clients

If you are migrating your DB2 clients to DB2 Version 9, you need to consider how to do the migration based on the migration options and restrictions for V9 DB2 clients. You also need to be aware of the support available for communication protocols and connectivity between different versions of DB2 clients and DB2 servers. Finally, evaluate the migration recommendations in order to plan your DB2 client migration.

### Migration options for DB2 clients

The migration options vary depending on the type of client that you want to install. The following table describes the migration options for each type of V9 DB2 client:

Table 10. Migration options for V9 DB2 clients

| Migrating from                                                                  | Migrating to                           | Migration support details                                                                                                                                                                                                                                                                         |
|---------------------------------------------------------------------------------|----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| V8 DB2 Administration Client or V8 DB2 Application Development Client (Windows) | V9 DB2 Client (Windows)                | You have two options: <ul style="list-style-type: none"><li>• Install V9 DB2 Client and choose the option migrate, the client instance is automatically migrated for you.</li><li>• Install a new copy of the V9 DB2 client, and then manually migrate existing V8 DB2 client instance.</li></ul> |
| V8 DB2 Run-Time Client and V8 DB2 Run-Time Client Lite (Windows)                | DB2 Runtime Client Version 9 (Windows) | <ul style="list-style-type: none"><li>• Install V9 DB2 Runtime Client as a new separate copy, and then manually migrate your existing V8 DB2 client instance.</li></ul>                                                                                                                           |
| All V8 DB2 clients (Linux or UNIX)                                              | All V9 DB2 clients (Linux or UNIX)     | <ul style="list-style-type: none"><li>• Install a new copy of any V9 DB2 client, and then manually migrate your existing V8 DB2 client instance after installation.</li></ul>                                                                                                                     |

When the client instance is created, the bit size is determined by the operating systems where you installed V9 DB2 client:

- 32 bit instances only on 32-bit kernels of Linux on x86, Windows on x86, or Windows on X64 using DB2 Version 9 32-bit product.
- 64-bit instances on 64-bit kernel of AIX, HP-UX, Solaris, Linux on zSeries, Linux on POWER, Linux on x86\_64, and Linux on Itanium Platform Family (IPF). 32-bit instances are not supported on these operating systems. However, support for 32-bit clients and 32-bit applications is included on all UNIX operating systems except Linux on IPF.

### Migration restrictions for DB2 clients

Review “Migration restrictions for DB2 servers” on page 20 regarding instance migration and operating system support. These restrictions also apply to DB2 clients and can impact the migration of your DB2 clients.

If you installed V8 DB2 client is located on the same system as a DB2 Version 9 server, local connections using Interprocess Communication (IPC) are not supported. You should migrate both the DB2 server and the DB2 client to DB2 Version 9 to access the migrated databases using the existing local database directory. If you do not migrate the V8 DB2 client, you can

only access migrated databases that were catalogued as remote databases. You need to recatalog migrated databases using IPC as remote.

### Communications protocol support

DB2 Version 9 supports the TCP/IP and named pipes communication protocols but no longer supports the NetBIOS and SNA protocols. After migration, you need to catalog your nodes and databases using a valid protocol such as TCP/IP. If you try to connect to any databases cataloged on a node using NetBIOS protocol, your connection request also returns an error because the protocol is invalid.

DB2 Version 9 also supports the Internet Protocol Version 6 (IPv6). After migration, the TCP/IP nodes in your node directory will continue to work and your V9 DB2 client will use either IPv6 or IPv4 connections. If you want to use a particular version of IP, you need to recatalog your nodes using the CATALOG TCPIP4 NODE command to explicitly specify IPv4 or the CATALOG TCPIP6 NODE command to explicitly specify IPv6.

### Connectivity support between DB2 clients and DB2 servers

In DB2 Version 9, the following connectivity support is available:

Table 11. DB2 Version 9 connectivity support

| DB2 client                            | DB2 server                                  | DB2 client connectivity support                                                |
|---------------------------------------|---------------------------------------------|--------------------------------------------------------------------------------|
| 32-bit or 64-bit DB2 Version 9        | 32-bit or 64-bit DB2 UDB Version 8          | Only DB2 UDB Version 8 functionality is available.                             |
| 32-bit or 64-bit DB2 UDB Version 8    | 32-bit or 64-bit DB2 Version 9              | Only DB2 UDB Version 8 functionality is available.                             |
| 32-bit DB2 UDB Version 7 <sup>1</sup> | 32-bit or 64-bit DB2 Version 9 <sup>2</sup> | Only SQL statements are supported. DB2 administration tools are not supported. |
| 64-bit DB2 UDB Version 7 <sup>1</sup> | 32-bit or 64-bit DB2 Version 9 <sup>3</sup> | Only SQL statements are supported. DB2 administration tools are not supported. |

#### Notes:

1. Support for V7 DB2 clients will be removed in future releases.
2. 32-bit V7 DB2 clients can connect to 64-bit DB2 Version 9 servers by connecting first to a 32-bit DB2 Version 9 or DB2 UDB Version 8 server.
3. 64-bit V7 DB2 clients can connect to 32-bit DB2 Version 9 servers by connecting first to a 64-bit DB2 Version 9 or DB2 UDB Version 8 server.

### Migration recommendations for DB2 clients

In general, the recommendation is to migrate DB2 servers first and then DB2 clients. V8 DB2 clients can connect to V9 DB2 servers. The only restriction is that DB2 Version 9 features are not available to the V8 DB2 clients. However, it is not likely that you need access to these new features because your existing applications do not use them.

If you have software that requires a V8 DB2 client, you should install a V9 DB2 client as a new copy and keep your existing V8 DB2 client to satisfy this requirement. You need to create a V9 client instance and keep your existing V8 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.

If you choose to migrate your DB2 clients first, you need to be aware that there are known limitations about the support for connectivity from a V9 DB2 client to DB2 UDB Version 8 servers. Check Supported and unsupported client configurations, see if these limitations apply to your application in order to take necessary actions.

Perform the pre- and post-migration tasks to ensure a successful migration.

**Related concepts:**

- “Post-migration tasks for DB2 clients” on page 123
- “Pre-migration tasks for DB2 clients” on page 111
- Chapter 11, “Migration overview for DB2 clients,” on page 105
- “Version 9 incompatibilities with previous releases and changed behaviors” in *Administration Guide: Planning*

**Related tasks:**

- “Planning migration for your DB2 clients” on page 9
- “Migrating a DB2 Client (Windows)” on page 113
- “Migrating a DB2 Runtime Client (Windows)” on page 115
- “Migrating DB2 clients (Linux and UNIX)” on page 119

**Related reference:**

- “Supported combinations of client and server versions” in *Quick Beginnings for DB2 Clients*



---

## Chapter 13. Pre-migration tasks

This chapter describes the pre-migration tasks for DB2 clients. It contains the following sections:

- “Pre-migration tasks for DB2 clients”
- “Saving DB2 client configuration information”

---

### Pre-migration tasks for DB2 clients

Before you migrate your DB2 clients, you should complete certain tasks to help ensure that your migration is successful.

Prepare for the migration of your DB2 clients by performing the following tasks:

1. Review the migration essentials for DB2 clients to determine which factors might impact your DB2 client migration.
2. Review the supported and non-supported client configurations.
3. Plan your migration strategy. For example, you might need to migrate your DB2 server first, then your DB2 clients.
4. Optional: Migrate your DB2 servers.
5. Save your DB2 client configuration information.

**Related concepts:**

- Chapter 12, “Migration essentials for DB2 clients,” on page 107

**Related tasks:**

- “Migrating DB2 clients (Linux and UNIX)” on page 119
- “Migrating a DB2 Client (Windows)” on page 113
- “Migrating a DB2 Runtime Client (Windows)” on page 115

---

### Saving DB2 client configuration information

Before you migrate, you should save 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 migration, if necessary.

**Prerequisites:**

Ensure that you have enough authority to access the DB2 client.

**Restrictions:**

This procedure describes how to save the configuration information for only one DB2 client. If you have different configuration settings on each DB2 client, you need to save the configuration information for each client.

**Procedure:**

To save your DB2 client configuration information:

1. Save your database manager configuration parameter settings by using the GET DATABASE MANAGER CONFIGURATION command to list your settings for the parameters and redirect the command output to a file as shown in the following example:

```
db2 GET DBM CFG > D:\migration\dbm_client.cfg
```

2. Save the information of cataloged databases by running the **db2cfexp** command to create a configuration profile:

```
db2cfexp cfg_profile BACKUP
```

The `cfg_profile` file is a client profile contains all of the instance configuration information, including the database manager configuration and registry profile settings, because the option `BACKUP` is specified. You can also use the DB2 Configuration Assistant to export your configuration profile.

**Related concepts:**

- “Pre-migration tasks for DB2 clients” on page 111

**Related tasks:**

- “Creating a client profile using the Configuration Assistant” in *Quick Beginnings for DB2 Clients*

**Related reference:**

- “db2cfexp - Connectivity configuration export tool command” in *Command Reference*
- “db2cfimp - Connectivity configuration import tool command” in *Command Reference*
- “GET DATABASE MANAGER CONFIGURATION command” in *Command Reference*



---

## Chapter 14. Migrating DB2 clients (Windows)

This chapter describes how to migrate your DB2 clients on Windows. It contains the following sections:

- “Migrating a DB2 Client (Windows)”
- “Migrating a DB2 Runtime Client (Windows)” on page 115
- “Migrating from DB2 clients Version 7 (Windows)” on page 116

---

### Migrating a DB2 Client (Windows)

This procedure applies to migration from the V8 DB2 Administration Client and the V8 DB2 Application Development Client to the V9 DB2 Client. Migrating your client instance to a V9 DB2 client ensures that you can connect to all your previously cataloged databases.

When you install a V9 DB2 Client, you can choose to automatically migrate an existing V8 DB2 Administration Client copy or V8 DB2 Application Development Client copy to the V9 DB2 Client copy. Your existing V8 client instance is migrated to the V9 DB2 Client copy and the V8 DB2 client copy is removed. You can also choose to install a new copy of V9 DB2 Client and then manually migrate your client instance.

#### Prerequisites:

- Ensure that you have SYSADM, SYSCTRL, or SYSMANT authority and Local Administrator authority to run the **db2imigr** and the **db2icrt** commands.
- Review supported connectivity between DB2 clients and DB2 servers in migration essentials for DB2 clients.
- Perform pre-migration tasks.

#### Restrictions:

- Direct migration is not supported from V7 or earlier DB2 clients to V9 DB2 clients. You must migrate first to a V8 DB2 client.
- The bit size of the client instance is determined by the operating systems where you install V9 DB2 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.

#### Procedure:

To migrate to a V9 DB2 Client on Windows:

1. Install V9 DB2 Client by running the **setup.exe** command to launch the DB2 Setup wizard. You have two options:
  - Select the option Migrate in the Install a Product panel. You can choose this option, if you have an existing copy of the DB2 Version 8 Administration Client or the DB2 Version 8 Application Development Client. Your V8 DB2 client copy is removed and your client instance is migrated. Proceed to step 5 on page 114.
  - Select the option Install New in the Install a Product panel. You should choose this option to create a new copy of V9 DB2 Client and keep your existing V8 DB2 client copy.

2. Log on as Local Administrator.
3. Migrate your existing V8 DB2 client instance. To manually migrate your client instance, run the **db2imigr** command:

```
"%DB2PATH%" \bin\db2imigr InstName
```

where DB2PATH is set to the location that you specified during the V9 DB2 client installation, and *InstName* is the name of the instance.

4. Optional: You can create a new V9 client instance instead of migrating the existing V8 client instance. You only need to create a new V9 client instance when you want to keep multiple DB2 copies running on the same DB2 server, or create a testing environment. To create a new V9 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-migration tasks.

5. Compare the migrated database manager configuration parameter values with the pre-migration values to ensure the changed values are compatible with your database applications.

After migrating your DB2 Client, perform the recommended post-migration tasks for DB2 clients, especially verifying migration for DB2 clients to ensure that your DB2 client migration was successful.

#### Related concepts:

- Chapter 12, “Migration essentials for DB2 clients,” on page 107
- “Pre-migration tasks for DB2 clients” on page 111
- “Post-migration tasks for DB2 clients” on page 123

#### Related tasks:

- “Verifying migration of DB2 clients” on page 125
- “Setting communication protocols for a DB2 instance” in *Installation and Configuration Supplement*
- “Installing DB2 clients (Windows)” in *Quick Beginnings for DB2 Clients*

#### Related reference:

- “db2cfexp - Connectivity configuration export tool command” in *Command Reference*
- “db2cfimp - Connectivity configuration import tool command” in *Command Reference*
- “db2icrt - Create instance command” in *Command Reference*
- “db2imigr - Migrate instance command” in *Command Reference*

---

## Migrating a DB2 Runtime Client (Windows)

This procedure applies to migration from the V8 DB2 Run-Time Client and the V8 DB2 Run-Time Client Lite to the V9 DB2 Runtime Client. After you install V9 DB2 Runtime Client, you can manually migrate your existing V8 client instance from a V8 DB2 Run-Time or V8 DB2 Run-Time Client Lite copy. Migrating your client instance to a V9 DB2 client ensures that you can connect to all your previously cataloged databases.

### Prerequisites:

- Ensure that you have SYSADM, SYSCTRL, or SYSMAINT authority and Local Administrator authority to run the **db2imigr** and the **db2icrt** commands.
- Review supported connectivity between DB2 clients and DB2 servers in migration essentials for DB2 clients.
- Perform pre-migration tasks.

### Restrictions:

- Direct migration is not supported from V7 or earlier DB2 clients to V9 DB2 clients. You must migrate first to a V8 DB2 client.
- The bit size of the client instance is determined by the operating systems where you install V9 DB2 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.

### Procedure:

To migrate to a V9 DB2 Runtime Client on Windows:

1. Install V9 DB2 Runtime Client. Run the **setup.exe** command to launch the DB2 Setup wizard and select the option Install New in the Install a Product panel.
2. Log on as Local Administrator.
3. Migrate your existing V8 DB2 client instance by running the **db2imigr** command:

```
"%DB2PATH%" \bin\db2imigr InstName
```

where DB2PATH is set to the location that you specified during the V9 DB2 client installation, and *InstName* is the name of the instance.

4. Optional: You can create a new V9 client instance instead of migrating the existing V8 client instance. You only need to create a new V9 client instance when you want to keep multiple DB2 copies running on the same DB2 server. To create a new V9 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-migration tasks.

5. Compare the migrated database manager configuration parameter values with the pre-migration values to ensure the changed values are compatible with your database applications.

After migrating your DB2 Runtime Client, perform the recommended post-migration tasks for DB2 clients, especially verifying migration for DB2 clients to ensure that your DB2 client migration was successful.

**Related concepts:**

- Chapter 12, “Migration essentials for DB2 clients,” on page 107
- “Pre-migration tasks for DB2 clients” on page 111
- “Post-migration tasks for DB2 clients” on page 123

**Related tasks:**

- “Verifying migration of DB2 clients” on page 125
- “Installing DB2 clients (Windows)” in *Quick Beginnings for DB2 Clients*
- “Migrating a DB2 Client (Windows)” on page 113
- “Migrating from DB2 clients Version 7 (Windows)” on page 116

**Related reference:**

- “db2cfexp - Connectivity configuration export tool command” in *Command Reference*
- “db2cfimp - Connectivity configuration import tool command” in *Command Reference*
- “db2icrt - Create instance command” in *Command Reference*
- “db2imigr - Migrate instance command” in *Command Reference*

---

## Migrating from DB2 clients Version 7 (Windows)

There is no direct migration to DB2 Version 9 from V7 DB2 clients. You must migrate first to a V8 DB2 client, and then migrate to a V9 DB2 client. You should migrate to the latest DB2 UDB Version 8.2 FixPak to take advantage of all fixes that could affect migration.

**Prerequisite:**

- Review supported client and server connectivity in Migration essentials for DB2 clients.

**Restrictions:**

- Migration to V9 DB2 clients is supported only from DB2 Version 8 clients:
  - Migration to V9 DB2 Client is supported from an existing installation of V8 DB2 Administration Client or V8 DB2 Application Development Client.
  - Migration to V9 DB2 Runtime Client from V8 DB2 Run-Time or V8 DB2 Run-Time Client Lite
- Migration to V8 DB2 clients is supported from DB2 Version 7 clients.

**Procedure:**

To migrate to a V9 DB2 client from a V7 DB2 client:

1. Save client configuration and connectivity settings. Use the **db2cfexp** tool to create a configuration profile:

```
db2cfexp client_profile backup
```

This profile contains all of the instance configuration information, including the database manager configuration and registry profile because the option backup is specified. You can also use the DB2 Configuration Assistant to export your configuration profile.

