IBM DB2 Change Management Expert for Multiplatforms

User's Guide

Version 1 Release 1

SC18-9958-01

IBM DB2 Change Management Expert for Multiplatforms

User's Guide

Version 1 Release 1

SC18-9958-01

Note:
Before using this information and the product it supports, read the information in “Notices” on page 125.

Second Edition (January 2007)

This edition applies to Version 1, Release 1 of IBM DB2 Change Management Expert for Multiplatforms (product number 5724-L29) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC18-9958-00.

Contents

Figures

Tables

About this book

This book provides information about using IBM® DB2 Change Management Expert for Multiplatforms to manage database change. The book offers conceptual, usage, and reference information to help you implement your change management system.

Examples and sample scenarios are provided to help you understand how this tool can assist you in your change management efforts.

This book is designed to help application programmers, application database administrators (DBAs), and production DBAs perform these tasks:

- Plan for the installation of DB2 Change Management Expert
- Install and customize DB2 Change Management Expert
- Use DB2 Change Management Expert to change and promote changes to DB2® data
- Diagnose and recover from DB2 Change Management Expert problems
- Design and write applications for DB2 Change Management Expert
- Use DB2 Change Management Expert with other IBM DB2 products

Always check the IBM DB2 and IMS™ Tools Library Web page for the most current version of this publication:

www.ibm.com/software/data/db2imstools/library.html

Who should read this book

This book is intended for application programmers, application database administrators (DBAs), and production DBAs who use DB2 Change Management Expert.

This book assumes a working knowledge of:

- Basic DB2 concepts and facilities
- Linux, UNIX, and Windows operating systems
- The Java™ stored procedures environment and concepts

Conventions and terminology used in this book

This book uses the following highlighting conventions:

- `Monospace type` indicates examples of text that you enter exactly as shown.
- *Italic type* indicates variables that you should replace with a value. It is also used to indicate book titles and to emphasize significant words.

The following labels identify significant elements within this book:

- **Definition:** is used to identify and define terminology unique to this product.
- **Example:** is used to identify example code or scenarios.
- **In this chapter:** is used to identify the significant subsections within each chapter.
- **Recommendation:** is used to provide guidance when more than one option is available.

- **Related reading:** is used to refer you to other publications that contain relevant information.
- **Requirement:** identifies a condition that must be met to ensure that the product is functional.
- **Restriction:** is used to identify a restriction or limitation with this product or an associated procedure.

Service updates and support information

To find service updates and support information, including software PTFs, Frequently Asked Questions (FAQs), technical notes, troubleshooting information, and downloads, refer to the following Web page:

www.ibm.com/software/data/db2imstools/support.html

Receiving information updates automatically

By registering with the IBM My Support service, you can automatically receive a weekly e-mail that notifies you when new DCF documents are released, when existing product documentation is updated, and when new product documentation is available. You can customize the service so that you receive information about only those IBM products that you specify.

To register with the My Support service:

1. Go to http://www.ibm.com/support/mysupport
2. Enter your IBM ID and password, or create one by clicking **register now**.
3. When the My Support page is displayed, click **add products** to select those products that you want to receive information updates about. The DB2 and IMS Tools category is located under **Software** -> **Data and Information Management** -> **Database Tools & Utilities**.
4. Click **Subscribe to email** to specify the types of updates that you would like to receive.
5. Click **Update** to save your profile.

Where to find information

The IBM DB2 and IMS Tools Library Web page provides current product documentation that you can view, print, and download. To locate publications with the most up-to-date information, refer to the following Web page:

www.ibm.com/software/data/db2imstools/library.html

IBM Redbooks™ that cover IBM DB2 and IMS Tools are available from the following Web page:

www.ibm.com/software/data/db2imstools/support.html

Accessibility features

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use a software product successfully. The major accessibility features in DB2 Change Management Expert enable users to:

- Use assistive technologies such as screen readers and screen magnifier software. Consult the assistive technology documentation for specific information when using it to access Multiplatform interfaces.
- Customize display attributes such as color, contrast, and font size.
- Operate specific or equivalent features by using only the keyboard.

Keyboard shortcuts in DB2 Change Management Expert

You can use many keyboard shortcuts when you work with DB2 Change Management Expert.

The following table shows the shortcut keys that are supported in the Deployment Script Editor.

Table 1. Keyboard shortcuts in the Deployment Script Editor

Keyboard shortcut	**Action**
Ctrl+S	Saves the deployment script
Tab	Moves focus to next item
Enter	Runs the action for the item in focus (for example, when the Edit target model hyperlink is in focus, the hyperlink is executed)
Alt+Shift+Q,X	Moves focus to the Problems view
Shift+F10	Displays the context menu

The following table shows the shortcut keys that are supported in the Model Editor.

Table 2. Keyboard shortcuts in the Model Editor

Keyboard shortcut	**Action**
Tab	Moves focus between objects
Space	Selects the currently focused object
Ctrl+Space	Adds the currently focused object to the selection (or removes the object if it is already in the selection)
Shift+Space	Adds all the items from the currently selected object to the focused object inclusive
Enter	Changes the current edit object
Alt+Enter	Moves focus to the Properties view
Alt+Shift+Q,O	Moves focus to the Outline view
Alt+Shift+Q,X	Moves focus to the Problems view
Alt+Rt_Arrow	Moves to last edit object
Alt+Lt_Arrow	Moves to previous edit object
Shift+F10	Displays context-sensitive menu
Ctrl+Shift+D	Displays the SQL Dialog Editor
Ctrl+S	Saves the model definition
F12	Puts focus on the editor

The following table shows the shortcut keys that are supported in the Generate Change Commands wizard.

Table 3. Keyboard shortcuts in the Generate Change Commands wizard

Keyboard shortcut	Action
Alt+n	Moves to next page of wizard
Alt+b	Moves to previous page of wizard
Alt+f	Finishes the wizard
Alt+d	On the DDL Generation Options page, checks the action for Generate Delta DDL or unchecks it if it is already selected
Alt+u	On the DDL Generation Options page, checks the action for Enable Undo or unchecks it if it is already selected
Alt+p	On the DDL Generation Options page, checks the action for Enable Data Preservation or unchecks it if it is already selected
Atl+l	On the Specify File Information page, moves to the Data File Location field
Atl+d	On the Specify File Information page, selects the Delimited Data File Format radio button
Atl+i	On the Specify File Information page, selects the Integrated Exchange Format (IXF) radio button
Atl+c	On the Specify File Information page, moves to the Code Page field
Atl+d	On the Mapping Tables page, moves to the Data Preservation tab
Atl+u	On the Mapping Tables page, moves to the Data Preservation for Undo tab
Shift+F10,a	On the Mapping Tables page, adds an entry
Shift+F10,e	On the Mapping Tables page, edits an entry
Shift+F10,d	On the Mapping Tables page, deletes an entry
Alt+c	On the Customize Import Commands page, moves to the Commands tab
Alt+n	On the Customize Import Commands page, moves to the Columns tab
Alt+m	On the Columns tab of the Customize Import Commands page, checks the Map Columns box or unchecks it if it is already selected
Alt+s	On the DB2 Maintenance Commands page, checks the Generate runstats commands box or unchecks it if it is already selected
Alt+f	On the DB2 Maintenance Commands page, checks the Generate flush package cache commands box or unchecks it if it is already selected
Alt+b	On the DB2 Maintenance Commands page, checks the Generate rebind commands box or unchecks it if it is already selected

Table 3. Keyboard shortcuts in the Generate Change Commands wizard (continued)

Keyboard shortcut	Action
Alt+I	On the DB2 Maintenance Commands page, checks the Limit scope of the generated rebind commands box or unchecks it if it is already selected

Keyboard shortcuts in the DB2 Change Management Expert information

When viewing the online product documentation, you can navigate the interface by keyboard.

The following key combinations are available:

- To go directly to the Topic pane (the right side), press Alt+K, and then press Tab.
- In the Topic pane, to go to the next link, press Tab.
- To go directly to the Search Results view in the left side, press Alt+R, and then press Enter or Up arrow to enter the view.
- To go directly to the Table of Contents view in the left side, press Alt+C, and then press Enter or Up arrow to enter the view.
- To expand and collapse a node in the table of contents, press the Right and Left arrows.
- To move to the next topic node, press the Down arrow or Tab.
- To move to the previous topic node, press the Up arrow or Shift+Tab.
- To go to the next link, button, or topic node from inside on of the views, press Tab.
- To scroll all the way up or down in a pane, press Home or End.
- To go back, press Alt+Left arrow; to go forward, press Alt+Right arrow.
- To go to the next pane, press F6.
- To move to the previous pane, press Shift+F6.
- To print the active pane, press Ctrl+P.

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other IBM DB2 and IMS Tools documentation:

- Use the online reader comment form located at:

  www.ibm.com/software/data/rcf/
- Send your comments by e-mail to comments@us.ibm.com. Be sure to include the name of the book, the part number of the book, the version of DB2 Change Management Expert, and, if applicable, the specific location of the text you are commenting on (for example, a page number or table number).

Summary of changes to this book

This section summarizes the significant improvements or enhancements for DB2 Change Management Expert and refers you to relevant sections of this book for more information.

Version 1, Release 1 January 2007

Information and maintenance has been added to several chapters. These additions are marked using the revision symbol | (a vertical bar).

- Installation information for DB2 Change Management Expert has been updated.
- Information on DB2 Change Management Expert and DB2 V9 support has been added.
- Information on new data features that include data migration between databases, support for HPU and LOAD utilities, and automatic data type casting has been added.
- Information for the new deployment states and the Deploy Now function has been added.
- Information on the integration of Visual Explain has been added.
- Information on the extended support for REORG and RUNSTATS commands has been added.

Chapter 1. The installation process for DB2 Change Management Expert

DB2 Change Management Expert can be installed on Linux® and Windows® 32-bit systems.

The installation process includes:

- Installing DB2 Change Management Expert
- Verifying that DB2 Change Management Expert was installed correctly
- Optionally, installing DB2 Grouper

After you have installed DB2 Change Management Expert, you can customize your user preferences at any time.

Installation requirements for DB2 Change Management Expert

The prerequisites for installing DB2 Change Management Expert include both hardware and software requirements.

Hardware and operating system

Tip: Always check the readme file for updates.

DB2 Change Management Expert can be installed on the following client platforms:

- Windows 2000, Windows 2003, and Windows XP (all 32-bit)
- Linux on xSeries® (32-bit)

DB2 Change Management Expert works with DB2 on the following servers:

- AIX® 5.2, AIX 5.3 (32 bit and 64-bit DB2 instances)
- Windows 2000, Windows 2003, and Windows XP (32-bit DB2 instances)
- Solaris V8, Solaris V9, Solaris V10 (32-bit and 64-bit DB2 instances)
- Linux on xSeries RH V3 and SUSE 8 (32-bit and 64-bit DB2 instances), Linux on pSeries® RH V3 (64-bit DB2 instances only), Linux on zSeries® RH V3 and SUSE 9 (64-bit DB2 instances only)

DB2 Change Management Expert has the following hardware requirements:

- 1 GB RAM (2 GB is recommended)
- 450 MB disk space
- 25 MB of free disk space
- Additional space for user data; the amount that you need depends on the size of your databases and the amount of change activity on your system

Software

DB2 Change Management Expert has the following software requirements on the server platform:

- IBM DB2 UDB V8 Fix Pack 11 for Linux, Unix, and Windows or later
- IBM DB2 9 for Linux, Unix, and Windows or later
- DB2 Database Administration Server

- Optional: IBM DB2 High Performance Unload for Multiplatforms (or Workgroups) V3.1 or later

  DB2 HPU must be installed to set and use HPU as the DB2 Change Management Expert Data Unload Provider preference. For more information, see the unload provider options in the Data preservation topic.

DB2 Change Management Expert has the following software requirements on the client platform:

- Connection to a locally installed DB2 database or connection to a DB2 UDB Administration Client (DAS)
- A Web browser (to view demonstrations of the product in the online documentation)
- IBM Java Runtime Environment (JRE) Version 1.5.0, which is included with DB2 Change Management Expert
- IBM Installation Manager Version 1.0, which is included with DB2 Change Management Expert

  The DB2 Change Management Expert installation program requires IBM Installation Manager. The installation program prompts you to install IBM Installation Manager if it is not already installed.

In addition, if DB2 Change Management Expert V1.1 is currently installed, ensure that it is uninstalled before you install V1.1.1.

You must ensure that the DB2 libraries are in the PATH environment variable (Windows) or in the LD_LIBRARY_PATH environment variable (UNIX®).

To run DB2 Change Management Expert, you must use the IBM Java Database Connectivity (JDBC) Type 4 driver. The JDBC Type 4 driver is included in DB2 Change Management Expert. You can select a different JDBC driver if necessary.

On Linux, if the class location of your JDBC driver (db2jcc.jar and license) is not automatically populated, check your environment. The DB2INSTANCE environment variable should be set to the local DB2 instance name, for example, export DB2INSTANCE=db2inst1 (this is case sensitive). Otherwise, you can specify the location manually.

To run DB2 Change Management Expert, you must also have connectivity to the DB2 UDB Administration Client (DAS) that is running on the same machine as the database. The DAS connectivity libraries can be provided by either of the following ways:

- A local DB2 instance
- The DB2 UDB Administration Client

In either case, make sure the bin directory is in the path before you start DB2 Change Management Expert. (In Linux, this is handled by sourcing $INST_HOME/sqllib/db2profile.) The DB2 UDB Administration Client is available from developerWorks® at http://www.ibm.com/developerworks.

For example, to install the DAS client on a Linux system, complete the following steps:

1. Install the DAS Client.
2. Create a DB2 instance.
3. Source db2profile in $INST_HOME/sqllib/ before you start DB2 Change Management Expert. (For example, add it to your .bashrc.)

Installing DB2 Change Management Expert

You can install DB2 Change Management Expert on a Windows system or on a Linux system.

DB2 Change Management Expert is based on Eclipse, an open source Integrated Development Environment (IDE). DB2 Change Management Expert uses an installation strategy that allows you to share the same Eclipse shell with other products that are also based on Eclipse. When you use shared Eclipse resources and install products into the same package group, you have all of the features of the products that you have installed available in the same user interface.

Recommendation: If you choose to share the same Eclipse shell with other products, do so only with products that fully support shell sharing. Otherwise, some capabilities might not be available. For example, you might not be able to safely uninstall or upgrate such environments.

The DB2 Change Management Expert installation program requires IBM Installation Manager. If the installation program cannot find IBM Installation Manager, it prompts you to install IBM Installation Manager, which is included with the installation program. When you install DB2 Change Management Expert, you have the option of:

- Creating a new Eclipse installation
- Using an existing Eclipse installation that was created for other products that were installed with IBM Installation Manager
- Using an existing Eclipse installation (version 3.2.1 or higher) that was created using a different method

Installing DB2 Change Management Expert on Windows systems

You can install DB2 Change Management Expert on a Windows system.

To install DB2 Change Management Expert on a Windows system, complete the following steps:

1. Start the Launchpad program. If you are installing from a CD, place the CD into the CD-ROM device. The Launchpad program starts automatically if the drive is set for autoplay. If the Launchpad program does not start automatically, use File Manager or Windows Explorer to locate your CD-ROM drive, and double-click the `launchpad.exe` file.

   If you are installing from a downloaded package, complete the following steps:

   a. Download the compressed .zip files to your hard drive.
   b. Extract the downloaded files into one temporary folder on your hard drive.
   c. Double-click the `launchpad.exe` file.

   The Welcome panel of the Launchpad program is displayed.
2. Click **Release Information** to read information about the release. Click **Install** to start the installation.
3. If IBM Installation Manager Version 1.0 is not already installed on your system, a program to install IBM Installation Manager runs automatically. Complete the following steps to install IBM Installation Manager:

   a. On the Welcome page, click **Next**.
   b. Read the licensing agreement, select **I accept the terms in the license agreement**, and click **Next**.

c. On the Destination Folder page, specify the folder where you want to install IBM Installation Manager. You can accept the default folder or specify your own. Click **Next**.

d. On the Ready to Install the Program page, click **Install**.

e. When the installation of IBM Installation Manager is complete, click **Finish**.

4. When IBM Installation Manager starts, click **Next**. DB2 Change Management Expert Version 1.1.1 is automatically pre-selected to be installed.
5. On the Licenses page, read the licensing agreement. Select **I accept the terms in the license agreement**, and click **Next**.
6. On the Location page, specify where to install the shared resources if you are prompted to do so. You can accept the default folder or specify your own. Click **Next**. You will be prompted to specify a location for the shared resources if you have not installed other products using IBM Installation Manager. The shared resources are the common set of Eclipse plugins that other products that are based on Eclipse, such as Rational® Data Architect or Rational Software Architect, can use. The name of the folder must not exceed 80 characters.
7. On the Location page, specify whether to install DB2 Change Management Expert into an existing package group or into a new package group. If you specify a new package group, specify a location for the package. If an existing directory is specified as the location, it must be empty. The name of the folder must not exceed 80 characters. Click **Next**.
8. On the Location page, select whether you want to install DB2 Change Management Expert into an existing Eclipse installation. If you choose to install into an existing Eclipse installation, specify the location of your Eclipse Integrated Development Environment (IDE) in the Eclipse IDE field, and leave the default value in the Eclipse IDE JVM field. Click **Next**. If you install DB2 Change Management Expert into an existing Eclipse installation, the directory into which you specified to install DB2 Change Management Expert on the Location page is ignored because the location will default to the existing Eclipse directory.
9. On the Features page, click **Next** to accept English. English is the only language that is supported.
10. On the Features page, click **Next**. DB2 Change Management Expert consists of a single feature, and there are no additional optional features that can be installed.
11. On the Summary page, verify the information. Click **Back** to change any specifications that were made on previous pages. Click **Install** to begin installing DB2 Change Management Expert on your system.
12. When the installation completes, click **Finish**. Before you click **Finish**, you can click **View the Log** to see the installation details.
13. Close the Launchpad program.

You are now ready to start DB2 Change Management Expert.

Installing DB2 Change Management Expert on Linux systems

You can install DB2 Change Management Expert on a Linux for xSeries system.

You must have root authority to complete the installation of DB2 Change Management Expert.

To install DB2 Change Management Expert on a Linux system, complete the following steps while logged in as the root user:

1. Start the Launchpad program by completing one of the following sets of steps, depending on whether you are installing from a CD or a download package. If you are installing from a CD:
   a. Place the CD into the CD-ROM device on the computer where you are installing DB2 Change Management Expert.
   b. Mount the CD on a suitable mount point, for example, mount -rv cdrfs /dev/cd0 /cdrom.
   c. Verify that X applications (such as xclock&) can use the desired display:
      1) Show the current setting of the DISPLAY variable by typing: echo $DISPLAY
      2) Export the display to the workstation where the X server is running by typing: export DISPLAY=hostname:0
      3) Verify that X applications can display correctly by typing: xclock&
      4) Close the display.
   d. Change to the directory where the CD-ROM is mounted by entering the following command where *cdrom* is the mount point of the DB2 Change Management Expert installation CD-ROM: `cd /cdrom`
   e. Enter the command `./launchpad.sh`. The Welcome panel of the Launchpad program is displayed.

   If you are installing from a download package:
   a. Download the compressed file (.tar.gz) to your computer.
   b. Extract the downloaded files into a temporary directory on your computer.
   c. Verify that X applications (such as xclock&) can use the desired display:
      1) Show the current setting of the DISPLAY variable by typing: echo $DISPLAY
      2) Export the display to the workstation where the X server is running by typing: export DISPLAY=hostname:0
      3) Verify that X applications can display correctly by typing: xclock&
      4) Close the display.
   d. Change to the temporary directory where the downloaded files are extracted by entering the following command where *temp* is the temporary directory: `cd temp`
   e. Enter the command `./launchpad.sh`. The Welcome panel of the Launchpad program is displayed.
2. Click **Release Information** to read information about the release. Click **Install** to start the installation.
3. If IBM Installation Manager Version 1.0 is not already installed on your system, a program to install IBM Installation Manager runs automatically. Complete the following steps to install IBM Installation Manager:
   a. On the Welcome page, click **Next**.
   b. Read the licensing agreement, select **I accept both IBM and non-IBM terms**, and click **Next**.
   c. On the Destination Folder page, specify the folder where you want to install IBM Installation Manager. You can accept the default folder or specify your own. Click **Next**.
   d. On the Ready to Install the Program page, click **Back** to modify your selections, or click **Install**.
   e. When the installation of IBM Installation Manager is complete, click **Finish**.

4. When IBM Installation Manager starts, click **Next**. DB2 Change Management Expert Version 1.1.1 is automatically pre-selected to be installed.
5. On the Licenses page, read the licensing agreement. Select **I accept the terms in the license agreement**, and click **Next**.
6. On the Location page, specify where to install the shared resources if you are prompted to do so. You can accept the default folder or specify your own. Click **Next**. You will be prompted to specify a location for the shared resources if you have not installed other products using IBM Installation Manager. The shared resources are the common set of Eclipse plugins that other products that are based on Eclipse, such as Rational Data Architect or Rational Software Architect, can use. The name of the folder must not exceed 1024 characters.
7. On the Location page, specify whether to install DB2 Change Management Expert into an existing package group or into a new package group. If you specify a new package group, specify a location for the package. If an existing directory is specified as the location, it must be empty. The name of the folder must not exceed 1024 characters. Click **Next**.
8. On the Location page, select whether you want to install DB2 Change Management Expert into an existing Eclipse installation. If you choose to install into an existing Eclipse installation, specify the location of your Eclipse Integrated Development Environment (IDE) in the Eclipse IDE field, and leave the default value in the Eclipse IDE JVM field. Click **Next**. If you install DB2 Change Management Expert into an existing Eclipse installation, the directory into which you specified to install DB2 Change Management Expert on the Location page is ignored because the location will default to the existing Eclipse directory.
9. On the Features page, click **Next** to accept English. English is the only language that is supported.
10. On the Features page, click **Next**. DB2 Change Management Expert consists of a single feature, and there are no additional optional features that can be installed.
11. On the Summary page, verify the information. Click **Back** to change any specifications that were made on previous pages. Click **Install** to begin installing DB2 Change Management Expert on your system.
12. When the installation completes, click **Finish**. Before you click **Finish**, you can click **View the Log** to see the installation details.
13. Close the Launchpad program.

Important: After DB2 Change Management Expert is installed on your Linux system, you can only create and use a workspace in the directories where you have write permission. If you try to create or use a workspace in a directory where you do not have write permission, the following error message will be displayed:

```
Workspace in use, choose a different one.
```

You must click **OK** and return to the dialog box to specify a valid workspace.

Verifying the installation of DB2 Change Management Expert

The ability to run DB2 Change Management Expert verifies that it was installed correctly.

To verify that you successfully installed DB2 Change Management Expert, complete the following steps:

1. Start DB2 Change Management Expert.

   To start DB2 Change Management Expert on a Windows system, click **Start** → **Programs** → **DB2 Change Management Expert** → **Change Management Expert**.

   To start DB2 Change Management Expert on a Linux system, enter the command: cme.sh
2. In the Workspace Launcher dialog, specify the workspace directory to use. You can accept the default or specify your own directory. Click **OK** to continue. The Welcome page is displayed.
3. On the Welcome page, click on any one of the icons to access an overview of the product, tutorials for the product, sample projects, and additional information resources. When you move your cursor over an icon, its description is displayed. You can return to the Welcome page at any time by clicking **Help** → **Welcome**.
4. Close the Welcome page to continue to the main views for DB2 Change Management Expert.

Installing IBM DB2 Grouper

IBM DB2 Grouper is client-server product that is an optional component of various DB2 products, including DB2 Change Management Expert. You can use DB2 Grouper to quickly locate tables and objects that are related to other tables in your database models.

For information about installing and configuring DB2 Grouper for Multiplatforms, see the *DB2 Grouper User's Guide* at http://www.ibm.com/software/data/db2imstools/db2tools-library.html,

Uninstalling DB2 Change Management Expert

You can uninstall DB2 Change Management Expert from your Windows or Linux system.

Uninstalling DB2 Change Management Expert from a Windows system

You can uninstall DB2 Change Management Expert from a Windows system.

To uninstall DB2 Change Management Expert, complete the following steps:

1. Click **Start** → **Programs** → **IBM Installation Manager** → **IBM Installation Manager**.
2. Click **Uninstall Packages**. The wizard to uninstall packages starts.
3. On the Uninstall Packages page, select DB2 Change Management Expert, and click **Next**.
4. Review the summary information, and click **Uninstall**.
5. When the uninstall wizard completes, click **Finish**.

Tip: A way to start the wizard to uninstall packages with DB2 Change Management Expert Version 1.1.1 pre-selected is to use File Manager or Windows Explorer to locate the `uninstall` subfolder of the directory where DB2 Change Management Expert is installed and double-click the `uninstall.exe` file. You can use that method instead of clicking **Start** → **Programs** → **IBM Installation Manager** → **IBM Installation Manager**.

Uninstalling DB2 Change Management Expert from a Linux system

You can uninstall DB2 Change Management Expert from a Linux system.

To uninstall DB2 Change Management Expert from a Linux system, complete the following steps while logged on as the root user:

1. Verify that X applications (such as xclock) can use the desired display:
   a. Show the current setting of the DISPLAY variable by typing: echo $DISPLAY
   b. Export the display to the workstation where the X server is running by typing: export DISPLAY=hostname:0
   c. Verify that X applications can display correctly by typing: xclock&
   d. Close the display.
2. Start IBM Installation Manager by going to the `eclipse` subdirectory of where you installed IBM Installation Manager and running the `launcher` program. For example, if you installed IBM Installation Manager in the default directory, go to /opt/IBM/InstallationManager/eclipse and type `launcher` on the command line.
3. Click **Uninstall Packages**. The wizard to uninstall packages starts.
4. On the Uninstall Packages page, select DB2 Change Management Expert, and click **Next**.
5. Review the summary information, and click **Uninstall**.
6. When the uninstall wizard completes, click **Finish**.

Tip: A way to start the wizard to uninstall packages with DB2 Change Management Expert Version 1.1.1 pre-selected is to go to the `uninstall` subdirectory of the directory where you installed DB2 Change Management Expert and run the `uninstall.bin` program. You can use that method instead of running the `launcher` program from the `eclipse` subdirectory.

Uninstalling IBM Installation Manager

You can uninstall IBM Installation Manager from your system.

To uninstall IBM Installation Manager, complete the appropriate set of steps for the type of system from which you are uninstalling the product:

1. To uninstall from a Windows system:
   a. Click **Start** → **Settings** → **Control Panel**.
   b. In the Control Panel window, double-click **Add or Remove Programs**.
   c. In the Add or Remove Programs window, select **IBM Installation Manager** and click **Remove**.
   d. In the confirmation message window, click **OK**.
2. To uninstall from a Linux system:
   a. Find the directory where you installed IBM Installation Manager.
   b. Go to the subdirectory that is named _uninst.
   c. Run the file that is named `uninstaller.bin`.
   d. Complete the steps in the uninstall wizard.