2. Install V8.2 DB2 client. Use the **setup.exe** command to launch the DB2 Setup wizard. Your existing V7 DB2 client and your client instance are automatically migrated as part of the installation.
3. Compare the migrated database manager configuration parameter values with pre-migration values. Ensure that any changes are compatible with your application.
4. Test connections to all your cataloged databases to confirm the migration was successful:

```
db2 CONNECT TO DATABASE database-name
```

Alternatively, you can use the Configuration Assistant to test your connections. You can also import your client profile, if you encounter any connection problems to your cataloged databases.

Ensure that you have network connectivity to the DB2 server, and that the DB2 server is up and running. Also, ensure that there are no connectivity issues due to 32-bit and 64-bit connection support.

5. Migrate to V9 DB2 Client on Windows or Migrate to V9 DB2 Run-Time Client on Windows.

**Related concepts:**

- Chapter 12, “Migration essentials for DB2 clients,” on page 107

**Related tasks:**

- “Migrating a DB2 Client (Windows)” on page 113
- “Installing DB2 clients (Windows)” in *Quick Beginnings for DB2 Clients*



---

## Chapter 15. Migrating DB2 clients (Linux and UNIX)

This chapter describes how to migrate your DB2 clients on Linux and UNIX. It contains the following sections:

- “Migrating DB2 clients (Linux and UNIX)”
- “Migrating from DB2 clients Version 7 (Linux and UNIX)” on page 121

---

### Migrating DB2 clients (Linux and UNIX)

This procedure applies to migration from the V8 DB2 Administration Client and the V8 DB2 Application Development Client to the V9 DB2 Client. It also applies to migration from the V8 DB2 Run-Time Client and the V8 DB2 Run-Time Client Lite to the V9 DB2 Runtime Client. Migrating your client instance to a V9 DB2 client ensures that you can connect to all your previously cataloged databases.

After you install a V9 DB2 client on a system where a V8 DB2 client is installed, you need to manually migrate your existing V8 client instance to ensure that you can connect to all your previously cataloged databases.

#### Prerequisites:

- Ensure that you have root access.
- Ensure that you have SYSADM, SYSCTRL, or SYSMAINT authority and root access to run the **db2imigr** and the **db2icrt** commands.
- Review the System requirements Web page for DB2 database product installation. Some operating systems require a 64-bit kernel.
- Review supported connectivity between DB2 clients and DB2 servers in migration essentials for DB2 clients.
- Perform pre-migration tasks.

#### Restrictions:

- Direct migration is not supported from V7 or earlier DB2 clients to V9 DB2 clients. You must migrate first to a V8 DB2 client.
- The bit size of the client instance is determined by the operating system where you install the V9 DB2 client. The client instance is 32-bit only in Linux on x86 and 64-bit on all the other supported Linux and UNIX operating systems.

#### Procedure:

To migrate a V8 DB2 client to a V9 DB2 client:

1. Install V9 DB2 Client or V9 DB2 Runtime Client.. Run the **db2setup** command and select Install New on the Install a Product panel to install a new copy of DB2 Version 9.
2. Log on as root.
3. Migrate your existing V8 DB2 client instances by running the **db2imigr** command:

```
$DB2DIR/instance/db2imigr InstName
```

where

**DB2DIR**

is set to the location that you specified during the V9 DB2 client installation. The default installation path for UNIX is /opt/IBM/db2/V9.1 and for Linux is /opt/ibm/db2/V9.1.

*InstName*

is the login name of the client instance owner.

4. Optional: You can also create a new V9 client instance instead of migrating the existing V8 client instance. You only need to create a new V9 client instance when you want to keep multiple DB2 copies running on the same DB2 server. To create a new V9 client instance, run the **db2icrt** command with the option **-s**:
 

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

where

**DB2DIR**

is set to the location you specified during the V9 DB2 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 save in the pre-migration tasks.

5. Compare the migrated database manager configuration parameter values with the pre-migration values to ensure the changed values are compatible with your database applications.

After migrating any DB2 client, perform the recommended post-migration tasks for DB2 clients, especially verifying migration for DB2 clients to ensure that your DB2 client migration was successful.

**Related concepts:**

- “Pre-migration tasks for DB2 clients” on page 111
- “Post-migration tasks for DB2 clients” on page 123
- Chapter 12, “Migration essentials for DB2 clients,” on page 107

**Related tasks:**

- “Verifying migration of DB2 clients” on page 125
- “Setting communication protocols for a DB2 instance” in *Installation and Configuration Supplement*
- “Installing DB2 clients (UNIX and Linux)” in *Quick Beginnings for DB2 Clients*

**Related reference:**

- “db2cfexp - Connectivity configuration export tool command” in *Command Reference*
- “db2cfimp - Connectivity configuration import tool command” in *Command Reference*
- “db2imigr - Migrate instance command” in *Command Reference*



---

## Migrating from DB2 clients Version 7 (Linux and UNIX)

There is no direct migration to DB2 Version 9 from V7 DB2 clients. You must migrate first to a V8 DB2 client, and then migrate to a V9 DB2 client. You should migrate to the latest DB2 UDB Version 8.2 FixPak to take advantage of all fixes that could affect migration.

### Prerequisites:

- Ensure that you have root authority.
- Ensure that you have SYSADM, SYSCTRL, or SYSMAINT authority and root authority to run the **db2imigr** and the **db2icrt** commands.
- Review supported client and server connectivity in Migration essentials for DB2 clients.

### Restrictions:

- Migration to V9 DB2 clients is supported only from DB2 Version 8 clients:
  - Migration to V9 DB2 Client is supported from an existing installation of V8 DB2 Administration Client or V8 DB2 Application Development Client.
  - Migration to V9 DB2 Runtime Client from V8 DB2 Run-Time or V8 DB2 Run-Time Client Lite.
- Migration to V8 DB2 clients is supported from DB2 Version 7 clients.

### Procedure:

To migrate to a V9 DB2 client from a V7 DB2 client:

1. Save client configuration and connectivity settings. Use the **db2cfexp** tool to create a configuration profile:

```
db2cfexp client_profile backup
```

This profile contains all of the instance configuration information, including the database manager configuration and registry profile because the option backup is specified. You can also use the DB2 Configuration Assistant to export your configuration profile.

2. Install DB2 client Version 8.2.
3. Log on as root.
4. Migrate your existing V7 DB2 client instances. Use the **db2imigr** command:

```
$DB2DIR/instance/db2imigr InstName
```

where DB2DIR indicates the location for V8 DB2 client installation, and *InstName* is the login name of the instance owner. The default V8 DB2 client installation path is /usr/opt/db2\_08\_01 on AIX and /opt/IBM/db2/V8.1 on all other UNIX operating systems.

5. Compare the migrated database manager configuration parameter values with pre-migration values. Ensure that any changes are compatible with your application.
6. Test connections to all your cataloged databases to confirm the migration was successful:

```
db2 CONNECT TO DATABASE database-name
```

Alternatively, you can use the Configuration Assistant to test your connections. You can also import your client profile, if you encounter any connection problems with your cataloged databases.

Ensure that you have network connectivity to the DB2 server, and that the DB2 server is up and running.

7. Migrate to V9 DB2 client on Linux and UNIX.

**Related concepts:**

- Chapter 12, “Migration essentials for DB2 clients,” on page 107

**Related tasks:**

- “Migrating DB2 clients (Linux and UNIX)” on page 119
- “Installing DB2 clients (UNIX and Linux)” in *Quick Beginnings for DB2 Clients*

---

## Chapter 16. Post-migration tasks

This chapter describes the post-migration tasks for DB2 clients. It contains the following sections:

- “Post-migration tasks for DB2 clients”
- “Recataloging nodes and databases that use NetBIOS and SNA protocols”
- “Verifying migration of DB2 clients” on page 125

---

### Post-migration tasks for DB2 clients

After migrating your DB2 clients, you should perform some post-migration tasks to ensure that your DB2 clients perform as expected and at their optimum level.

Perform the following post-migration tasks that apply to your DB2 clients:

1. Recatalog nodes and databases if you catalogued them using the NetBIOS and SNA protocols in DB2 UDB Version 8. DB2 Version 9 does not support the NetBIOS and SNA protocols.
2. Review changes in DB2 registry variables and configuration parameters to modify your settings where required. There are new registry variables, new configuration parameters, and new default values for registry variables and configuration parameters introduced in DB2 Version 9 that can impact the behavior of your application.
3. Verify that migration of your DB2 clients was successful.

**Related tasks:**

- “Migrating DB2 clients (Linux and UNIX)” on page 119
- “Migrating a DB2 Client (Windows)” on page 113
- “Migrating a DB2 Runtime Client (Windows)” on page 115

---

### Recataloging nodes and databases that use NetBIOS and SNA protocols

DB2 Version 9 does not support the NetBIOS and SNA protocols. You need to recatalog, using a valid protocol, any nodes that you cataloged with the NetBIOS and SNA protocols. If you try to connect to any databases cataloged on a node that uses the NetBIOS or SNA protocol, your connection request returns an error because these protocols are invalid.

This procedure describes how to recatalog nodes using the TCP/IP protocol. If you change the node name, you need to recatalog the databases as well to use the new node name.

**Prerequisites:**

- Ensure that you have SYSADM or SYSCTRL authority.
- Ensure that you have network connectivity from the DB2 client to the DB2 server.

**Restrictions:**

The only protocols available in DB2 Version 9 are TCP/IP and Named Pipes.

**Procedure:**

To recatalog nodes and databases specifying the TCP/IP protocol:

1. Determine which nodes use the NetBIOS or SNA protocol by issuing the LIST NODE DIRECTORY command:

```
db2 LIST NODE DIRECTORY show detail > node_list.log
```

Redirect the output of this command to a file and keep it, because the information is useful to recatalog your nodes.

2. Remove all nodes that use NetBIOS or SNA protocol from the node directory by issuing the UNCATALOG NODE command:

```
db2 UNCATALOG NODE node-name
```

3. Determine which databases use the nodes that you cataloged specifying the NetBIOS or SNA protocol by issuing the LIST DATABASE DIRECTORY command:

```
db2 LIST DATABASE DIRECTORY show detail > database_list.log
```

4. If you are going recatalog your nodes using a different node name, remove all databases using those nodes by issuing the UNCATALOG DATABASE command:

```
db2 UNCATALOG DATABASE database-name
```

5. Recatalog your nodes specifying TCP/IP as the protocol. If you use the original node name, you do not need to recatalog your databases.

```
db2 CATALOG TCPIP NODE new-node REMOTE host-name
SERVER instance-svcename REMOTE_INSTANCE instance-name
```

You can determine the value of *instance-svcename* by looking at the value of the *svcename* database manager configuration parameter for that instance.

6. If you did not recatalog your nodes using the original node names, recatalog your databases using the new node name.

```
db2 CATALOG DATABASE db-name [AS alias-db-name]
AT NODE new-node
```

**Related concepts:**

- “Post-migration tasks for DB2 clients” on page 123

**Related tasks:**

- “Cataloging a database from a client using the CLP” in *Quick Beginnings for DB2 Clients*
- “Cataloging a TCP/IP node from a client using the CLP” in *Quick Beginnings for DB2 Clients*

**Related reference:**

- “CATALOG DATABASE command” in *Command Reference*
- “CATALOG TCPIP/TCPIP4/TCPIP6 NODE command” in *Command Reference*
- “LIST DATABASE DIRECTORY command” in *Command Reference*
- “LIST NODE DIRECTORY command” in *Command Reference*
- “UNCATALOG DATABASE command” in *Command Reference*
- “UNCATALOG NODE command” in *Command Reference*

---

## Verifying migration of DB2 clients

When the migration of your DB2 client is complete, it is a good practice to run some tests in the new migrated environment to verify that your DB2 client is working as expected. These tests can consist of running batch programs that connect to databases in a DB2 server or any programs or scripts that you use for benchmarking.

### Prerequisites:

- Ensure that you have network connectivity from the DB2 client to the DB2 server.
- Ensure that the DB2 servers and instances are up and running.

### Procedure:

To verify that your DB2 client migration 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 DATABASE sample user mickey using mouse
```

```
Database Connection Information
```

```
Database server = DB2/AIX64 9.1.0
SQL authorization ID = TESTDB2
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-migration task to re-create the same client connectivity environment you had prior to migration.
3. Run your client database applications or scripts that connect to your databases to ensure they are working as expected.

### Related concepts:

- “Post-migration tasks for DB2 clients” on page 123

### Related tasks:

- “Saving DB2 client configuration information” on page 111

### Related reference:

- “CONNECT (Type 1) statement” in *SQL Reference, Volume 2*



---

## Part 4. Migrating database applications and routines

This part of the book contains the following chapters:

Chapter 17, "Migration overview for database applications and routines," on page 129

Chapter 18, "Migration essentials for database applications," on page 131

Chapter 19, "Migration essentials for routines," on page 137

Chapter 20, "Pre-migration tasks for database applications and routines," on page 139

Chapter 21, "Migrating database applications," on page 141

Chapter 22, "Migrating routines," on page 153

Chapter 23, "Post-migration tasks for database applications and routines," on page 165





---

## Chapter 17. Migration overview for database applications and routines

Migrating your DB2 servers from DB2 UDB Version 8 to DB2 Version 9 might require that you migrate your database applications and routines to run in DB2 Version 9.

Migrating your applications and routines involves the following actions:

- Test whether your applications and routines perform as expected in a DB2 Version 9 testing environment. You do not need to migrate your applications and routines if they run successfully.
- If your applications or routines have errors running in DB2 Version 9, you should:
  - Review migration essentials for database applications to identify any changes in DB2 Version 9 that can impact your applications.
  - Review migration essentials for routines to identify any changes in DB2 Version 9 that can impact your routines.
  - Plan how to modify your applications and routines to handle these changes. Determine the steps you need to perform by reviewing the Migrating database applications or Migrating routines tasks.
  - Modify your applications and routines according to your plan.
  - Test your applications and routines in your DB2 Version 9 testing environment.
- Verify that your applications and routines perform as expected in your DB2 Version 9 production environment before deploying them.

If your applications and routines make use of any functionality that is deprecated in DB2 Version 9, you should plan how to remove this functionality from your application code in the near future.

Also, you should consider using new features available in DB2 Version 9 to enhance functionality and improve performance.

### Related concepts:

- “What’s new for V9.1: Application development enhancements summary” in *What’s New*
- “What’s new for V9.1: Deprecated functionality summary” in *What’s New*
- “Functionality deprecated or discontinued in DB2 database products that impacts migration” on page 29

### Related tasks:

- “Migrating database applications” on page 141
- “Migrating routines” on page 153
- “Planning migration for your database applications and routines” on page 11



---

## Chapter 18. Migration essentials for database applications

Migration essentials describe changes in application development support, changes to support new features, unsupported features, and deprecated features that might impact your database applications, scripts and tools.

### Changes to operating system support

Certain versions of UNIX, Linux and Windows operating systems are not supported in DB2 Version 9, such as AIX Version 4.3.3, Solaris 8, and Windows NT<sup>®</sup>. A complete list of supported operating systems is available at DB2 system requirements Web page. If your current version of operating system is not supported, you must upgrade it before you install DB2 Version 9.

In UNIX operating systems, only 64-bit kernels are supported. Your DB2 UDB Version 8 32-bit instances are migrated to DB2 Version 9 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 migrate to DB2 Version 9 so that they use the new run-time libraries in the operating system.

### Changes to application drivers

In DB2 Version 9, ODBC data sources include the installation name. After you migrate your instances, the existing ODBC data source names (DSN) are migrated to use the new IBM ODBC driver name. If your application uses the ODBC Driver manager to create a new DSN, you only have access to databases under the instances from the default copy.

The DB2 .NET Data Provider now supports the Microsoft .NET Framework, Version 2.0, and has additional new features such as support for the System.Data.Common base classes, DB2Types classes, scrollable and updatable result sets, data paging, bulk data copy, and running SQL statements as a batch. Modify your database applications if you want to take advantage of these new features to improve the performance of your application.

The DB2 JDBC type 3 driver is not supported in DB2 Version 9. Modify your Java applications and applets that use this driver to use the IBM DB2 Driver for JDBC and SQLJ with type 4 connections instead.

You must be aware of the following behavioral differences between DB2 UDB Version 8 and DB2 Version 9 IBM DB2 Driver for JDBC and SQLJ.

#### Default LOB data retrieval method has changed

By default, LOB retrieval is done using progressive streaming instead of using LOB locators where progressive streaming is supported by the DB2 server. The database manager dynamically determines the most efficient method for returning LOB data based on the size of the LOB and will make use of progressive streaming whenever possible.

#### Specifying a method for client-side conversion of input data types

By default, application data types are converted to column data types when column data type information is available. You can now choose to disable this conversion.

### Method using JDBC 1.0 positioned UPDATE operations is deprecated

The IBM DB2 Driver for JDBC and SQLJ supports a method of defining and using updatable **ResultSets** that follows the JDBC 1.0 standard. That method is deprecated and is not recommended. The JDBC 1.0 method involves using the `ResultSet.getCursorName` method to obtain the name of the cursor for the `ResultSet`, and defining a positioned UPDATE statement of the following form:

```
UPDATE table SET col1=value1,...coln=valueN
WHERE CURRENT OF cursorname
```

If you use the JDBC 1.0 method to update data on a database server that supports multiple-row FETCH, such as DB2 for z/OS Version 8 or later, the positioned UPDATE statement might update multiple rows, when you expect it to update a single row.

### Changes to 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 for a specific list of what is no longer supported.

The new IBM Database Add-Ins for Microsoft Visual Studio 2005 provides tools for rapid application development, database schema development, and debugging in this development environment.

### Changes to DB2 APIs and DB2 commands

The DB2 command line processor (CLP) and system commands have changed in DB2 Version 9. These changes include new options, and options that are not available anymore. The following table lists changes that might impact applications and scripts:

Table 12. Changes to DB2 CLP commands and system commands

| Command          | Summary of changes                                                                                                                                                                               |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CREATE DATABASE  | There is a new AUTOMATIC STORAGE clause. This command automatically runs the AUTOCONFIGURE command.                                                                                              |
| GET SNAPSHOT     | The command output includes new fields, and the values for existing fields have changed.                                                                                                         |
| INSPECT          | There is a new ROWCOMPESTIMATE clause to estimate the effectiveness of row compression for a table.                                                                                              |
| EXPORT           | The LOBSINFILE and CODEPAGE file type modifiers can be specified together. The generated lob file name appends the .lob extension after the 3-digit sequence number.                             |
| IMPORT           | The LOBSINFILE and CODEPAGE file type modifiers can be specified together. A row is rejected if a LOB file is not found (The same applies to the LOAD command).                                  |
| LIST TABLESPACES | The Content field in the command output has new values: <ul style="list-style-type: none"> <li>All permanent data. Regular table space</li> <li>All permanent data. Large table space</li> </ul> |

Table 12. Changes to DB2 CLP commands and system commands (continued)

| Command              | Summary of changes                                                                                                                                                                |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RECOVER DATABASE     | The RESTART option forces the recover utility to redo the restore phase and to ignore any prior recover operation that failed to complete                                         |
| REORG INDEXES/TABLE  | New options available are RESETDICTIONARY and KEEPDICTIONARY.                                                                                                                     |
| RESET DB CFG         | The command resets the database configuration parameters to the pre-database configuration values. The SELF_TUNING_MEMORY and the AUTO_RUNSTATS options will be set to ON.        |
| RESTORE DATABASE     | The new GENERATE SCRIPT option generates a redirected restore script from an existing backup image.                                                                               |
| ROLLFORWARD DATABASE | Both Coordinated Universal Time (UTC) and local time are valid input formats. The format of the output timestamp is the same as the specified input format.                       |
| db2atld              | This command is no longer supported. Refer to for details on how to use the LOAD command instead.                                                                                 |
| db2batch             | The command runs only in CLI mode. The -p and -cli options are removed. There are many new options including -iso to specify the isolation level.                                 |
| db2icrt and db2iupdt | The -w option is no longer available.                                                                                                                                             |
| db2licm              | The -l option has a new optional SHOW DETAIL clause. The -c, -g, and -x options are new.                                                                                          |
| db2look              | New -xs option exports all files necessary to register XML schemas and DTDs at the target database, and generates appropriate commands for registering them.                      |
| db2pd                | The -stack and -dump options are new.                                                                                                                                             |
| db2sampl             | The -path and -k options are no longer available. There are new options to specify the directory in which to create the database files, the name of the sample database and more. |

The corresponding DB2 API calls to the commands listed in Table 12 on page 132 have also changed. Review the changes for each DB2 API function and modify the corresponding function calls in your database applications.

### Changes to SQL statement syntax

The syntax of certain SQL statements has changed. The following table lists the syntax changes for SQL statements that might impact applications and scripts:

Table 13. Syntax changes to SQL statements

| SQL statement | Summary of changes                                                                                                                                                                                                                                                                                            |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ALTER TABLE   | <ul style="list-style-type: none"> <li>New clause DROP DISTRIBUTION replaces DROP PARTITIONING KEY, and new clause PARTITIONING KEY ADD DISTRIBUTE BY HASH replaces ADD PARTITIONING KEY.</li> <li>Other new clauses are DROP COLUMN, ALTER COLUMN SET DATA TYPE, SET NOT NULL, and DROP NOT NULL.</li> </ul> |
| CREATE INDEX  | Up to 64 columns can be specified as index key. The new NOT PARTITIONED clause indicates that a single index should be created that spans all of the data partitions defined for the table. There are new clauses for XML index specification.                                                                |

Table 13. Syntax changes to SQL statements (continued)

| SQL statement | Summary of changes                                                                                                                                                                                                                                                                                               |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SET INTEGRITY | The FOR clause has new and changed options. Online integrity processing is available by using the new ALLOW NO ACCESS, ALLOW READ ACCESS, and ALLOW WRITE ACCESS options to define table access while it is being processed for integrity. Other new options are GENERATE IDENTITY and ALLOW QUERY OPTIMIZATION. |

### Changes to SQL Administrative views and routines, and catalog views

After database migration to DB2 Version 9, the catalog views under the SYSCAT schema remain compatible with catalog views defined in DB2 UDB Version 8. However, the COLNAMES column in SYSIBM.INDEXES and SYSCAT.INDEXES is deprecated in DB2 Version 9 and will be removed in a future release. To obtain the same information, query the table SYSCAT.INDEXCOLUSE instead. Also, although the SYSCAT.BUFFERPOOLS view structure has not changed, the ESTORE column value is set to 'N' during database migration because extended storage for buffer pools is no longer supported in DB2 Version 9. The ESTORE column has been removed from SYSIBM.SYSBUFFERPOOLS.

SQL Administrative routines include changes such as new parameters, new columns returned, and replacement by system-defined table functions and views. Also, all the system-defined table functions with names that start with SNAPSHOT\_ are deprecated in DB2 Version 9. Review the list of the Deprecated SQL administrative routines and their replacement routines or views to determine the changes that might impact your applications.

### Database packages

When you migrate 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 are dependent on a database object 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 when the optimizer needs access, you should rebind your database packages to control when rebinding occurs.

### Changes to 32-bit and 64-bit support

On Linux and UNIX operating systems excluding Linux on x86, DB2 Version 9 supports only 64-bit kernels and 64-bit instances. Therefore, when you migrate to DB2 Version 9, your DB2 UDB Version 8 32-bit instances are migrated to 64-bit instances.

The DB2 Version 9 64-bit instances on the Linux and UNIX operating systems, except for the Linux operating system on Itanium Platform Family (IPF), include 32-bit shared libraries. The DB2 Version 9 64-bit instances on Linux on IPF include only 64-bit shared libraries. If you have 32-bit applications that access databases on these 64-bit instances, you need to ensure that you link your application to the correct shared library path to run successfully.

The following table indicates applications that will run after you migrate to DB2 Version 9:

Table 14. Shared library paths used in DB2 database applications

| Application | Instance                     | Embedded shared library paths                                   |
|-------------|------------------------------|-----------------------------------------------------------------|
| 32-bit      | 32-bit                       | \$INSTHOME/sql11ib/lib<br>\$INSTHOME/sql11ib/lib32              |
| 32-bit      | 64-bit (except Linux on IPF) | \$INSTHOME/sql11ib/lib <sup>1</sup><br>\$INSTHOME/sql11ib/lib32 |
| 64-bit      | 64-bit                       | \$INSTHOME/sql11ib/lib <sup>2</sup><br>\$INSTHOME/sql11ib/lib64 |

**Notes:**

1. \$INSTHOME/sql11ib/lib is a symbolic link to \$INSTHOME/sql11ib/lib32
2. \$INSTHOME/sql11ib/lib is a symbolic link to \$INSTHOME/sql11ib/lib64

Where INSTHOME is your instance home directory.

During DB2 Version 9 installation, statements are added to the db2profile file to set the environment variables for library search path. The following table shows the settings of the library path environment variables for each operating system:

Table 15. Environment variables settings for library search path

| Environment variables                                                                                                                                          | Application               | Shared library paths                 |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|--------------------------------------|
| <ul style="list-style-type: none"> <li>• LIBPATH (AIX)</li> <li>• LD_LIBRARY_PATH (HP-UX, Linux, and Solaris)</li> <li>• SHLIB_PATH (HP-UX on PA32)</li> </ul> | 32-bit                    | INSTHOME <sup>1</sup> /sql11ib/lib32 |
| <ul style="list-style-type: none"> <li>• LIBPATH (AIX)</li> <li>• LD_LIBRARY_PATH (HP-UX, Linux, and Solaris)</li> <li>• SHLIB_PATH (HP-UX on PA32)</li> </ul> | 64-bit                    | INSTHOME/sql11ib/lib64               |
| LIB (Windows)                                                                                                                                                  | 32-bit on 64-bit instance | DB2PATH <sup>2</sup> \lib\Win32      |
| LIB (Windows)                                                                                                                                                  | 32-bit or 64-bit          | DB2PATH\lib                          |

These environment variables specify additional locations where DB2 shared libraries can be loaded at application runtime, allowing your application to run after you migrate to DB2 Version 9 but if you not specify the correct shared library path.

**Changes to DB2 server behavior**

There are new registry variables, new database and database manager configuration parameters, and new default values for those parameters, and that could impact your application behavior or performance. Review Changes in DB2 registry variables, configuration parameters and database physical design characteristics to assess the impact.

After migration of your DB2 server, you need to compare your registry variable and configuration parameter values to your values before migration.

**DB2 clients connectivity support**

Your applications can use Version 8 DB2 clients to access databases in DB2 Version 9 servers. However, only DB2 UDB Version 8 functionality is

available to your application. Review migration essentials for DB2 clients for details on client connectivity and to identify changes that in support that can impact your DB2 clients.

**Related concepts:**

- Chapter 17, “Migration overview for database applications and routines,” on page 129
- Chapter 23, “Post-migration tasks for database applications and routines,” on page 165
- Chapter 20, “Pre-migration tasks for database applications and routines,” on page 139
- “LOBs in JDBC applications with the IBM DB2 Driver for JDBC and SQLJ” in *Developing Java Applications*

**Related tasks:**

- “Migrating database applications” on page 141
- “Migrating routines” on page 153
- “Planning migration for your database applications and routines” on page 11

**Related reference:**

- “Changed APIs and data structures” in *Administrative API Reference*



---

## Chapter 19. Migration essentials for routines

Migration essentials describe changes in application development support, changes to support new features, unsupported features, and deprecated features that might impact your routines. The changes described in the migration essentials for database applications could also impact your routines.

### Development software support

The information about development software support in the migration essentials for database applications applies to external stored procedures and user-defined functions (UDFs).

### 32-bit external stored procedures and UDFs

Unfenced 32-bit external stored procedures and UDFs are only supported in DB2 Version 9 32-bit instances, they will not run in DB2 Version 9 64-bit instances. If you migrated from a DB2 Version 8 32-bit instance to a DB2 Version 9 64-bit instance, you must change them to fenced stored procedures or UDFs or rebuild them as 64-bit routine libraries.

The implementation for LOB locators depends on DB2 database product installed. If you migrate from a DB2 UDB Version 8 32-bit instance to a DB2 Version 9 64-bit instance, you must rebuild 32-bit external routines that use LOB locators as 64-bit routine libraries.

Support for default function entry points in external routine libraries is deprecated in DB2 Version 9. If you migrated from a DB2 UDB Version 8 32-bit instance on AIX or Windows operating systems, you should specify an explicit entry point for your routine library.

### SQL stored procedures

SQL stored procedures that you created in DB2 UDB Version 8.1 will run in DB2 Version 9 if you migrate from a DB2 UDB Version 8 32-bit instance to a DB2 Version 9 32-bit instance, provided that they do not reference any unsupported features. The same applies if you migrate from a DB2 UDB Version 8 64-bit instance to a DB2 Version 9 64-bit instance. However, if you migrate from a DB2 UDB Version 8 32-bit instance to a DB2 Version 9 64-bit instance, your SQL procedures do not run because external C routines declared unfenced are no longer supported in DB2 Version 9 and the shared library path points to 32-bit libraries. You must drop and re-create these SQL procedures.

If you created SQL stored procedures in DB2 UDB Version 8.2 and migrate your databases to DB2 Version 9, your SQL stored procedures are migrated to the DB2 Version 9 executable code and will function successfully provided that they do not reference any unsupported features. If any issues arise after migration, you can simply drop and re-create these stored procedures.

### Java external routines

DB2 Version 9 installs a 32-bit JVM by default on Linux on x86 and Windows (when DB2 Version 9 32-bit product is installed) operating systems. For all other supported operating systems, DB2 Version 9 installs a 64-bit JVM. On Linux and UNIX operating systems, this JVM is linked to the `INSTHOME/sql1lib/java/jdk32` directory for a 32-bit instance and the

INSTHOME/sql11ib/java/jdk64 directory for a 64-bit instance, where INSTHOME is set to the instance home owner directory.

If you migrate an instance to DB2 Version 9, the *jdk\_path* parameter is set to the following value:

Table 16. *jdk\_path* configuration parameter settings

| DB2 Version 9 instance    | Operating System | <i>jdk_path</i> value       |
|---------------------------|------------------|-----------------------------|
| 32-bit instance           | Linux and UNIX   | INSTHOME/sql11ib/java/jdk32 |
| 64-bit instance           | Linux and UNIX   | INSTHOME/sql11ib/java/jdk64 |
| 32-bit or 64-bit instance | Windows          | DB2PATH\java\jdk            |

In DB2 Version 9 64-bit instances, Java external routines require that the *jdk\_path* parameter is set to a 64-bit JVM installation path to run successfully. 32-bit Java stored procedures and UDFs will fail to run in a DB2 Version 9 64-bit instance.

### Developer Workbench replaces the Development Center

In DB2 Version 9, the Developer Workbench replaces the Development Center from DB2 UDB Version 8. The Developer Workbench is an Eclipse-based tool that includes functionality that is comparable to the Development Center along with additional features. The Developer Workbench provides a wizard to migrate existing Development Center projects into Developer Workbench.

#### Related concepts:

- Chapter 17, “Migration overview for database applications and routines,” on page 129
- Chapter 23, “Post-migration tasks for database applications and routines,” on page 165
- Chapter 20, “Pre-migration tasks for database applications and routines,” on page 139
- “What’s new for V9.1: Developer Workbench replaces the Development Center” in *What’s New*

#### Related tasks:

- “Migrating routines” on page 153
- “Planning migration for your database applications and routines” on page 11
- “Rebinding packages in migrated databases” on page 95

---

## Chapter 20. Pre-migration tasks for database applications and routines

Before you migrate your database applications and routines, you should perform certain tasks to help ensure a successful migration.

Prepare for the migration of your database applications and routines by performing the following tasks:

1. Review Chapter 18, “Migration essentials for database applications” to determine which changes might impact your database applications.
2. Review Chapter 19, “Migration essentials for routines” to determine which changes might impact your routines.
3. Plan your migration strategy.
4. Upgrade your operating system to a supported level, if necessary.
5. Upgrade your supported level, if necessary.
6. Optional: Migrate your DB2 client or install a Version 9 application driver if your application requires one. Although DB2 Version 9 provides connectivity support for earlier DB2 clients, migrating to a Version 9 DB2 client eliminates any limitations and incompatibilities between releases.
7. Test your database applications in a DB2 Version 9 testing environment. If testing is successful, you do not need to migrate your applications. However, review the Migrating database applications task and consider performing any steps that can help you improve performance.
8. Test your routines in a DB2 Version 9 testing environment. If testing is successful, you do not need to migrate your routines. However, review the Migrating routines task and consider performing any steps that can help you improve performance.

### Related concepts:

- “Benchmark testing” in *Performance Guide*
- Chapter 17, “Migration overview for database applications and routines,” on page 129
- “Sample files” in *Samples Topics*

### Related tasks:

- “Verifying migration of DB2 servers” on page 98



---

## Chapter 21. Migrating database applications

This chapter describes how to migrate your database applications. It contains the following sections:

- “Migrating database applications”
- “Migrating embedded SQL and CLI applications” on page 143
- “Migrating Java applications that use IBM DB2 Driver for JDBC and SQLJ” on page 145
- “Migrating Java applications that use DB2 JDBC Type 2 or 3 driver” on page 146
- “Migrating ADO.NET applications” on page 148
- “Migrating scripts” on page 149
- “Migrating 32-bit database applications to run on 64-bit instances” on page 151

---

### Migrating database applications

If you migrate your DB2 servers from DB2 UDB Version 8 to DB2 Version 9, you might need to migrate your database applications to support the changes in DB2 Version 9. Migrating your applications involves modifying your code and rebuilding your applications.

Test your database applications in a DB2 Version 9 testing environment. If they run successfully you are not required to change them. You only need to modify your application code to remove the use of deprecated or unsupported functionality in DB2 Version 9 or if you want to use new features.