Uninstalling IBM DB2 Grouper

If you have installed IBM DB2 Grouper, which is an optional component of DB2 Change Management Expert, you might want to uninstall it after you have uninstalled DB2 Change Management Expert

DB2 Grouper is also an optional component of other IBM products. Check with your Grouper Administrator to determine the usages of DB2 Grouper and whether it can be uninstalled.

For information about uninstalling DB2 Grouper, see the *DB2 Grouper User's Guide* at http://www.ibm.com/software/data/db2imstools/db2tools-library.html.

Migrating from a previous release of DB2 Change Management Expert

You can migrate from one release of DB2 Change Management Expert to the next.

To migrate from DB2 Change Management Expert Version 1.1.0 to Version 1.1.1, complete the following steps:

1. Uninstall Version 1.1.0 from your system, noting the location of your workspace directory before you begin. The workspace directory will not be deleted when the product is uninstalled.
2. Install DB2 Change Management Expert Version 1.1.1.
3. Start DB2 Change Management Expert and, when you are prompted for the workspace directory to use, specify a new workspace.
4. Import the projects from the workspace directory that you used for Version 1.1.0 into the new workspace directory for Version 1.1.1 by completing the following steps:
   a. Click **File** → **Import**.
   b. In the **Import** wizard, expand the General folder, click **Existing Projects into Workspace**, and follow the steps in the wizard to select the projects from the workspace directory that you used for Version 1.1.0.

Chapter 2. Customization of DB2 Change Management Expert

You can change the default options for some DB2 Change Management Expert functionality. Specifically, you can set preferences for working with SQL scripts and for working with the Data Model Editor.

You can change your preferences any time after you install and start using DB2 Change Management Expert. You can also reset the preferences to the default DB2 Change Management Expert settings.

Changing preferences for SQL scripts

You can change preferences for how DB2 Change Management Expert processes SQL scripts.

To change your preference settings for SQL scripts, complete the following steps:

1. From the main menu, click **Window** → **Preferences**.
2. In the Preferences window, expand **Data** and click **Change Management Options**.
3. Change your preferences. The following table shows the preferences that you can change:

Table 4. User preferences for SQL Scripts

Item	Description
Choose an SQL termination character	Sets the character that will be used as the termination character for SQL statements.
Maximum parser errors for diagnosis	Sets the maximum number of parser errors that are reported when an SQL script file is opened. The value must be an integer that is greater than 0. When SQL script files are large, it can be helpful to limit the number of errors that are diagnosed.
Maximum parser time for diagnosis	Sets the maximum amount of time for finding errors to report when an SQL script file is opened. The value must be an integer that is greater than 0. When SQL script files are large, it can be helpful to limit the time that is spent diagnosing errors.
Default DB2 version	Sets the DB2 version level at which syntax is parsed when a script is opened in the Change Command Editor.
Ignore semantic errors when reverse engineering	Specifies that semantic errors will be ignored when the models are created from SQL scripts.
Ignore semantic errors when applying DDL to models	Specifies that semantic errors will be ignored when changing models with SQL scripts. If you choose to ignore errors, you can later validate a model to find problems.
Implicit objects used when reverse engineering	When you create SQL scripts from models, specifies the default values that will be used for the statements for implicitly created objects. You can specify values for the partition group, the temporary partition group, the buffer pool, and the table space.

Table 4. User preferences for SQL Scripts (continued)

Item	Description
Forward engineering options	Unchecking the **Order the create statements for derived objects** option might leave create statements for derived objects such as views, aliases and MQTs in the wrong order. If you uncheck this option, remember to inspect the generated change commands to make sure that each derived object is created before it is used by another derived object.
Unload provider	Specifies the initial value of the Data Unload Provider field on the Mapping Tables page of Generate Change Commands wizard. Data will be unloaded for data preservation with this provider unless you select a different provider on the wizard page.
Load provider	Specifies the initial value of the Data Load Provider field on the Mapping Tables page of Generate Change Commands wizard. Data will be loaded with this provider unless you select a different provider on the wizard page.

Changing preferences for the Data Model Editor

You can change preferences for the validations options, appearance of the breadcrumb trail, and appearance of the Relationships section in the Data Model Editor.

To change your preference settings for the Data Model Editor, complete the following steps:

1. From the main menu, click **Window** → **Preferences**.
2. In the Preferences window, expand **Data** and click **Data Model Editor Options**.
3. Change your preferences. As the following table shows, you can change your preferences for the validation options, the breadcrumb trail, and the Relationships section.

Table 5. User preferences for the Data Model Editor

Item	Preference	Description
Validation Options	On Save	Validate the model when it is saved.
	On Change	Validate the model when it is changed.
	Disable	Prohibit validation of models.
Bread Crumb Trail	Show Bread Crumbl	Show the path to the current edit object.
	Show Title in Bread Crumb	Show the title of the current edit object at the end the breadcrumb trail instead of separately above the breadcrumb trail.
	Annotate Title	When the title is shown above the breadcrumb trail, show stereotype text and expanded information in the title. For example, the title will include the object type and the name of the model to which the object belongs.

Table 5. User preferences for the Data Model Editor (continued)

Item	Preference	Description
Related Objects	Show Relationships	Show the relationships section, which shows the objects, by category, that are related to the current edit object.
	Show Decorators	Show data type and length information in labels.
	Show Empty Relationships	Show all the categories in the relationships section, including those that are empty.
	Show Parent Relationships	Show the parent relationships to the current edit object.

To access model validation options in general, select **Model Validation** → **Constraints**.

Chapter 3. Introduction and concepts

The following topics provide an overview of managing database changes with DB2 Change Management Expert. These topics introduce DB2 Change Management Expert concepts, terminology, and common scenarios.

Introduction to database change management

Database change management is the process of determining what changes need to be made to a database, specifying those changes, and then deploying them.

Changes to database schemas can be required for a number of reasons, including new business requirements, mergers, legislative changes, and application environment changes. Schema changes can involve changes to both *logical database objects* (for example, tables, columns, primary keys, constraints) and to *physical database objects* (for example, databases, table spaces, buffer pools, indexes). Changing database objects, regardless of their type, is often not a trivial operation. Changes often impact dependent objects and sometimes even underlying data. The process of analyzing and maintaining these dependencies is traditionally time-consuming and prone to error.

Consider a typical database environment, shown in Figure 1, in which new applications and database design changes are initially introduced on a dedicated development system, are then validated on a test system, and are finally deployed to the organization's production system.

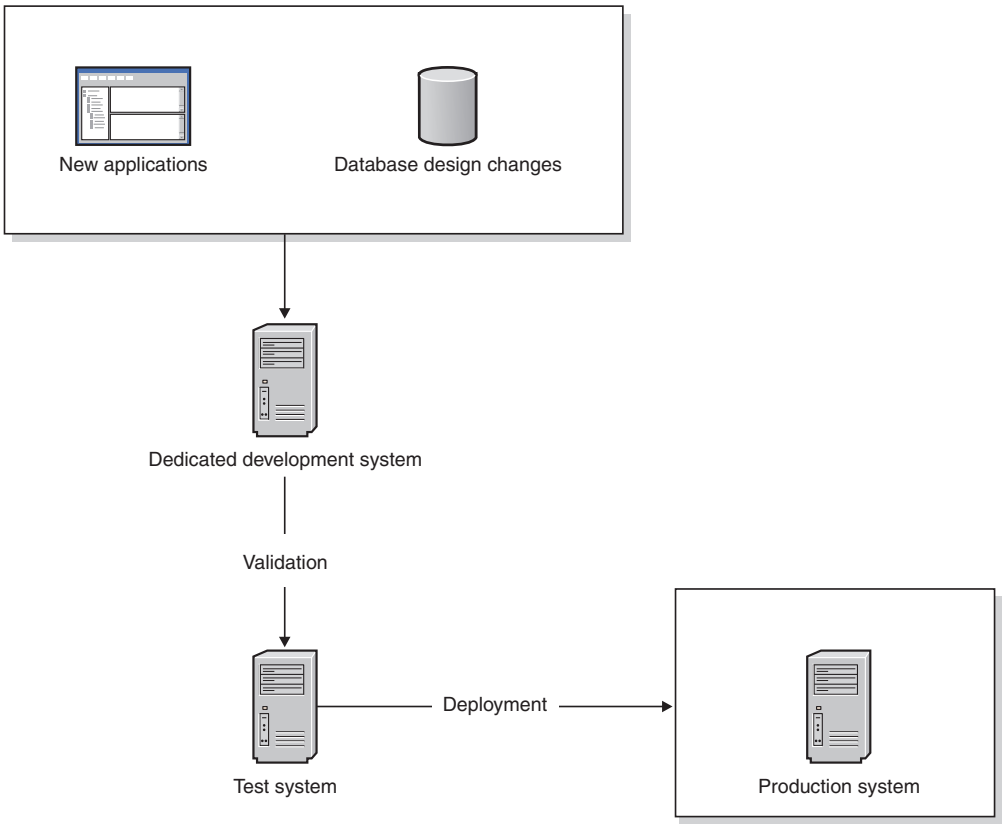

Figure 1. Typical database environment

Although the overall design of the development, test, and production systems is typically quite similar, the business rules that govern each system are likely to be different. The production database operates under strict business rules and must be running 24 hours a day, 7 days a week. The test database also operates under strict business rules to assure that what is tested will run properly in production; however, the test database does not require the same level of availability that the production system requires. In contrast to the production and test systems, the development database often has fewer business rules because developers need to constantly make changes. The process of managing these disparate database systems often requires a DBA to:

- Synchronize the development system or the test system to be a point-in-time copy of the production system
- Promote (or migrate) changes from one system to another system
- Undo changes that have been made to a database environment
- Create a historical base model for future reference
- Audit changes to understand their effects
- Manage the life cycle of structural changes to databases
- Compare two sets of objects to determine how they differ
- Analyze the impact of a proposed change on a database
- Manage the deployment of changes to the target database
- Load, unload, and move data
- Rebind packages that have become inoperative as a result of your changes
- Refresh or redefine dependent objects

Change management is often a difficult and time-consuming process for a database administrator because it presents the following challenges:

- Failing to recognize a schema change is dangerous to system integrity.
- Finding all related schema change elements is difficult.
- Analyzing the impact of a schema change is time consuming.
- Planning the migration requires deep consideration.
- Applying changes to a database necessitates expert precision.
- Learning SQL syntax can be difficult.

Change management software can make the process of change management easier because it increases reliability and reduces human error.

DB2 Change Management Expert overview

DB2 Change Management Expert is a powerful and flexible tool that simplifies the process of identifying, analyzing, and implementing database schema changes for DB2 UDB for Linux, UNIX, and Windows.

DB2 Change Management Expert simplifies the process of managing schema changes. You can use DB2 Change Management Expert to visualize your changes before you make them.

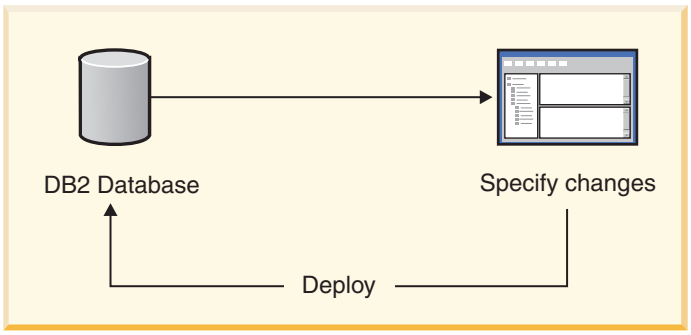

Figure 2. Simplifying database changes with DB2 Change Management Expert

You can use DB2 Change Management Expert to apply change commands to physical data models, which enables you to see the results of the change commands before you deploy the changes against the actual DB2 database. Applying a change command to a model is similar to issuing a command against DB2.

DB2 Change Management Expert helps solve or alleviate many of the problems that a DBA faces:

Table 6. Problems that DB2 Change Management Expert helps solve

Typical DBA problem	**DB2 Change Management Expert solution and value**
Failing to recognize a schema change can be dangerous to system integrity.	Provides an internal model of database dependencies to enable the DBA to fully understand the effects of the changes before they are made.
Finding all related schema change elements is difficult.	Displays related schema changes to help the DBA better understand the change. Finding schema changes is a difficult task to perform manually. This feature saves time.
Analyzing the impact of a schema change is time consuming.	Aggregates summary information and allows the DBA to drill down into the model. This feature enables the DBA to fully understand the issues that relate to impact and migration.
Planning the migration requires deep consideration.	Visualization capabilities such as graphical compare, graphical dependency viewing, SQL source text formatting, and context-sensitive help. These capabilities allow the DBA to see exactly how a change will affect a database.
Applying changes to a database necessitates expert precision. Finding which objects will be affected by a change is difficult and time consuming.	DB2 Change Management Expert makes the process easier by automatically propagating changes. Automatically propagating changes also increases reliability and reduces human error when analyzing, migrating, and issuing changes to the database schema.

DB2 Change Management Expert provides the following features:

- An interactive user interface that allows you to make changes iteratively and see the results from different viewpoints
- Provides graphical wizards for commonly performed tasks

- Generates delta DDL to change the database, which can then be viewed, edited, and validated before it is run
- Provides automated forward and reverse schema engineering between the models, the database, and the DDL
- Synchronizes, copies, or merges database schema definitions from the base model to the target model
- Integrates remote DB2 Command execution, such as utility execution, application binding and rebinding, and data loading and unloading
- Creates new schemas or loads them from a variety of different sources, such as databases and DDL
- Reports on the impact of proposed changes, which mitigates an otherwise time consuming process
- Creates a historical base model for future reference
- Is based on a modern infrastructure that integrates with data management components such as IBM DB2 Grouper
- Exploits modern development tools and techniques for faster, more efficient development and a more scalable infrastructure
- Leverages IBM Data Tooling and Eclipse technology to provide an extensible user interface

DB2 Change Management Expert can also manage database objects in an Enterprise Server Edition (ESE) and Database Partitioning Feature (DPF) environment. However, DB2 Change Management Expert does not change manage DPF objects or configurations.

Overview of managing changes to a database

The three most important tasks that a DBA can perform with DB2 Change Management Expert are change specification, deployment, and auditing.

The following figure depicts the sequence of steps that are required to manage changes to a database:

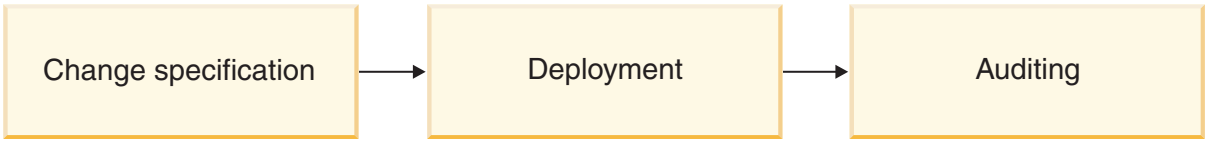

Figure 3. Steps required to manage changes

Change specification
The process of defining a set of changes that you want to apply to a database. This phase involves creating models of the database that you want to change and using the models to define the changes, analyze the impact of the changes, and generate the operations to implement the changes against the database.

Deployment
The process of changing the database to reflect the conditions that were specified in the model specification phase. Deployment encompasses the process of applying the changes to the actual database, and if necessary, undoing the changes.

Auditing

The process of looking back after a change is made to understand the evolution of the database model. This phase is important because the catalog reflects only the current state of a database and does not keep a record of its history. Auditing helps you keep track of who performed what actions and helps you to prevent unknown or unanticipated access to the data.

DB2 Change Management Expert terminology

Understanding DB2 Change Management Expert terminology will help you understand the change management process with DB2 Change Management Expert.

- Instead of making changes directly to the DB2 catalog, you begin by making changes to a *model*. A model is a representation of the database catalog that can be stored in memory or in the local file system. You can use models to plan changes to a database, analyze the impact of those changes, and deploy those changes to the actual database. You can also use models to identify the differences between databases and move those differences from one database to another database. Models can represent the same database at different points in time or different databases.
- A *base model* is a representation of the database before you apply changes; it describes the current catalog. The base model is important because it is the version of the catalog on which the change is based. At the time of deployment, the base version and the catalog should match. If not, the deployment might not be successful or the desired results might not be achieved. DB2 Change Management Expert does not lock the database schema; therefore, the database can be changed during the time that you are constructing your change. You can check whether the base model and the catalog have diverged before the change is applied. If they are not the same, you can evaluate the impact of the discrepancies on your change to assure that your change does not fail and that the desired results will be achieved. In addition, if you apply a change and then need to undo it, you can use the base model to revert to a previous version of the catalog.
- A *target model* is a model representation of what the catalog will look like after the change is deployed. For example, if you want to move changes from a development environment to a test environment, the development database acts as the source model, and the test database is the target model.
- A *model version* is a representation of the DB2 catalog at a specific point in time. A base model is a special type of model version. Model versions are useful because the DB2 catalog does not keep a history of the changes that you make. Model versions provide a historical perspective of the database.
- A *source model* is a potential source of database objects that can be copied to the target model. While base and target models represent versions of the same database, source models can be from different databases.
- *Copying* a database is the process of copying an object or objects and the entire internal structure from the source database to the target database. During the copying process, you can narrow the scope of the changes to specific database objects. You can narrow the scope of the copying or fine-tune any changes. You can use the copy option in the following situations:
  - To make the development environment an exact copy of the production environment before beginning development
  - To copy a production environment to a test environment

  - To copy the test system to the development system when you do not need to fine-tune any changes
- A *change command* is an ordered collection of operations, such as DDL, DML, DCL, running utilities, or rebind commands, that is applied to a model to change it. Change commands can be SQL, DB2 commands, or DB2 utility invocations. You can generate change commands between the base model and the target model by using the Deployment Script Editor, create them manually by using the Change Command Editor, or import them from a script file. Change commands can be applied to models to visually represent their impact, or they can be implemented immediately by deploying them against your DB2 database. You can use change commands to:
  - Change the database catalog
  - Visualize the results of DDL on a model

  In some cases, applying change commands might leave a model with invalid, undefined, or inoperative objects.
- An *undefined* object is a DB2 object that is valid and operative, but undefined at run time. For example, alias objects become undefined when they do not point to a valid object. When an object becomes undefined, it is annotated and is displayed in the Problems view as a warning.
- An *inoperative* object is a DB2 object that is valid, but must be re-created to be used. Inoperative markers point to objects that need to be created because of a change that you have made to the model. For example, views become inoperative if the tables that they reference are modified. When you generate change commands, DB2 Change Management Expert automatically makes objects operative in the Deployment Script Editor. When an object becomes inoperative, it is annotated and is displayed in the Problems view as a warning.
- *Deployment* is the second stage in the change management process. It is the process of changing the database catalog to reflect the conditions that you specified in the model specification phase. Deployment encompasses the process of applying the changes to the actual target database and, if necessary, undoing the changes.
- *Migration* is the process of moving an object or objects of one database to match another database. During migration, only part of the internal structure is copied from the source database to the target database. You can migrate objects by comparing models of the source and target objects, copying the differences to the target model or changing the target model directly, generating the DDL that will make the target database match the source database, and deploying the changes to the target database. You can also migrate database objects, attributes, and dependencies within the same database or between different database environments, such as between development environments and test environments.

  In addition to migrating objects using the migration action and the Comparison Editor on the Models page of the Deployment Script Editor, you can also use the Data Migration wizard that is available from the Deploy menu or the Change Commands page of the Deployment Script Editor.
- You can change the database catalog by using a *deployment script*, which is a set of changes that a user, or a group of users, wants to deploy to a database at a point in time. The deployment script organizes the various resources and actions that are required to deploy changes to the database catalog. The deployment script describes specification resources, change commands that are intended to implement the change, and metadata about how that change will be issued. You can create deployment scripts manually by using the Deployment Script Editor or DB2 Change Management Expert can generate them from the

proposed model and the target base model. You can use the deployment script to define your changes and to *deploy* the changes to the target database.

- You perform a *comparison* by analyzing the differences between two models, typically the base and target models. During the comparison process, DB2 Change Management Expert compares objects, attributes, and references between the two different models. The differences between the models are displayed in the Comparison Editor.

The DB2 Change Management Expert user interface

DB2 Change Management Expert is based on Eclipse, an open source Integrated Development Environment (IDE).

DB2 Change Management Expert opens in the Eclipse Workbench window, which consists of perspectives, views, and editors. A *perspective* is a particular set of views and editors.

You typically use a *view* to navigate information within the Workbench, open an editor, or display properties for the active editor. The changes that you make in a view are saved immediately.

You use an *editor* to browse or modify a resource. Modifications made in an editor follow an open-save-close lifecycle model.

The following figure shows that the DB2 Change Management Expert interface uses the Data perspective. The figure also provides an example of some of the views and editors in this perspective. The Database Explorer view, which displays the connections to databases, is in the lower left corner. The Project Explorer view, which shows the resources in the various projects, is in the upper left corner. More views, such as Properties, Tasks, and Problems, are located along the bottom. The Deployment Script Editor is located in the editor area in the middle, and it is the active editor. The Data Model Editor is tiled behind the Deployment Script Editor.

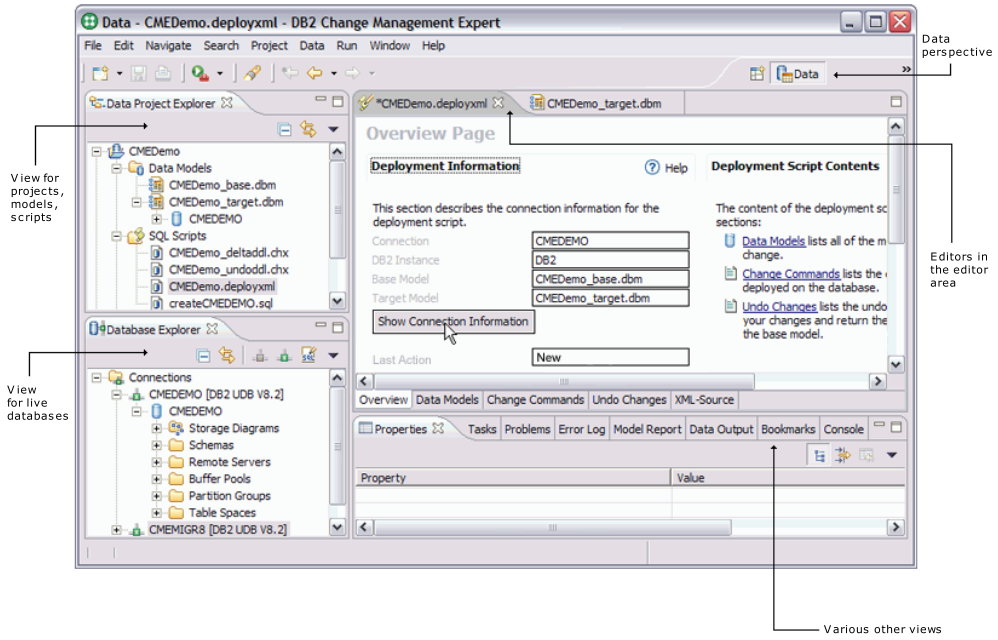

Figure 4. The DB2 Change Management Expert integrated development environment

Views

The main views that DB2 Change Management Expert uses include:

Database Explorer
Displays database connection information. You can use this view to manage database connections and view objects in your databases.

Data Project Explorer
Displays your database management projects, which contain the resources (models, scripts, and other files) that are required to make changes to a database.

Outline
Displays an outline of the resource file that is currently open in the editor area and lists structural elements. The contents of the Outline view are specific to the editor. The Outline view is linked to the Data Model Editor, the Diagram Editor, and the Change Command Editor.

Properties
Displays property names and values for a selected item. For example, for a connection that is selected in the Database Explorer view, the Properties view displays information about that connection. For a table that is selected within a model in the Data Project Explorer view, the Properties view displays information about the columns, the partition key, and the table space.

Tasks Displays system-generated problems, warnings, and errors. You can add your own entries to the table to build a list of tasks or to-do items.

Error Log
Displays all the internal warnings and errors that are issued by the Eclipse platform and by plug-in development code.

Problems
Displays problems with the validity of models, syntax errors in change commands, and the validity of deployment scripts. When you double-click a problem in the list, the editor is opened at the location of the problem.

Model Report
Displays the details about the relationship between an object and its dependent objects when you perform impact analysis to determine the effects of changing an object.

Data Output
Displays the results of deploying change commands and scripts to the database.

Bookmarks
Displays user-defined bookmarks. For example, you can bookmark the objects in a model from the Data Model Editor or the Outline view to make it easier to find the object later.

Editors

The main editors that DB2 Change Management Expert uses include:

Deployment Script Editor
Presents a consolidated way to work with the changes that you want deploy to your database. The Deployment Script Editor uses these tabs that contain the actions that you can use to access objects and perform database changes:
- Data Model tab
- Change Commands tab
- Undo Scripts tab
- XML Source tab
- Multiple Deployment tab

Data Model Editor
Edit schema definitions.

Diagram Editor
Visualize or edit the schema definitions of a model through its diagram.

Comparison Editor
Compare two models and merge the differences between them.

Change Command Editor
Edit change commands.

Tips

The following tips can help you work with views and editors:

- Two or more editors can be in the editor area. They are tiled by default.
- You can rearrange views and editors to customize the layout. However, you cannot drag views into the editor area, nor can you drag editors outside the editor area.
- Instead of closing views, you can use Fast views, which are hidden views, to free up screen space.
- You can open a view by selecting **Window → Show View**.
- You can reset the Data perspective to its default views and layout by selecting **Window → Reset Perspective**

How DB2 Change Management Expert works with Entity Relationship Modeling tools

DB2 Change Management Expert works with different Entity Relationship Modeling (ERM) tools such as Rational Data Architect (RDA). It is helpful to understand how the tools differ and how they work together.

DB2 Change Management Expert and Rational Data Architect (RDA)

DB2 Change Management Expert and RDA help address closely related business problems. For example,

With DB2 Change Management Expert you can:
Migrate data objects between environments, compare data models, and visualize dependencies. You can also change data objects in place, analyze your data models, and preserve data.

With RDA you can:
Create logical models and view the model entities and relationships in the Data Design Project.

With RDA and DB2 Change Management Expert you can:
Manage the life cycle of your database. RDA is useful for Business Analysts and Database Architects where DB2 Change Management Expert is useful for Database Administrators.

Product differences by role

DB2 Change Management Expert and RDA can be used in different role-oriented contexts.

Business Analysts and Data Architects can use RDA to:

- Work together on specifications
- Create database changes
- Use diagramming and visualization
- Analyze the impact of changes
- Develop data models iteratively
- Develop logical models
- Compare data models

Database Administrators can use DB2 Change Management Expert to:

- Migrate data object changes throughout Enterprise environments.
- Compare data models
- Create complex database changes
- Analyze the impact of changes
- Preserve data
- Preserve authorizations
- Preserve database application bindings
- Deploy complex database changes

Operations Managers can use DB2 Change Management Expert to:

- Audit changes
- Handle contingencies by using the undo and redo capabilities

Product differences by object

Certain objects are handled differently by the two products:

Data models

RDA supports logical data models and physical data models.

DB2 Change Management Expert supports physical data models.

You can translate the logical data models used by RDA into physical data models that can then be incorporated into DB2 Change Management Expert. Transformation from logical data models to physical data models must be completed within RDA. When you transform a logical model to a physical model, the logical entities map to physical objects. For example, entities become tables, attributes become columns, and relationships become referential integrity.