#### Prerequisites:

- Review the migration essentials for database applications to identify any changes that apply to your applications.
- Ensure that you have access to a DB2 Version 9 server, including instances and databases. The DB2 server can be part of a testing environment.
- Ensure that the operating system is at a version level that is supported by DB2 database products.
- Ensure that the development software is at a version level that is supported by DB2 database products.
- Perform the pre-migration tasks for database applications.

#### Restrictions:

This procedure only applies to database applications programmed in C, C++, COBOL, FORTRAN, Java, Perl, PHP, REXX, and .NET languages.

#### Procedure:

To migrate your database applications to DB2 Version 9:

1. If you identified changes in DB2 Version 9 that impact your applications, edit your application code or scripts to modify:
  - SQL statements syntax

- SQL statements using catalog views and SQL Administrative views and routines
  - DB2 API calls
  - Application programming interface calls such as JDBC, ODBC and CLI
2. If you identified changes specific to the development environment that impact your applications, modify them to support these changes. Migrate your:
    - Embedded SQL applications
    - Java applications that use the IBM DB2 Driver for JDBC and SQLJ or that use the DB2 JDBC Type 2 or Type 3 driver.
    - ADO and .NET applications.
    - scripts that use DB2 CLP commands.
    - 32-bit database applications to run on 64-bit instances
  3. Rebuild all changed database applications programmed in C/C++, COBOL, FORTRAN, and REXX, using the appropriate DB2 build file and specifying the appropriate DB2 shared library path as shown in Table 15 on page 135.
  4. Test your database applications to ensure that they run as expected using DB2 Version 9.

After migrating your database applications, perform the recommended post-migration tasks for database applications to ensure that your migration was successful.

**Related concepts:**

- Chapter 18, “Migration essentials for database applications,” on page 131
- Chapter 20, “Pre-migration tasks for database applications and routines,” on page 139
- “Getting started with database application development” in *Getting Started with Database Application Development*

**Related tasks:**

- “Migrating embedded SQL and CLI applications” on page 143
- “Migrating Java applications that use IBM DB2 Driver for JDBC and SQLJ” on page 145
- “Migrating Java applications that use DB2 JDBC Type 2 or 3 driver” on page 146
- “Migrating ADO.NET applications” on page 148
- “Migrating scripts” on page 149
- “Migrating 32-bit database applications to run on 64-bit instances” on page 151
- “Setting up the UNIX application development environment” in *Getting Started with Database Application Development*
- “Setting up the Windows application development environment” in *Getting Started with Database Application Development*

**Related reference:**

- “Supported programming languages and compilers for database application development” in *Getting Started with Database Application Development*
- “Changed APIs and data structures” in *Administrative API Reference*

---

## Migrating embedded SQL and CLI applications

To ensure that the embedded SQL and CLI applications that you built for DB2 UDB Version 8 will work with DB2 Version 9, you must migrate these applications.

The steps that you should perform to migrate your applications depend on:

- The operating system
- The bit-width of the application (32-bit or 64-bit)
- The bit-width of the DB2 instance to which the applications will be deployed (32-bit or 64-bit)

### Prerequisites:

- Review the migration essentials for applications to identify key changes that might apply to your routines.
- Ensure that you have access to a DB2 Version 9 server, including instances and databases. The DB2 server can be part of a testing environment.
- Ensure that the operating system is at a version level that is supported by DB2 database products.
- Ensure that the C, C++, COBOL, FORTRAN, or REXX development software is at a version level that is supported by DB2 database products.
- Perform the pre-migration tasks for database applications.

### Restrictions:

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

### Procedure:

To migrate your embedded SQL and CLI applications to DB2 Version 9:

1. If you modified the library path environment variables, ensure that those variables include the correct DB2 shared library path for your applications as shown in Table 15 on page 135. The environment variables listed in this table specify additional paths to enable your applications to find the appropriate DB2 shared library at runtime (in most cases).

**On the Linux operating system:** if you link an application using the RPATH link option without also specifying the RUNPATH link option, the LD\_LIBRARY\_PATH environment variable will be ignored at application run time, which can cause your application to fail.

2. Test your embedded SQL applications in a DB2 Version 9 testing environment. If testing is successful, you do not need to perform any additional steps.
3. If you identified changes in DB2 Version 9 that impact your applications, edit your application code to modify:
  - SQL statements syntax
  - SQL statements using catalog views and SQL Administrative views and routines
  - DB2 API calls
  - Application programming interface calls such as ODBC and CLI

You can run the DB2 precompiler to check the syntax of your SQL statements in your database applications and determine whether you need to make any changes.

4. To explicitly specify the correct DB2 shared library path for your applications, do one of the following:
  - If the application source code is available, rebuild the application. Specify the required DB2 shared library path as shown in Table 15 on page 135. This is the best option.
  - Create a wrapper script to run your application. In the wrapper script, explicitly set the library path environment variable to the required DB2 shared library path as shown in Table 15 on page 135.
  - 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.
5. For applications that you did not rebuild but that you modified, rebind the embedded SQL application packages to the target DB2 database. During database migration, the existing packages for your embedded SQL applications are invalidated.
6. Test your embedded SQL and CLI applications to ensure that they run as expected using DB2 Version 9.

After migrating your embedded SQL and CLI applications, perform the recommended post-migration tasks for database applications.

**Related concepts:**

- Chapter 19, “Migration essentials for routines,” on page 137
- Chapter 20, “Pre-migration tasks for database applications and routines,” on page 139
- “Building embedded SQL applications” in *Developing Embedded SQL Applications*
- “Introduction to DB2 CLI and ODBC” in *Call Level Interface Guide and Reference, Volume 1*
- “Introduction to embedded SQL” in *Developing Embedded SQL Applications*
- Chapter 18, “Migration essentials for database applications,” on page 131

**Related tasks:**

- “Rebinding packages in migrated databases” on page 95
- “Building CLI applications on UNIX” in *Call Level Interface Guide and Reference, Volume 1*
- “Building CLI applications on Windows” in *Call Level Interface Guide and Reference, Volume 1*
- “Setting up the CLI environment” in *Call Level Interface Guide and Reference, Volume 1*
- “Setting up the embedded SQL development environment” in *Developing Embedded SQL Applications*
- “Migrating C, C++, and COBOL routines” on page 155

**Related reference:**

- “Supported programming languages and compilers for database application development” in *Getting Started with Database Application Development*
- “DB2 CLI and ODBC API support in the IBM DB2 Driver for ODBC and CLI” in *Call Level Interface Guide and Reference, Volume 1*



---

## Migrating Java applications that use IBM DB2 Driver for JDBC and SQLJ

To ensure that the Java database applications that you built for DB2 UDB Version 8 using the IBM DB2 Driver for JDBC and SQLJ function successfully with DB2 Version 9, you must migrate them.

### Prerequisites:

- Review the migration essentials for applications to identify key changes that might impact your Java database applications.
- Ensure that you have access to a DB2 Version 9 server, including instances and databases. The DB2 server can be part of a testing environment.
- Ensure that the operating system is at a version level that is supported by DB2 database products.
- Ensure that the Java application development software is at a version level that is supported by DB2 database products.
- Perform the pre-migration tasks for database applications.

### Restrictions:

- The minimum supported Java SDK is Java SDK 1.4.2.
- This procedure applies only to Java applications using the IBM DB2 Driver for JDBC and SQLJ.

### Procedure:

To migrate your Java database applications using the IBM DB2 Driver for JDBC and SQLJ to DB2 Version 9:

1. If required, update the applications to manage the following differences between DB2 UDB Version 8 and DB2 Version 9. You can manage some of these differences by setting `Connection` or `DataSource` object properties without any modifications to application source code:
  - To continue using LOB locators instead of LOB retrieval using progressive streaming, set the `fullyMaterializeLobData` property to `false` and set the `progressiveStreaming` property to `DB2BaseDataSource.NO` in the `Connection` or `DataSource` object.
  - To maintain the same default value, set the `sendDataAsIs` value to `false` so that the IBM DB2 Driver for JDBC and SQLJ converts input parameter values to the target column data types. If you set the `sendDataAsIs` property to `true`, the IBM DB2 Driver for JDBC and SQLJ will not convert input data to the target column data types, even if the information is available in the `Connection` or `DataSource` object.
  - To remove the use of the JDBC 1.0 positioned update before it becomes unsupported, you should modify your applications to use the JDBC 2.0 method
2. If you identified changes in DB2 Version 9 that impact your applications, update your application code to modify:
  - SQL statements syntax
  - SQL statements using catalog views and SQL Administrative views and routines
  - JDBC calls

3. If you changed your Java application source code, rebuild your Java application. Refer to one of the following tasks for details on how to rebuild them:
  - Building JDBC applications
  - Building SQLJ applications
4. Test your Java applications to ensure that they run as expected using DB2 Version 9.

Upon completion of this task, your Java application should perform successfully using DB2 Version 9.

After migrating your Java applications, perform the recommended post-migration tasks for database applications.

**Related concepts:**

- Chapter 19, “Migration essentials for routines,” on page 137
- “Supported Java application development software” in *Developing Java Applications*
- Chapter 20, “Pre-migration tasks for database applications and routines,” on page 139
- “LOBs in JDBC applications with the IBM DB2 Driver for JDBC and SQLJ” in *Developing Java Applications*
- “Introduction to Java application development for DB2” in *Developing Java Applications*
- Chapter 18, “Migration essentials for database applications,” on page 131

**Related tasks:**

- “Specifying updatability, scrollability, and holdability for ResultSets in JDBC applications” in *Developing Java Applications*
- “Building JDBC applications” in *Developing Java Applications*
- “Building SQLJ programs” in *Developing Java Applications*
- “Installing the IBM DB2 Driver for JDBC and SQLJ” in *Developing Java Applications*

---

## Migrating Java applications that use DB2 JDBC Type 2 or 3 driver

The DB2 JDBC Type 3 driver, which was deprecated in DB2 Version 8, is no longer supported and therefore not shipped with DB2 Version 9. You must migrate your Java applications to the IBM DB2 Driver for JDBC and SQLJ.

The DB2 JDBC Type 2 driver is deprecated. Although your Java applications that use the DB2 JDBC Type 2 driver will function successfully with DB2 Version 9, migrating those applications to the IBM DB2 Driver for JDBC and SQLJ as soon as possible will help you to avoid possible support problems in future releases.

**Prerequisites:**

- Review the migration essentials for applications to identify key changes that might impact your Java database applications.
- Ensure that you have access to a DB2 Version 9 server, including instances and databases. The DB2 server can be part of a testing environment.

- Ensure that the operating system is at a version level that is supported by DB2 database products.
- Ensure that the Java application development software is at a version level that is supported by DB2 database products.
- Perform the pre-migration tasks for database applications.

**Restriction:**

- The minimum supported Java SDK is Java SDK 1.4.2.

**Procedure:**

To migrate your Java database applications to DB2 Version 9:

1. Install the IBM DB2 Driver for JDBC and SQLJ.
2. Update your Java applications to use the IBM DB2 Driver for JDBC and SQLJ.
3. Review the information in the following topics to identify behavioral differences between drivers that might impact your Java application:
  - Driver support for JDBC APIs
  - JDBC differences between the IBM DB2 Driver for JDBC and SQLJ and other DB2 JDBC drivers
  - SQLJ differences between the IBM DB2 Driver for JDBC and SQLJ and other DB2 JDBC drivers
4. Update your Java applications to resolve any issues created by the behavioral differences that you identified in the previous step. These changes might include modifying existing method calls and removing the use of unsupported features in DB2 Version 9.
5. If you identified changes in DB2 Version 9 that impact your Java applications, update your application code to modify:
  - SQL statements syntax
  - SQL statements using catalog views and SQL Administrative views and routines
6. If you changed your Java application source code in any of the previous steps, rebuild your Java applications. Refer to one of the following tasks for details on how to rebuild them:
  - Building JDBC applications
  - Building SQLJ applications
7. Test your Java applications to ensure that they run as expected using DB2 Version 9.

After migrating your Java applications, perform the recommended post-migration tasks for database applications to ensure that your migration was successful.

**Related concepts:**

- Chapter 19, “Migration essentials for routines,” on page 137
- “Supported Java application development software” in *Developing Java Applications*
- Chapter 20, “Pre-migration tasks for database applications and routines,” on page 139
- “Introduction to Java application development for DB2” in *Developing Java Applications*
- Chapter 18, “Migration essentials for database applications,” on page 131

**Related tasks:**

- “Installing the IBM DB2 Driver for JDBC and SQLJ” in *Developing Java Applications*
- “Connecting to a data source using the DriverManager interface with the IBM DB2 Driver for JDBC and SQLJ” in *Developing Java Applications*
- “Building JDBC applications” in *Developing Java Applications*
- “Building SQLJ programs” in *Developing Java Applications*
- “Migrating Java applications that use IBM DB2 Driver for JDBC and SQLJ” on page 145
- “Migrating Java routines” on page 157

**Related reference:**

- “Driver support for JDBC APIs” in *Developing Java Applications*
- “JDBC differences between the IBM DB2 Driver for JDBC and SQLJ and other DB2 JDBC drivers” in *Developing Java Applications*
- “SQLJ differences between the IBM DB2 Driver for JDBC and SQLJ and other DB2 JDBC drivers” in *Developing Java Applications*

---

## Migrating ADO.NET applications

To ensure that the ADO.NET applications that you built for DB2 UDB Version 8 function successfully with DB2 Version 9, you need to migrate them. Migrating your ADO.NET applications might require that you rebuild them.

You do not need to rebuild ADO.NET applications that use the OLE DB .NET Data Provider or the ODBC .NET Data Provider to run with DB2 Version 9. However, migrating these applications to the DB2 .NET Data Provider can be beneficial for the following reasons:

- The DB2 .NET Data Provider has a far more extensive set of APIs than the OLE DB and ODBC .NET data providers.
- Access to the IBM database development productivity tools integrated with Visual Studio.
- Use of the DB2 .NET Data Provider can bring significant performance improvements.

**Prerequisites:**

- Review the migration essentials for applications to identify key changes that might apply to your ADO.NET applications.
- Ensure that you have access to a DB2 Version 9 server, including instances and databases. The DB2 server can be part of a testing environment.
- Ensure that the operating system is at a version level that is supported by DB2 database products.
- Ensure that a supported version of the .NET Framework software is installed on the DB2 database client computer.
- Perform the pre-migration tasks for database applications.

**Procedure:**

To migrate your ADO.NET applications to DB2 Version 9:

1. If you identified changes in DB2 Version 9 that impact your applications, edit your application code to modify:

- SQL statements syntax
  - SQL statements using catalog views and SQL Administrative views and routines
2. Rebuild your ADO.NET applications that use the DB2 .NET Data Provider, referencing a DB2 .NET Data Provider for DB2 Version 9. DB2 Version 9 features two DB2 .NET Data Provider versions:
    - one for the .NET Framework Version 1.1
    - one for the .NET Framework Version 2.0, which is the most highly optimized data provider for DB2 Version 9, and it has a greatly extended set of features, in case that you plan to further develop your applications.
  3. Test your ADO.NET applications to ensure that they run as expected using DB2 Version 9.

After migrating your ADO.NET applications, perform the recommended post-migration tasks for database applications.

**Related concepts:**

- “Supported .NET development software” in *Developing ADO.NET and OLE DB Applications*
- Chapter 20, “Pre-migration tasks for database applications and routines,” on page 139
- Chapter 18, “Migration essentials for database applications,” on page 131
- “ADO.NET development for DB2 databases” in *Developing ADO.NET and OLE DB Applications*
- “DB2 .NET Data Provider” in *Developing ADO.NET and OLE DB Applications*
- “IBM DB2 Development Add-In overview” in *Developing ADO.NET and OLE DB Applications*
- “Limitations for ADO applications” in *Developing ADO.NET and OLE DB Applications*
- “ODBC .NET Data Provider” in *Developing ADO.NET and OLE DB Applications*
- “OLE DB .NET Data Provider” in *Developing ADO.NET and OLE DB Applications*

---

## Migrating scripts

You might need to migrate your scripts that use DB2 command line processor (CLP) commands, DB2 system commands or SQL statements due to changes in DB2 Version 9 related to SQL statements, DB2 CLP and system commands, SQL Administrative views and routines, and catalog views.

**Prerequisites:**

- Review the migration essentials for database applications to identify any changes that apply to your scripts.
- Ensure that you have access to a DB2 Version 9 server, including instances and databases.
- Ensure that a DB2 Version 9 client is installed.
- Ensure that the operating system is at a version level that is supported by DB2 database products and development software.
- Ensure that the development software is at a version level that is supported by DB2 database products.
- Perform the pre-migration tasks for database applications.

**Restrictions:**

This procedure only applies to scripts that use DB2 CLP commands, DB2 system commands or SQL statements.