The physical models can be copied between the two products.

Projects

Both RDA and DB2 Change Management Expert use Data Design Projects.

Scripts

Both RDA and DB2 Change Management Expert support SQL scripts such as DDL, DML, and DCL. DB2 Change Management Expert supports additional DB2 commands and utilities such as import, export, create database, reorg, and rebind.

If a script contains commands that DB2 Change Management Expert supports, but that RDA does not, it will have a file extension of .chx.

In summary, you can:

- Use physical data models that were created by using RDA in DB2 Change Management Expert.
- Check RDA data design projects out of library control and use them in DB2 Change Management Expert.
- Use SQL scripts generated with RDA or other ERM tools and reverse engineer them with DB2 Change Management Expert to implement databases or make changes to database schemas.

Incorporating changes made with Entity Relationship Modeling tools into DB2 Change Management Expert

You can incorporate changes that were made with Entity Relationship Modeling (ERM) tools into DB2 Change Management Expert.

To incorporate changes made with an ERMs, complete the following steps:

1. Forward engineer the DDL from the ERM tool.
2. Reverse engineer a physical data model in DB2 Change Management Expert from the DDL.
3. Create a deployment script to configure the change that you want to make.
4. Add the new ERM-based physical data model as a source model on the **Data Models** tab of the Deployment Script Editor.
5. Right-click the ERM-based physical data model and select **Migrate**.
6. Use the Comparison Editor to incorporate changes from the ERM-based physical data model into the DB2 Change Management Expert target model.

Chapter 4. DB2 Change Management Expert scenarios

The following scenarios illustrate how DB2 Change Management Expert handles processes such as copying a database, making a simple change to a database, applying a change from one database environment to another environment, and undoing changes to the catalog.

Scenario: Changing in place

Suppose that you want to make changes to two tables, EMPLOYEE and MANAGERS, in the test database. Specifically, you want to make the information in the EMPLOYEE table more comprehensible and easier to access. At the moment, the ADDRESS column in the EMPLOYEE table contains the street, city, state, and zip code where an employee lives. You want to split the information in the ADDRESS column into four separate columns: STREET, CITY, STATE, and ZIPCODE.

Also, you want to add an index, NAME_INDEX, to the MANAGERS table to help improve performance.

DB2 Change Management Expert will help you:

1. Create a Data Design project
2. Identify which database objects you want to change
3. Create a deployment script to specify the change
4. Implement those changes on a model
5. Identify which views, functions, procedures, and triggers will be rendered inoperative by the change and which ones will still be operative after the change
6. Use the deployment script to generate SQL to apply the changes
7. Preserve the data by using export and import commands
8. Deploy the changes to the test database

The following steps show you how to use DB2 Change Management Expert to change the columns in a table and to add an index to a table:

1. Create a database connection for the test database by opening the Database Explorer view, right-clicking in the view, and clicking **New Connection**. The New Connection wizard starts. Enter information about the test database, including the user ID and password. Select the version of DB2 that you want to use and specify the JDBC driver class location. Click **Test Connection** to verify that the connection has been created successfully, and then click **Finish**.

   Tip: You can also create database connections from the Deployment Script wizard when you create a deployment script.
2. Create a deployment script to handle the changes. Deployment scripts are sets of changes that you want to deploy to a database at a point in time. When you create a deployment script, the Deployment Script wizard creates the base and target models, which are representations of what the test database will look like before and after the changes, respectively. The Deployment Script wizard can also automatically create a DB2 Change Management Expert project to hold the deployment script, models, and other scripts that are related to the change if a project does not already exist.
   a. From the main menu bar, click **File** → **New** → **Deployment Script**. The Deployment Script wizard starts.

b. Enter `change_scenario` as the name of the project. If a project by that name does not already exist, DB2 Change Management Expert will create one for you.

c. Name the deployment script `inplace_deploy`.

d. Select **Use Existing Connection** and select the target database, which is the test database.

e. Select the schemas that you want the models to have.

f. **Optional:** Specify names for the base and target models. Name the base model `TEST_BASE`, and name the target model `TEST_TARGET`.

g. Click **Finish**. DB2 Change Management Expert creates the base and target models of the test database and the deployment script. The Deployment Script Editor is displayed.

The Deployment Script Editor is your central resource for managing your change. The four main areas on the Overview page of the deployment script contain the following information and actions:

- Deployment Information
- Deployment Script Contents
- Verifying
- Deploying

The tabs in the Deployment Script Editor organize the database change management process into tasks. For example, you can work with the undo commands that generated to reverse your changes from the Undo tab.

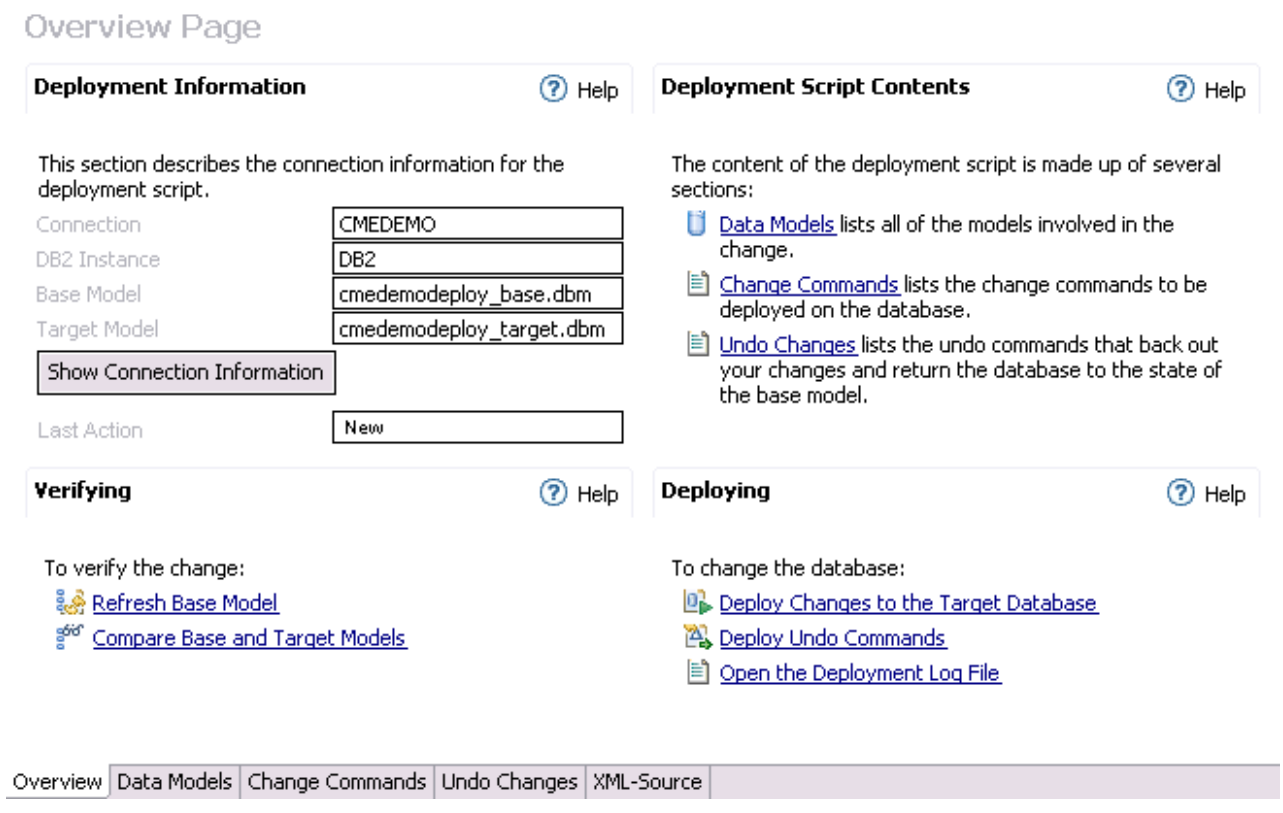

Figure 5. The Overview page of the Deployment Script Editor for inplace_deploy.deployxml

3. Add the new columns to the EMPLOYEE table in TEST_TARGET.

a. In the Deployment Script Editor, on the **Data Models** tab, click **Edit Target Model**. The Data Model Editor is displayed with information about TEST_TARGET.

   The Data Model Editor provides you with a hierarchical view of the objects in a database model. You can use the Data Model Editor to view and change the objects in a model.

b. Double-click the schema in which the EMPLOYEE table is located and select the EMPLOYEE table. You can view the columns in the EMPLOYEE table by clicking the **Columns** tab in the Properties view.

   Tip: Instead of drilling down through schema objects in the Data Model Editor, you can use **Ctrl+f** or **Main menu** → **Edit** → **Find/Replace** to search an open model.

c. Click **New** in the Properties view. Name your column `STREET`.

d. Repeat the process to create the other three columns, `CITY`, `STATE`, and `ZIPCODE`.

e. In the Properties view, select the ADDRESS column and click the **Delete** icon. DB2 Change Management Expert deletes the ADDRESS column.

4. Add the new index, NAME_INDEX, to the MANAGERS table.

   a. Find and select the MANAGERS table. Right-click the table and click **New** → **Index**. The Properties view displays the information about the new index.

   b. On the **General** tab in the Properties view, rename the index to `NAME_INDEX`.

   c. On the **Details** tab, define the columns in and the attributes for the index.

5. Save the target model.

6. Generate the change commands to apply to the target database. In the Deployment Script Editor, on the **Change Commands** tab, click **Generate Change Commands**. The Generate Change Commands wizard starts.

7. Complete the steps in the Generate Change Commands wizard. DB2 Change Management Expert automatically identifies if your changes require data preservation commands or DB2 maintenance commands and will provide the appropriate wizard pages.

   In this scenario, you are changing EMPLOYEE from a table with one column for the address information into a table with four columns for the address information. Therefore, you need export and import commands to preserve your data, and you need to map the column translation. For example:

   ```
   CREATE TABLE EMPLOYEE (NAME CHAR(26), ADDRESS CHAR(67))
   ```

   is changed to:

   ```
   CREATE TABLE EMPLOYEE (NAME CHARACTER (26), STREET CHARACTER (30),
   CITY CHARACTER (30), STATE CHARACTER (2), ZIPCODE DECIMAL(5,0))!
   ```

   a. On the DDL Generation Options page, verify that the **Generate delta DDL**, **Enable Undo**, **Enable Data Preservation** options are checked. **Enable Undo** generates the commands that are necessary to undo your changes.

   b. On the Specify File Information page, specify a data file location for the export and import commands. You must specify the full path. Specify your data file format and accept the default export and import file names.

   c. On the Mapping Tables page, there is a mapping entry that identifies the EMPLOYEE table as the table from which you will export data and then later into which you will import data. Set the Customize field for the entry to Yes so that you can later customize the SELECT statement in the export command.

d. On the Customize Export Commands page, modify the SELECT statement so that the query extracts the data for the ADDRESS column in four substrings with definitions that correspond to the four new columns for STREET, CITY, STATE, and ZIPCODE.

e. On the Customize Import Commands page, verify the mapping of the export columns to the import columns. The substrings for the ADDRESS field should map to the four new columns for STREET, CITY, STATE, and ZIPCODE.

f. On the DB2 maintenance command page, check whether you want to generate commands to preserve the authorization on objects, to rebind packages, to generate runstats, or to flush the package cache. No options are selected by default.

   If you choose to rebind packages, the schema for which the packages should be rebound are selected by default on the Select Schema page.

g. Click **Finish** to generate your change commands. You can view the change commands on the Change Commands page of the Deployment Script Editor.

8. In the Deployment Script Editor, click the **Overview** tab. Click **Deploy Changes to the Target Database**. DB2 Change Management Expert deploys the changes to the database catalog.

Your database catalog is now updated and includes the changes that you specified.

The Last Action field displays **Deployed**.

Scenario: Applying changes from one database environment to another

You are asked to promote changes that were made to a very large development database that contains a large number of tables to the formal test database. As the following figure shows, this task requires you to determine the differences between the development system and test system, and to change only those objects that are different.

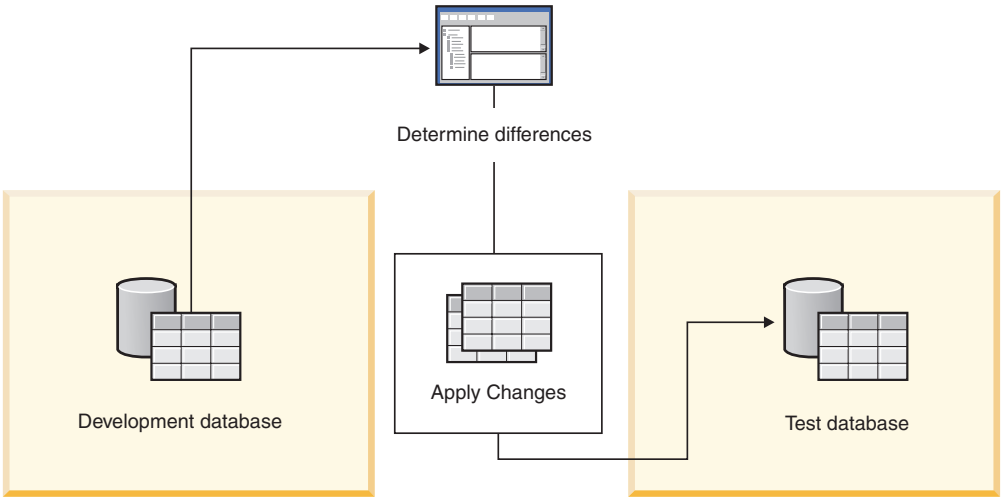

Figure 6. The process of promoting changes from one database environment to another

Specifically, the EMPLOYEE table of the development system has been updated with a new column, OFFICENO, and the DIVISION column has been removed. You must identify these changes and apply them to the test database.

The following steps show you how you can use DB2 Change Management Expert to:

1. Create a project to manage the resources for migrating the changes
2. Create models of the development and test databases, including creating a base model of the test database for fallback purposes
3. Identify the differences between the development and test database models and move the changes that are identified in the development database to the test database
4. Create the DDL to apply the changes to the test database
5. Finally, deploy the changes to the test database

1. Create a project by opening the Data Project Explorer view, right-clicking in the view, and clicking **New** → **Project**. In the New Project wizard, select **Data Design Project** and follow the steps throughout the wizard.
2. Create a database connection for the development database by opening the Database Explorer view, right-clicking in the view, and clicking **New Connection**. The New Connection wizard starts. Enter the information about the development database, including the user ID and password. Click **Test Connection** to verify that the connection has been created successfully, and click **Finish**.
3. Create a model of the development database. Models are local representations of a database at a particular point in time, and the development database will serve as the source for the changes.
   a. Click **File** → **New** → **Physical Data Model**. The New Physical Data Model wizard starts. Complete the steps in the wizard to create the model.
   b. In the **Destination folder** field, enter the name of the project that you created in step 1. In the **File name** field, type `Development`. Indicate the type and version of the development database by selecting a type from the **Database** menu and a version from the **Version** menu. Select **Create from reverse engineering**, and click **Next**.
   c. Select **Database** as the source, and click **Next**.
   d. Select **Use an existing connection**, select the connection that you created in Step 2, and click **Next**.
   e. Enter the user ID and password for the development database, and click **Next**.
   f. Select the schemas that you want to include in the model, and click **Finish**.

   A physical model of the development database is created.
4. Create a deployment script to handle the changes to the test database. When you create a deployment script, the Deployment Script wizard creates base and target models of the test database. These models are representations of what the test database will look like before and after the changes, respectively. In a later step, you will change the target model of the test database and then generate the actual DDL commands that are necessary to implement the changes.
   a. In the Data Project Explorer view, right-click the project and click **New** → **Deployment Script**. The Deployment Script wizard starts.
   b. Enter a name for the script in the **Script name** field, and click **Next**.

c. Select **Use an existing connection**, select the connection to the test database, and click **Next**.

d. Select the appropriate schemas, and click **Next**.

e. Enter `test_base` in the **Base Model Name** field and `test_target` in the **Target Model Name** field. Click **Finish**.

A deployment script and two models of the test database are created. You can use the base model version of the test database for fallback purposes, for historical analysis, or for auditing. The Deployment Script Editor is also displayed.

5. Synchronize the structure of the test database model with the development database model by finding the differences between the models and changing the target test database model.

   a. On the **Data Models** tab of the Deployment Script Editor, click **Add**.

   b. In the Source Models selection dialog, select the development database model, and click **OK** . The model is displayed in the Define Source Models field.

   c. Select the model, and click **Migrate**. The Comparison Editor is displayed with the development database model on the left and the test database target model on the right.

      Tip: Double-click on the Comparison Editor tab to maximize it. Another double-click will minimize the window.

   d. Use the down arrow in the toolbar to tab through the differences between the two models.

      In this scenario, the Comparison Editor shows that the EMPLOYEE table of the development database has been updated with a new column, OFFICENO, and the DIVISION column has been removed.

   e. Move or apply the changes to the target test database model by clicking the right arrow when a difference is selected. For example, you can select OFFICENO and click the right arrow to add the column to the test database model.

      Tip: Apply the changes in order.

   f. Click **File** → **Save all** to save your changes. Close the Comparison Editor.

6. Verify the changes that you are making to the test database by comparing the base model and target model of the test database. The base model represents the test database before changes, and the target model represents the database after the changes. From the Deployment Script Editor, click **Compare Base and Target Models** to display the Comparison Editor and view the differences between the models.

   Tip: You can also compare your models by using options outside of the Deployment Script Editor. From the Data Project Explorer view, you can select two models, right-click, and click **Compare with** → **Each other**.

7. Generate the change commands that will implement your changes. From the **Change Commands** tab of the Deployment Script Editor, click **Generate Change Commands**. The Generate Change Commands wizard starts. Complete the steps in the wizard:

   a. On the DDL Generation Options page, verify that the correct changes will be made. Ensure that **Enable Undo** is checked so that commands to undo your changes, if necessary, are generated.

b. On the Specify File Information page, specify a data file location for the export and import commands. You must specify the full path. Specify your data file format and accept the default export and import file names.

c. On the Columns tab of the Customize Import Commands page, check **Map Columns** and drag and drop the columns from the import table to map the relationship between the export columns and the import columns.

d. On the Mapping Tables page, there is a mapping entry that identifies the EMPLOYEE table as the table from which you will export data and then later into which you will import data. Ensure that the Customize field is set to **Yes** so that you can later customize the mapping of the columns for the import command. You can also override the default Load and Unload providers here.

e. On the Customize Export Commands page, click **Next** to continue. You do not need to customize the export command.

f. On the Customize Import Commands page, verify the mapping of the export columns to the import columns and correct the column mappings to adjust for the DIVISION column being dropped from the middle of the table.

g. On the DB2 Maintenance Command page, check whether you want to generate commands to preserve the authorization on objects, to rebind packages, to generate runstats, or to flush the package cache. No options are selected by default.

   If you choose to rebind packages, the schema for which the packages should be rebound are selected by default on the Select Schema page.

h. Click **Finish** to generate your change commands. You can view the change commands on the Change Commands page of the Deployment Script Editor.

8. Deploy your changes to the DB2 test database catalog. Click **Deploy Changes to the Target Database**.

   When you deploy the changes, DB2 Change Management Expert will check to see if the test database has changed since you created the base model of the test database. If the database has changed, you need to refresh the base model and regenerate the change commands.

Scenario: Undoing changes to the catalog

You deployed a deployment script against the test database. In the deployment script, TEST_BASE is the base model and TEST_TARGET is the target model. The Generate Change Command wizard generates both change commands and undo commands provided that you have enabled these two options. You can use these undo commands if you decide to undo the changes to the test database.

Undoing a change reverts the database catalog back to your base model. The base model, TEST_BASE, can now be thought of as the target model because it acts as a representation of the catalog after the undo process is completed. The target model, TEST_TARGET, can be thought of as the base model because it acts as a representation of the catalog before the undo process begins.

1. In the Data Project Explorer view, select your deployment script. The Deployment Script Editor opens.
2. Click the **Undo Changes** tab of the Deployment Script Editor. The Undo Changes page is displayed. The Define Undo Commands section lists the change commands that DB2 Change Management Expert uses to undo your changes. You can view these undo commands in a text editor by selecting the command from the list and clicking **Edit**.

3. Click **Deploy Undo Commands** to start the Undo wizard, which will deploy the undo commands and roll back your changes from the database catalog.

   Optional: You can also click Deploy Undo Commands directly from the Overview tab of the Deployment Script Editor if you do not need to review the contents of the commands. The Undo wizard starts.
4. Complete the steps in the Undo wizard and click **Finish**. DB2 Change Management Expert deploys the undo commands and reverses the changes that you previously deployed to the catalog. The database catalog is restored to the state it was in before you implemented the change.

Chapter 5. The Database Explorer and database connections

The following topics discuss how to use the Database Explorer view and how to work with your database connections.

Database Explorer

In the Database Explorer, you can connect to existing databases and view their designs and objects.

You can browse database designs and import them to the Data Project Explorer, where you can extend or modify the designs. You can also run stored procedures and user-defined functions and view the results in the Data Output view.

Using the Database Explorer, you can:

- Create a database connection
- Define connection filters
- Connect to existing databases and view their designs
- Reconnect to a database if your database connection was disconnected
- Disconnect from a database connection if you are connected
- Use saved connection information to view objects in a database even if you are not actively connected
- Create new data objects using the SQL editor
- Compare two data objects
- Deploy stored procedures or user-defined functions to the same database or to a different database
- Analyze the impact and dependency of data objects
- Analyze a data model to ensure model integrity
- Import database designs to a Data Design project in the Data Project Explorer, where you can extend or modify the designs
- Copy stored procedures or user-defined functions into a Data Development project, where you can modify and deploy the objects
- Refresh the database objects that are defined for a database connection
- Delete a database connection
- Drag and drop or copy database objects to the Database Explorer
- Generate DDL from a database, schema, or data object, and select the model objects that you want to include in the DDL script

Data object filters

You can create, view, change, and remove filters.

You can use filters to exclude data objects (such as tables, schemas, stored procedures and user-defined functions) from your view. Only the data objects that match the filter condition are shown.

You can filter data objects and JDBC connections.

A data object filter excludes data objects from the tree view of the database. These filters can be set or removed at any time. You can apply data object filters to your views.

JDBC connection filters are set when you create a connection to a database by using the New Connection wizard. These filters can be edited for an existing connection.

Tip: Connections to a remote server for all database vendors can take a considerable amount of time to establish. To reduce the time for remote connections, configure appropriate connection filters by excluding parts of the database that are not required.

Creating a data object filter

You can create a data object filter at any time. A data object filter excludes data objects (such as, tables, schemas, stored procedures and user-defined functions) from the Database Explorer view.

You can use one or more percent sign (%) characters for a wildcard search character in the data object filter. The asterisk (*) wildcard character is not supported.

To create a data object filter:

1. In the Database Explorer, right-click a schema or a folder under the schema and select **Filter** to open the Filter window.
2. Specify filtering options. You can specify a filtering expression or a list of data objects to either include or exclude.
3. Click **OK**.

Viewing and changing a data object filter

You can view and change a data object filter to modify the filtering conditions for data objects.

To view and change a data object filter:

1. In the Database Explorer, right-click a data object folder and select **Filter** to open the Filter window.
2. Make the appropriate filter changes, and then click **OK** to close the window and save your changes.

Removing a data object filter

Remove a data object filter so that you can see filtered objects in the tree view.

To remove a data object filter:

1. In the Database Explorer, right-click a data object folder and select **Filter** to open the Filter window.
2. Click **Disable filter**, and then click **OK** to close the window and save your changes.

Overview of database connections

Use the New Database Connection wizard or pages that are embedded in other wizards to create a JDBC connection to a database.

Every database vendor has its own list of available JDBC drivers. To use an unsupported JDBC driver, in the New Database Connection wizard or the Database Connection page of another wizard, select the appropriate database vendor, and then select **Other Driver** from the **JDBC driver** pull-down list. To connect by using this option, you must enter the name of the JDBC driver class, the location of the class, and the correct form of the connection URL. Because unsupported drivers are not tested, they are not recommended.

The following restrictions apply to database connections in the workbench.

- The New Database Connection wizard and the pages that are embedded in other wizards do not create connections that can be used at run time.
- If you create a connection to a database and that connection is cancelled in some way from outside this product (for example, in DB2 Universal Database™ by using the `force applications all` command), the workbench cannot detect that the connection has been dropped.

Connecting to a database by using a wizard

You can browse existing data objects by connecting to a database in the Database Explorer. This topic explains how to open the New Database Connection wizard from the Database Explorer view. The New Database Connection wizard pages can also be embedded in other wizards in the workbench.

To connect to a database by using a wizard:

1. Right-click in the Database Explorer, and select **New Connection** from the pop-up menu.
2. On the first page of the wizard, select a database manager, a JDBC driver, and specify other connection details. See one of the related topics for specific information about the database that you want to connect to.
3. Optional: On the **Limit the objects retrieved from the database** page, specify filtering options. For best performance, you should use filters when you are connecting to a large database.
4. Complete all other wizard steps and click **Finish**.

The connection is displayed in the Database Explorer.

Connecting to a specific database

DB2 Universal Database

License requirements for all DB2 Universal Database database connections:

License files are required to provide connectivity to DB2 Universal Database databases.

DB2 connection license requirements

The following table describes the license requirements for connecting to DB2 databases.

License file	Description
db2jcc_license_cu.jar	Permits JDBC/SQLJ connectivity to all DB2 Universal Database for Linux, UNIX, and Windows servers. This is the standard license provided with DB2 Universal Database for Linux, UNIX, and Windows. db2jcc_license_cu.jar is included with all of the editions of DB2 Universal Database for Linux, UNIX, and Windows.
db2jcc_license_cisuz.jar	Permits JDBC/SQLJ connectivity to all DB2 servers, including: DB2 Universal Database for z/OS®; DB2 Universal Database for iSeries™; DB2 Universal Database for Linux, UNIX, and Windows. This license is provided to DB2 Connect™ licensees only. db2jcc_license_cisuz.jar is included with all DB2 Connect editions.

Support and restrictions for DB2 Universal Database for Linux, UNIX, and Windows database connections:

Use the New Database Connection wizard or pages that are embedded in other wizards to create a JDBC connection to DB2 Universal Database for Linux, UNIX, and Windows.

The DB2 Universal Driver is supported for DB2 Universal Database for Linux, UNIX, and Windows.

DB2 JDBC Drivers

You can download the DB2 JDBC drivers at www.ibm.com/software/data/db2/.The DB2 Universal Driver is included with the DB2 Application Development Client. This driver is a two tier pure Java JDBC driver that allows a Java client to communicate directly with DB2 UDB servers through the DRDA® protocol. This driver supports both Type 2 connections (to databases that are cataloged on your local computer) and Type 4 connections (using a host, database name, and port). This driver is designed to replace the Type 3 Net driver.

Troubleshooting database connection problems

This topic lists known database connection problems and solutions.

A connection appears to be active but is not

If a database connection is canceled from outside of the workbench (for example, in DB2 Universal Database by using the FORCE APPLICATIONS ALL command), the workbench cannot detect that the connection is dropped.

Some tasks you perform in this product require a live connection to the database. Any task that requires an active connection to the database (such as running an SQL statement in the SQL query builder) will not work properly if the connection is

dropped. If a connection appears to be active in the workbench but is not active, first try to disconnect from the database and then reconnect.

If the connection is still not active, then follow these steps:

1. Disconnect from the database.
2. Restart the database server.
3. Reconnect to the database.

If the database connection is still not active, then shut down the workbench and restart it.

Connection to the database server fails

If the connection to the database server fails, you might receive an error message similar to this one: `IWAS0126E Problems encountered while trying to establish connection &1. Reason: '&2'`. For more information about the reason code (&2), refer to the documentation that comes with your database server.

The application requester cannot establish the connection

For DB2 Universal Database for iSeries, the Toolbox JDBC driver uses the system name, user ID, and password provided in the call to `DriverManager.getConnection()` to establish a connection to the iSeries database. If any of these values are not provided, the Toolbox JDBC driver displays a sign-on prompt. If the Toolbox JDBC driver cannot sign on to the system for any reason, you will see this message: `The application requester cannot establish the connection`. Here is a list of potential causes for this error:

- The system name is incorrect. Ping the specified system name to verify the name. If the ping fails, then there is a TCP/IP configuration problem between your client and the server.
- The user ID or password is incorrect.
- The iSeries database host server is not started. To start the host server, run the following command on a CL command line: `STRHOSTSVR *DATABASE`

  You can verify that the iSeries database host server is running by running the NETSTAT *CNN command. When the Work with TCP/IP Connection Status screen is displayed, look for the as-database entry under the **Local Port** heading and ensure that the entry is in Listen status.

Managing database connections

You can refresh, reconnect, disconnect, and delete your database connection. You can also view the properties of a database connection.

Reconnecting to a database

You can reconnect to a database if the connection is defined but not connected.

If a database connection is canceled from outside of the workbench (for example, in DB2 Universal Database by using the FORCE APPLICATIONS ALL command), the workbench cannot detect that the connection is dropped.

To reconnect to a database:

1. In the Database Explorer, right-click on the database connection that you want to reconnect to.
2. Click **Reconnect**.

Viewing the properties of a database connection

You can view the properties of a database connection.

To view the properties of a database connection:

1. Select a database connection in the Database Explorer.
2. View the properties for the connection in the Properties view.

Editing a database connection

You can edit a database connection in the Database Explorer view.

To edit a database connection:

1. In the Database Explorer, right-click on the database connection that you want to edit.
2. Select **Edit Connection**. The Edit Connection wizard opens with the current connection information displayed in each field.
3. Modify the connection and filter information in the wizard, and click **Finish**.

The connection is modified.

Refreshing the catalog metadata for a database connection

You can refresh the catalog metadata if a connection is defined and you are connected. If other users have modified the metadata, you can see the latest changes.

To refresh the catalog metadata for a database connection:

1. In the Database Explorer, right-click on the database connection that you want to refresh.
2. Select **Refresh**.

Disconnecting from a database

You can manually disconnect from a database.

When you close this product, you are automatically disconnected from all databases, but the connection information is saved. You can restore the connection the next time that you use this product by selecting **Reconnect** from the connection's pop-up menu.

To disconnect from the database:

1. In the Database Explorer, right-click on the database connection that you want to disconnect.
2. Select **Disconnect**.

Deleting a database connection

You can delete a database connection from the Database Explorer view.

To delete a database connection:

1. In the Database Explorer, right-click on the database connection that you want to delete.
2. Select **Delete**.

Hints for using the Database Explorer

The Database Explorer is automatically populated with aliases, for example, from the local DB2 instance. Connections that you define are stored in the Eclipse workspace and not in your project.

These behaviors have the following implications:

- Your workspaces might not always be portable between DB2 Change Management Expert versions and Fix Packs. If the workspaces are not portable, you can import the project that you want to move into the new workspace and then redefine your connections.
- The connections in the Database Explorer are referenced by an alias. That way different CONNECT statements can uniquely identify databases with the same name that are on different instances. For example, the name that is displayed in the Database Explorer might not necessarily be the same as the database name.
- Connection names in the Database Explorer need not be the same in different workspaces amongst different users. If there are CONNECT TO statements in the change command (.chx) files, the connection being referenced will be the one that is displayed in the Database Explorer. Take care to use the same connection aliases in the Database Explorer when you share scripts between multiple users or workspaces. This ensures that all users are deploying to the correct target environment.

Chapter 6. Data Design Projects

A Data Design Project is a container for storing the resources that you will use to plan, analyze, archive, and make changes to your databases. Projects are displayed in the Data Project Explorer view, and they correspond to directories in the file system.

The resources that you will use to make database changes include models and scripts. When you create a project, the following virtual subfolders are displayed within the project to organize the following resources:

Data Diagrams folder
Holds the diagram files for your base and target models.

Data Models folder
Holds the models. Models have a file extension of .dbm.

SQL Scripts folder
Holds the types of script files that are shown in the following table:

Table 7. Types of scripts in the SQL scripts folder

Type of script	File extension	Description
Change command script	chx	Can contain SQL statements, DB2 commands, and utility invocations. These scripts can be used to make or undo changes to a database, such as change the structure of a database (DDL), change the data in the database (EXPORT, IMPORT, and DML), modify the database packages (REBIND), or run statistics.
Deployment script	deployxml	Contains a set of changes to make to a database. The deployment script brings together the specification resources (models), change commands, and metadata about the change and organizes them into a coherent form.
SQL script	sql ddl	Contains SQL statements.

You can select **Run SQL** by right-clicking a script in the Data Project Explorer. The run options that are available, depend on the file extension of the script. The following list describes the available options:

- The **Run SQL** option is available for scripts that contain commands that are specific to DB2 Change Management Expert. Those scripts will have a .chx extension.
- The **Run SQL** option never appears for deployment scripts with a .deployxml file extension, as those commands are issued through the Deploy wizard.
- The **Run SQL** option is available for other files in the scripts folder that only contain commands that can be deployed through a JDBC. For example, DDL, DCL, and DML commands. These scrpits typically have a .ddl or .sql file extension.

Other Files folder
Holds the files that do not belong in either the SQL Scripts or Data Models folders. For example, the deployment log files with a file extension .log, are placed in this folder.

The following figure shows an example of a project with its resources:

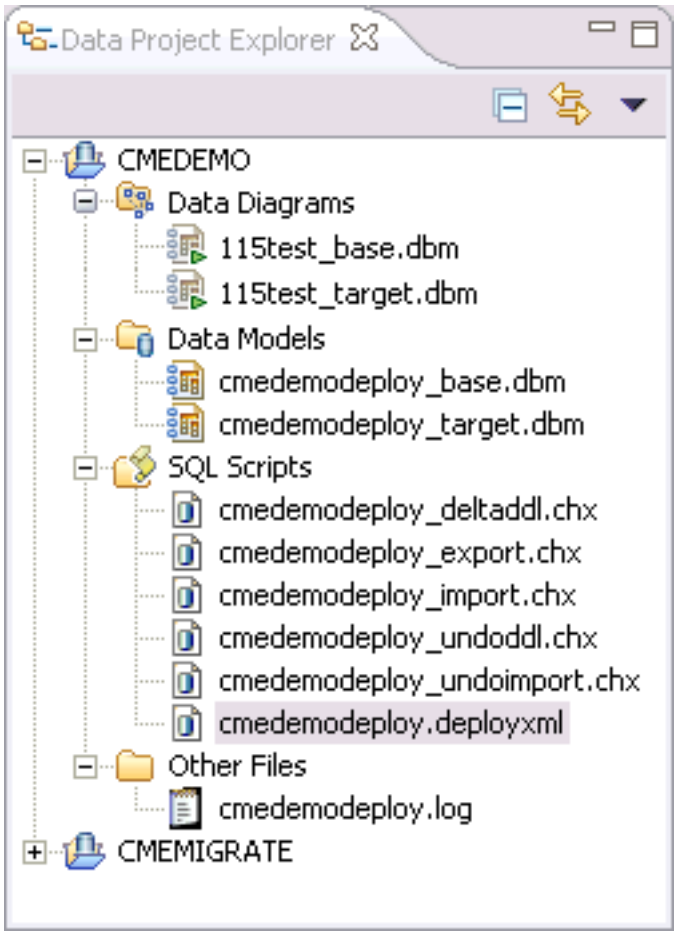

Figure 7. Example of a project in the Data Project Explorer view

A project must exist before you can use any of the other functionality in DB2 Change Management Expert. Some functions in DB2 Change Management Expert will create a project if one does not already exist. For example, when you create a deployment script, you can specify that a project be created for you. You can also create a project explicitly.

Creating a project

You can create a project explicitly or have DB2 Change Management Expert create one automatically when you create a deployment script.

To explicitly create a project, complete the following steps:

1. From the main menu bar, click **File** → **New** → **Project**. In the **New Project** wizard, select **Data Design Project** and follow the steps throughout the wizard to create your project.

You can also create a Data Design Project by right-clicking in the Data Project Explorer view when it is empty or right-click an existing project, and click **New** → **Project**.

2. In the **Project name** field, type the name of your project.
3. Leave the box checked to use the default location for your new project. Click **Finish** when you are done. Your new project is displayed in the Data Project Explorer view with a folder for data diagrams, data models, a folder for SQL scripts, and a folder for other files.

A project can also be created when you create a deployment script. The Deployment Script wizard will prompt you to specify the name of the project in which to create the deployment script. If the specified project does not exist, it will be created for you.

You can now create the resources that you need to plan, analyze, and make changes to your databases.

Managing project resources

The resources in a project, as well as the project itself, can be copied, renamed, moved, and deleted by using pop-up menu operations. In addition, you can use Eclipse team to store and retrieve projects and project resources in a library control system so that multiple users can share the resources and use version control.

Copying project resources

You can copy the resources in a project.

To copy a resource, complete the following steps:

1. In the Data Project Explorer view, right-click the resource to copy, such as a model in the Data Models folder.
2. Click **Copy**.
3. Right-click the project into which to copy the resource, and click **Paste**.
4. If you are copying the resource into the same project, type in a new name for the resource, and press Enter.

You can use the same technique to copy a project.

Renaming project resources

You can rename the resources in a project.

To rename a resource, complete the following steps:

1. In the Data Project Explorer view, right-click the resource to rename, such as a model in the Data Models folder.
2. Click **Rename**. The name of the resource is overlaid with a text field.
3. Type in the new name for the resource, and press Enter.

You can use the same technique to rename a project.

Moving project resources

You can move resources between projects.

To move a resource from one project to another, complete the following steps:

1. In the Data Project Explorer view, right-click the resource to move, such as a model in the Data Models folder.
2. Click **Move**. You are prompted to select a folder destination
3. Select the project to which to move the resource.

Deleting project resources

You can delete resources from a project.

To delete a resource, complete the following steps:

1. In the Data Project Explorer view, right-click the resource to delete, such as a model in the Data Models folder
2. Click **Delete**.
3. Click **Yes** to confirm that you want to delete the resource.

You can use the same technique to delete a project.

Library control systems

Library control systems let you share your projects and project resources with other users and provide you with a mechanism for version control. You can use Eclipse team to manage your projects in a library control system.

You can invoke the Eclipse team menu by right-clicking a project and clicking **Team** on the pop-up menu. For more information about using Eclipse team, see the topics *Team CVS tutorial* and *Working in the team environment with CVS* within the Workbench User Guide.

Chapter 7. Change specification

Change specification is the process of defining the changes that you want to make to a database.

With DB2 Change Management Expert, you create models of the database that you want to change, and use those models to define what the database should look like before and after the changes and to generate the DDL to implement the changes. Before deploying the changes to the actual database, you also create commands to define how the data should be preserved in or migrated from the database when there are structural changes and to rebind the packages that might be affected by the changes.

As you change a model to define the set of changes that you want to make, you might want to compare the model to another model and move differences, validate the model definition, determine the impact that the changes might have on other objects in the database, and verify your changes by comparing the model of the database after changes to a model of the database before changes.

DB2 Change Management Expert provides two approaches to change specification. You can use:

- The Deployment Script Editor, which is the primary approach and the recommended way to get started. The Deployment Script Editor presents a consolidated way to work with the resources for the changes that you want to deploy. With the Deployment Script Editor, you can use the Overview page to specify the change specification resources, the Data Models page to define structural changes, the Change Commands page to generate the DDL for the changes and deploy them to the database, and the Undo Changes page to back out changes after they are deployed.
- The actions that are available from the Data Project Explorer view. As change specification resources are created, they appear in database change projects in the Data Project Explorer. You can use this view and the actions that are available to work with change specification resources.

Each approach to change specification provides some functionality that is not provided by the other. Therefore, although the Deployment Script Editor is the primary approach, you will often use both approaches together to plan, analyze, and deploy your changes.

Models

A *model* is a representation of a database catalog that you can use to make changes to the catalog. A model can be stored in memory or in the local file system.

Models make changing a database easy because you can make changes without having to remember complex SQL. In addition, DB2 Change Management Expert provides a variety of user interfaces, depending on the task.

Instead of making changes directly to a DB2 catalog, you make changes to models. You use models to plan changes to a database, analyze the impact of the changes, and then deploy those changes to the actual database. You can also use models to identify the differences between databases and move those differences from one database to another database.

DB2 Change Management Expert uses several types of models:

- A *model version* represents a DB2 database at a particular point in time. Model versions are important for auditing because the DB2 catalog does not keep a history of changes.
- A *base model* is a special type of version model, specifically the version model on which a change is based. A base model represents a DB2 database before any changes are applied.
- A *target model* represents a DB2 database after the changes are applied.
- A *source* is a model that serves as the basis of change for another model.

As an example of how models are used, assume that you want to increase the length of a column and drop a column in a particular table in a development database. You can create a deployment script to handle the change. When you create a deployment script, the Deployment Script wizard creates a base model and a target model of the development database. You can then change the target model to increase the length of a column, to drop a column, and to generate the change commands to create the DDL that you will use to change the actual development database. The change commands will be based on the differences between the base and target models (the before and after representations of the development database).

As another example, assume that you made some changes to the development database and now want to migrate those changes to the test database. You can create a model of the development database, which will be the source model for your changes to the test database. You can then create a deployment script to handle the changes to the test database, which also causes a base and target model of the test database to be created. You can then compare the development database model (the source) to the target model of the test database and move the changes from the development database model to the target test database model. Finally, you can generate the change commands to create the DDL that will change the actual test database. These change commands will be based on the differences between the base model of the test database and the target model of the test database, which you changed by moving over differences for the development database).

Methods of creating a model

There are several ways to create a model. Models let you plan database changes and analyze the impact of those changes before deploying the changes to the database.

When you complete the steps in the Deployment Script wizard to create a deployment script, DB2 Change Management Expert automatically creates a base model and a target model of the database that you plan to change.

In addition, you can use any of the following methods to create a model manually:

- Creating a model from a DB2 catalog
- Creating a model from a SQL script
- Creating an empty model
- Copying an existing model

Creating a model from a DB2 catalog

You can create a model by extracting information from a DB2 catalog and importing the information into DB2 Change Management Expert.

Prerequisites: You must have a project in the Data Project Explorer view before creating a model.

To create a model from a DB2 catalog, complete the following steps:

1. From the main menu bar, click **File** → **New** → **Physical Data Model**. The New Physical Data Model wizard starts.

   You can also invoke the New Physical Data Model wizard by using either of the following methods:

   - Click the down arrow on the **New** button on the main menu bar, and click **Physical Data Model**.
   - Right-click a project in the Data Project Explorer view, and click **New** → **Physical Data Model**.
2. Complete the steps in the New Physical Data Model wizard to create a new model:
   a. Select the destination folder (or project) into which to place the model.
   b. Choose to create the model from reverse engineering with database as the source.
   c. Select the schemas that you want to include so that you can narrow the scope of the model that you want to work with.
   d. Specify the database elements to include in the model.
   e. Evaluate whether you want to select the options to infer implicit relationships or infer implicit primary keys on the last page of the wizard. In many cases, you do not want the model to include the implicit objects that were created by DB2.

The model is added to your project, and it is displayed in the navigation tree. You are now ready to work with the model.

Creating a model from an SQL script

You can create a model by parsing an SQL script, which generates the schema definitions for the model.

Prerequisites: You must have an existing SQL script and a project in the Data Project Explorer view before creating a model.

To create a model from an SQL script, complete the following steps:

1. From the main menu bar, click **File** → **New** → **Physical Data Model**. The New Physical Data Model wizard starts.

   You can also invoke the New Physical Data Model wizard by using either of the following methods:

   - Click the down arrow on the **New** button on the main menu bar, and click **Physical Data Model**.
   - Right-click a project in the Data Project Explorer view, and click **New** → **Physical Data Model**.
2. Complete the steps in the New Physical Data Model wizard to create a new model:
   a. Select the destination folder (or project) into which to place the model.
   b. Choose to create the model from reverse engineering with DDL script as the source.

   Restriction: The following restrictions apply when you create a model from a SQL script:

- If the SQL script creates a table with a primary key, a unique index is not automatically created for the primary key. Therefore, if you compare a model that was created from the DB2 catalog with a model that was created from equivalent DDL, the Comparison Editor will detect that the model from the catalog has a unique index for the primary key and the model from the SQL script does not.
- If the SQL script creates an alias for a table, the script must also include a CREATE statement for the table on which the alias is defined. Otherwise, because no information for the table columns exists, the columns for the alias cannot be created.
- The SQL script must not include a CREATE statement for table space SYSCATSPACE, which is a system-generated table space. If the DDL script contains ALTER statements for table space SYSCATSPACE, DB2 Change Management Expert creates a table space named SYSCATSPACE in the model, sets the attributes with the values that are specified in the statement, and uses the default value with which the table space is created for the attributes that are not specified. Thus, the values in the model of SYSCATSPACE that is created from the SQL script might not match the values in the actual database.

The model is added to your project, and it is displayed in the navigation tree. You are now ready to work with the model.

Creating an empty model

You can create a model that is empty (or contains no schemas).

Prerequisites: You must have a project in the Data Project Explorer view before creating an empty model.

To create an empty model, complete the following steps:

1. From the main menu bar, click **File** → **New** → **Physical Data Model**. The New Physical Data Model wizard starts.

   You can also invoke the New Physical Data Model wizard by using either of the following methods:

   - Click the down arrow on the **New** button on the main menu bar, and click **Physical Data Model**.
   - Right-click a project in the Data Project Explorer view, and click **New** → **Physical Data Model**.
2. Complete the steps in New Physical Data Model wizard to create a new model:
   a. Choose to create the model from a template
   b. Select Empty Physical Model Template as the template.

The model is added to your project, and it is displayed in the navigation tree. You are now ready to work with the model. For example, you can edit the model or apply an SQL script to populate the model.

Methods of changing a model

One of the key aspects of the change specification phase is to make changes to a model and assess the impact of those changes before deploying the changes to the actual database. You can edit models directly or you can apply scripts to make changes.

You can use any of the following methods to change a model:

- Changing the model with the Data Model Editor
- Changing the model in the Data Project Explorer view
- Changing the diagram of a model
- Comparing two models and moving the differences from one model to the other
- Applying changes from another deployment script to a target model
- Applying change commands

Changing a model with the Data Model Editor

You can use the Data Model Editor to change a model. For example, to accommodate new application requirements, you might want to add a new table, change a column in a table, or redefine the partitioning key of a table. The Data Model Editor displays the objects in a model and allows you to edit the objects one at a time, including related objects.

Prerequisites: A model of the object that you want to change must exist.

To edit a model with the Data Model Editor, complete the following steps:

1. In the Data Project Explorer view, double-click the model that you want to edit. The Data Model Editor opens.

   You can also invoke the Data Model Editor from the Deployment Script Editor. You can only modify the target model. The base model is read-only.

   To open the target model in the Data Model Editor, click the **Edit Target Model** link on the Data Models page.

   The Data Model Editor has the following characteristics:

   - The left-hand side of the editor displays a Relationships section and the right-hand side of the editor displays a Dependents section. You can type text in the filter fields to filter the lists and quickly move to an object.
   - The Relationships section of the editor shows the objects, by category, that are related to the object that is currently being edited. Each relationship category can be expanded and collapsed.
   - Each object in the Dependents section represents an object in the database. The Data Model Editor lets you edit one object at a time.
   - As you drill down into the database, a breadcrumb trail is displayed.
   - You single-click an object to select it. You can select multiple objects, for example, to copy or delete a group of objects. The currently selected object is highlighted.

     You double-click an object to make it the object that is currently being edited.

     When you click or double-click an object, it becomes the current object in the Properties view.
   - Global edit actions are available from the Edit menu on the main menu bar, and search actions are available from the Search menu.

     You can find and replace objects names in the Data Model Editor by clicking **Edit** → **Find/Replace**. You can click **Find** to cycle the Editor through the matching objects.

You can use the search capability to find objects across models by selecting **Search** and then clicking the **Data Model Search** tab. When you search for the name of an object, the search results are shown in a Search view. You can pin the Search view to the Data Model Editor and then use the up and down arrows in the Search view to cycle the editor through the matching objects.

- Context-sensitive actions are available by right-clicking an object. When you right-click, a pop-up menu with available actions is displayed. You use the pop-up menu to add new objects to the model.
- Tools are available from the main tool bar. For example, you can expand or collapse all the Relationship categories at once, sort the Relationship categories and objects alphabetically, and filter the content that is displayed.
- You can drag and drop objects between the Data Model Editor and other views.
- You can bookmark objects in the Data Model Editor to facilitate quick access to critical objects through the Bookmarks view.

2. Use the functionality in the Data Model Editor, the Properties view, and the Outline view to make changes to the model. You can also edit the SQL for the objects to change them.

   You can right-click on objects in the Data Model Editor and select **Edit** → **DDL** from the context sensitive menu. This action will generate DDL to create the entire data model. You can change the model by modifying the DDL and then view the changes by comparing your base and target models. The Generate Change Commands wizard enables you to redefine the database object by using CREATE statements. DB2 Change Management Expert will redefine your model according to the modified script.

   Using the Generate Change Commands wizard relieves you of having to keep up with all the new ALTER options. You can focus on the CREATE statements. By using the Generate Change Commands wizard, DB2 Change Management Expert generates the ALTER statements where appropriate. This wizard gives you the benefits of specifying table definitions using DDL, but eliminates the need to remember when to use ALTER and when a DROP or CREATE statement is required.

   The Properties view facilitates editing the attributes of an object. The attributes that are displayed in the Properties view are context sensitive and depend on the type of object that is selected. Changes that are made through the Properties view are reflected immediately in the Data Model Editor, the Outline view, and the Data Project Explorer.

   An Outline view is provided for the Data Model Editor. Context-sensitive actions are available by right-clicking an object in the Outline view. Objects can also be dragged and dropped. You can synch the Outline view with the editor so that the current objects in both are the same.

   Annotations, markers, and decorators are displayed in the editor, the Properties view, and the Outline view to indicate errors.

3. Click **File** → **Save** or press Ctrl + S to save your changes. An asterisk in the tab of the Data Model Editor indicates that a model has unsaved changes. The asterisk disappears when changes are saved.

Example

Assume that you have a model of your HR database and you want add a new index to the EMPLOYEE table and change the data type of the EMPNO column in the table. You can use these steps to make the changes:

1. In the Data Project Explorer view, double-click your model of the HR database. HR should be displayed in the Data Model Editor.
2. Double-click HR under Schema.
3. To add the index:
   a. Right-click the EMPLOYEE table in the Data Model Editor, and click **New** → **Index**. An index object is created under the list of Index relationships.
   b. Select the new index object and go to the Properties view.
   c. In the Properties view, use the **General** tab to rename the index to a meaningful name and the **Details** tab to define the attributes of and columns for the index.
4. To change the data type of the EMPNO column:
   a. Double-click the EMPLOYEE table in the Data Model Editor.
   b. Select the EMPNO column and go to the Properties view.
   c. In the Properties view, use the **Type** tab to change the data type of the EMPNO.
5. Click **File** → **Save** to save the changes that you made.

Restriction: The Outline view fails if the currently selected object is deleted from Edit DDL.

Adding a table, view, or other new object:

You can use the Data Model Editor to add tables, views, aliases, structured types, distinct types, user-defined functions, procedures, views, sequences, and materialized query tables to a model of a database.

To add a table, view, or other new object to the model of a database, complete the following steps:

1. Open the model of the database in the Data Model Editor.
2. Right-click the schema object to be changed, select **New** and select the object to add from the pop-up menu. A new object is added to the model, and it is displayed under the appropriate Relationship category in the Data Model Editor.

You can now view the added object in the Properties view and then select various options to rename and define the added element.

Adding a column, index, foreign key, constraint, or trigger:

You can use the Data Model Editor to add columns, indexes, foreign keys, constraints, and triggers to the model of a table.

To add a column, index, foreign key, constraint, or trigger to a table, complete the following steps:

1. Open the model that contains the table in the Data Model Editor and find the table to be changed.
2. Right-click the table to be changed, select **New** and then select the element to add from the menu. A new element is added to the table, and it is displayed under the appropriate Relationship category in the Data Model Editor.

You can now view the added element in the Properties view and then select various options to rename and define the added element.

Deleting objects:

You can delete objects from a model using the Data Model Editor.

To delete an object in the Data Model Editor, right-click the object to be deleted, and click **Delete**.

Changing model object properties:

You can change the properties of objects in the Data Model Editor by using the Properties View. For example, you can use the Properties view to change the name and attributes of a column in a table or to reorder the columns of an index.

To change the properties of an object, complete the following steps:

1. With the model open in the Data Model Editor, right-click the object whose properties you want to change, and click **Show Properties**. Alternatively, you can double-click the object and go to the Properties view.
2. Use the tabs in the Properties view to change the properties that are associated with the object. The following table shows the properties that you can change:

Table 8. Object properties that can be changed

Object	Tab option	Description
Schema Buffer pool Partition group Alias	General	You can edit the name, label, and various other attributes of the object.
	Documentation	You can provide documentation.
	Annotation	You can add or delete properties and edit their values.
Table space	General	You can edit the name, label, and various other attributes of the object.
	Tables	You can specify the tables.
	Documentation	You can provide documentation.
	Annotation	You can add or delete properties and edit their values.
Table	General	You can edit the name, label, and various other attributes of the object.
	Columns	You can add or delete columns, reorder the columns by using the up and down arrows, rename columns, select the primary key, select the domain, select the data type, edit the length, specify scale, select not null, select generated, and specify default value/generate expression.
	Partition Key	You can specify how the key is to be maintained and edit the columns for the key, including the order of the columns.
	Table spaces	You can select the table space to which the table belongs.
	Documentation	You can provide documentation.
	Annotation	You can add or delete properties and edit their values.