**Procedure:**

To migrate your scripts with DB2 CLP commands to DB2 Version 9:

1. Run your scripts to detect any incompatibilities with DB2 Version 9. If your scripts run successfully, you do not need to perform any additional steps. However, consider performing steps 2 to 5 to remove deprecated features in DB2 Version 9 before they become unsupported or to use new command functionality.
2. Edit your scripts and correct the syntax for SQL statements.
3. Edit your scripts and correct the syntax for DB2 CLP and system commands. Remove the command options that are no longer available. Use new options, where available, to obtain new details in the command output or to access new functionality.
4. If your scripts read from the command output, you need to modify your scripts to be able to read the changes in the output. The command output text for the GET SNAPSHOT and LIST TABLESPACES commands have changed. Review migration essentials for database application for details on the DB2 command changes.
5. If your scripts perform snapshot or event monitoring, you need to modify your scripts to remove references to unsupported monitor elements or use a new name when they have been replaced by a new monitor element.
6. Edit your scripts to use the new SQL Administrative views and routines in DB2 Version 9. Although the DB2 Version 8 SQL Administrative views and routines are available for compatibility, you should modify your scripts to use the new Version 9 views and routines before the Version 8 views and routines become unsupported. Using the new 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 new view or routine.
  - Remove column names from your queries for columns that are not available in the new view.
  - Replace \* in your queries for a specific list of column names that you want to receive as a result set because the new 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 new routine or querying a new view that returns additional columns.
7. Test your scripts to ensure that they run as expected using DB2 Version 9.

After migrating your database applications, perform the recommended post-migration tasks for database applications.

**Related concepts:**

- Chapter 18, “Migration essentials for database applications,” on page 131
- Chapter 20, “Pre-migration tasks for database applications and routines,” on page 139

**Related reference:**

- “Supported programming languages and compilers for database application development” in *Getting Started with Database Application Development*
- “Command line processor options” in *Command Reference*
- “db2 - Command line processor invocation command” in *Command Reference*
- “Command Line Processor (CLP) samples” in *Samples Topics*

---

## Migrating 32-bit database applications to run on 64-bit instances

If you migrate from a DB2 Version 8 32-bit instance to a DB2 Version 9 64-bit instance that includes 32-bit shared libraries, you need to ensure that your 32-bit database applications are linked to the appropriate shared library path to run them successfully.

You do not need to modify your 32-bit database applications if you linked them to the \$INSTHOME/sqllib/lib32 shared library path on Linux and UNIX or the DB2PATH\lib\Win32 shared library path on Windows, where INSTHOME is the instance home directory and DB2PATH is the location of the DB2 copy.

**Prerequisites:**

- Review the migration essentials for database applications to identify any changes that apply to your scripts.
- Ensure that you have access to a DB2 UDB Version 8 32-bit instance that you migrated to a DB2 Version 9 64-bit instance that includes 32-bit shared libraries.
- Ensure that the operating system is at a version level that is supported by DB2 database products.
- Ensure that the development software is at a version level that is supported by DB2 database products.
- Perform the pre-migration tasks for database applications.

**Restrictions:**

- This procedure applies only to 32-bit database applications programmed in C/C++, COBOL, FORTRAN, and REXX.
- This procedure indicates only the changes that are required to run 32-bit database applications on a 64-bit instance that includes 32-bit shared libraries. Check migration essentials for database application to see whether you need to apply additional changes to your applications.

**Procedure:**

To migrate 32-bit database applications to run on a DB2 Version 9 64-bit instance:

1. Ensure that the library path environment variables include the correct DB2 shared library path for 32-bit libraries as shown in Table 15 on page 135, so that the correct library can be loaded at runtime.
2. Test your 32-bit applications in a DB2 Version 9 testing environment. If testing is successful, you do not need to perform any additional steps. However, consider performing step 4 on page 152 or 5 on page 152 if they apply to your applications, to improve their support by using the correct client and shared library path.
3. If you identified changes in DB2 Version 9 that impact your 32-bit applications, edit your application code to modify:
  - SQL statements syntax

- SQL statements using catalog views and SQL Administrative views and routines
  - DB2 API calls
  - Application programming interface calls such as ODBC and CLI
4. Specify the correct library path by linking or rebuilding your 32-bit applications using the DB2 shared library paths for 32-bit libraries shown in Table 15 on page 135.
  5. Optional: If you no longer have the source code to rebuild your applications or the use environmental variables is not possible, you can run the **db2chglbpath** command to change the DB2 shared library path to `$INSTHOME/sql11ib/lib32` on your application binary file as long as it has an embedded runtime path. The embedded runtime path can be changed to a new path with the same length or less.
  6. Test your 32-bit applications to ensure that they run as expected using DB2 Version 9.

After migrating your 32-bit database applications, perform the recommended post-migration tasks for database applications.

**Related concepts:**

- Chapter 18, “Migration essentials for database applications,” on page 131
- Chapter 20, “Pre-migration tasks for database applications and routines,” on page 139
- “Support changes for 32-bit and 64-bit DB2 servers” on page 27

**Related reference:**

- “Supported programming languages and compilers for database application development” in *Getting Started with Database Application Development*
- “db2chglbpath - Modify the embedded runtime library search path command” in *Command Reference*



---

## Chapter 22. Migrating routines

This chapter describes how to migrate your database applications. It contains the following sections:

- “Migrating routines”
- “Migrating C, C++, and COBOL routines” on page 155
- “Migrating Java routines” on page 157
- “Migrating .NET CLR routines” on page 159
- “Migrating SQL procedures” on page 160
- “Migrating 32-bit external routines to run on 64-bit instances” on page 162

---

### Migrating routines

If you migrate your databases from DB2 UDB Version 8 to DB2 Version 9, you might need to migrate your routines to support the changes in DB2 Version 9. Migrating routines involves modifying your routine code, rebuilding your external routines, recreating your external routines in the database and recreating SQL routines.

Test your routines in a DB2 Version 9 testing environment. If they run successfully you are not required to change them. You only need to modify your routines to remove the use of deprecated or unsupported functionality in DB2 Version 9 or if you want to use new features.

#### Prerequisites:

- Review migration essentials for routines to identify any changes that apply to your routines.
- Ensure that you have access to migrated DB2 Version 9 databases. These can be test databases.
- Ensure that the operating system is at a version level that is supported by DB2 database products.
- Ensure that the development software is at a version level that is supported by DB2 database products.
- Perform the pre-migration tasks for routines.
- Ensure that you have SYSADM or DBADM authority to use the following SQL statements:
  - ALTER FUNCTION
  - ALTER PROCEDURE

Other authorizations allowed are listed in the Command Reference.

#### Restrictions:

This procedure only applies to SQL routines and external routines programmed in C/C++, COBOL (procedures only), Java, and .NET languages.

#### Procedure:

To migrate your routines to DB2 Version 9 databases:

1. If you identified changes in DB2 Version 9 that impact your routines, edit your routine code and modify:
  - SQL statement syntax
  - SQL statements using SQL Administrative views and routines, and catalog views
  - 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. Migrate your:
  - C, C++, and COBOL routines
  - Java routines.
  - .NET CLR routines.
  - SQL stored procedures, if you created your SQL procedures in DB2 Version 8.1 and you migrated from a 32-bit DB2 Version 8 instance to a DB2 Version 9 64-bit instance.
  - 32-bit external routines to run on 64-bit instances.
3. If you migrated from a DB2 Version 8 32-bit instance to a DB2 Version 9 64-bit instance, change unfenced 32-bit external stored procedures or user-defined functions (UDFs) to fenced using the ALTER FUNCTION or ALTER PROCEDURE statements. For example, the following statement changes an external procedure to fenced:

```
ALTER SPECIFIC PROCEDURE schema-name.specific-name FENCED
```

where *schema-name* is the schema that owns the external procedure and *specific-name* is the specific name that uniquely identifies a procedure that was indicated or assigned by default at creation.

4. If you migrated from a DB2 UDB Version 8 32-bit instance to a DB2 Version 9 64-bit instance, you must rebuild 32-bit external routines that use LOB locators as 64-bit routine libraries.
5. Rebuild all changed external routine libraries programmed in C and COBOL by using the appropriate DB2 build file and specifying the appropriate DB2 shared library path. For 32-bit routines, specify \$INSTHOME/sql11ib/lib32 as the shared library path, and for 64-bit routines, specify \$INSTHOME/sql11ib/lib64 as the shared library path, where INSTHOME is the instance home directory.
6. Test your routines to ensure that they run as expected using DB2 Version 9.

This procedure outlines the steps required to migrate your routines at a high level. See migration sub-tasks in step 2 for more details.

After migrating your routines, perform the recommended post-migration tasks for routines.

**Related concepts:**

- Chapter 19, “Migration essentials for routines,” on page 137
- Chapter 20, “Pre-migration tasks for database applications and routines,” on page 139
- “Supported routine programming languages” in *SQL Guide*

**Related tasks:**

- “Migrating C, C++, and COBOL routines” on page 155
- “Migrating Java routines” on page 157
- “Migrating .NET CLR routines” on page 159

- “Migrating SQL procedures” on page 160
- “Migrating 32-bit external routines to run on 64-bit instances” on page 162
- “Creating OLE automation routines” in *SQL Guide*
- “Developing routines” in *SQL Guide*

**Related reference:**

- “Supported programming languages and compilers for database application development” in *Getting Started with Database Application Development*
- “ALTER FUNCTION statement” in *SQL Reference, Volume 2*
- “ALTER PROCEDURE statement” in *SQL Reference, Volume 2*
- “DROP statement” in *SQL Reference, Volume 2*

---

## Migrating C, C++, and COBOL routines

To ensure that C, C++, or COBOL routines that you created prior to DB2 Version 9 function successfully, you must migrate them.

**Prerequisites:**

- Review the migration essentials for routines to identify key changes that might apply to your routines.
- Ensure that you have access to a DB2 Version 9 server, including instances and databases. The DB2 server can be part of a testing environment.
- Ensure that the operating system is at a version level that is supported by DB2 database products.
- 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:
  - Supported C routine development software
  - Supported C++ routine development software
  - Supported COBOL routine development software
- Perform the pre-migration tasks for routines.
- Ensure that you have SYSADM or DBADM authority to use the following statements:
  - ALTER FUNCTION
  - ALTER PROCEDURE

Other authorizations allowed are listed in the Command Reference.

**Restrictions:**

This procedure only applies to external routines programmed in C/C++, and COBOL (procedures only).

**Procedure:**

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

1. If you identify changes in DB2 Version 9 that impact your routines, edit your routine code and modify:
  - SQL statement syntax
  - SQL statements using SQL Administrative views and routines, and catalog views

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

Table 17. Migrating C, C++, and COBOL routines to a Version 9 64-bit instance

| Routine definition                                                  | Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|---------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>unfenced</i> 32-bit routine library                              | <p>Do one of the following:</p> <ul style="list-style-type: none"> <li>Define the routine as both fenced and not threadsafe using the ALTER PROCEDURE or ALTER FUNCTION statement with the FENCED clause and the NOT THREADSAFE clause. You can not use this option if LOB locators are referenced in the routine.</li> <li>Rebuild the routine source code into a 64-bit library using the DB2 Version 9 b1drtn script and redeploy the library to the DB2 server. If LOB locators are referenced in the routine, you must use this option. An advantage of this approach is that using a 64-bit library results in better routine runtime performance than using a 32-bit library.</li> </ul> |
| <i>fenced</i> 32-bit routine library                                | <p>Do one of the following:</p> <ul style="list-style-type: none"> <li>Define the routine as not threadsafe using the ALTER PROCEDURE or ALTER FUNCTION statement with the NOT THREADSAFE clause.</li> <li>Rebuild the routine source code into a 64-bit library using the DB2 Version 9 b1drtn scripts and redeploy the library to the DB2 server.</li> </ul>                                                                                                                                                                                                                                                                                                                                  |
| migrated from a DB2 UDB Version 8 32-bit instance (AIX and Windows) | <p>You should specify a default entry point for your routine library using the ALTER PROCEDURE or ALTER FUNCTION statement. For example, to explicitly specify the entry point for an existing procedure use the following statement:</p> <pre>ALTER SPECIFIC PROCEDURE <i>schema-name.specific-name</i>     EXTERNAL NAME '<i>library-name!function-name</i>'</pre> <p>where <i>library-name</i> is the library to be loaded and <i>function-name</i> is the explicit entry point for the function associated with the routine.</p>                                                                                                                                                            |

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

- For routines that you did not rebuild but that you modified, rebind the routine packages to the target DB2 database. During database migration, the existing packages for your routines are invalidated.
- Test your routines to verify your changes and to ensure that the routines run as expected using DB2 Version 9.

After migrating your routines, perform the recommended post-migration tasks for routines.

#### Related concepts:

- Chapter 19, "Migration essentials for routines," on page 137
- "Support for external routine development in C" in *SQL Guide*
- "Support for external routine development in C++" in *SQL Guide*
- Chapter 20, "Pre-migration tasks for database applications and routines," on page 139
- "Building applications and routines written in C and C++" in *Developing Embedded SQL Applications*
- "Building applications and routines written in COBOL" in *Developing Embedded SQL Applications*

- “COBOL procedures” in *SQL Guide*

**Related tasks:**

- “Rebinding packages in migrated databases” on page 95
- “Building CLI routines on UNIX” in *Call Level Interface Guide and Reference, Volume 1*
- “Building CLI routines on Windows” in *Call Level Interface Guide and Reference, Volume 1*
- “Setting up the CLI environment” in *Call Level Interface Guide and Reference, Volume 1*
- “Setting up the embedded SQL development environment” in *Developing Embedded SQL Applications*
- “Building C and C++ routine code” in *SQL Guide*

**Related reference:**

- “Support for external procedure development in COBOL” in *SQL Guide*
- “ALTER FUNCTION statement” in *SQL Reference, Volume 2*
- “ALTER PROCEDURE statement” in *SQL Reference, Volume 2*

---

## Migrating Java routines

After you migrate your databases, you must migrate Java routines that you created prior to DB2 Version 9 to ensure that they function as expected.

**Prerequisites:**

The following prerequisites must be met to perform this task:

- Review migration essentials for routines to identify key changes that might apply to your Java routines.
- Ensure that you have access to a DB2 Version 9 server, including instances and databases. The DB2 server can be a test system.
- Ensure that the operating system is at a version level that is supported by DB2 database products.
- Ensure that the Java routine development software is at a version level that is supported by DB2 database products.
- Ensure that you are using supported DB2 drivers for JDBC and SQLJ APIs.
- Perform the pre-migration tasks for routines.
- Ensure that you have SYSADM or DBADM authority to use the following statements:
  - ALTER FUNCTION
  - ALTER PROCEDURE

Other authorizations allowed are listed in the Command Reference.

**Procedure:**

To migrate your Java routines:

1. Ensure the *jdk\_path* database manager configuration parameter specifies the correct JVM to run your routines. Determine the current value by issuing the following command:

```
db2 GET DBM CFG
```

By default the *jdk\_path* database manager configuration parameter value is set during instance migration to the values shown in Table 16 on page 138. If you want to use a JVM other than the one installed in your DB2 Version 9 copy, you must set this configuration parameter to that JVM path with the same bit width as the DB2 instance by updating the *jdk\_path* parameter:

```
db2 UPDATE DBM CFG USING jdk_path <JVM-path>
```

2. Determine and set the Java driver for Java routine runtime support. Determine which Java driver is being used for Java routines by issuing the following command:

```
db2set DB2_USE_DB2JCCT2_JROUTINE
```

If the value of this registry variable is:

- OFF or not set, which is the default, your Java routines are using the IBM DB2 JDBC Type 2 driver.
- ON, your Java routines are using the IBM DB2 Driver for JDBC and SQLJ.

Use the IBM DB2 Driver for JDBC and SQLJ if you want access to features particular to this driver or if you anticipate using XML parameters. Indicate that your Java routines will use this driver by issuing the following command to set the DB2\_USE\_DB2JCCT2\_JROUTINE registry variable to ON at the global level:

```
db2set -g DB2_USE_DB2JCCT2_JROUTINE=ON
```

The -g option indicates that this value applies to all instances running on the same DB2 Version 9 copy.

3. Test your Java routines in your DB2 Version 9 database. If testing is successful and your Java routine perform as expected, you do not need to perform any additional steps.
4. If you are using the IBM DB2 Driver for JDBC and SQLJ and found any differences in the behavior of your Java routines, review the migrating Java applications task to learn how to manage those differences.
5. If you identify changes in DB2 Version 9 that impact your Java routines, edit your routine code and modify:
  - SQL statement syntax
  - SQL statements using SQL Administrative views and routines, and catalog views
  - JDBC calls
6. Explicitly define your Java routines as fenced using the ALTER FUNCTION or ALTER PROCEDURE statement with the FENCED clause. All Java routines run as fenced, regardless of how you defined them, but defining your Java routine definitions as fenced improves routine manageability and maintenance.
7. Optional: If your Java routine class is included within a JAR file that has been installed into a DB2 instance using a specific JAR file ID, ensure that the Java class is resolved more quickly by the DB2 database manager, ensure that you specify the JAR file ID as part of the EXTERNAL NAME clause in the routine definition. Use the ALTER PROCEDURE or ALTER FUNCTION statements to update the EXTERNAL NAME clause if required.
8. If you created projects in the Development Center to develop your Java routines, migrate any existing projects to the Developer Workbench using the migration wizard.
9. Test your Java routines to ensure that they run as expected with DB2 Version 9.