Table 8. Object properties that can be changed (continued)

Object	Tab option	Description
Columns (in a table)	General	You can edit the name and label.
	Type	You can select the data type and properties of the column (primary key, null or not, and identity).
	Documentation	You can provide documentation.
	Annotation	You can add or delete properties and edit their values.
Index	General	You can edit the name, label, and schema.
	Details	You can specify whether the index is unique or clustered, add or delete key or include columns, and change the order of the index columns.
	Documentation	You can provide documentation.
	Annotation	You can add or delete properties and edit their values.
Primary key	General	You can edit the name, label, and various other attributes of the object.
	Columns	You can add or delete columns in the key and reorder the columns by using the up and down arrows.
	Documentation	You can provide documentation.
	Annotation	You can add or delete properties and edit their values.
Foreign key	General	You can edit the name, label, and various other attributes of the object.
	Details	You can add, delete, or reorder columns in the foreign key.
	Referential integrity	You can change update and delete preferences.
	Documentation	You can provide documentation.
	Annotation	You can add or delete properties and edit their values.
Materialized query table	General	You can edit the name, label, and the attributes for how the table is to be refreshed and maintained.
	Columns	You can view the columns and their data types.
	SQL	You can edit, and validate the SQL expression that is used to define the materialized query table.
	Documentation	You can provide documentation.
	Annotation	You can add or delete properties and edit their values.
View	General	You can edit the name, label, and various other attributes of the object.
	Columns	You can view the columns and their data types.
	SQL	You can edit and validate the SQL expression that is used to define the view.
	Documentation	You can provide documentation.
	Annotation	You can add or delete properties and edit their values.

Table 8. Object properties that can be changed (continued)

Object	Tab option	Description
Sequence	General	You can edit the name, label, and data type.
	Type	You can specify the start, increment, minimum and maximum values of the sequence, and whether the sequence is to cycle.
	Documentation	You can provide documentation.
	Annotation	You can add or delete properties and edit their values.
Stored Procedure	General	You can edit the name, label, result set, language, parameter style, and external name.
	Parameters	You can change the parameters for the stored procedure.
	Source	For an SQL procedure, you can change the procedure on which the stored procedure is sourced.
	Documentation	You can provide documentation.
	Annotation	You can add or delete properties and edit their values.
User-defined function	General	You can edit the name, label, language, parameter style, and external name.
	Return Type	You can edit the data type of the returned value.
	Parameters	You can change the parameters for the function
	Source	For an SQL procedure, you can change the procedure on which the stored procedure is sourced.
	Documentation	You can provide documentation.
	Annotation	You can add or delete properties and edit their values.
User-defined type	General	You can edit the name and label. For user-defined data types, you can also specify the data type, including precision and scale where applicable.
	Custom data types	You can add, delete, and order custom data types. The tab is not applicable for user-defined data types.
	Documentation	You can provide documentation.
	Annotation	You can add or delete properties and edit their values.

3. Click **File** → **Save** or press Ctrl + S to save your changes.

Changing a model in the Data Project Explorer view

You can change a model in the Data Project Explorer view by using context-sensitive menus.

To change a model in the Data Project Explorer view, complete the following steps:

1. Expand the Data Models folder in your project and right-click the object that you would like to change.
2. In the context-sensitive menu that appears, select an action for the object.

Changing the diagram of a model

You can use diagrams to visualize and edit a model.

Diagrams can represent all of a model or only elements that you select. You can use diagrams to edit a model or you can choose to edit only the diagram. You can format and annotate the diagram to provide additional information and then print or save an image file of the diagram.

Creating a physical data model diagram:

You can create a diagram of a physical data model to visualize or edit the model.

You must create or open a data design project and a physical data model before you can create a physical data model diagram.

Physical data model diagrams can use Information Engineering (IE) or Unified Modeling Language (UML) notation. You can change the default notation style for physical data model diagrams in the preferences. You cannot change the notation style after you have created the diagram.

To create a physical data model diagram:

1. Expand a physical data model project in the Data Project Explorer.
2. Right-click the Diagrams folder and choose one of the following options:
   - To create a diagram that automatically includes some or all elements of the physical data model, click **New Overview Diagram**. Then select elements to be included and click **OK**.
   - To create an empty diagram, click **New Blank Diagram**. You can then use the palette to add elements or click and drag objects from the Data Project Explorer onto the diagram.

Adding or deleting objects from a physical data model diagram:

You can use the diagram palette, the Properties view, and context menus to add or delete objects from a physical data model diagram. You can also copy and paste an existing data object to create a new data object with the same properties.

Adding a table or a view:

You can add tables or views to physical data model diagrams.

To add a table or a view to a physical data model diagram:

1. Complete one of the following steps:
   - To add a table, click **Table** in the palette.
   - To add a view, click **View** in the palette.
2. Click the location in the diagram where you want to add the table or the view.

Deleting a data object from a diagram:

You can use the context menus in the diagram editor to delete data objects from the model or from the diagram.

Data objects are part of the underlying data model that a diagram is based on. These objects are on the diagram palette in the **Data** compartment. When you delete a data object, you can either delete the object from both the diagram and the underlying model, or you can delete the object from the diagram and retain the object in the underlying data model.

In contrast, an annotation object (for example, a note) is not part of the underlying data model. When you delete an annotation object, you can only delete the object from the diagram.

To delete a data object from a diagram:

Right-click the data object to be deleted and complete one of the following options:

- To delete the data object from the model, click **Delete from Model**. This option deletes the object both from the diagram and the underlying model.
- To delete the data object from the diagram, click **Delete from Diagram**. This option removes the object from the diagram but retains the object in the underlying data model.

Adding keys, columns, indexes, and triggers to tables:

You can add keys, columns, indexes, and triggers to tables in physical data model diagrams.

To add a key, column, index, and trigger to a table:

1. Select a table. An action bar appears.
2. Click one of the following action bar icons:
   - Click **Add new Key**.
   - Click **Add new Column**.
   - Click **Add new Index**.
   - Click **Add new Trigger**.

   You can also add these items by right-clicking the table, selecting **Add New Data Object**, and selecting one of the objects from the sub menu.

You can then select options in the Properties view.

Adding columns or triggers to views:

You can add columns or triggers to views in physical data model diagrams.

To add a column or trigger to a view:

1. Select a view to which you want to add a column or trigger.
2. An action bar appears.
   - Click **Add new Column**.
   - Click **Add new Trigger**.

   You can also add these items by right-clicking the table, selecting **Add New Data Object**, and selecting one of the objects from the sub menu.

You can then select options in the Properties view.

Deleting elements from tables and views:

You can delete elements from tables and views in physical data model diagrams.

To delete an element from a table or view:

1. Select the element to be deleted.
2. Press Delete.

Applying another deployment script to a target model

From the Data Models Page of the Deployment Script Editor, you can apply the DDL changes from another deployment script to the target model.

Prerequisite: A deployment script for changes to another database must exist.

Applying another deployment script to the target model of a database allows you to make the same changes to the database that were made to another database. For example, assume that two developers each have their own development database for an application. You want to move the changes that each developer made to the test database for the application. If you have the deployment scripts that the developers used to change their databases, you can apply those deployment scripts to a model of the test database, and then eventually deploy the consolidated changes to the test database.

Restriction: The Apply Deployment Script action, on the Data Models tab of the Deployment Script Editor, might not always leave the target model in a valid state. When the base models of the two databases are different, you might have to resolve errors that DB2 Change Management Expert will flag to account for missing objects. However, invalid models might result if there are extensive differences, especially in the unchanged parts of the models that have dependencies on the objects that were changed. Take care to thoroughly understand the changes to the target model before generating your change commands. Otherwise the commands might be invalid. When you are managing changes between databases with different base models, you should merge the changes by using the Comparison Editor. Always check your target model for errors before generating change commands and deploying those changes.

To apply another deployment script to a target model, complete the following steps:

1. Go to the Data Models Page in the Deployment Script Editor for the database that you want to change.
2. Click **Apply another deployment script** to display the Select Deployment Script wizard.
3. Complete the steps in the wizard to select the deployment script to apply to your target model. DB2 Change Management Expert extracts the DDL changes from the deployment script and applies them to your target model.

   Tip: If you encounter errors when applying another deployment script to the target model, the errors might be caused by the deployment script trying to:
   - Add an object that already exists in the target model
   - Delete an object that does not exist in the target model
   - Change an object that does not exist in the target model.
   - Add an object whose parent object does not exist in the target model.

   You will see a page that shows each error. You can choose to ignore the error or force the change to be applied. If you delete an object that does not exist or if you add an object that does exist, forcing the change to be applied is the same as ignoring the change. In all the other cases, forcing the change to be applied will copy the missing objects from the target model in the other deployment script and then reapply the failed changes.

   Verify that the deployment script is the set of changes that you want to make to

your target model. Check your target model, make any necessary changes to target model to correct for the possible causes of error, and then apply the deployment script again.

You are now ready to compare the base model and target model of the database that you are changing, to analyze the differences between the models, and to verify the changes that you are making. You can then generate the change commands to create the DDL that can be deployed to change the actual database.

Data model analysis

At any time when you are building a data model, you can analyze the model to verify that it is compliant with the defined constraints.

Data model analysis preferences

Use the Preferences window to set preferences for how data models are analyzed.

Setting preferences for constraints for data model analysis

Use the **Constraints** Preferences page to set default validation constraints for data model analysis.

You can use the Preferences window to specify the constraints that are enabled by default during a data model analysis.

To set preferences for constraints for data model analysis:

1. Click **Window** → **Preferences** from the main menu.
2. Click **Modeling** → **Validation** → **Constraints**.
3. Select constraints or constraint categories in the data modeling categories.

The constraints or constraint categories that you select are selected automatically when you perform a data model analysis.

Analyzing a data model

Use the Analyze Model wizard to analyze a data model to ensure that it meets certain specifications. Model analysis helps to ensure model integrity and helps to improve model quality by providing design suggestions and best practices.

You can use the Analyze Model wizard to analyze data models. You can analyze the following types of data models:

From the Data Project Explorer:

- A database or schema in a physical data model

From the Database Explorer:

- A database or schema in a database connection

To analyze a data model:

1. Right-click one of the supported data objects, and click **Analyze Model** from the pop-up menu. The Analyze Model wizard opens.
2. Complete the steps of the wizard, then . On the first page of the wizard, you select the rules that you want to use during the analysis. If you select naming standard rules, you can specify glossary models on the next page of the wizard.
3. Click **Finish**.

The model is analyzed, and errors and warnings are displayed in the Problems view.

Navigating to data model elements from the Problems view

After you perform a data model analysis, the results are displayed in the Problems view. You can navigate from a problem in the Problems view to the affected resource.

To navigate to a data model element from the Problems view:

Double-click an error or warning message in the Problems view.

The data element that is referenced in the error or warning is highlighted in either the Data Project Explorer or the Database Explorer, depending on which view that you used to perform the data model analysis.

Impact analysis

Impact analysis is the process of finding the dependencies that exist for an object. As you make changes to an object, you can generate and view a list of the related objects that might be impacted by your changes.

Analyzing impact and dependency

Use the Impact Analysis window to specify direction and expansion options for impact analysis.

You can use the Impact Analysis window to find dependencies. For example, if you want to copy a schema from the Database Explorer to the Data Project Explorer, you can find dependencies on the schema to ensure that all references are resolved. You can also use the Impact Analysis window to analyze impact. For example, if you modify an object, you can view the list of objects that might be impacted. You can analyze data objects in the Database Explorer, the Data Project Explorer, or in the data diagram.

To analyze impact and dependency:

1. Right-click a supported data object, and click **Analyze Impact** from the pop-up menu. The Analyze Impact window opens.
2. Specify options for the analysis.

The data object is analyzed, and the results are displayed in the Model Report view. Also, an impact analysis diagram is generated to show relationships.

Model comparison and synchronization

After you make changes to a model, you might want to compare the model to an earlier version of a model. Or, you might want to compare models of two different databases and move the differences between the models.

When you compare two models, you can easily see and navigate through the differences between the models. Comparing a changed model to an earlier version of the model lets you:

- Check your work and verify the changes are what you intended
- Use the information as an auditing tool to understand the differences between the two model versions

You can also use model comparison when you want to migrate the changes from one database environment to another. Comparing the model of the source database to the model of the target database lets you:

- Analyze the differences between the two models
- Move the changes from the source database model to the target database model

Model comparison can also be important when you are about to deploy your change commands to the database and you check to ensure that the current database catalog and base model still match. If the current database has changed and you refresh the base model to reflect the changes, you should compare the base and target models again to verify your planned changes. After you verify the changes, you can regenerate the change commands.

Data object comparison

When a comparison of data objects is performed, a comparison editor opens in the editor area. The differences between data objects are highlighted in the comparison editor, so that you can browse and copy changes between the compared resources.

The following types of data objects can be compared with each other:

- Objects that are contained in a connection in the Database Explorer
- Objects that are contained in a physical data model
- Physical data models (*.ddm)

The following data models are supported only if you have installed DB2 Change Management Expert into the same Eclipse shell as IBM Rational Data Architect:

- Logical data models (*.ldm)
- Domain models (*.ddm)
- Glossary models (*.ndm)

When you compare data objects, the standard compare editor is customized. For data objects, you can perform basic compare tasks, plus the following additional tasks:

Compare With → Another Data Object
Compares a data object with another data object. If you select this option, a window opens so that you can select another data object for the comparison.

Compare With → Original Source
Compares a data object with the original version of the object. This option is available only if you select an object that was reverse engineered from another data source.

Compare editor for data objects

When you perform a data object comparison, the compare editor is customized. The editor supports standard editor actions such as undo and redo.

The compare editor contains the following views when you compare data objects:

Structural Compare
Use this view to navigate and merge structural differences between data objects. The differences are presented in table format. Double-click a row containing a change and use the toolbar buttons to merge the changes from one object to the other.

Property Compare
Use this view to merge properties changes between data objects. The properties for each object that is selected in the Structural Compare view can be viewed and modified in the Property Compare view.

The following additional actions are available in the compare editor only when you compare data objects:

Analyze Left Impact 
Performs an impact analysis for the object on the left side of the editor.

Analyze Right Impact 
Performs an impact analysis for the object on the right side of the editor.

Generate Left Delta DDL 
Generates DDL for the changes that you made to the object on the left side of the editor. This action is available only if you make changes to the object. If you perform this action, you must specify a data design project to store the DDL file.

Generate Right Delta DDL 
Generates DDL for the changes that you made to the object on the right side of the editor. This action is available only if you make changes to the object. If you perform this action, you must specify a data design project to store the DDL file.

Export 
Exports the structural differences to an XML file on the file system. If you select this action, a window opens so that you can specify a name and location for the XML file.

The following basic compare actions are not available when you compare data objects:

- Copy Current Change from Left to Right
- Copy Current Change from Right to Left

Merging the structural differences of data objects in the compare editor

The Structural Compare view in the compare editor shows the structural differences between data objects. You can use this view to merge structural changes from one object to another. This view is available only when you perform a comparison among data objects.

The following basic comparison actions are available in the Structural Compare view of the compare editor when you compare data objects:

Copy from Left to Right 
Copies the structural attributes of the object on the left side of the editor to the object on the right side of the editor.

Copy from Right to Left 
Copies the structural attributes of the object on the right side of the editor to the object on the left side of the editor.

If you make changes to a model object, the changes are automatically added to the model; then you must save the model to save the changes. You can also choose not to save the changes, and generate delta DDL statements instead.

If you make changes to a server object, then you must generate delta DDL statements for the object and then deploy it to the server to save the changes.

To merge the structural differences of data objects in the compare editor:

1. Navigate to structural differences in the table by using the **Go to Next Difference** and **Go to Previous Difference** toolbar buttons on the main toolbar. You can also navigate by clicking in the table, pressing the arrow keys on the keyboard, and pressing Enter.
2. Merge changes from one object to another:
   - To merge changes in a single row and any rows that are contained by that row, double-click the row, then click the **Copy from Left to Right** or **Copy from Right to Left** toolbar buttons on the view toolbar to merge the change. If the row you select contains other rows (for example, if you select a table that contains columns), then all of the changes in the contained objects are also merged.
   - To merge all changes, double-click in a row that is the root of the object that you are merging changes for (for example, if you are merging changes in a schema, double-click the schema row), then use the view toolbar buttons to merge all changes.
3. Optional: Display objects that are impacted by your changes by clicking the **Analyze Left Impact** or **Analyze Right Impact** toolbar buttons on the view toolbar.
4. Save the changes:
   - To save changes to a model object, save the model, or click **Generate Left Delta DDL** or **Generate Right Delta DDL** to generate a DDL file that contains the changes.
   - To save changes to a server object, click **Generate Left Delta DDL** or **Generate Right Delta DDL** to generate a DDL file that contains the changes.

Merging data object properties differences in the compare editor

The Property Compare view in the compare editor shows the differences in the properties of data objects. You can use this view to merge property changes from one object to another. This view is available only when you perform a comparison among data objects.

If you make changes to a model object, the changes are automatically added to the model, then you must save the model to save the changes. You can also choose to not save the changes, and generate delta DDL statements instead.

If you make changes to a server object, then you must generate delta DDL statements for the object and then deploy it to the server to save the changes.

To merge data object properties differences in the compare editor:

1. Click the tabs in the Property Compare view to see the properties of the objects.

2. Use the controls in the Property Compare view to merge changes. You can type in the fields, or copy and paste properties values from one object to another.
3. Optional: Display objects that are impacted by your changes by clicking the **Analyze Left Impact** or **Analyze Right Impact** toolbar buttons on the view toolbar.
4. Save the changes:
   - To save changes to a model object, save the model, or click **Generate Left Delta DDL** or **Generate Right Delta DDL** to generate a DDL file that contains the changes.
   - To save changes to a server object, click **Generate Left Delta DDL** or **Generate Right Delta DDL** to generate a DDL file that contains the changes.

Delta DDL generation from the compare editor

After you merge changes to data objects in the compare editor, you can generate a delta DDL script that contains a series of statements that you can run to change an object from its original state to the state after the merge.

This feature is available only when you compare the following types of data objects:

- Objects that are contained in physical data models
- Objects that are contained in a connection in the Database Explorer

Generating and running delta DDL is the only way to save changes to a server object that you make changes to by using the compare editor. If you make changes to a model object in a project in the compare editor, you can either save the object or generate a delta DDL script.

The generated DDL scripts are optimized so that dropping and recreating tables is minimized when you are targeting DB2 Universal Database for Linux, UNIX, and Windows, or DB2 Universal Database for z/OS.

The **Generate Left DDL** and **Generate Right DDL** toolbar buttons are available only after you make changes to a data object by using the compare editor. When you click the toolbar buttons, a wizard opens so that you can specify a name for the DDL script and a project to store the script. In the wizard, you can also select to run the DDL script. If you select this option, you must provide connection information on the next pages of the wizard. If you do not select this option, you can review and edit the script in the SQL editor and then run it later outside of the workbench.

After you complete the wizard, the DDL script is saved and displayed in the **SQL Scripts** folder under the project that you specified.

Exporting changes to an XML file from the compare editor

After you merge changes to data objects in the compare editor, you can export the changes to an XML file.

This feature is available only when you compare data objects.

To export changes to an XML file from the compare editor:

1. Merge changes to data objects using either the Structural Compare view or the Property Compare view.
2. Click **Export**  from the compare editor toolbar.
3. In the window that opens, specify a name and a local directory for the XML file, then click **OK**. The XML file cannot be saved to a project in the workbench.

The XML file is saved to the directory that you specified.

The following code example shows the type of XML file that is generated.

```
<?xml version="1.0" encoding="UTF-8" ?>
<compare timestamp="Fri Jul 15 11:52:54 PDT 2005"
    left="design3/SAMPLE.dbm/SAMPLE.SAMP"
    right="SAMPLE:SAMPLE.SAMP">
<item name="Schema" left="<Schema> SAMP"  right="<Schema> SAMP">
 <item name="Table" left="<Table> Table1"   right="<Table> Table1">
 </item>
 <item name="Table" left="<Table> Table2">
 </item>
 <item name="Table" left="<Table> Table3">
 </item>
 <item name="Table" left="<Table> Table4">
 </item>
</item>
</compare>
```

Migration analysis

A common task that you might have as a DBA is to migrate the changes from one database environment to another. For example, you might be asked to promote the changes that were made to a large development database (the source database) to the formal test database (the target database). *Migration analysis* is the process of comparing two databases and identifying the differences between them.

Without DB2 Change Management Expert, you would need to keep track of the differences yourself by using mechanisms such as catalog queries to find the differences or by using tools to reverse engineer the catalogs into SQL and to identify the differences between the SQL. After finding the differences, you would then need to create and run the DDL to apply the appropriate changes to the test database.

With DB2 Change Management Expert, you can create models of the development database and the test database and compare the models to determine which objects are different. The Comparison Editor makes it easy to visualize the differences between models. You can also use the Comparison Editor to synchronize the structure of the test database model to the development database model (the source model) by moving the differences. After the test database model has the appropriate changes, you can then create a deployment script, using the test database model as the target model, to apply the changes to the actual test database.

Typically, a database is not exactly the same between environments. For example, in the previous scenario, the development database name might be different from the test database name. The development database might also have only one table space, no indexes, and relatively little data, whereas the test database might have several table spaces, a few indexes, and more data. So, as you analyze the differences, you need to identify those differences that are of real interest.

Comparing and synchronizing models of the source and target databases is one way to migrate changes between database environments. Another way to migrate changes is to apply the deployment script that was used to change the source database to a target model of the target database. DB2 Change Management Expert extracts the DDL changes from the deployment script and applies them to the target model. You can then regenerate change commands, handle data preservation, rebind changes, and deploy the changes to the actual database.

Comparing models, analyzing all the differences, and synchronizing the target model with the relevant differences can be time-consuming and labor intensive. Therefore, applying a deployment script to a target model to migrate changes can be an easier approach. The deployment script contains only the DDL for what changed in the source database; any other differences are ignored.

Methods of copying databases

You can copy databases in different ways both inside an outside of DB2 Change Management Expert. Copying a database involves copying an object or objects, the entire internal structure, and data from the source database to the target database.

You can use several different methods to copy a database, some of which are performed by using DB2 Change Management Expert and some of which are performed by using other DB2 tools. The methods that are supported by DB2 Change Management Expert include:

1. Copying a database by using a Deployment Script
2. Copying a database by using a script file

The method that you choose depends on the database environment and what needs to be copied. In some cases, it might be appropriate to use another DB2 tool, such as DB2 Control Center, DB2 utilities, or the DB2 Command Line Processor, to perform all or part of the database copy. The method that you choose depends on the database environment and on what needs to be copied. The following table compares each method:

Table 9. Methods for copying databases and the behavior in DB2 Change Management Expert

Method	Description	Creates database	Copies database structure	Copies data
Copying by using a Deployment Script	Copy the database by migrating database objects and data.	No. Performing a copy using a Deployment Script requires you to create the new database manually.	Yes	Yes
Manually creating script files	Manually create scripts with DB2 Change Management Expert to copy the database.	Yes	Yes	Yes

Table 9. Methods for copying databases and the behavior in DB2 Change Management Expert (continued)

Method	Description	Creates database	Copies database structure	Copies data
Copying a database outside of DB2 Change Management Expert	Use techniques outside DB2 Change Management Expert to copy the database.	Yes	Yes	Yes

Copying databases using a deployment script

Copying databases using a deployment script is a technique that allows flexibility and provides more structure and automation than manually creating script files.

Prerequisite: Creating a deployment script requires being able to connect to the database. Therefore you must create the new database manually.

Once the database is created, create a deployment script to configure the database copy. Copy the structural database objects from the source model to the target model. If copying data is desired, export the source data. Implement change commands (what amounts to the DDL. If copying data is desired, generate commands to import the data into the target.) Click **Generate Change Commands** on the Change Commands tab of the Deployment Script Editor to generate these statements. Deploy the change commands that are defined in the deployment script.

To copy a database using a deployment script, complete the following steps:

1. Create a deployment script. For instructions on how to create a deployment script, see “Creating a deployment script” on page 79.
2. Copy your database objects. To copy the structural objects, add the source model that you want to copy to the Physical Models page by clicking **Add**.

   Optional: You can also get the source model by dragging it from the Database Explorer into your project or by selecting **File** → **New** → **Physical Data Model**.

   Select the model and click **Migrate** to display the Comparison Editor. Use the Comparison Editor to move objects from the source to the target. If additional changes or configuration are required, those changes can be made by clicking **Edit Target Model** on the Physical Models page of the deployment script.
3. Migrate your data. For more information about migrating data, see the “Migrating data” on page 89 topic.
4. Generate your change commands. To generate commands, open the deployment script for the target database. On the Change Commands tab, click **Generate Change Commands**. Click **Generate Delta DDL** to generate DDL commands that create the structural objects defined in the target (that were either copied from the source model in the comparison editor or changed manually in the target model). Because this is a new database you might not need to select the **Enable Undo** option. Similarly, you do not need to select **Enable Data-Preservation** because the database most likely does not contain any data. If you do want to move data, the import script that was generated can be added to the Change Commands page of the target deployment script.

5. Deploy your change commands After you have created all the change commands, click **Deploy Change Commands** to copy the database structure and data.

Copying databases by manually creating script files

DB2 Change Management Expert can be used to generate some of the script files and to deploy the change commands that are necessary to copy a database. You can forward engineer the source database to create a script that will create or recreate all of the physical database objects.

To copy a database by manually creating script files, complete the following steps:

1. From the Database Explorer view, open a connection to your source database.
2. Select the database and click **Generate DDL** to generate a script. You could also add a create database statement to the top of the script.
3. To copy your data, create a script to export the data from the source database and import the data into the target database. You can deploy this script from the Data Project Explorer view.
4. Select each script, right click and select **Run SQL** to copy your database.

Change commands

Change commands are statements that can be issued against a DB2 database. Change commands are the central resource for managing change with DB2 Change Management Expert. They can be SQL, DB2 commands, or utility commands (such as export and import commands).

Change commands can be generated between the base model and the target model, manually created, or imported from a script file. DB2 Change Management Expert allows the SQL, DB2 commands, or utility commands to all be specified in a single script file.

DB2 Change Management Expert checks the syntax of change commands in the Change Command Editor. A change command can be invalid. Change commands are generated based on a particular database state. If these change commands are applied to the database in a different state, semantic errors can occur. DB2 Change Management Expert checks the semantics of the change command when the change commands are applied to the model. Error messages are generated if a change command has put the DB2 Change Management Expert model in an invalid state. DB2 Change Management Expert cannot issue change commands that are not syntactically correct. DB2 Change Management Expert can issue commands that might be invalid for semantic reasons; however, proper use of the deployment script will help to prevent this possibility.

Tip: There are a variety of editors avail in the Eclipse environment to help you edit SQL, however the Change Command Editor is probably the best to use when you are using DB2 Change Management Expert because it supports syntax checking for all of the commands that DB2 Change Management Expert supports. Other editors might not support all of the commands or might not check your syntax. You can specify the default editor based on your file extensions by selecting **Windows** → **Preferences** → **General** → **Editors** → **File Associations**. You should use the Change Command Editor as the default editor for *.chx, *.ddl, and *.sql files.

In addition, DB2 Change Management Expert provides the Generate Change Commands wizard that generates your change commands for you. For example, you can manipulate export and import utility commands by using the DB2 Change Management Expert Generate Change Commands wizard.

You use change commands to change your target database. You create a deployment script, which includes all of your change commands, to deploy your changes to the DB2 catalog. DB2 Change Management Expert automatically determines how to issue those changes.

A change command that is applied against a DB2 Change Management Expert model might succeed, but the same command issued against DB2 might fail. DB2 Change Management Expert can be more forgiving than DB2 because DB2 Change Management Expert shows you the proposed model of the database and you can decide if the actions taken by DB2 Change Management Expert were appropriate. For example, if between the time that you generate change commands and the time that you deploy them, the DB2 catalog might have changed underneath. Although the change commands that you applied to the model were successful, the change commands might fail when you deploy them to DB2 if the base catalog has been altered.

Tip: The Deployment wizard will warn you of situations where your commands might fail or if there is a conflict with the database catalog. You can refresh your base model before you deploy your changes. If a difference between your base model and the DB2 catalog was found, you must regenerate your change commands.

Creating change commands

You must create change commands to apply a change to a DB2 database.

To create change commands, click **File** → **New** → **Change Commands**.

Complete the steps in the New Change Commands wizard to create your change commands.

The New Change Commands wizard can create a change command file several different ways. The wizard can:

- Create a blank change command file
- Generate commands based on a model (creates everything in the model)
- Generate drop commands based on a model (drops everything in the model)
- Generate delta ddl based on two models (migrates information in one model to another model)

A change command script with the name that you specified is created under the Scripts folder in the Data Project Explorer view. You can view your change commands from the Data Project Explorer by opening your script with the Change Command Editor. You should use the Change Command Editor to edit and view all .sql, .ddl, and .chx files that are used in DB2 Change Management Expert.

Types of change commands

There are many different change commands and several ways to construct them in DB2 Change Management Expert. DDL, data preservation commands (export and import commands), and rebind commands are all different types of change commands.

DDL can be:

- Generated between any two models. This is called *Delta DDL*. Typically this is done between the base model and target model, more generally two versions of the same model. But they can be generated between two heterogeneous models in some cases.
- Generated from a single model. This is useful to recreate a model.
- Generated outside of DB2 Change Management Expert by using another tool. For example, db2look.
- Created manually by using the Change Command Editor.
- Imported into the project and into a deployment script by using the Change Commands tab.

Supported DB2 commands, utilities, and system commands can be:

- Generated based on DDL using a wizard in the deployment script.
- Composed manually by using the Change Command Editor.
- Imported into the project and into a deployment script by using the Change Commands tab.

The list of supported DB2 commands, utilities, and system commands includes:

- export
- import
- load
- HPU
- rebind
- reorg
- runstats

Importing change commands from a script file

You can import change commands from a script file. Script files can be applied to models.

Prerequisite: The script file must exist in your workspace or a directory outside of your workspace.

To import a change command from a script file, complete the following steps:

1. From the Data Project Explorer, select your project and right-click to display the context specific menu options.
2. Click **Import** to open the Import wizard. Complete the steps throughout the Import wizard to import your script file. For example, select File system to browse your directories and import your script. **Optional:** You can also drag and drop your script directly into your project.

Your new change commands have been generated from the script file and are now available in your workspace.

The Change Command Editor helps you identify script errors. A script with syntactic errors cannot be deployed against DB2.

Editing change commands

You can edit your change commands from the Change Command Editor.

You can create a new change command in the Change Command Editor, or you can edit a change command that exists in your project.

To edit your change commands from the Change Command Editor, complete the following steps:

1. Select your change command script from the Data Project Explorer.
2. Right-click the script and click **Open with** → **Change Command Editor** to display your change commands in the Change Command Editor.

   You can also view your change commands in the Outline view. Use this view to decide which changes you want to make. You cannot make changes within the Outline view. The Outline view is read-only.

   Tip: If your change commands do not appear in the Outline view, there is a syntax error in the script. Check the Problems view to resolve the syntax error.
3. Edit your change command by changing the SQL directly in the Change Command Editor.

   Tip: You can use Visual Explain functionality to view helpful information for SQL statements when you create stored procedures. To access Visual Explain, complete the following steps:

   a. In the Change Commands Editor, type a SELECT statement.
   b. Select that statement, right-click and select **Visual Explain**.

   The tables must already be created in the database and the connection to the database must already exist in order for you to be able to use Visual Explain. For more information about Visual Explain, see the Using Visual Explain for SQL statements topic in the DB2 Information Center.
4. Press Ctrl + S to save your changes. Your project is automatically refreshed when you issue the save command.

Example: You can add SQL statements to your change command:

```
CREATE INDEX I1 ON CMETEST.CMETEST10 (Col1, Col2, Col3, Col4,)!
CREATE FUNCTION F1 (F CHAR)
 RETURNS ROW (P1 INTEGER, P2 DOUBLE)
 LANGUAGE SQL
 CONTAINS SQL
 NO EXTERNAL ACTION
 DETERMINISTIC
 RETURN VALUES (1,0.1)!
```

You can also add comments such as:

```
--Definition of CMETEST1 database
--CMETEST schema
```

Tip: You can use the following comment to change the statement terminator:

```
--<scriptOptions statementTerminator = "!" />
```

Your change commands are now updated and reflect your changes.

An alternative method is to select **DDL** → **Edit**from the Data Model Editor to bring up a dialog with the DDL statements for all of the database objects contained in the model.

Change command syntax

DB2 Change Management Expert verifies the syntax of your change commands before they are applied to the database. DB2 Change Management Expert indicates suspected problems by placing an indicator (red 'x') next to the problem.

DB2 Change Management Expert verifies the validity of your change commands automatically, whether you create change commands with the Data Project Explorer or you import them from a file.DB2 Change Management Expert supports the same change command syntax as DB2 for Linux, UNIX, and Windows.

Remember: Problems are also reported in the Problems view.

If you notice syntax errors in your document, it might be caused by an incorrect statement terminator. Right-click in the Change Command Editor and select **Set Statement Terminator** to change the statement terminator.

If you use input that is generated by other products, ensure that the input is valid. The most common problem is with statement terminators, which are used to separate multiple statements. Any UCS-2 character in the range 0x0000 to 0x0080 inclusive can be used as the statement terminator provided that the statement terminator does not appear any where else in the statement. For example, you cannot use a semicolon as a statement terminator for an SQL statement to create a trigger because semicolons are required within the definition of the trigger body. You can change the statement terminator throughout the input by placing a script options statement terminator directive before the statements.

Applying change commands

You can use DB2 Change Management Expert to visualize your changes before you make them. DB2 Change Management Expert can apply DDL to physical data models that enable you to see their result before you deploy the changes against the actual DB2 database. This process is called *applying change commands to a model*. Applying a change command to a model is similar to issuing a command against DB2.

To view the outcome of your changes to a database model, you must apply your change commands. To apply your change commands to your target model, complete the following steps:

1. In the Data Project Explorer, select the change command script that you want to apply.
2. Right-click the script and click **Apply to Model** to display the Apply to Model wizard. Enter or select the project for your model and type the name of the model to which you want to apply your changes. Click **Finish** to apply your changes. The Data Model Editor will appear.
3. Save the resulting model. You might want to save this model with a new name to reflect your changes.

The resulting database model includes all of the change commands that were applied.

Merging change commands

If you make additional changes to your DDL after you have generated your change commands, you have the option to merge these changes. DB2 Change Management Expert will detect additional changes and prompt you to merge them by using the Merge Editor.

If you have already generated your change commands, and make an additional change, you must use the Merge Editor to merge those changes into one change command script. For example, if you add a GRANT statement to your existing DDL script by using the Change Command Editor and then generate change commands, DB2 Change Management Expert will prompt you to merge your changes.

To merge your change commands, complete the following steps:

1. After you have made one change to the target model, on the **Change Commands** tab of the Deployment Script Editor, click **Generate Change Commands**. Your new change commands script will appear under the SQL Scripts folder in the **Data Project Explorer** view.
2. If you make an additional change to your target model, you will not lose that change. For example, if you added another statement to the change command script or added a column to your target model. When you generate your change commands again, you will be prompted to merge your changes.
3. On the **Change Commands** tab of the Deployment Script Editor, click **Generate Change Commands**. DB2 Change Management Expert will prompt you with a message asking you if you want to merge your changes. Click **Yes** to display the Merge Editor.
4. In the Merge Editor, select the additional change from the Calculated Change Commands side of the editor and click the left arrow button to move your change to the Existing Change Commands side of the editor. Click **OK** to merge your changes.

You have now successfully merged your changes into one change command script.

Deploying change commands

Change commands can be deployed directly against a DB2 database.

Prerequisite: A working connection to a database is required to deploy change commands.

A properly configured Database Administration Server (DAS) is not necessary to deploy most change commands, unless they involve creating a database or deploying certain DB2 system commands. Most DDL statements go through JDBC for deployment.

To deploy change commands, complete the following steps:

1. In the Data Project Explorer, select your change command script. For example, select sample_deltaddl.chx.
2. Right-click the script, and click **Run SQL**.

The database catalog is updated and will contain your changes. You can save the result of your deployed change commands to a file.

Working with the deployment script

A *deployment script* describes specification resources, change commands intended to implement the change, and metadata about how that change will be issued.

In other words, the deployment script is the set of changes that you want to make to your database. When you deploy your changes from the deployment script, you

are issuing commands to change your DB2 database. You can use the Deployment Script Editor to deploy your changes to a DB2 database.

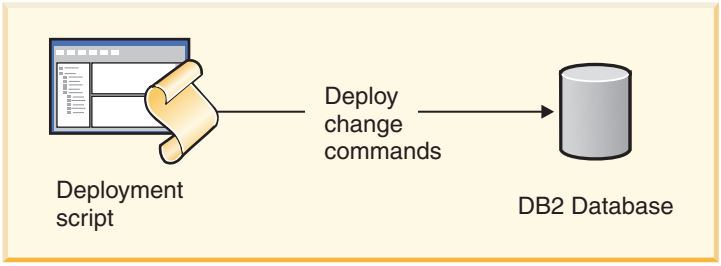

Figure 8. Deploying change commands from the Deployment Script Editor.

The Deployment Script Editor presents a consolidated way to work with the change that you want to deploy. It allows you to access all of the objects and actions that are relevant to the change that is being deployed.

Within the Deployment Script Editor you can:

- Specify your database connection information
- Edit your target physical data models
- Compare the base and target models
- Migrate objects from other databases to the target model
- Create change commands, edit existing change commands, and import other change command files
- Create export and import commands to preserve your data
- Migrate data between environments
- Create runstats commands
- Create flush package case commands
- Rebind your application packages
- Create REORG commands
- Preserve authorizations
- Refresh your base model
- Deploy your changes to the DB2 catalog
- Undo your changes

Overview of using a deployment script to change a database schema

You can change a database by using a deployment script, which is a set of changes that a user, or group of users, wants to effect at a single point in time.

You can use the Deployment Script Editor to change a database schema. The following figure illustrates the Overview Page of the Deployment Script Editor:

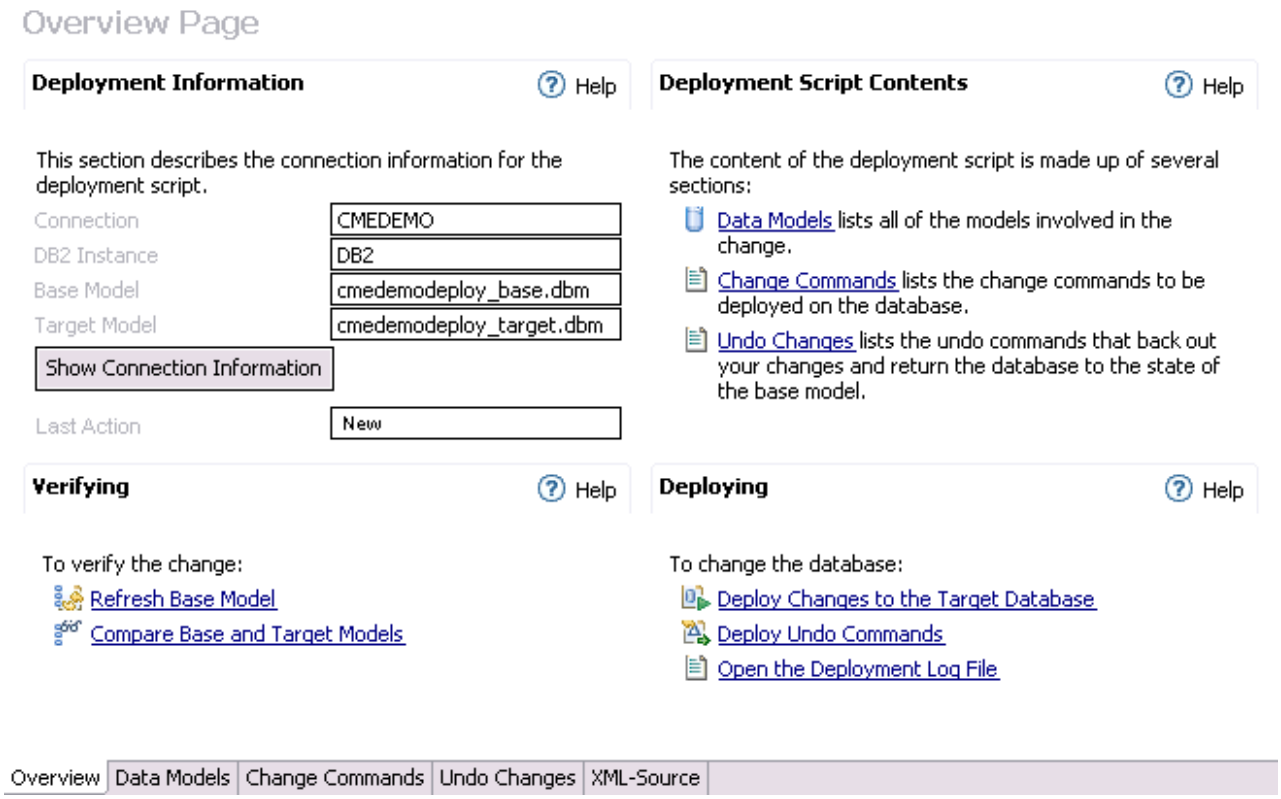

Figure 9. The Deployment Script Overview Page

The tabs of the Deployment Script Editor contain actions that you can use to perform your database change.

Overview tab

The overview tab provides an overview of your change management project. You can view your connection information, the last action that was performed in the change cycle, your deployment script contents, as well as deployment and verification options.

The purpose of the Last Action, or deployment script state, is to indicate the last action performed and help to indicate the next action to perform. A history of actions for any particular change is stored in the deployment script XML. The sequence of states can be reviewed later on if the deployment script for each change is being saved for auditing purposes. The states are cleared when the deployment script is reset. The following is a list of states that might be displayed in the Last Action field:

New The deployment script is in the new state when it is first created or after is has been reset. This state indicates little customization has been done to the script. The likely actions from this state are to edit the target model or add a source model on the Models tab of the Deployment Script Editor.

Modeled
The modeled state is reached when you edit the target model. The likely action from this state is to generate your change commands.

Migrated
The deployment script is put in this state after you use the migrate

option to copy data objects from a source model to the target model. The likely action from this state is to generate your change commands.

Refreshed Base Model
The deployment script is put in this state after the base model has been refreshed. The base model would need to be refreshed if a change has been made to the database catalog since the time that the deployment script was constructed. The baseline is checked against the database catalog when you attempt to deploy your change commands. The likely action from this state is to compare your base and target models in order to forward fit any necessary changes.

Generated
The Generated state is reached when you either generate your change commands by using the Generate Change Commands wizard or when you select the data migration option. The likely actions from this state are to deploy your change commands or to generate data migration commands.

Deployed
The deployed state is reached after you deploy your change commands. The next possible actions are to reset the deployment script, commit your changes and tag the files in your library control system, undo your changes, or refresh your base model.

Undo The undo state is reached after you have reversed your changes. The next possible actions after undo are to reset the deployment script, commit your changes using Eclipse Team, or to fix the problem that made it necessary to undo your changes.

Multiple Provisioned
The multiple provision state is reached after you have deployed your changes to multiple databases.

The Outline view also displays the state of your deployment script in parentheses as well as a check mark next to each completed step.

Data Models tab
You can specify and compare your base and target models from the **Data Models** tab. Click **Compare Base and Target Models** to display the Comparison Editor. The models that you are comparing are displayed as follows:

Table 10. How models are displayed in the Comparison Editor

Type of Analysis	Left side of the Comparison Editor	Right side of the Comparison Editor
Impact analysis	Base model	Target model
Migration analysis	Model from another environment	Target model

Change Commands tab
Use the **Change Commands** tab to generate the change commands that are necessary to change a database (DDL, export and import commands, and maintenance commands for runstats, rebind, and flush package). Specify the order of your commands for deployment, and deploy the commands against the database catalog. The order of commands is important. For example, export commands must be deployed before the

change command delta DDL files because the data needs to be exported before the destructive changes occur. Import commands and maintenance commands should be issued at the end and must be at the bottom of the list.

You can also use this tab to generate the change commands (DDL, export and import commands, and maintenance commands) that are necessary to undo the change.

Undo Changes tab

Use the **Undo Changes** tab to verify the undo commands necessary to back out your changes, specify the order of your commands for deployment, and deploy the undo commands against the database catalog.

XML Source tab

Use the **XML Source** tab to view the XML for the change commands that DB2 Change Management Expert will deploy against your database catalog.

Multiple Deployment tab

Use the **Multiple Deployment** tab to deploy your changes to multiple databases at one time. This tab is only displayed after a deployment script is created and then selected for deployment to multiple databases. DB2 Change Management Expert also supports generation of data preservation commands and maintenance commands during multiple deployment.

In general, the process of using a deployment script to change a database schema consists of the following steps:

1. Changing your target model
2. Generating your change commands
3. Deploying your change commands

The details of the process are as follows:

Changing your target model

The deployment script always references a base and a target model. These are models of the same database but potentially at different points in time.

You can edit your target model in a variety of different ways. It can be edited by using the Data Project Explorer, the Data Model Editor, the Comparison Editor, or the Diagram Editor. You can edit the properties of your database objects by using the Properties View.

You can also migrate object definitions from other databases. The primary way to migrate objects from other databases is to add the source model to the Data Models page of the Deployment Script Editor and click **Migrate**. The migrate action displays the Comparison Editor.

Compare your base and target models before you generate your change commands. This will allow you to quickly review the differences that your changes will create between your models.

Generating your change commands

The Generate Change Commands wizard is located on the Change Commands page of the Deployment Script Editor. If you make changes to your target model, you should regenerate the change commands before you deploy them to the target database. You can specify your data preservation commands in the Generate Change Commands wizard.

Deploying your change commands

From the Overview page of the Deployment Script Editor, click **Deploy changes to the target database** to deploy your changes to the database catalog. The change commands are issued against your DB2 database. You can also deploy your changes from the **Change Commands** tab.

Resetting and reusing a deployment script

After you have deployed your changes to a database, you can reuse the same deployment script for deploying another set of changes. To do this, you will have to reset the deployment script by selecting **Deploy** → **Reset**. The reset operation will reload the base and target models so that they are current. You can also specify new schema and database element filters for the new deployment process. The other deployment script elements, such as delta DDL file, export command file, and so on, that were created during the previous change management process will be deleted. The state of the deployment script will be reset to **New**.

Creating a deployment script

You need to create a deployment script to deploy your changes to your target database.

To create a new deployment script, complete the following steps:

1. Click **File** → **New** → **Deployment Script** to start the Deployment Script wizard.

   You can also create a Deployment Script by using the following methods:

   - Click the down arrow for the **New** icon and click **Deployment Script** to start the New Deployment Script wizard.
   - Right-click in the Data Project Explorer, and select **New** → **Deployment Script** to start the New Deployment Script wizard.

2. Complete the steps in the New Deployment Script wizard to create a new deployment script.

   DB2 Change Management Expert creates the base and target models for you. You can filter by schema name so that you can narrow the scope of the two models that you want to work with.

   Tip: DB2 Change Management Expert uses the base model to ensure the database has not changed between the time that the change was created, defined, and deployed. Always check to see that the base model matches the catalog when deploying a script. However, never refresh the base model without forward fitting changes to the target model because the deployment script might backout changes from another change.

   If you have DB2 Grouper installed, DB2 Change Management Expert checks for your Grouper groups and allows you to use those groups when you create a new deployment script. When you create a new deployment script, the New Deployment Script wizard will display your Grouper set names on the **Select Setname** page. Select the Grouper group that you want to include in your deployment script and click **Next**. The schemas that belong to those Grouper groups will be preselected on the **Select Schema** page of the wizard.

   For more information about installing and using DB2 Grouper, see the *DB2 Grouper User's Guide*.

3. The Deployment Script Overview page is displayed after you complete the steps in the Deployment Script wizard.

From the Deployment Script Overview page, you can manage most of your change management operations. Click the **Data Models** tab and the **Change Commands** tab to work with your deployment script.

Tip: The DB2 instance name of the database must be correct in the New Deployment Script wizard. With DB2 version 8.1 fix pack 9 and above, DB2 Change Management Expert can detect the DB2 instance name. If DB2 Change Management Expert is unable to detect the instance name, then DB2 Change Management Expert will use DB2 as the default instance name. You can edit the instance name on the Overview page of the Deployment Script Editor.

Working with models from the Deployment Script Editor

DB2 Change Management Expert allows you to easily identify changes and create more changes throughout your database environment. DB2 Change Management Expert uses models to help you transition database structural changes throughout your database environment.

Instead of making changes directly to the DB2 catalog, you make changes to *models*. A model is a representation of the database catalog. Models can be used to plan changes to a database, analyze the impact of those changes, and to eventually deploy those changes to the actual database.

On the Data Models page, you can:

- Organize changes
- Identify and define changes to the target database by comparing your base and target models in the Comparison Editor
- Identify and merge differences between source and target databases
- Add a source database to the list
- Select the newly added database
- Merge changes from your source model into your target model
- Edit the target model directly
- Automatically apply changes from another deployment script

To work with models from the Deployment Script Editor, complete the following steps:

1. On the Overview page in the Deployment Script Editor, click the **Data Models** tab to display the Data Models page.
2. In the Define Source Models area, click **Add** to display the Source Models selection window.
   a. Expand your project folder and select the model that you want to reference.
   b. Click **OK** to add this model to your deployment script.
3. Click **Edit Target Model** to display the Model Editor and make changes to your target model.
4. Click **Compare Base and Target Models** to display the Comparison Editor, which you can use to analyze the differences between your base and target models.

After you have defined your base and target models, proceed to the **Change Commands** tab to generate and define your change commands.

Generating change commands from the Deployment Script Editor

You can use the Generate Change Commands wizard, which can be invoked repeatedly from the Deployment Script Editor, to generate the change commands (the DDL, DCL, and DB2 commands, such as export, import, rebind, and runstats) that are necessary to apply changes to a DB2 database.

Prerequisite: A deployment script for the database that you plan to change must exist.

To generate change commands from the Deployment Script Editor, complete the following tasks:

1. Click the **Change Commands** tab to display the Change Commands page.
2. On the Change Commands page, click **Generate Change Commands**. The Generate Change Commands wizard starts.

   Complete the steps in the wizard to generate your commands. The wizard takes you through steps to specify information for:

   a. Generating the DDL.

      You must select Generate Delta DDL unless a delta DDL file for the change has been created previously. You can also select Enable Undo to generate the DDL that is required to undo the change. Both options are checked by default; however, the last selection that you make in this wizard is preserved. If you un-checked any of these options the last time that you ran the wizard, they will not be checked when you return to the wizard.

      Restriction: DB2 Change Management Expert might not always generate rename statements in situations where a rename is possible.

      However, the delta DDL is fully editable and you can change any of the scripts manually. You can replace the destructive changes with rename statements where it is appropriate. You should use the **Table Rename** tab on the first page of the Generate Change Commands wizard to specify what the table has been renamed to and let the Delta DDL generator determine if the rename is appropriate or not. The Change Command Editor will verify that the script is syntactically correct. Check all of the generated commands before you deploy them to your database.

   b. Renaming a table.

      If you create a deployment script that uses TEST as the base model, and then rename the ORG table to ORG_new in the Test target database, DB2 Change Management Expert records this rename activity. When you generate your change commands by using the Generate Change Commands wizard, the **Table Rename** tab displays the existing table name on the left side and the new table name that you specified on the right side of the page. When you generate the change commands, a RENAME statement similar to the one in the following example will be generated:

      ```
      RENAME SCHEMA.ORG TO "ORG_new"
      ```

      If you rename a table by using some method other than the deployment script, you must specify the new table name on the **Table Rename** tab of the Generate Change Commands wizard. For example, if you create a physical model and rename the ORG table to ORG_new, you must specify the new table name to ensure that DB2 Change Management Expert will generate the correct change commands. Click the Add icon in the upper

right corner of the **Table Rename** page, select ORG from the source model list on the left side, and select ORG_new from the target model list on the right side. DB2 Change Management Expert will generate a RENAME statement instead of the DROP TABLE ORG and CREATE TABLE ORG_new commands that would have been generated if you did not specify the rename activity on the **Table Rename** tab.

You can add or delete your table rename selections by using the Add and Delete icons in the upper right corner of the **Table Rename** page.

Make sure that you check the option to **Enable data preservation** for the data in the table that you are renaming.

c. Generating export and import commands for data preservation.

When you select the **Enable data preservation** option on the DDL Generation Options page of the wizard, the wizard includes pages to specify information for generating export and import commands. You can create export commands to save the data from a dropped table to a file and import commands to insert data from the file into a newly created table.

If you selected to generate the DDL to undo the changes, export and import commands will also be generated for undoing the changes.

If DB2 Change Management Expert determines that data needs to be preserved, the **Enable data preservation** option in the Generate Change Commands wizard is checked by default. To skip data preservation, uncheck the option.

d. Generating DB2 maintenance commands.

You can generate statements to grant and revoke authorization and commands to rebind packages, run statistics for tables, and flush the package cache so that SQL statements will use updated statistics.

If you selected to generate the DDL to undo the changes, maintenance commands will also be generated for undoing the changes.

You can choose to not generate any DB2 maintenance commands.

e. Merging change commands.

If you have already generated your DDL, export and import commands, or DB2 maintenance commands and regenerate any of these, you are prompted to merge the changes from the **Previous Change Commands** side of the Merge Editor to the **Proposed Change Commands** side of the editor. You should consider merging in the following cases:

- If you want to preserve customization made to a change command file.
- If both generate data migration and generate change commands are being used, it might be necessary to merge the export and import files.

You can always return to the Generate Change Commands wizard to specify the generation of commands that were not previously generated or to make changes to commands that were previously generated.

3. Verify that the generated script files are displayed in the Define Change Command field on the Change Commands page. If you generated commands to undo the changes, you can click the **Undo Changes** tab to see the generated undo script files, which are displayed on the Undo Changes page.