After migrating your Java routines, perform the recommended post-migration tasks for routines.

**Related concepts:**

- Chapter 19, “Migration essentials for routines,” on page 137
- “Supported Java routine development software” in *SQL Guide*
- “Supported drivers for JDBC and SQLJ” in *Developing Java Applications*
- Chapter 20, “Pre-migration tasks for database applications and routines,” on page 139
- “Java routines” in *SQL Guide*
- “Using SQLJ and JDBC in the same application” in *Developing Java Applications*
- “What’s new for V9.1: Java routine class loader enhancements” in *What’s New*

**Related tasks:**

- “Building JDBC routines” in *Developing Java Applications*
- “Building SQLJ routines” in *Developing Java Applications*
- “Migrating Java applications that use DB2 JDBC Type 2 or 3 driver” on page 146
- “Migrating Java applications that use IBM DB2 Driver for JDBC and SQLJ” on page 145

**Related reference:**

- “ALTER FUNCTION statement” in *SQL Reference, Volume 2*
- “ALTER PROCEDURE statement” in *SQL Reference, Volume 2*

---

## Migrating .NET CLR routines

After migrating a DB2 instance and databases to DB2 Version 9, you must migrate your .NET CLR routines that you created prior to Version 9 to ensure that they continue to function successfully and perform as expected.

**Prerequisites:**

- Review the migration essentials for routines to identify key changes that might apply to your .NET CLR routines.
- Ensure that you have access to a DB2 Version 9 server, including instances and databases. The DB2 server can be part of a testing environment.
- Ensure that the operating system is at a version level that is supported by DB2 database products.
- Ensure that a supported version of the .NET Framework software is installed on the DB2 server.
- Perform the pre-migration tasks for routines.

**Procedure:**

To migrate your .NET CLR routines to DB2 Version 9:

1. If you identified changes in DB2 Version 9 that impact your routines, edit your routine code and modify:
  - SQL statement syntax
  - SQL statements using SQL Administrative views and routines, and catalog views
2. Connect to the DB2 Version 9 database in which you defined the .NET CLR.

3. Rebuild your .NET CLR routine source code using the compile and link options specified in bldrtn.bat, the DB2 sample script for building .NET CLR routines.
4. Deploy the routine assembly to the DB2 server in the same location specified by the EXTERNAL clause in the routine definition.
5. Test your .NET CLR routines. The routines should function successfully, with no differences in between DB2 UDB Version 8 and DB2 Version 9 behavior.

After migrating your .NET CLR routines, perform the recommended post-migration tasks for routines.

**Related concepts:**

- Chapter 19, “Migration essentials for routines,” on page 137
- “Supported .NET development software” in *Developing ADO.NET and OLE DB Applications*
- Chapter 20, “Pre-migration tasks for database applications and routines,” on page 139
- “DB2 .NET Data Provider database system requirements” in *Developing ADO.NET and OLE DB Applications*
- “Designing .NET CLR routines” in *SQL Guide*
- “ODBC .NET Data Provider” in *Developing ADO.NET and OLE DB Applications*
- “DB2 .NET Data Provider” in *Developing ADO.NET and OLE DB Applications*

**Related tasks:**

- “Building .NET common language runtime (CLR) routine code” in *SQL Guide*

**Related reference:**

- “CLR .NET routine compile and link options” in *Developing ADO.NET and OLE DB Applications*

---

## Migrating SQL procedures

If you created your SQL procedures in DB2 UDB Version 8.2, your SQL procedures are migrated when you migrate your databases.

If you migrated from a DB2 UDB Version 8 instance to a DB2 Version 9 instance with the same bit width, your routines will run successfully in DB2 Version 9. However, if you created your SQL procedures in DB2 UDB Version 8.1 and migrated from a DB2 UDB Version 8 32-bit instance to a DB2 Version 9 64-bit instance, you must drop and re-create those SQL procedures.

**Prerequisites:**

- Review the migration essentials for routines to identify key changes that might apply to your SQL procedures.
- Ensure that you have access to your migrated database on DB2 Version 9.
- Ensure that you have the necessary authorizations and privileges to use the CREATE PROCEDURE and DROP PROCEDURE statements. You can find the complete list of authorizations and privileges required in the SQL Reference.
- Perform the pre-migration tasks for routines.

**Restriction:**



This procedure applies only to SQL procedures that were created in DB2 UDB Version 8.1 prior to FixPak 7 or Version 8.2.

### Procedure:

To migrate your SQL procedures to DB2 Version 9:

1. Connect to the migrated database.
2. Run the following query to identify the SQL procedures that you need to re-create:

```
SELECT procschema, specificname
FROM syscat.procedures
WHERE language = 'SQL'and fenced = 'N'AND
 substr(IMPLEMENTATION, 10,6) = 'pgsjmp'
```

Take note of the schema and specific name values returned by this query, because you will need this information to perform subsequent steps.

3. Run the **db2look** tool to generate a DDL script for all your database objects:

```
db2look -d sample -e -o db2look.sql
```

where *sample* is the database name, *-e* generates DDL statements for database objects, and *-o db2look.sql* indicates the output file that will contain the DDL statements.

Edit the *db2look.sql* file to keep only the DDL statements necessary to create the SQL procedures that you identified in step 2.

4. For each SQL stored procedures that you identified in step 2, use the DROP PROCEDURE statement indicating the schema name and specific name to uniquely identify each procedure:

```
DROP SPECIFIC PROCEDURE <schema-name>.<specific-name>
```

Alternatively, if you have a DDL script that drops and re-creates your SQL procedures, edit it to drop and re-create only the SQL procedures identified in step 2, and run it. Then proceed to step 6.

5. Re-create the SQL procedures identified in step 2 using the CREATE PROCEDURE statement. Alternatively, you can run your own DDL script or the *db2look.sql* file that you created in step 3.
6. Test your SQL procedures to ensure that they run as expected under DB2 Version 9. You can use the Developer Workbench or the Command Line Processor (CLP) interface to test them. The following example illustrates how to invoke an SQL procedure using the CLP :

```
CONNECT TO sample
```

```
Database Connection Information
```

```
Database server = DB2/AIX64 9.1.0
SQL authorization ID = TESTDB2
Local database alias = SAMPLE
```

```
CALL <schema-name>.<procedure-name> ([<parameter-list>])
```

7. If you created projects in the Development Center to develop your SQL procedures, migrate any existing projects to the Developer Workbench using the migration wizard.

After migrating your SQL procedures, perform the recommended post-migration tasks for routines to ensure that your migration was successful.

**Related concepts:**

- Chapter 20, “Pre-migration tasks for database applications and routines,” on page 139
- Chapter 18, “Migration essentials for database applications,” on page 131
- Chapter 19, “Migration essentials for routines,” on page 137
- “SQL procedure design considerations” in *SQL Guide*
- “SQL routine development in the DB2 Developer Workbench” in *SQL Guide*
- “SQL routines” in *SQL Guide*

**Related tasks:**

- “Creating SQL procedures” in *SQL Guide*
- “Migrating routines” on page 153

**Related reference:**

- “db2 - Command line processor invocation command” in *Command Reference*
- “db2look - DB2 statistics and DDL extraction tool command” in *Command Reference*
- “CREATE PROCEDURE (SQL) statement” in *SQL Reference, Volume 2*
- “DROP statement” in *SQL Reference, Volume 2*

---

## Migrating 32-bit external routines to run on 64-bit instances

If you migrate from a DB2 Version 8 32-bit instance to a DB2 Version 9 64-bit instance that includes 32-bit shared libraries, you need to ensure that your 32-bit external routines are linked to the appropriate shared library path to run successfully.

You do not need to modify your 32-bit external routines if you linked them to the \$INSTHOME/sql1lib/lib32 library path on Linux and UNIX or the DB2PATH\lib\Win32 library path on Windows, where INSTHOME is the instance home directory and DB2PATH is the location of the DB2 copy.

**Prerequisites:**

- Review the migration essentials for routines to identify key changes that might apply to your 32-bit external routines.
- Ensure that you have access to a DB2 UDB Version 8 32-bit instance that you migrated to a DB2 Version 9 64-bit instance that includes 32-bit shared libraries.
- Ensure that the operating system is at a version level that is supported by DB2 database products.
- Ensure that the development software is at a version level that is supported by DB2 database products.
- Perform the pre-migration tasks for routines.
- Ensure that you have SYSADM or DBADM authority to use the following SQL statements:
  - ALTER FUNCTION
  - ALTER PROCEDURE

Other authorizations allowed are listed in the Command Reference.

**Restrictions:**

- This procedure applies only to 32-bit external routines programmed on C and COBOL.
- This procedure indicates only the changes that are required to run 32-bit external routines on a 64-bit instance that include 32-bit shared libraries.

**Procedure:**

To migrate 32-bit external routines to run on a DB2 Version 9 64-bit instance:

1. Ensure that the library path environment variables include the correct DB2 shared library path for 32-bit libraries as shown in Table 15 on page 135, so that the correct library can be loaded at runtime.
2. Test your routines in a DB2 Version 9 testing environment. If testing is successful, you do not need to perform any additional steps. However, consider performing steps 4 to 6 if they apply to your routine for better support by using the correct library path and development software.
3. If changes in DB2 Version 9 impact your applications, edit your application code to modify:
  - SQL statements syntax
  - SQL statements using catalog views and SQL Administrative views and routines
4. Specify the correct library path by linking or rebuilding your 32-bit external routines using the DB2 shared library paths for 32-bit libraries shown in Table 15 on page 135. If you migrated from a DB2 UDB Version 8 32-bit instance to a DB2 Version 9 64-bit instance, you must rebuild 32-bit external routines that use LOB locators as 64-bit routine libraries.
5. Optional: If you no longer have the source code to rebuild your routine library or use environmental variables, you can use the **db2chglbpath** command to change the DB2 shared library path to \$INSTHOME/sql1lib/lib32 on your routine binary file as long as it has an embedded runtime path. The embedded runtime path can be changed to a new path with the same length or less.
6. If you migrated from a DB2 Version 8 32-bit instance to a DB2 Version 9 64-bit instance, change unfenced 32-bit external stored procedures or user-defined functions (UDFs) to fenced using the ALTER FUNCTION or ALTER PROCEDURE statements. For example, the following statement changes an external procedure to fenced:

```
ALTER SPECIFIC PROCEDURE <schema-name>.<specific-name> FENCED
```

where *<schema-name>* is the schema that owns the external procedure and *<specific-name>* is the specific name that uniquely identify a procedure indicated or assigned by default at creation.

7. Test your 32-bit external routines. The following example illustrates how to run an external procedure using the command line processor interface:

```
CONNECT TO sample
```

```
Database Connection Information
```

```
Database server = DB2/AIX64 9.1.0
SQL authorization ID = TESTDB2
Local database alias = SAMPLE
```

```
CALL schema-name.procedure-name ([parameter-list])
```

After migrating your 32-bit external routines, perform the recommended post-migration tasks for routines.

**Related concepts:**

- Chapter 19, “Migration essentials for routines,” on page 137
- Chapter 20, “Pre-migration tasks for database applications and routines,” on page 139
- Chapter 18, “Migration essentials for database applications,” on page 131
- “Support changes for 32-bit and 64-bit DB2 servers” on page 27
- “COBOL procedures” in *SQL Guide*
- “Support for external routine development in C” in *SQL Guide*
- “Support for external routine development in C++” in *SQL Guide*

**Related tasks:**

- “Building C and C++ routine code using sample bldrtn scripts” in *SQL Guide*

**Related reference:**

- “Supported programming languages and compilers for database application development” in *Getting Started with Database Application Development*
- “Support for external procedure development in COBOL” in *SQL Guide*
- “db2chglbpath - Modify the embedded runtime library search path command” in *Command Reference*

---

## Chapter 23. Post-migration tasks for database applications and routines

After migrating your database applications and routines, you should perform several post-migration tasks to ensure that your database applications and routines perform as expected and at their optimum levels.

Perform the following post-migration tasks that apply to your database applications and routines:

1. Tune your database applications. Review important guidelines related to:

- Character conversion
- Optimization class
- Specifying the isolation level
- Locks and concurrency
- Parallel processing for applications
- Query tuning

Refer to the Administration Guide: Performance for complete details on how to tune applications.

2. 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 guideline on optimization classes, locks, concurrency, and query tuning.

3. Remove dependencies on features that are deprecated in DB2 Version 9 in your database applications and routines before those features become unsupported.
4. Start using new features in DB2 Version 9 for application development where appropriate to improve performance. Check the Sample files to understand how the new features work.

### Related concepts:

- “Performance of routines” in *SQL Guide*
- “Performance of embedded SQL applications” in *Developing Embedded SQL Applications*
- “Benchmark testing” in *Performance Guide*
- “Quick-start tips for performance tuning” in *Performance Guide*
- “Sample files” in *Samples Topics*

### Related tasks:

- “Migrating database applications” on page 141
- “Migrating routines” on page 153



---

## Appendixes

This part of the book contains the following appendixes:

Appendix A, "DB2 Information Center Version 9," on page 169

Appendix B, "Important references," on page 171

Appendix C, "DB2 Database technical information," on page 173

Appendix D, "Notices," on page 183





---

## Appendix A. DB2 Information Center Version 9

You cannot migrate a DB2 Information Center Version 8 copy to DB2 Version 9. However, a DB2 Information Center Version 8 copy can coexist with a DB2 Information Center Version 9 copy, as long as you specify a different location when installing the Version 9 copy.

Consider the following information when installing a DB2 Information Center Version 9 copy:

- If you have a DB2 Information Center Version 8 copy, it remains untouched when installing a DB2 Information Center Version 9 copy. You must uninstall the Version 8 copy if you no longer need it or if you wish to install a Version 9 copy in the same location as the Version 8 copy.
- If you have a DB2 Information Center Version 8 copy installed in the default directory, you must specify a different location and a different port number when you install a DB2 Information Center Version 9 copy. The DB2 setup wizard prompts you for an alternative location and indicates the next available port number.
- You can install only one DB2 Information Center Version 9 copy on any given system.

The following table shows the URL for specific versions of the DB2 Information Center:

*Table 18. URLs for online DB2 Information Centers*

| DB2 Information Center version | URL                                                                                                                                     |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Version 8.2                    | <a href="http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp">http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp</a> |
| Version 9                      | <a href="http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp">http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp</a> |
| Most recently released version | <a href="http://publib.boulder.ibm.com/infocenter/db2help/index.jsp">http://publib.boulder.ibm.com/infocenter/db2help/index.jsp</a>     |



---

## Appendix B. Important references

The following list of references can help you with migration of your DB2 environment:

### **DB2 operating system requirements Web page**

The operating system and hardware requirements for DB2 Version 9 are available at <http://www.ibm.com/software/data/db2/udb/sysreqs.html>.

### **DB2 migration portal**

The DB2 migration portal at <http://www.ibm.com/support/docview.wss?rs=73&uid=swg21200005> provides you with a single place for accessing up-to-date information about the migration process and additional resources as they become available.

### **DB2 Information Center**

You can find the information in this book in the online DB2 Information Center at <http://publib.boulder.ibm.com/infocenter/db2help/index.jsp> under the Migrating topic. The Migrating to DB2 Version 9 title under Database systems is the high level topic. The online DB2 Information Center also contains information on migration-related topics such as DB2 database product installation.

### **DB2 Version 9 manuals in PDF format**

DB2 Version 9 manuals in PDF format are available for complimentary download at <http://www.ibm.com/software/data/db2/udb/support/manualsv9.html> hosted by the DB2 Technical Support Web site.

### **DB2 database product education**

Review the list of complimentary DB2 database product self-study courses that can help you build skills at your own pace at <http://www.ibm.com/software/data/education/selfstudy.html>. The Information Management Training Web site at <http://www.ibm.com/software/data/education/> offers a wide variety of training options and the list of skills resources and communities to help you find the educational resources that are right for you.

### **developerWorks Information Management Web site**

The developerWorks Information Management Web site at <http://www.ibm.com/developerworks/db2> 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 features 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

### **DB2 forums**

The DB2 forums, hosted by developerWorks at [http://www.ibm.com/developerworks/forums/db2\\_forums.jsp](http://www.ibm.com/developerworks/forums/db2_forums.jsp), are places to exchange ideas and

share solutions with your peers in the IBM DB2 product community. In addition, DB2 forums include forums that are mirrors to DB2 newsgroups, such as the `ibm.software.db2.udb` and `ibm.software.db2.udb.beta` newsgroups.

---

## Appendix C. DB2 Database technical information

---

### Overview of the DB2 technical information

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

- DB2 Information Center
  - Topics
  - Help for DB2 tools
  - Sample programs
  - Tutorials
- DB2 books
  - PDF files (downloadable)
  - PDF files (from the DB2 PDF CD)
  - printed books
- Command line help
  - Command help
  - Message help
- Sample programs

IBM periodically makes documentation updates available. If you access the online version on the DB2 Information Center at [ibm.com](http://ibm.com)<sup>®</sup>, you do not need to install documentation updates because this version is kept up-to-date by IBM. If you have installed the DB2 Information Center, it is recommended that you install the documentation updates. Documentation updates allow you to update the information that you installed from the *DB2 Information Center CD* or downloaded from Passport Advantage as new information becomes available.

**Note:** The DB2 Information Center topics are updated more frequently than either the PDF or the hard-copy 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 Redbooks<sup>™</sup> 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 we can improve the DB2 documentation, send an e-mail 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 e-mail address to contact DB2 Customer Support. If you have a DB2 technical issue that the documentation does not resolve, contact your local IBM service center for assistance.

**Related concepts:**

- “Features of the DB2 Information Center” in *Online DB2 Information Center*
- “Sample files” in *Samples Topics*

**Related tasks:**

- “Invoking command help from the command line processor” in *Command Reference*
- “Invoking message help from the command line processor” in *Command Reference*
- “Updating the DB2 Information Center installed on your computer or intranet server” on page 179

**Related reference:**

- “DB2 technical library in PDF format” on page 174

---

## DB2 technical library in PDF format

The following tables describe the DB2 library available from the IBM Publications Center at [www.ibm.com/shop/publications/order](http://www.ibm.com/shop/publications/order).

Although the tables identify books available in print, the books might not be available in your country or region.

The information in these books is fundamental to all DB2 users; you will find this information useful whether you are a programmer, a database administrator, or someone who works with DB2 Connect or other DB2 products.

*Table 19. DB2 technical information*

| <b>Name</b>                                                    | <b>Form Number</b> | <b>Available in print</b> |
|----------------------------------------------------------------|--------------------|---------------------------|
| <i>Administration Guide: Implementation</i>                    | SC10-4221          | Yes                       |
| <i>Administration Guide: Planning</i>                          | SC10-4223          | Yes                       |
| <i>Administrative API Reference</i>                            | SC10-4231          | Yes                       |
| <i>Administrative SQL Routines and Views</i>                   | SC10-4293          | No                        |
| <i>Call Level Interface Guide and Reference, Volume 1</i>      | SC10-4224          | Yes                       |
| <i>Call Level Interface Guide and Reference, Volume 2</i>      | SC10-4225          | Yes                       |
| <i>Command Reference</i>                                       | SC10-4226          | No                        |
| <i>Data Movement Utilities Guide and Reference</i>             | SC10-4227          | Yes                       |
| <i>Data Recovery and High Availability Guide and Reference</i> | SC10-4228          | Yes                       |
| <i>Developing ADO.NET and OLE DB Applications</i>              | SC10-4230          | Yes                       |
| <i>Developing Embedded SQL Applications</i>                    | SC10-4232          | Yes                       |
| <i>Developing SQL and External Routines</i>                    | SC10-4373          | No                        |

Table 19. DB2 technical information (continued)

| Name                                                                                                                                               | Form Number | Available in print |
|----------------------------------------------------------------------------------------------------------------------------------------------------|-------------|--------------------|
| <i>Developing Java Applications</i>                                                                                                                | SC10-4233   | Yes                |
| <i>Developing Perl and PHP Applications</i>                                                                                                        | SC10-4234   | No                 |
| <i>Getting Started with Database Application Development</i>                                                                                       | SC10-4252   | Yes                |
| <i>Getting started with DB2 installation and administration on Linux and Windows</i>                                                               | GC10-4247   | Yes                |
| <i>Message Reference Volume 1</i>                                                                                                                  | SC10-4238   | No                 |
| <i>Message Reference Volume 2</i>                                                                                                                  | SC10-4239   | No                 |
| <i>Migration Guide</i>                                                                                                                             | GC10-4237   | Yes                |
| <i>Net Search Extender Administration and User's Guide</i><br><b>Note:</b> HTML for this document is not installed from the HTML documentation CD. | SH12-6842   | Yes                |
| <i>Performance Guide</i>                                                                                                                           | SC10-4222   | Yes                |
| <i>Query Patroller Administration and User's Guide</i>                                                                                             | GC10-4241   | Yes                |
| <i>Quick Beginnings for DB2 Clients</i>                                                                                                            | GC10-4242   | No                 |
| <i>Quick Beginnings for DB2 Servers</i>                                                                                                            | GC10-4246   | Yes                |
| <i>Spatial Extender and Geodetic Data Management Feature User's Guide and Reference</i>                                                            | SC18-9749   | Yes                |
| <i>SQL Guide</i>                                                                                                                                   | SC10-4248   | Yes                |
| <i>SQL Reference, Volume 1</i>                                                                                                                     | SC10-4249   | Yes                |
| <i>SQL Reference, Volume 2</i>                                                                                                                     | SC10-4250   | Yes                |
| <i>System Monitor Guide and Reference</i>                                                                                                          | SC10-4251   | Yes                |
| <i>Troubleshooting Guide</i>                                                                                                                       | GC10-4240   | No                 |
| <i>Visual Explain Tutorial</i>                                                                                                                     | SC10-4319   | No                 |
| <i>What's New</i>                                                                                                                                  | SC10-4253   | Yes                |
| <i>XML Extender Administration and Programming</i>                                                                                                 | SC18-9750   | Yes                |
| <i>XML Guide</i>                                                                                                                                   | SC10-4254   | Yes                |
| <i>XQuery Reference</i>                                                                                                                            | SC18-9796   | Yes                |

Table 20. DB2 Connect-specific technical information

| Name                                                     | Form Number | Available in print |
|----------------------------------------------------------|-------------|--------------------|
| <i>DB2 Connect User's Guide</i>                          | SC10-4229   | Yes                |
| <i>Quick Beginnings for DB2 Connect Personal Edition</i> | GC10-4244   | Yes                |

Table 20. DB2 Connect-specific technical information (continued)

| Name                                            | Form Number | Available in print |
|-------------------------------------------------|-------------|--------------------|
| <i>Quick Beginnings for DB2 Connect Servers</i> | GC10-4243   | Yes                |

Table 21. WebSphere Information Integration technical information

| Name                                                                                                    | Form Number | Available in print |
|---------------------------------------------------------------------------------------------------------|-------------|--------------------|
| <i>WebSphere® Information Integration: Administration Guide for Federated Systems</i>                   | SC19-1001   | Yes                |
| <i>WebSphere Information Integration: ASNCLP Program Reference for Replication and Event Publishing</i> | SC19-1000   | Yes                |
| <i>WebSphere Information Integration: Configuration Guide for Federated Data Sources</i>                | SC19-1034   | No                 |
| <i>WebSphere Information Integration: SQL Replication Guide and Reference</i>                           | SC19-1002   | Yes                |

**Note:** The DB2 Release Notes provide additional information specific to your product's release and fix pack level. For more information, see the related links.

**Related concepts:**

- "Overview of the DB2 technical information" on page 173
- "About the Release Notes" in *Release notes*

**Related tasks:**

- "Ordering printed DB2 books" on page 176

---

## Ordering printed DB2 books

If you require printed DB2 books, you can buy them online in many but not all countries or regions. You can always order printed DB2 books from your local IBM representative. Keep in mind that some softcopy books on the *DB2 PDF Documentation CD* are unavailable in print. For example, neither volume of the *DB2 Message Reference* is available as a printed book.

Printed versions of many of the DB2 books available on the DB2 PDF Documentation CD can be ordered for a fee from IBM. Depending on where you are placing your order from, you may be able to order books online, from the IBM Publications Center. If online ordering is not available in your country or region, you can always order printed DB2 books from your local IBM representative. Note that not all books on the DB2 PDF Documentation CD are available in print.

**Note:** The most up-to-date and complete DB2 documentation is maintained in the DB2 Information Center at <http://publib.boulder.ibm.com/infocenter/db2help/>.

**Procedure:**



To order printed DB2 books:

- To find out whether you can order printed DB2 books online in your country or region, check the IBM Publications Center at <http://www.ibm.com/shop/publications/order>. You must select a country, region, or language to access publication ordering information and then follow the ordering instructions for your location.
- To order printed DB2 books from your local IBM representative:
  - Locate the contact information for your local representative from one of the following Web sites:
    - The IBM directory of world wide contacts at [www.ibm.com/planetwide](http://www.ibm.com/planetwide)
    - The IBM Publications Web site at <http://www.ibm.com/shop/publications/order>. You will need to select your country, region, or language to the access appropriate publications home page for your location. From this page, follow the "About this site" link.
  - When you call, specify that you want to order a DB2 publication.
  - Provide your representative with the titles and form numbers of the books that you want to order.

**Related concepts:**

- "Overview of the DB2 technical information" on page 173

**Related reference:**

- "DB2 technical library in PDF format" on page 174

---

## Displaying SQL state help from the command line processor

DB2 returns an SQLSTATE value for conditions that could be the result of an SQL statement. SQLSTATE help explains the meanings of SQL states and SQL state class codes.

**Procedure:**

To invoke 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.

**Related tasks:**

- "Invoking command help from the command line processor" in *Command Reference*
- "Invoking message help from the command line processor" in *Command Reference*

---

## Accessing different versions of the DB2 Information Center

For DB2 Version 9 topics, the DB2 Information Center URL is <http://publib.boulder.ibm.com/infocenter/db2luw/v9/>.

For DB2 Version 8 topics, go to the Version 8 Information Center URL at:  
<http://publib.boulder.ibm.com/infocenter/db2luw/v8/>.

**Related tasks:**

- “Updating the DB2 Information Center installed on your computer or intranet server” on page 179

---

## Displaying topics in your preferred language in the DB2 Information Center

The DB2 Information Center attempts to display topics in the language specified in your browser preferences. If a topic has not been translated into your preferred language, the DB2 Information Center displays the topic in English.

**Procedure:**

To display topics in your preferred language in the Internet Explorer browser:

1. In Internet Explorer, click the **Tools** → **Internet Options** → **Languages...** button. The Language Preferences window opens.
2. Ensure your preferred language is specified as the first entry in the list of languages.
  - To add a new language to the list, click the **Add...** button.

**Note:** Adding a language does not guarantee that the computer has the fonts required to display the topics in the preferred language.

- To move a language to the top of the list, select the language and click the **Move Up** button until the language is first in the list of languages.
3. Clear the browser cache and then refresh the page to display the DB2 Information Center in your preferred language.

To display topics in your preferred language in a Firefox or Mozilla browser:

1. Select the **Tools** → **Options** → **Languages** button. The Languages panel is displayed in the Preferences window.
2. Ensure your preferred language is specified as the first entry in the list of languages.
  - To add a new language to the list, click the **Add...** button to select a language from the Add Languages window.
  - To move a language to the top of the list, select the language and click the **Move Up** button until the language is first in the list of languages.
3. Clear the browser cache and then refresh the page to display the DB2 Information Center in your preferred language.

On some browser and operating system combinations, you might have to also change the regional settings of your operating system to the locale and language of your choice.

**Related concepts:**

- “Overview of the DB2 technical information” on page 173

---

## Updating the DB2 Information Center installed on your computer or intranet server

If you have a locally-installed DB2 Information Center, updated topics can be available for download. The 'Last updated' value found at the bottom of most topics indicates the current level for that topic.

To determine if there is an update available for the entire DB2 Information Center, look for the 'Last updated' value on the Information Center home page. Compare the value in your locally installed home page to the latest value which is available on the IBM hosted Information Center home page. If they are the same, you have the latest documentation level and no update is required. If they are not the same, you should update your locally-installed Information Center.

Updating your locally-installed DB2 Information Center requires that you:

1. Stop the DB2 Information Center on your computer, and restart the Information Center in stand-alone mode. Running the Information Center in stand-alone mode prevents other users on your network from accessing the Information Center, and allows you to download and apply updates.
2. Use the Update feature to determine if update packages are available from IBM. If update packages are available, use the Update feature to download the packages. (The Update feature is only available in stand-alone mode.)
3. Stop the stand-alone Information Center, and restart the DB2 Information Center service on your computer.

### Procedure:

To update the DB2 Information Center installed on your computer or intranet server:

1. Stop the DB2 Information Center service.
  - On Windows, click **Start** → **Control Panel** → **Administrative Tools** → **Services**. Then right-click on **DB2 Information Center** service and select **Stop**.
  - On Linux, enter the following command:  
`/etc/init.d/db2icdv9 stop`
2. Start the Information Center in stand-alone mode.
  - On Windows:
    - a. Open a command window.
    - b. Navigate to the path where the Information Center is installed. By default, the DB2 Information Center is installed in the `C:\Program Files\IBM\DB2 Information Center\Version 9` directory.
    - c. Run the `help_start.bat` file using the fully qualified path for the DB2 Information Center:  
`<DB2 Information Center dir>\doc\bin\help_start.bat`
  - On Linux:
    - a. Navigate to the path where the Information Center is installed. By default, the DB2 Information Center is installed in the `/opt/ibm/db2ic/V9` directory.
    - b. Run the `help_start.sh` file using the fully qualified path for the DB2 Information Center:  
`<DB2 Information Center dir>/doc/bin/help_start`

The systems default Web browser launches to display the stand-alone Information Center.

3. Click the Update button (🔄). On the right hand panel of the Information Center, click **Find Updates**. A list of updates for existing documentation displays.
4. To initiate the download process, check the selections you want to download, then click **Install Updates**.
5. After the download and installation process has completed, click **Finish**.
6. Stop the stand-alone Information Center.
  - On Windows, run the `help_end.bat` file using the fully qualified path for the DB2 Information Center:  

```
<DB2 Information Center dir>\doc\bin\help_end.bat
```
  - On Linux, run the `help_end.sh` file using the fully qualified path for the DB2 Information Center:  

```
<DB2 Information Center dir>/doc/bin/help_end
```
7. Restart the DB2 Information Center service.
  - On Windows, click **Start** → **Control Panel** → **Administrative Tools** → **Services**. Then right-click on **DB2 Information Center** service and select **Start**.
  - On Linux, enter the following command:  

```
/etc/init.d/db2icdv9 start
```

The updated DB2 Information Center displays the new and updated topics.

#### Related concepts:

- “DB2 Information Center installation options” in *Quick Beginnings for DB2 Servers*

#### Related tasks:

- “Installing the DB2 Information Center using the DB2 Setup wizard (Linux)” in *Quick Beginnings for DB2 Servers*
- “Installing the DB2 Information Center using the DB2 Setup wizard (Windows)” in *Quick Beginnings for DB2 Servers*

---

## DB2 Visual Explain tutorial

The DB2 Visual Explain tutorial helps you learn about analyzing, optimizing, and tuning SQL statements for better performance. Lessons provide step-by-step instructions.

#### Before you begin:

You can view the XHTML version of the tutorial from the Information Center at <http://publib.boulder.ibm.com/infocenter/db2help/>.

Some lessons use sample data or code. See the tutorial for a description of any prerequisites for its specific tasks.

#### DB2 Visual Explain tutorial:

To view the tutorial, click on the title.

##### *Visual Explain Tutorial*

Analyze, optimize, and tune SQL statements for better performance using Visual Explain.

**Related concepts:**

- “Visual Explain overview” in *Administration Guide: Implementation*

---

## DB2 troubleshooting information

A wide variety of troubleshooting and problem determination information is available to assist you in using DB2 products.

**DB2 documentation**

Troubleshooting information can be found in the DB2 Troubleshooting Guide or the Support and Troubleshooting section of the DB2 Information Center. There you will find information on how to isolate and identify problems using DB2 diagnostic tools and utilities, solutions to some of the most common problems, and other advice on how to solve problems you might encounter with your DB2 products.

**DB2 Technical Support Web site**

Refer to the DB2 Technical Support Web site if you are experiencing problems and want help finding possible causes and solutions. The Technical Support site has links to the latest DB2 publications, TechNotes, Authorized Program Analysis Reports (APARs or bug fixes), fix packs, and other resources. You can search through this knowledge base to find possible solutions to your problems.

Access the DB2 Technical Support Web site at <http://www.ibm.com/software/data/db2/udb/support.html>

**Related concepts:**

- “Introduction to problem determination” in *Troubleshooting Guide*
- “Overview of the DB2 technical information” on page 173

---

## Terms and Conditions

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

**Personal use:** You may reproduce these Publications for your personal, non commercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these Publications, or any portion thereof, without the express consent of IBM.

**Commercial use:** You may reproduce, distribute and display these Publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these Publications, or reproduce, distribute or display these Publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the Publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the Publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

---

## Appendix D. Notices

IBM may not offer the products, services, or features discussed in this document in all countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give 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 (DBCS) information, contact the IBM Intellectual Property Department in your country/region or send inquiries, in writing, to:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, Japan

**The following paragraph does not apply to the United Kingdom or any other country/region where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product, and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information that has been exchanged, should contact:

IBM Canada Limited  
Office of the Lab Director  
8200 Warden Avenue  
Markham, Ontario  
L6G 1C7  
CANADA

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems, and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements, or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious, and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information may contain 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.