The following table shows the default file names that DB2 Change Management Expert uses for making database changes:

Table 11. Default file names for change command files

Default script file names	Contains
*deployment_script_name*_export.chx	Data unload commands

Table 11. Default file names for change command files (continued)

Default script file names	Contains
*deployment_script_name*_deltaddl.chx	DDL to make the changes
*deployment_script_name*_import.chx	Data load commands
*deployment_script_name*_maint.chx	Rebind commands, runstats commands, REORG commands, and commands to flush the package cache
*deployment_script_name*_auth.chx	Authorization statements

The following table shows the default file names that DB2 Change Management Expert uses for undoing database changes:

Table 12. Default file names for undo change command files

Default script file names	Contains
*deployment_script_name*_undoddl.chx	DDL to undo the changes
*deployment_script_name*_undoimport.chx	Import commands for undoing the changes
*deployment_script_name*_undoauth.chx	Authorization statements
*deployment_script_name*_undomaint.chx	Rebind commands, runstats commands, and commands to flush the package cache for undoing the changes

Important: The script files are deployed in the order that they appear in the list. Change commands are deployed in the order that they appear in the script files.

4. To view the contents of any change command script file, select the file and click **Edit**.

Your change commands are ready for use, and you can go to the Overview page to deploy them.

Important: If your target model is invalid, the generated DDL and undo commands might have syntax errors. You can correct these errors by using the Change Commands Editor. It is recommended that you fix any errors in the target model before you generate the change commands.

Data preservation

Data preservation allows you to control how data appears in your target database when there are changes to the data structures or metadata. Whenever change commands are generated that involve CREATE and DROP table statements, that data will likely need to be preserved. You can create the necessary export and import commands by using the data preservation options in DB2 Change Management Expert.

DB2 Change Management Expert allows you to preserve your data from the Change Commands page in the Deployment Script Editor.

Data preservation is useful in the following situations:

When you drop a table

When you drop a table, you might want to store the data from that table in a file for future use. DB2 Change Management Expert automatically generates this file, but this functionality can be disabled.

When you create a new table

When you create a new table, you might want to populate the table with data from an existing data file. You can map the data that you want to export from one table and import into another table by using the Generate Change Commands wizard.

After you generate the change commands, you can manually edit the change commands to import the data from the data file. If you edit the change commands manually, verify that the import commands are added to the change command list after the other change commands.

When you perform a destructive change

If you rename a table by using a DROP and CREATE statement, you will need to export the data in that table and then import it. DB2 Change Management Expert automatically creates data preservation entries, one with the export commands to drop the table (old name), and another with the import commands to create the table (new table name). You can use the pair function to combine these two entries into one. This way, the export and import commands have the same data file.

You can toggle the Customize field to make data preservation entries read-only or customizable. When an entry is customizable, additional wizard pages that assist with the customization of export and import commands are available. Customization of the data preservation change commands is necessary to correctly deploy certain changes.

Restriction: When you generate change commands, the default data preservation entries are always regenerated. Previously generated data preservation commands are not brought in as part of rerunning the Generate Change Commands wizard.

To skip the data preservation process, uncheck the **Enable data preservation** option in the Generate Change Commands wizard.

Advanced data preservation techniques

Advanced data preservation techniques include:

Dropping columns

Dropping columns can be easily managed by creating a new data preservation entry. One approach is to accept the default during export customization. For example, if you enable undo by selecting that check box, an export command will automatically be generated. However, when you customize your import commands, you must map the exported columns to the imported columns.

Adding NOT NULL columns

Adding NOT NULL columns is simple provided that the column has a default value. If there is no default value, customize the export command by using the Customize Export Command page of the Generate Change Commands wizard or use the Generate Data Migration wizard.

Normalizing tables

There are many strategies for normalizing a table. These techniques range from creating new normalized tables and transforming them in the database to transforming during the export and import of data.

Moving data between databases

Moving data between databases requires exporting data from one database

and importing data into another. DB2 Change Management Expert supports data migration with the Generate Data Migration wizard.

LOAD Providers

DB2 Change Management Expert now supports LOAD commands as well as IMPORT commands. You can specify the default load provider for the data load operations by using the preferences page (**Windows** → **Preferences**). You can change the load provider for a particular load operation by selecting **Data Load Provider** from the drop-down menu under Mapping Details on the Mapping Tables page. DB2 Change Management Expert also supports the IMPORT provider. You can specify the command options for all providers on the **Specify File Information** page of the Generate Change Commands wizard.

For more information about the DB2 LOAD and the DB2 IMPORT commands, see the *IBM DB2 Command Reference*.

UNLOAD providers

Data Unload Providers generate different types of unload commands. The following table shows the three different providers that DB2 Change Management Expert supports and the corresponding file formats for the EXPORT commands:

Table 13. Supported unload providers and their corresponding file formats

Unload provider	Corresponding file format for EXPORT commands
EXPORT_DEL	DEL data file format
EXPORT_IXF	IXF data file format
HPU	HPU unload commmands

In order to deploy the generated HPU unload commands, the DB2 High Performance Unload (HPU) for Multiplatforms (or Workgroups) product must be installed.

Attention: DB2 Change Management Expert supports the DB2 High Performance Unload (HPU) for Multiplatforms (or Workgroups) product to unload your data. However, note that HPU is a separately priced tool and must be installed before setting the DB2 Change Management Expert option for Data Unload Provider to HPU. If you set the Data Unload Provider to HPU and HPU is not installed, the generated unload commands will fail.
You can unload data using either the HPU command or a DB2 EXPORT command. You can specify the default unload provider on the preferences page by selecting **Windows** → **Preferences** → **Data** → **Change Management Options**. If you select HPU as the default provider, then DB2 Change Management Expert generates HPU commands for all of the unload operations. You can override this behavior for any particular unload operation by selecting a different **Data Unload Provider** on the Mapping Tables page of the Generate Change Commands wizard.

For more information about High Performance Unload, see the IBM DB2 High Performance Unload for Multiplatforms and Workgroups User's Guide V3.1.

For more information about EXPORT commands, see the *IBM DB2 Command Reference*.

Using REORG commands

DB2 Change Management Expert now supports the REORG maintenance command. DB2 Change Management Expert can generate REORG commands for you through the Generate Change Commands wizard. When a table space is altered, DB2 Change Management Expert generates a REORG command for all of the indexes and tables contained within the table space. When a table is altered, DB2 Change Management Expert generates a REORG command for the indexes defined on the table. REORG commands are optional.

Using the Auto CAST function

If the data types of the import and export columns are mismatched, DB2 Change Management Expert resolves them automatically by adding a CAST column function in the SELECT clause of the EXPORT statement. This can be done by selecting **Auto Cast** on the Customize Export Commands page of the Generate Change Commands wizard. There is another option, **Default Query** that allows you to revert to the default SELECT clause.

Restriction: If you make any additional changes to the SELECT clause, they will be lost when you select the **Auto Cast** or **Default Query** options.

Generating data unload and load commands for data preservation:

Your changes might require that some tables be dropped, created, or dropped and recreated, such as when you drop a column from the middle of a table. You might need data unload commands to preserve data from the dropped table and data load commands to load the appropriate data into the newly created table when you make database changes, including undoing a change. You can generate data unload and load commands and map the tables from which to unload and then load the data from the Deployment Script Editor.

To generate data unload and load commands from the Deployment Script Editor, complete the following steps:

1. On the Change Commands tab of the Deployment Script Editor, click **Generate Change Commands**. The Generate Change Commands wizard starts.
2. Ensure that the **Enable data preservation** option on the first page of the wizard is checked, and complete the steps in the wizard for generating data unload and load commands.

   If data preservation is unchecked DB2 Change Management Expert will not attempt to generate data preservation commands even if they are needed. If data preservation is not needed DB2 Change Management Expert will not generate any data preservation commands.

   DB2 Change Management Expert generates default data preservation entries when it detects that data needs to be preserved. You can also add your own entries to associate data unload commands with their corresponding load commands. When the **Customize** field for an entry is set to **Yes**, the wizard includes pages that allow you to customize the data unload and load commands and map how data should be unloaded out of one table and loaded into another table. You can toggle the **Customize** field. DB2 Change Management Expert automatically sets the **Customize** field to **Yes** if it detects that you might need to customize the commands or map the columns.

Important: Ensure that the data file names for your data unload and load commands are correct. Incorrect data file names can result in data loss. All data unload commands must specify unique data file names.

Also, on the first page of the Deployment wizard, ensure that the **Stop deployment if errors occur** option is checked. If a command fails, you want to stop the deployment from continuing. If you do not stop the deployment on errors, data will be lost.

3. When you return to the **Change Commands** page, verify that data unload and load command files appear in the **Define Change Commands** field.

Your data unload and load commands are displayed in the appropriate order in the change commands list. Data unload commands are added first because you need to unload your data before you drop a column or table. Data load commands are added last.

You can view the syntax of your data unload and load commands in a text editor by selecting the file from within the Deployment Script Editor and clicking **Edit**.

Generating DB2 maintenance commands

Throughout the change management process, specific database packages might become inoperative and statistics might become inaccurate. For example, DB2 marks packages as invalid or inoperative when you drop objects. You might need to issue rebind commands to re-create the packages based on the most current statistics, or you might need to restore the authorization for objects that are re-created. When you generate change commands from the Deployment Script Editor, you can specify that DB2 Change Management Expert automatically generate DB2 maintenance commands.

You have the option to generate the following commands:

Authorization statements

If your database changes require that objects be dropped and recreated, you might need to restore the authorizations that were granted on those objects. You might also want to specify authorizations for new objects that you add to the database manually by using the Change Command Editor.

Tip: Ensure that the script that contains these changes is displayed on the Change Commands page of the Deployment Script Editor.

Runstats commands

It is important to regenerate statistics when you make many changes to the database and also after your data has been loaded into tables. Runstats commands are generated for all the change command files in the deployment script, including the newly generated ones.

Reorg commands

You should reorganize all indexes when a table is altered. You should also reorganize all tables and indexes when a tablespace is altered. DB2 Change Management Expert will automatically generate runstats commands to refresh the statistics after a table is reorganized. You should also rebind all of your application packages to take advantage of the reorganized data.

For more information about the impact of reorg commands, see the *DB2 Command Reference*.

Rebind commands

You will need to rebind packages if your change commands contain the following DROP statements:

- TABLE
- TRIGGER
- MQT
- UDF
- VIEW
- ALIAS
- INDEX
- STRUCTURE TYPE

If you are making several changes, it is also recommended that you rebind your packages to improve performance. DB2 Change Management Expert generates rebind commands for all the packages in the database for the selected schemas by default. To limit the scope to only those packages with the DDL in the change command files, you can check the **Limit the scope of the generated rebind commands to the selected change command** files option.

Flush package cache commands

After the statistics are updated, you can flush the package cache so that the dynamic SQL statements will use the updated statistics, which will enhance the performance.

These commands should be run after the changes have been made to your database and after the data has been loaded into various tables.

To generate DB2 maintenance commands from the Deployment Script Editor, complete the following steps:

1. On the Change Commands tab of the Deployment Script Editor, click **Generate Change Commands**. The Generate Change Commands wizard starts.
2. Complete the page in the wizard to specify whether to preserve any authorities and privileges that exist on the objects and to generate runstats, rebind, and flush cache package commands. No options are checked by default.

   If you choose to preserve authorization, DB2 Change Management Expert generates the authorization statements in a separate script file from the other maintenance commands. By default, the file name is *deployment_script_name*_auth.chx. If you choose to generate rebind, runstats, or flush package cache commands, DB2 Change Management Expert generates these commands and puts them a single script file in the following order:
   a. Runstats commands
   b. Rebind commands
   c. Flush package cache commands

   You can use the default file name for the script file for the rebind, runstats, and flush package commands, or specify your own file name.
3. When you return to the **Change Commands** page, verify that script files for the authorization statements and the other maintenance commands are displayed in the Define Change Commands field.

Generating undo commands

If a change to the target database fails or produces undesirable results, you can roll back that change to recover the previous database by creating *undo commands*.

Undo commands are change commands that you use to roll back your changes. You can generate, view, and deploy undo commands from the Deployment Script Editor.

When you generate the change commands to apply to your database, DB2 Change Management Expert can automatically generate the commands that are necessary to undo the changes. To generate undo commands from the Deployment Script Editor, complete the following steps:

1. On the Change Commands page of the Deployment Script Editor, click **Generate Change Commands**.
2. Ensure that **Enable Undo** is checked on the DDL Generation Options page of the Generate Change Commands wizard to automatically generate the undo commands. This option is checked by default. If the **Enable Undo** check box is cleared, no undo script is generated. You can generate undo commands only by using the Generate Change Commands wizard.
3. Complete the steps in the Generate Change Commands wizard. DB2 Change Management Expert uses the base and target models to calculate the undo commands and adds them to the deployment script.
4. Return to the Undo Changes page of the Deployment Script Editor. On the Undo Changes page, your undo scripts are displayed in the **Define Undo Commands** section.
5. You can add and specify the order of the undo scripts in the **Define Undo Commands** section. Click **Add** if you want to add additional scripts to the set of automatically generated undo commands.

Your undo commands are ready for you to use in case you need to return your database to its original state.

Restriction: If you modify the undo commands file and then run the Generate Change Commands wizard again, the automatically generated undo commands will overwrite your manual changes. However, you can merge the commands using the Merge page of the Generate Change Commands wizard.

Migrating data

You can use the Migrate Data wizard, which can be invoked from the Deployment Script Editor, to generate the change commands that are necessary to copy data. Those commands can then be deployed to the database.

Prerequisite: A deployment script for the database to which you plan to move data must exist.

To copy data, complete the following steps:

1. Open the deployment script in the Deployment Script Editor.
2. Click the **Change Commands** tab to display the Change Commands page.
3. On the Change Commands page, click **Generate Data Migration**. The Migrate Data wizard starts.

   Complete the steps in the wizard to generate your commands. The wizard takes you through steps to specify information for:

   a. Selecting the data migration options.

      You can either generate the change commands so that the files are standalone or so that they are integrated into the deployment script. When you generate them to be standalone, you deploy them to the database by

right-clicking the change command file in the Data Project Explorer, and clicking **Run SQL**. When you have them integrated into the deployment script, you can deploy them to the database from the Deployment Script Editor.

b. Selecting the source connection.

   You must identify a connection for the database from which you want to copy data.

c. Specifying file information.

   You must specify a directory for the export and import commands that will be generated for copying the data from the source database to the target database. You can change the method that will be used to unload and reload the data.

d. Specifying the mapping tables.

   The wizard includes pages for specifying information for generating the export and import commands. You must identify the pairs of tables from which to save data and into which to load the saved data. Use the **Add New Entry** icon to add a table entry.

e. Customizing export and import commands for data migration.

   You can customize the commands that unload and reload the data and map the columns between tables.

f. Merging change commands.

   If you chose to have the change commands generated so that they are integrated into the deployment script and the deployment script contains other change command files, you are prompted to merge the changes from the **Previous Change Commands** side of the Merge Editor to the **Proposed Change Commands** side of the editor. Merge the commands, ensuring that referential integrity is preserved. That is, ensure that the parent table in any referential integrity relationship is loaded before the child table.

4. Deploy the change commands to copy the data from the source database to the target database, using one of the following methods:
   - If you generated the change commands to be standalone:
     a. In the Data Project Explorer, right-click the change commands file that contains the export commands, click **Run SQL**, and complete the steps in the Deploy Change Commands wizard. Repeat this for each export file.
     b. Right-click the change commands file that contains the import commands, click **Run SQL**, and complete the steps in the Deploy Change Commands wizard. Repeat this for each import file.
   - If you generated the change commands to be integrated with the deployment script:
     a. With the deployment script open in the Deployment Script Editor, click the **Change Commands** tab to display the Change Commands page.
     b. On the Change Commands page, click **Deploy Changes to the Target Database** .
     c. Complete the steps in the Deployment wizard to deploy the changes. The changes for all of the change command files that are contained in the deployment script are applied.

There is no way to automatically back out migrated data because you cannot generate undo commands for data migration. However, if you deploy undo

commands to roll back the changes that you have made to a database and a table is dropped and re-created as part of the undo, any data that you have copied over is effectively lost.

Working with change commands from the Deployment Script Editor

The change commands that are generated for the target model are listed on the Change Commands page in the Deployment Script Editor. You can add additional change commands by creating them manually or by importing them from your workspace.

On the Change Commands page, you can create new change commands, add existing change commands, and specify the order of your change commands. The following actions are available from the Change Commands page in the **Define Change Commands** work space:

- Click **Add** to add a change command file to your deployment script. You can import an existing change command file from your workspace or create additional change command files manually in the Change Command Editor.
- Click **Up** or **Down** to rearrange the order of your change commands in the list.
- Select a change command file and click **Edit** to manually make changes.

Working with XML source

You can view and make changes directly to your XML source from the Deployment Script Editor. If you know XML well and are comfortable with DB2 Change Management Expert, use the XML Source page as a quick way to make your changes.

To view and change your XML source, complete the following steps:

1. On the Overview page in the Deployment Script Editor, click the **XML Source** tab to display the XML Source page. On the XML Source page, you can:
   - View the XML source for your changes
   - Directly edit and change the XML source file without going through the DB2 Change Management Expert process
2. Click and type directly onto the XML Source page to edit and change your XML. The color coding indicates syntax checking for your XML.
3. Save the changes in the XML Source page by clicking **File** → **Save**.

Any changes that you make in the XML Source page will be deployed against the target database when you deploy your changes.

Refreshing a base model

Refreshing your base model will synchronize it with the current state of the database catalog. However, synchronizing the base model with the catalog does not automatically incorporate differences between the catalog and base model into the target model.

Important: Always check that the base model is in synch with the catalog before you deploy your changes. The database might have been changed between the time that you created your change commands and the time that you want to deploy those change commands.

The option to **Check if the base model has changed** is selected by default in the Deploy Change Commands wizard. You should not clear this check box unless you know that the base model and catalog are in synch.

If you update the base without forward fitting the changes into the target model, you run the risk of overwriting other changes.

Tip: After you refresh the base model, return to the Data Models page and click **Compare Base and Target Models**. Forward fit any changes from the base model into the target model.

To refresh the base model of your database, complete the following steps:

1. From the Deployment Script Editor, click **Refresh Base Model**.
2. In the Specify User Information window, specify your DB2 user ID and password, and click **Next**.
3. On the Select Schema page, select the schema that you originally selected to create your models, and click **Finish**.

   The Last Action field on the Overview page of the Deployment Script Editor will display **Refreshed Base Model**.

A status bar is displayed that indicates the progress of the refresh process.

Your base model now matches the most current state of the database catalog. If there is a difference with the previous base model, regenerate your change commands.

For more information on what to do if the base model is out of synch with the database catalog see the Common Problems topic.

Chapter 8. Deployment

These topics provide step-by-step instructions for using the DB2 Change Management Expert deployment script to specify and deploy changes to your target database.

The deployment process

The deployment script is the central resource in the deployment process. When you deploy a deployment script you are issuing commands to change a DB2 database.

The following figure illustrates the process of deploying changes to a target database:

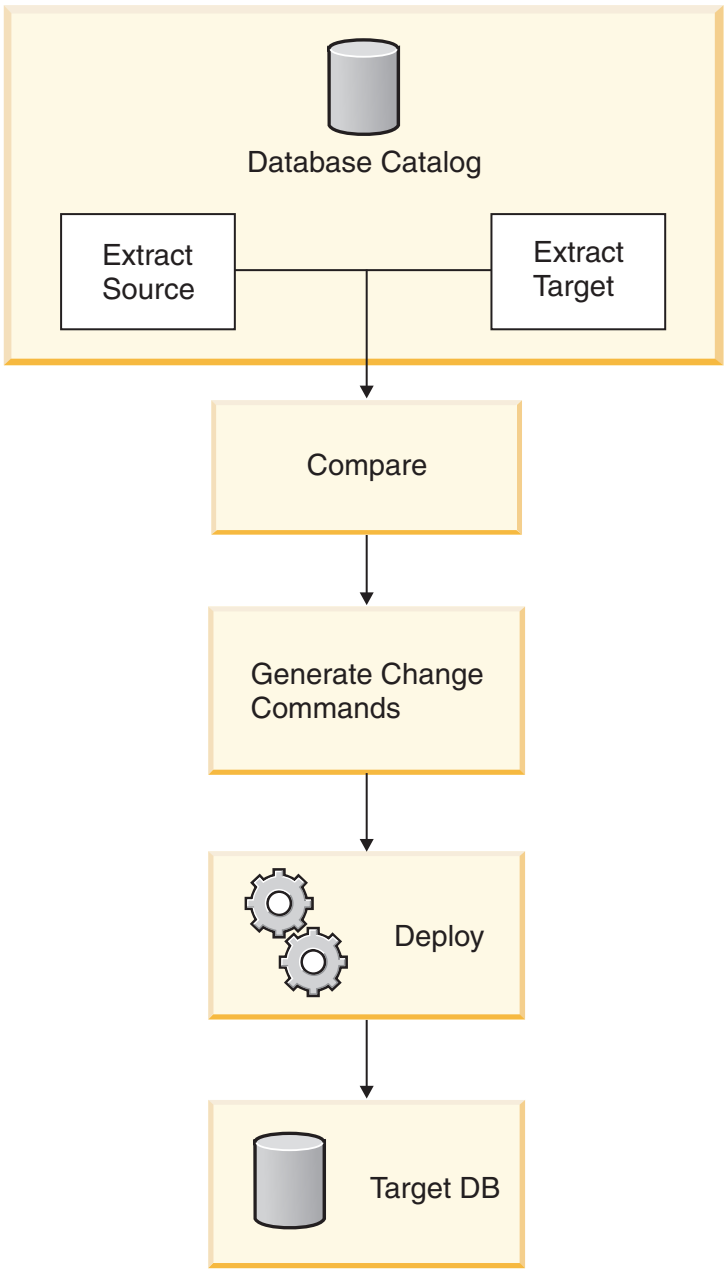

Figure 10. The process of deploying changes to your target database

The deployment script binds the change management resources together and organizes those resources into a coherent form. You use the deployment script to deploy changes to your DB2 databases.

The following figure illustrates the process of using a deployment script to deploy your changes to a database catalog.

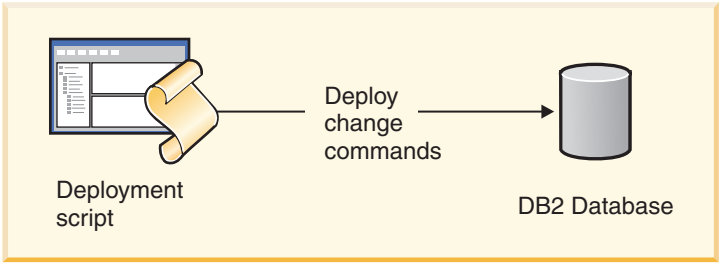

Figure 11. Using the deployment script to deploy your changes

The deployment script describes specification resources, change commands, and metadata about how the change will be issued.

Specification resources

Specification resources are the models and deltas that make up a change. The most important of these resources are the base and target models of the database. Another important resource are source models. Source models contain changes to migrate as part of the change and can be associated by using the deployment script.

The goal of the specification phase is to create the post-condition of the schema after the change is implemented. Part of that post-condition might include migrating objects from other database catalogs or integrating changes that are specified by a delta. These additional resources can be associated with the change. By associating additional resources with the change you can:

- Verify your work
- Help others understand the motivation for the change
- Historically tie resources together at points in time

Change commands

Change commands describe how the change will be implemented. The deployment script enables you to order these commands and provides wizards and editors to manipulate the commands.

Metadata

The deployment script provides metadata about the change. Use the Overview page of the Deployment Script Editor to specify the metadata and resources that are pertinent to the change, such as the target database connection, models, and change command files.

Deploying changes to a database

Deploying changes to a database is the process of changing the database catalog to reflect the conditions that you specified in the model specification phase. Deployment affects the database catalog, not the models.

To deploy changes to a database, complete the following steps:

1. Create a deployment script. The New Deployment Script wizard guides you through the process of creating a deployment script. When you create a deployment script, the wizard prompts you to create a project and automatically creates a base and target model of the database.
2. Make your changes to your target model. Models can be modified from the **Data Models** tab in the Deployment Script Editor.

Optional: Compare your base and target models. Click **Compare Base and Target Models** to display the Comparison Editor. Use the Comparison Editor to verify that the proposed changes will produce the desired result.

3. Generate your change commands from the **Change Commands** tab in the Deployment Script Editor.

   The Generate Change Commands wizard also helps you to generate data preservation and DB2 maintenance commands, which include import, export, authorization, rebind, runstats, and flush package cache commands.

   Optional: You can use the Generate Change Commands wizard to create undo commands in case you need to undo your changes. You can also use the **Deploy now** option in the Generate Change Commands wizard to quickly deploy your changes. Check the **Deploy now** option on the Merge page to automatically start the Deployment wizard. Only use the **Deploy now** option if you are confident that all of your changes are correct and you are aware of the impact of those changes.
4. Use the deployment script to verify certain issues before deploying your changes. You should verify that:
   - The base model has not changed. If the base model has not changed, you can continue to deploy your changes to the database. If the base model has changed, you can choose to ignore the change and continue deploying your changes, or you can investigate the updated base model.
   - A connection to the database exists.
   - The change commands are correct.
5. Deploy changes to the database catalog. DB2 Change Management Expert automatically manages the deployment of your changes and reports the results.

   In addition to using the deployment script, you can also deploy changes one at a time by using the projects in the Data Project Explorer view.
   a. Select a script file.
   b. Right-click the script file and click **Run SQL** from the context-sensitive menu. The script file extension should be .chx because the **Run SQL** option has certain restrictions when you use it with .sql and .ddl commands. For more information about the Run SQL restrictions see the Restrictions topic.

The changes that you have defined are deployed to the database catalog.

Deploying change commands from the Deployment Script Editor

Change commands can be deployed from the Deployment Script Editor and issued against a DB2 database catalog.

Prerequisite: You must have a working connection to your existing databases and models of those databases in the Data Project Explorer before deploying your changes.

To deploy changes to the database catalog, complete the following steps:

1. From the Deployment Script Editor, review and verify the information for your deployment script.
2. Click **Deploy Changes to the Target Database** from the Overview page or the Change Commands page to start the Deployment wizard.
3. Complete the steps in the Deployment wizard to deploy your changes.

The Deployment wizard checks to see if your base model matches the database catalog. If the base model does not match the catalog the change might not deploy correctly.

For more information on what to do if the base model is out of synch with the database catalog see the Common Problems topic.

Your changes have now been issued to the DB2 database catalog. The DB2 database catalog is updated and contains your changes. You can view the deployment results for each change command in the Data Output view. DB2 Change Management Expert also creates a deployment log file.

DB2 Change Management Expert manages the deployment and records the model version and other information for audit purposes.

Deploying changes to multiple databases

DB2 Change Management Expert provides the ability to deploy changes to multiple databases simultaneously. This functionality is called *multiple provisioning*. Deploying changes to multiple databases in a single operation can save time and increase productivity.

Prerequisite: You must have connections to all of the databases to which you want to deploy your changes in the Database Explorer.

Important: Before you deploy changes to multiple databases, validate those changes thoroughly on a test system.

The following figure illustrates the process of deploying changes to multiple databases from the deployment script.