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

Company, product, or service names identified in the documents of the DB2 Version 9 documentation library may be trademarks or service marks of International Business Machines Corporation or other companies. Information on the trademarks of IBM Corporation in the United States, other countries, or both is located at <http://www.ibm.com/legal/copytrade.shtml>.

The following terms are trademarks or registered trademarks of other companies and have been used in at least one of the documents in the DB2 documentation library:

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Itanium<sup>®</sup>, Pentium<sup>®</sup>, and Xeon<sup>®</sup> are trademarks of Intel Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds 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 CLR routines
- migrating 159

## Numerics

- 32-bit and 64-bit support
  - migrating 27
- 32-bit applications
  - migrating 151
- 32-bit external routines
  - migrating 162
- 32-bit to 64-bit Linux and UNIX
  - migrating DB2 servers 65
- 32-bit to 64-bit Windows
  - migrating DB2 servers 64
- 64-bit instance
  - migrating
    - 32-bit applications 151
- 64-bit instances
  - migrating
    - 32-bit external routines 162

## A

- ACTIVATE DATABASE command
  - activating database and services 84
- adjusting log space
  - larger RID 83
- Administrative SQL routines and views
  - migrating 29
- ADO .NET
  - migrating
    - applications 148
- applications
  - migrating 3, 129, 131, 141
  - migration support 131
  - planning migration 11
  - post-migration tasks 165
    - tuning applications and routines 165
  - pre-migration tasks 139
    - migrating DB2 clients 139
    - testing 139
    - upgrading operating system and development software 139

## B

- backing up configuration
  - DB2 clients 111
  - DB2 servers
    - pre-migration tasks 37
- backing up databases
  - pre-migration tasks 36
- BACKUP DATABASE command
  - pre-migration tasks 36

## C

- C, C++, and COBOL
  - migrating
    - applications 143
    - routines 155
- changing raw devices to block devices
  - pre-migration tasks
    - DB2 servers 41
- CLI applications
  - migrating 143
- commands
  - ACTIVATE DATABASE 84
  - BACKUP DATABASE 36
  - dasmigr 50, 58
  - db2batch 98
  - db2ckmig 34
  - db2exmig 96
  - db2imigr 20, 48, 56
  - db2iupdt 65
  - db2rbind 95
  - db2support 37
  - db2uidl 93
  - MIGRATE DATABASE 20, 52, 60
  - REBIND 95
  - REORG INDEXES/TABLE 93
  - RESTORE DATABASE 67
- complex environments
  - migrating
    - DB2 servers 63
- configuration parameters
  - migrating 85
  - saving settings
    - pre-migration tasks 37
- contacting IBM 189
- CREATE TABLESPACE statement 97

## D

- dasmigr command
  - migrating DAS 50, 58
- Data Links
  - migrating 76
- Data Warehouse Manager
  - migrating 29
- database physical characteristics
  - migrating 85
- databases
  - migrating 52, 60
- DataLinks
  - migrating 29
- DB2 Administration Server (DAS)
  - migrating 50, 58
- DB2 Client
  - migrating
    - Windows 113
- DB2 clients
  - earlier version support 31
  - migrating 3, 105, 107
    - Linux and UNIX 119
    - Windows 113, 115
- DB2 clients (*continued*)
  - migration support 107
  - planning migration 9
  - post-migration tasks 123
    - recataloging nodes 123
    - reviewing configuration
      - parameters and registry variables 123
    - verifying migration 125
  - pre-migration tasks 111
    - backing up configuration 111
    - migrating DB2 servers 111
    - reviewing migration
      - essentials 111
- DB2 CLP and system command scripts
  - migrating 149
- DB2 environments
  - planning migration 5
- DB2 Information Center
  - updating 179
  - versions 177
  - viewing in different languages 178
- DB2 JDBC Type 2 or 3 driver
  - migrating 146
- DB2 JDBC type 3 driver
  - migrating 29
- DB2 Net Search Extender (NSE)
  - migrating 76
- DB2 products
  - migrating 29
- DB2 Runtime Client
  - migrating
    - Windows 115
- DB2 servers
  - migrating 3, 17, 19, 23, 46
    - 32-bit to 64-bit Linux and UNIX 65
    - 32-bit to 64-bit Windows 64
  - alternate fixpak installations 71
  - databases 52, 60
  - DB2 Administration Server (DAS) 50, 58
  - HADR 99
  - instances 48, 56
  - Linux and UNIX 55
  - multiple DB2 copies 71
  - new machine 67
  - partitioned database
    - environments 69
  - rebinding packages 95
  - verifying migration 98
  - Windows 47
  - migration support 19
  - planning migration 6
  - post-migration tasks 81
    - activating database and services 84
    - adjusting log space 83
    - checking configuration parameters, registry variables and physical characteristics 85

- DB2 servers (*continued*)
  - post-migration tasks (*continued*)
    - converting type-1 indexes to type-2 indexes 93
    - ensuring system temporary table spaces page sizes large enough 97
    - migrating explain tables 96
    - rebinding packages 95
    - revoking the EXECUTE privilege on PUBLIC 94
    - verifying migration 98
  - pre-migration tasks 33
    - backing up configuration 37
    - backing up databases 36
    - changing raw devices to block devices 41
    - increasing log space and table space 39
    - migrating in test environments 43
    - setting diagnostic error level 44
    - taking servers offline 46
    - verifying databases ready to migrate 34
  - reversing migration 101
- DB2 Spatial Extender
  - migrating 23
- DB2 Version 9
  - discontinued and deprecated functionality 29
  - migrating 3
  - NetBIOS and SNA protocols unsupported 123
- DB2\_USE\_DB2JCCT2\_JROUTINE
  - registry variables 157
- db2batch command
  - testing applications and scripts 98
- db2ckmig command
  - verifying databases ready to migrate 34
- db2exmig
  - migrating explain tables 96
- db2imigr command 20
  - migrating instances 48, 56
  - support 20
- db2iupdt command
  - updating to 64-bit instances 65
- db2rbind command
  - rebinding packages 95
- db2support command
  - saving configuration 37
- db2uiddl command
  - converting unique indexes 93
- deprecated functionality
  - DB2 Version 9 29
- diaglevel
  - setting diagnostic error level 44
- diagpath
  - database manager configuration parameter 44
- direct I/O
  - migrating
    - Linux 41
- discontinued functionality
  - DB2 Version 9 29

- disk space
  - migrating
    - DB2 servers 26
- documentation 173, 174
  - terms and conditions of use 181

**E**

- earlier version support
  - DB2 clients 31
- embedded SQL applications
  - migrating 143
- ensuring system temporary table spaces
  - page sizes large enough
    - larger RID 97
- explain tables
  - migrating 96
- extended storage
  - migrating 29

**F**

- FORTRAN applications
  - migrating 143

**H**

- HADR
  - migrating 23
  - migrating DB2 servers 99
- help
  - displaying 178
  - for SQL statements 177

**I**

- IBM DB2 Driver for JDBC and SQLJ
  - migrating 145
- implementing Version 9 features
  - post-migration tasks
    - applications and routines 165
- increasing log space
  - migrating DB2 servers 39
- Information Center
  - updating 179
  - versions 177
  - viewing in different languages 178
- instances
  - 32-bit and 64-bit support 27
  - migrating 48, 56

**J**

- Java
  - migrating
    - applications 145, 146
    - routines 157
- JDBC Type 2 or 3 driver
  - migrating 146
- JDBC type 3 driver
  - migrating 29
- jdk\_path
  - database manager configuration parameter 157

**L**

- larger RID
  - adjusting log space 83
  - ensuring system temporary table spaces page sizes large enough 97
- Linux
  - migrating
    - DB2 clients 119
    - DB2 servers 55
    - raw devices 41
    - Version 7 DB2 clients 121
- Linux and UNIX
  - migrating
    - Version 7 DB2 servers 73
- log file space
  - migrating DB2 servers 26

**M**

- Microsoft Cluster servers
  - migrating 74
- Microsoft SQL server
  - migrating 79
- MIGRATE DATABASE command 20, 52, 60
- migrating
  - .NET CLR routines 159
  - 32-bit and 64-bit support 20
    - instances 27
  - 32-bit applications 151
  - 32-bit external routines 162
  - applications
    - ADO .NET 148
    - C, C++, COBOL, Fortran, or REXX 143
    - embedded SQL and CLI 143
    - Java using DB2 JDBC Type 2 or 3 driver 146
    - Java using IBM DB2 Driver for JDBC and SQLJ 145
  - applications and routines 3, 129, 131
    - planning 11
    - pre-migration tasks 139
  - applications and scripts 141
  - commands 20
  - Data Links 76
  - databases 52, 60
  - DB2 Administration Server (DAS) 50, 58
  - DB2 Client
    - Windows 113
  - DB2 clients 3, 105
    - Linux and UNIX 119
    - planning 9
    - post-migration tasks 123
    - pre-migration tasks 111
  - DB2 Runtime Client
    - Windows 115
  - DB2 server performance 23
  - DB2 servers 3, 17, 19, 23
    - 32-bit to 64-bit Linux and UNIX 65
    - 32-bit to 64-bit Windows 64
    - adjusting log space 83
    - alternate fixpak installations 71
    - complex environments 63

- migrating (*continued*)
  - DB2 servers (*continued*)
    - configuration parameters, registry variables and physical characteristics 85
    - discontinued features 20
    - HADR 99
    - Linux and UNIX 55
    - log space and table space requirements 26
    - multiple DB2 copies 71
    - new machine 67
    - partitioned database environments 69
    - planning 6
    - post-migration tasks 81
    - pre-migration tasks 33
    - restrictions 20
    - reversing migration 101
    - scenarios 63
    - taking servers offline 46
    - Windows 47
  - DB2 Spatial Extender 23
  - DB2 Version 9 3
  - enabling
    - autonomic computing features 23
  - explain tables 96
  - HADR 20, 23
  - index extensions 20
  - instances 48, 56
  - Java
    - routines 157
  - Linux
    - raw devices 41
  - Microsoft SQL server 79
  - MSCS 74
  - NSE 76
  - Oracle 79
  - planning 5
  - primary database 99
  - Query Patroller 20
  - replication environments 20
  - routines 137, 153
    - C, C++, and COBOL 155
    - revoking the EXECUTE privilege on PUBLIC 94
  - scripts 131, 149
  - sql procedures 160
  - SQL replication environments 23
  - stored procedures 160
  - Sybase 79
  - test environment 43
  - Version 7 DB2 clients
    - Linux and UNIX 121
    - Windows 116
  - Version 7 DB2 servers
    - Linux and UNIX 73
    - Windows 73
  - XML Extender 78
  - XML type 78
- migrating to DB2
  - resources 79
- migration support
  - applications and routines 131
  - DB2 clients 107
  - DB2 servers 19
  - routines 137

- MSCS
  - migrating 74
- multiple DB2 copies
  - migrating DB2 servers 71

## N

- NetBIOS nodes
  - migrating 29, 123
- new machine
  - migrating DB2 servers 67
- notices 183

## O

- O\_DIRECT
  - migrating
    - Linux 41
- Oracle
  - migrating 79
- ordering DB2 books 176

## P

- partitioned database environments
  - migrating 69
- planning
  - migrating 5
- planning migration
  - applications and routines 11
  - DB2 clients 9
  - DB2 environments 5
  - DB2 servers 6
- post-migration tasks
  - applications and routines 165
    - implementing Version 9 features 165
    - removing Version 9 deprecated features 165
    - tuning applications and routines 165
  - DB2 clients 123
  - recataloging nodes 123
  - reviewing configuration parameters and registry variables 123
  - verifying migration 125
  - DB2 servers 81
    - activating database and services 84
    - adjusting log space 83
    - checking configuration parameters, registry variables and physical characteristics 85
    - converting type-1 indexes to type-2 indexes 93
    - ensuring system temporary table spaces page sizes large enough 97
    - migrating explain tables 96
    - rebinding packages 95
    - revoking the EXECUTE privilege on PUBLIC 94
    - verifying migration 98
- pre-migration tasks
  - applications and routines 139

- pre-migration tasks (*continued*)
  - migrating DB2 clients 139
  - reviewing migration
    - essentials 139
    - testing 139
  - upgrading operating system and development software 139
  - DB2 clients 111
    - backing up configuration 111
    - migrating DB2 servers 111
    - reviewing migration
      - essentials 111
  - DB2 servers 33
    - backing up configuration 37
    - backing up databases 36
    - changing raw devices to block devices 41
    - increasing log space 39
    - migrating in test environments 43
    - setting diagnostic error level 44
    - taking servers offline 46
    - verifying databases ready to migrate 34
  - previous version
    - DB2 clients 31
  - primary database
    - migrating 99
  - printed books
    - ordering 176
  - problem determination
    - online information 181
    - tutorials 181
- R
  - raw I/O
    - migrating
      - Linux 41
  - raw logs
    - migrating 29
  - REBIND command
    - rebinding packages 95
  - rebinding packages
    - migrating DB2 servers 95
  - recataloging nodes
    - NetBIOS and SNA protocol
      - post-migration tasks 123
  - registry variables
    - migrating 85
    - saving settings
      - pre-migration tasks 37
  - removing Version 9 deprecated features
    - post-migration tasks
      - applications and routines 165
  - REORG INDEXES command
    - reorganizing indexes for a table 93
  - REORG TABLE command
    - reorganizing table 93
  - RESTORE DATABASE command 67
  - reversing migration
    - DB2 servers 101
  - reviewing configuration parameters and registry variables
    - post-migration tasks
      - DB2 clients 123

- reviewing migration essentials
  - post-migration tasks
    - applications and routines 139
- REVOKE statement
  - revoking the EXECUTE privilege on PUBLIC 94
- revoking the EXECUTE privilege on PUBLIC
  - post-migration tasks 94
- REXX applications
  - migrating 143
- routines
  - migrating 3, 129, 131, 137, 153
    - C, C++, and COBOL 155
    - revoking the EXECUTE privilege on PUBLIC 94
  - migration support 137
  - planning migration 11
  - post-migration tasks 165
    - implementing Version 9
      - features 165
    - removing Version 9 deprecated
      - features 165
    - tuning applications and routines 165
  - pre-migration tasks 139
    - migrating DB2 clients 139
    - testing 139
    - upgrading operating system and development software 139

## S

- scenarios
  - migrating
    - DB2 servers 63
- scripts
  - migrating 131, 141, 149
- setting diagnostic error level
  - pre-migration tasks
    - DB2 servers 44
- SNA nodes
  - migrating 29, 123
- SQL Administrative views and routines
  - migrating 149
- SQL procedures
  - migrating 160
- SQL replication environments
  - migrating 23
- SQL statements
  - displaying help 177
  - migrating 149
- statements
  - CREATE TABLESPACE 97
  - REVOKE 94
- stored procedures
  - migrating 137, 153, 160
- Sybase
  - migrating 79

## T

- table spaces requirements
  - migrating
    - DB2 servers 26

- taking servers offline
  - pre-migration tasks 46
- terms and conditions
  - use of publications 181
- test environment
  - migrating 43
- troubleshooting
  - online information 181
  - tutorials 181
- tuning applications and routines
  - post-migration tasks
    - applications and routines 165
- tutorials
  - troubleshooting and problem determination 181
  - Visual Explain 180

## U

- UNIX
  - migrating
    - DB2 clients 119
    - DB2 servers 55
    - Version 7 DB2 clients 121
- updates
  - DB2 Information Center 179
  - Information Center 179
- upgrading development software
  - pre-migration tasks
    - applications and routines 139
- upgrading operating system
  - pre-migration tasks
    - applications and routines 139
- user-defined functions
  - migrating 137, 153

## V

- verifying databases ready to migrate
  - pre-migration tasks
    - DB2 servers 34
- verifying migration
  - post-migration tasks
    - DB2 clients 125
    - DB2 servers 98
- Version 7 DB2 clients
  - migrating
    - Linux and UNIX 121
    - Windows 116
- Version 7 DB2 servers
  - migrating
    - Linux and UNIX 73
    - windows 73
- Visual Explain
  - tutorial 180

## W

- Web sites
  - DB2 Migrate Now! 79
  - developerWorks Information Management 79
  - IBM Virtual Innovation Center 79
  - migration portal 17

- Windows
  - migrating
    - DB2 Client 113
    - DB2 Runtime Client 115
    - DB2 server 47
    - Version 7 DB2 clients 116
    - version 7 DB2 servers 73

## X

- XML Extender
  - migrating 78
- XML type
  - migrating 78

---

## Contacting IBM

To contact IBM in your country or region, check the IBM Directory of Worldwide Contacts at <http://www.ibm.com/planetwide>

To learn more about DB2 products, go to <http://www.ibm.com/software/data/db2/>.









Printed in U.S.A.

GC10-4237-00



DB2 9 BETA

Spine information:

IBM DB2  
DB2 9.5  
DB2 9.5

DB2 Version 9

Migration Guide