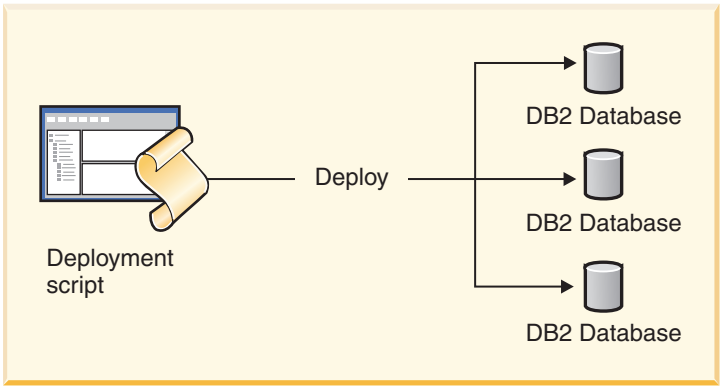

Figure 12. Deploying changes to multiple databases

Restriction: Undo scripts are not generated when you deploy to multiple databases.

To deploy your changes to multiple databases, complete the following steps:

1. Open the deployment script that you want to work with. Select **Deploy** → **Deploy to Multiple Databases** to display the **Multiple Deployment** tab on the Deployment Script Editor.
2. On the Multiple Deployment page, click **Add** to open the Add Connections window. From the list of connections, select the connection that you want to add. For example, select **SAMPLE** from the list of active database connections that is displayed.

You can add connections to any of the databases that are listed in the Add Connections window. However, the maximum number of connections that DB2 permits is eight. The eight database limit is the default for the number of concurrently active databases on a single DB2 instance. If the connections are to different DB2 servers or different instances on the same server, or if the you have changed the NUMDB parameter, this limit might be higher.

You can use a wizard to generate the data-preservation and maintenance commands that can be deployed against the selected database connection. To access the wizard, select a connection, right-click and then select **Generate commands**.

In addition, if you select a non-active connection, a dialog will display during deployment and prompt you for your user ID and password to connect to that database.

Repeat this step for each additional database that you want to add.

3. Click **Deploy Change Commands to All Connections** to deploy your changes to all of the database connections in your Connections List at one time. DB2 Change Management Expert deploys your change commands to the databases in the list.

   Tip: The status bar in the lower right corner displays your deployment progress.

   Optional: If you have many database connections in the Connections List, you can select one of them at a time and click **Deploy** to deploy your changes to the selected database only.
4. On the Multiple Deployment page, click **Open the Deployment Log File** to review the deployment log file. This log file contains the results of your change commands on the selected databases.

A deployment log file is shown in the following example:

```
!SESSION Sep 06, 2005 17:16:08.288 -----------------------------------------
eclipse.buildId=M200503110845
java.fullversion=J2RE 1.4.2 IBM Windows 32 build cn142-20040926 (JIT enabled: jitc)

!ENTRY  Sep 06, 2005 17:16:08.288
!MESSAGE Executing " ALTER TABLE CME.PROJECT  ADD COLUMN "test" VARCHAR (32) " for
Database: SAMPLE
SUCCESSFUL!!!
```

Your changes have now been deployed to all of the databases that you specified.

Deploying undo commands

If a change to the target database fails or produces an undesirable result, you can roll back that change to recover the previous database. DB2 Change Management Expert provides the capability to undo your changes. The undo capability is enabled by default.

Prerequisite: Undo commands must exist in the deployment script before you attempt to deploy them. For information about creating undo commands see the “Generating undo commands” on page 88 topic.

Restriction: You can deploy undo commands to only one database at a time.

To undo changes that you have promoted to your target database, complete the following steps:

1. On the Overview page in the Deployment Script Editor, click **Deploy Undo Commands** to start the Undo wizard. The Undo wizard contains the base model information, target model information, and default options that are necessary to restore the original database. In the Undo wizard, you can check your connection, enter your user ID and password, verify your change commands, and check that your deployment script contains the proper authorizations.

   You can also deploy your undo commands by clicking **Deploy Undo Commands** from the Undo Changes page of the Deployment Script Editor.
2. Click **Finish** to undo the changes that you have made to the target database.

The target database returns to the state that it was in before you deployed your change. You can view the undo commands that were deployed in the Text Editor. You can also view the deployment results for each undo command in the Database Output view.

The Last Action field on the Overview page of the Deployment Script Editor will display **Undo**.

Chapter 9. Auditing

Eclipse Team integration is a critical component to DB2 Change Management Expert auditing. After a change has been made, the DB2 Change Management Expert project, including all of the contained resources, should be checked into library control and tagged or labeled.

Library control systems, that plug in to Eclipse such as IBM Rational Clear Case, provide the best integration with DB2 Change Management Expert. However, because DB2 Change Management Expert stores all of its data files and folders on the local file systems, even source code control systems that do not integrate with Eclipse can be used to manage DB2 Change Management Expert resources. Changes are represented by DB2 Change Management Expert projects at particular points of time.

Archiving Data Design Projects

You can archive your Data Design Projects by using the Eclipse Team capabilities. Archiving your projects is useful to track and keep a record of your changes.

To archive your Data Design project, complete the following steps:

1. Share your Data Design project by using the Eclipse Team functionality. For more information about Eclipse Team, see the Eclipse Help.
2. After you deploy a change, save, commit, and check in the entire Data Design Project including all of the associated resources. The manner in which you check in your project will depend on the specific library control settings of your Eclipse environment.
3. Tag or label all the resources; assign a description of what is being archived in the library control system.

Repeat this process after each new change has been deployed.

Extracting Data Design Projects to audit changes

You can use your Data Design Projects to audit the changes that have been made to your databases.

To extract your projects and to review the changes that have been made, complete the following steps:

1. Extract the label or tag for the specific change that needs to be audited.
2. Open the Deployment Script Editor to audit the change.
3. Review the metadata and resources that are defined in the deployment script to audit the following aspects of the change:
   - Source models
   - Base and target models
   - All change commands
   - A log of the command execution
   - The scripts that are necessary to undo the change
4. Compare your base and target models to determine the database objects that were changed.

5. Review the change commands to see the actual DDL, utilities, and DB2 commands that are executed as part of the change.
6. Review the connection information that shows the live database from which the deployment script was developed.
7. Use the Last Action field to determine the last action that was completed with this deployment script. You can find the history of the actions that were completed in the XML for the deployment script on the **XML Source** tab of the Deployment Script Editor. For example, if the deployment script has been deployed, you will see the following:

```
<history>
    <event id="deployed" />
</history>
```

The relationship between Data Design Projects, databases, and library control

You can use library control in different ways to manage your database projects.

You can use a Data Design Project to track the life cycle of a database. The project can be shared by using the Eclipse team functionality so that several database administrators (DBAs) can collaborate on a change. Once a change is deployed, the resources will usually be committed to library control and tagged or labeled. The tag or label can be used to get back to the save point. For example, to undo or audit the change.

In more complex databases, a Data Design Project can be used to mange the life cycle of a particular database application. In some shops, tables or schemas are broken up and managed by particular DBAs or DBA teams. Data Design Projects can be adapted to match these environments. In this way, a single database can be broken up and managed by several Data Design Projects.

A single Data Design Project can be used to manage multiple databases if those databases are essentially copies of a primary database. This is referred to as *multiple provisioning* which means that the change is first constructed for a single database and is then deployed to multiple databases.

Using a library control system to back out changes

You might want to back out multiple changes after they have been deployed. You can use your library control system to undo those changes.

To back out multiple changes, complete the following steps:

1. Extract the label or tag for your latest change.
2. Use the Undo section of the Deployment Script Editor to roll back your changes.
3. Extract the next, most recent change and repeat these steps until all of the changes have been reversed.

After you back out your changes, your database catalog is returned to the particular state that you wanted.

Important: If data preservation was enabled, the data files that will be imported after the undo DDL were created when the change was deployed. If updates were made to the database since the change that you are trying to undo was made, take care not to overwrite important changes.

Depending on the situation, you might need to take additional action to ensure that the data being imported is current.

Chapter 10. Troubleshooting

These topics provide information about diagnosing and solving problems that you might encounter when installing or using DB2 Change Management Expert.

Common problems

There are some common problems and issues that you should be aware of when using DB2 Change Management Expert.

Finding the DB2 instance name

Change commands that require the DAS will request the DB2 instance name. With DB2 Version 8.2 fix pack 11 and higher, DB2 Change Management Expert will detect the instance name from the connection. However, with earlier versions of DB2, you will have to look at the server to determine the instance name. On Windows systems, the instance name is typically DB2. On UNIX systems, the instance name is typically the ID that DB2 was installed under. For example, db2inst82. Installation environments can vary.

Finding the JDBC type 4 driver

In DB2 Version 8, the DB2 type driver is called db2jcc.jar. It can typically be found under the DB2 installation directory in the Java subdirectory. To use the type 4 driver, you will also need a license file in your CLASSPATH. You can enter both of these values in the New Connection wizard by clicking **Browse** and selecting the files.

An additional problem related to the JDBC driver involvesh migrating workspaces between different versions of DB2 Change Management Expert. The connection information stored in the Database Explorer contains driver information that is not portable. When you migrate a workspace you must delete and recreate all of your custom connections in the Database Explorer to properly update the driver information.

Forward engineering

When you forward engineer changes, be sure to check the DDL for objects that are derived from other objects, such as views, materialized query tables, and aliases, to ensure that the correct dependent objects will be resolved. DB2 stores the CREATE statements for some derived objects in the catalog exactly as they were specified. If the creator of these statements did not fully specify schema names, the current login is assumed. The CREATE statement might fail if the current login is different during the forward engineering process. To resolve this problem either use the same creator ID when you change objects, fully specify the dependent objects in the DDL, or add the syntax `SET CURRENT SCHEMA` before the command.

For example, if you generate a CREATE VIEW command like `CREATE VIEW MYSCHEMA.VIEW AS SELECT * FROM MYTABLE;` and MYTABLE actually belongs to a schema called DB2, then when this command is deployed, DB2 will look for MYTABLE under the current login name. You might get an error saying *loginname*.MYTABLE is not defined. If you add a `SET CURRENT SCHEMA=DB2` statement before the `CREATE VIEW MYSCHEMA.VIEW AS SELECT * FROM MYTABLE` syntax, then all of the referenced objects will be resolved correctly.

When DB2 Change Management Expert is calculating the correct order for the CREATE statements of derived objects that contain names of other

derived objects without fully qualified schema names, it is possible that DB2 Change Management Expert will put those statements in an incorrect order. To correct this problem you might need to:

- Ensure that the login that is used to generate the Change Commands is the same as the login that was used to create the database objects.
- Open the generated Delta DDL in the Change Command Editor and put the statements in the correct order.

The base model is out of synch with the catalog when deploying the change

The Deploy Change Commands wizard detects whether the base model matches the current database catalog. This is a good safeguard because the database catalog might now be different from the data model that this script was constructed to change. If a difference is detected, complete the following steps:

1. Click **Cancel** to close the Deploy Change Commands wizard.
2. Return to the **Overview** tab of the Deployment Script Editor.
3. Click **Refresh Base Model** to refresh your base model. This will update your base model with any changes that have been made to the database catalog.
4. Click **Compare Base and Target Models** to view the differences between the current catalog and the target model. Forward fit the changes as necessary.
5. If you have made any changes, click **Generate Change Commands** to regenerate your change commands. At this point, you are ready to deploy your changes to the database catalog.
6. Click **Deploy Changes to the Target Database**.

Testing server connectivity

You can test your server connectivity by creating a new connection in the Database Explorer view. You must be in the Data Perspective for that view to be visible.

Model validation

If DB2 Change Management Expert detects an error in the physical data model, it is recorded in the Problems view. If the Problems view is not visible, you can open it from the main menu by selecting **Window → Show View → Problems**. Or you can open it by typing Alt+shift+Q,X from the keyboard. Double-click the problem to view more details.

Deployment script validation

If DB2 Change Management Expert detects an error in the deployment script, it is recorded in the Problems view. If the Problems view is not visible, you can open it from the main menu by selecting **Window → Show View → Problems**. Or you can open it by typing Alt+shift+Q,X from the keyboard. Double-click the problem to view more details.

Change command validation

If DB2 Change Management Expert detects an error in a change command file, it is recorded in the Problems view. If the Problems view is not visible, you can open it from the main menu by selecting **Window → Show View → Problems**. Or you can open it by typing Alt+shift+Q,X from the keyboard. Double-click the problem to view more details.

Problems deleting resources in the Data Project Explorer

At times, you might still see resources even after you have deleted them in the Data Project Explorer. You might have to close and reopen the editor to

refresh it. The editor can be opened from the main menu by selecting **Window → Show View → Data Project Explorer**.

There is not enough screen real estate to use the Comparison Editor

The Comparison Editor can take up quite a bit of space. To maximize the Comparison Editor, double-click the tab of the open editor. On Windows systems, the quick key is Alt+-,X.

Finding data objects

If you know the physical data model in which the data object is located, open that model in the Data Model Editor. In this editor, you can type text in the filters field to filter the list and quickly move to an object, use Find (Ctrl+F) to locate an object, or drill down to locate an object. If you do not know what model the object is in, use Search (Ctrl+H). The results will be displayed in the Search View. Double-click the object to open it in an editor.

The Run SQL wizard depends on the extension

Change command files in the SQL Scripts folder must have a file extension of `chx`. In addition to SQL statements, the change command files can contain some DB2 commands and utilities. The different actions for the Run SQL wizard depends on the extension. Trying to run DB2 commands and utilities from a script file with an extension other than `chx` can cause errors.

Deploying export and import commands

Whenever there is an error while deploying export or import scripts, the SQL code is always -22220, which means that the DAS (DB2 Administration server) has encountered a script error. The exact reason for the error is specified by the error code. The codes will appear in the data output view. The detailed description for a particular error code can be found in the DB2 Message Reference manual.

Import command errors

Error code 3088

The source column that was specified to be loaded into the database column name is not compatible with the database column, but the database column is not nullable.

```
Source table tblA:  col1 int, col2 int, col3 varchar(5), col4
num(6, 2)
Target table (table being loaded) tblB: col2 int, col3 varchar(5),
col4 num(6, 2) not null.
```

If you selected all of the rows during the export, then, due to a data type mismatch, the VARCHAR values will fail to be inserted into col4 of the target table during the import.

This can be fixed in the following ways:

- Edit the data file.
- Use the customization wizard to select the last three columns from tblA, in the export statement. Use the customization wizard to add column functions as needed.

Error code 3124

The field value in row ″2″ and column ″3″ cannot be converted to a PACKED DECIMAL value, but the target column is not nullable. The row was not loaded.

Error code 3196

The input file was not found. If the data file is located on a remote machine where the export command was run, then you must issue

the import command again from the Data Project Explorer. Ensure that the data file has been transferred to the machine where the import command is run, then select the import command file, for example xxx_import.chx, from the Data Project Explorer, right-click and select **Run SQL** to complete the import.

Export command errors

Error code 3025
A parameter that specifies a file name or path is not valid. Because the import and export commands will most likely be run on remote systems, DB2 Change Management Expert does not verify that the specified path exists. To avoid data loss, ensure that the path is valid. You can also run the export commands from the project explorer before you deploy the scripts.

Error code 3304
The table does not exist. It is important to issue the export commands before the tables are dropped. The order of the commands needs to be verified on the Change Commands tab of the Deployment Script Editor before deploying the scripts. The default data file type is DEL (Delimited ASCII, for exchange with many database managers and file managers). You can change the file type to IXF if necessary.

If the error code does not appear in the data output view, it can be found in the Administration notification log file db2dasdiag.log. Details such as the location of this file can be found in the DB2 Information Center .

Editing data preservation entries
The double-click action does not work well on Linux systems. You can use the quick key access to edit your data preservation entries instead. Type the 'e' key or type Shift-F10 to edit the selected entry. This will put the focus on the cell editor of the first column (For example, Export Commands). Pressing the 'Enter" key will move the focus to the cell editor of the second column (Import Command).

Data migration
You use the Generate Data Migration wizard to either generate change command scripts that are to be run manually or that are embedded into the deployment script.

The deployment script includes both the Generate Change Commands wizard and the Generate Data Migration wizard. When you need both data migration and data preservation and are managing the data migration with the deployment script, you must merge the import and export commands manually by using the file merge wizard page. The order in which you invoke the wizards does not matter, because both wizards have the file merge wizard page.

Syntax errors in the Change Command Editor
When you open change commands in the Change Command Editor, syntax errors might be caused by the level of the DB2 version that DB2 Change Management Expert uses to parse the DDL. By default, DB2 V9.1 is set as the level for parsing DDL. For example, if you change the default to a lower release of DB2 and then open change commands that include syntax that was introduced for DB2 V9.1, you will get errors. You can click **Window** → **Preferences** and then expand **Data**, click **Change Management Options**, and set the value in the **Default DB2 version** field back to V9.1.

When you apply change commands to a model, DB2 Change Management Expert always uses the level of the DB2 version for the model for parsing the DDL.

Deploying change commands fails due to the maximum number of lock requests

If you receive the error code SQL0912N while deploying your change commands, you can solve the problem by:

- Committing more often
- Increasing your lock size

Important: Stop and restart DB2 before deploying your change commands. If you do not restart DB2, you might still receive the SQL0912N error.

Deploying change commands fails because of too many active database connections

DB2 limits the number of database connections that can be active at one time. The database manager configuration parameter NUMDB determines the maximum number, which is 8 by default. After you reach the maximum number of connections in the Database Explorer, you will not be able to create a new connection. Therefore, when you attempt to deploy your change commands, they might fail. The error message from DB2 is ″error in the script file.″ This error is caused by having too many connections to DB2. Delete one of your unused or less-used connections from the Database Explorer or consider increasing the size of NUMDB.

The Database Explorer does not automatically reflect database changes

Database changes are not automatically reflected in the Database Explorer. To see the updates, you need to refresh the connection for the database.

Restrictions

The following restrictions apply to DB2 Change Management Expert.

- If you receive the following error message when you attempt to rebind packages with a RI database, the DB2 CLI packages need to be bound with an appropriate value for the bind parameter CLIPKG:

  ```
  SQL0805N error, Package.Nullid.SYSLH203 was not found
  ```

  For more information about this problem and the solution, see the IBM Software Support Web site.
- Database authorizations are generated by using the db2look utility. Results may vary depending on that tool. Review the authorization commands carefully. Authorizations generated for undo processing contain all the authorizations returned by the db2look utility. If an object exists, authorization commands will succeed (even if they are not required). For this reason, DB2 Change Management Expert runs all authorizations. Authorization preservation for a change starts with the GRANT and REVOKE statements that are returned by db2look. However, the object of each authorization command is cross checked in the target model. If the authorization object does not exist in the target it is removed; the GRANT or REVOKE statement is removed from the script. The db2look utility has known problems when generating authorization commands for table spaces. You might need to review and correct the authorizations in that case.
- DB2 Change Management Expert does not support these features or objects that were introduced with DB2 V9.1:

  - Label-based access control (LBAC)
  - XSR objects
  - Domains (for example, CREATE DOMAIN)
  - Character sets (for example, CREATE CHARACTER SET)
  - Collation (for example, CREATE COLLATION)
  - Ability to invoke external table functions in parallel (for example, CREATE FUNCTION with DISALLOW PARALLEL or ALLOW PARALLEL EXECUTE ON ALL DATABASE PARTITIONS RESULT TABLE DISTRIBUTED COLLATION)
- DB2 Change Management Expert does not fully support DB2 method objects. In some cases, physical data models will not be correctly reverse engineered from the catalog. In addition, DB2 Change Management Expert does not fully forward engineer method objects. When you are managing changes to method objects, you should check your physical data models and change command files to ensure that the commands are correct.
- DB2 Change Management Expert does not forward engineer federated objects such as servers, wrappers, function mapping, or nicknames.
- DB2 Change Management Expert does not preserve some attributes for tables and views when you create a model from the DB2 catalog. Thus, if you later forward engineer DDL for the model, the attributes are not included. The attributes that are not preserved are:
  - Tables
    - CCSID ASCII or CCSID UNICODE
    - SECURITY POLICY
    - DB2SECURITY LABEL
    - COLUMN SECURED WITH
    - DATA CAPTURE
    - COMPRESS SYSTEM DEFAULT
    - ORGANIZE BY KEY SEQUENCE
  - Views
    - ENABLE QUERY OPTIMIZATION or DISABLE QUERY OPTIMIZATION
    - WITH CASCADED/LOCAL CHECK OPTION
    - WITH ROW MOVEMENT or WITH NO ROW MOVEMENT
- DB2 Change Management Expert does not properly recognize labeled durations that use the MINUTES keyword. The Change Command Editor marks the keyword as an error. The Deployment Script Editor is unable to convert DDL that contains the MINUTES keyword into individual change commands, and DB2 Change Management Expert is unable to deploy DDL that contains the MINUTES keyword.

  To work around this problem, change occurrences of the MINUTES keyword to MINUTE.
- DB2 Change Management Expert will not deploy your changes correctly if you have changes that involve a materialized query table (MQT) that is based on an alias that is based on another alias that is again derived from another alias that is based on a real table. You might want to consider deploying changes in this type of situation outside of DB2 Change Management Expert. This situation is uncommon.
- DB2 Change Management Expert relies on db2look functionality to maintain object authorizations during a change. In some releases of DB2, db2look lacks support for certain database objects. A known db2look problem exists when you generate grants for table spaces. You should review the authorization scripts thoroughly before you issue them.

Chapter 11. Working with diagrams

You can use diagrams to visualize and edit data projects.

Opening and deleting diagrams

You can open or delete diagrams from the Data Project Explorer and open read-only overview diagrams from the Database Explorer.

To open or delete a diagram from the Data Project Explorer or Database Explorer:

Complete one of the following options:

- To open a diagram from the Data Project Explorer or the Database Explorer, double-click the diagram.
- To delete a diagram from the Data Project Explorer, right-click the diagram and then click **Delete**.

Changing the default options for the appearance of diagrams

You can change the default options for the appearance of diagrams.

To change the default options for the appearance of diagrams:

1. Click **Window** → **Preferences**.
2. In the Preferences window, expand **Modeling** and click **Appearance**.
3. Select new default options.
4. Click **OK**.

Changing diagram properties

You can change diagram properties in the Properties view for Information Engineering (IE) notation, Unified Modeling Language (UML), or storage diagrams.

To change diagram properties:

Click an empty space on the diagram to access the Properties view for the diagram. In the Properties view for the diagram, you can change the following properties:

Option	Description
General tab for IE, UML, or storage diagrams	You can edit the name.
General tab for IE diagrams	You can view the notation style.
Filters tab for IE diagrams	You can filter elements, including compartment display options, foreign key relationship display options, table display options, and column display options. You can also filter these elements for a specific data object by right-clicking a data object on the diagram surface, selecting **Filter**, and selecting one of the filter options from the submenu.

Option	Description
Format tab for IE diagrams	You can change formatting options, including table and view background colors, foreign key column foreground color, and implicit foreign key relationship line color.

Working with shapes and connectors

You can change the appearance of shapes and connectors and reposition them to meet your data project needs.

Changing the font and font color for shapes

You can change the font and font color for individual shapes to add visual interest or to convey conceptual groupings between them.

To change the font or font color for shapes:

1. Select a shape or a group of shapes.
2. Right-click and select **Format** → **Font**.
3. On the Font page, from the **Font** list, select a font.
4. From the **Font style** list, select a font style.
5. From the **Size** list, select a size.
6. From the **Color** list, select a font color and click **OK**.

Tip: For Information Engineering (IE) notation diagrams, you can also use the Properties page to semantically format colors based on type, including view, table, implicit foreign key relationship, and foreign key columns.

Changing the background color for shapes

You can change the background color for individual shapes to add visual interest or to convey conceptual groupings between them.

To change the background color for shapes:

1. Select a shape or a group of shapes.
2. Click **Diagram** → **Fill Color** and click a color.

Tip: For Information Engineering (IE) notation diagrams, you can also use the Properties page to semantically format colors based on type, including view, table, implicit foreign key relationship, and foreign key columns.

Applying the appearance of a shape to other shapes

You can apply the appearance of a shape to other shapes based on the primary selection so that the group of shapes looks identical in a diagram.

To apply the appearance of a shape to other shapes:

1. Select the shape that you want to copy the appearance properties of.
2. Hold down the shift key to select one or more shapes that you want to apply the appearance properties to.
3. Click **Diagram** → **Apply Appearance Properties**.

Making shapes the same size

You can change multiple shapes to make them the same height and width, same height, or same width, to improve the readability and organization of a diagram.

You can resize selected shapes to match the primary shape that has filled sizing handles. The selection method that you use determines which shape is the primary one that has filled sizing handles:

- If you select shapes by clicking Select All, the shapes are resized to match the last shape that you added to the diagram.
- If you select shapes by using the selection tool in the palette, the shapes are resized to match the shape that is nearest the selection border's starting point.
- If you select shapes by pressing Ctrl or Shift, the shapes are resized to match the last shape that you select.

To make shapes the same size:

1. Select the shapes that you want to resize and then select the primary shape.
2. Right-click a selected shape.
3. Click **Format** → **Make Same Size** and then click one of the following options:
   - **Both** to make shapes the same height and width.
   - **Height** to make shapes the same height.
   - **Width** to make shapes the same width.

Repositioning shapes

You can arrange all shapes or selected shapes, align shapes with other shapes, resize shapes, and move shapes in diagrams.

Automatically arranging all shapes and connectors

You can automatically arrange all shapes and connectors in data diagrams for better viewing results.

To automatically arrange all shapes and connectors:

1. Right-click an empty space in the diagram.
2. Click **Arrange All**.

The shapes and connectors in the diagram are automatically arranged in a hierarchical manner.

Arranging selected shapes and connectors

You can arrange selected shapes and connectors in diagrams for better viewing results.

To arrange selected shapes and connectors:

1. Select the shapes and connectors that you want to arrange.
2. Right-click a selected shape or connector.
3. Click **Format** → **Arrange** → **Selection**, to arrange selected shapes and connectors, or **Format** → **Arrange** → **Arrange Selected Connectors** to arrange selected connectors to minimize the bend point. The **Arrange Selected Connectors** action is only available when you select only connectors.

The selected shapes and connectors are automatically arranged in a hierarchical manner.

Aligning shapes

You can align shapes by their left, right, or center vertically or by their top, bottom, or middle horizontally in diagrams.

Tip: Selected shapes align to the primary selected shape. The primary shape has filled sizing handles. Grouped shapes (that are selected by the Marquee selection tool) are aligned by the boundary of the group, not by the individual shapes in the group.

Note: When you perform this action, objects might overlap on the diagram surface.

To align shapes:

1. Select the shapes that you want to align.
2. Right-click a selected shape.
3. Click **Format** → **Align** and click a type of alignment.

Resizing shapes

You can manually resize shapes in diagrams.

To resize a shape:

1. Select a shape or a group of shapes.
2. Complete one of the following steps:
   - To resize the shape in a specific direction, drag the appropriate sizing handle.
   - To resize the shape while maintaining the shape's center of view, press the Shift key and drag a sizing handle.
   - To resize the shape in aspect ratio, press Ctrl and drag a sizing handle.
   - To resize the shape while maintaining the shape's aspect ratio and center of view, press Shift+Ctrl and drag a sizing handle.

Changing the order of shapes in stacks

Shapes can be stacked on top of each other in layers in diagrams. You can move shapes one layer up or down in a stack. You can also bring shapes to the front or send them to the back of a stack.

To change the order of shapes in stacks:

1. Right-click a shape.
2. Click **Format** → **Order** and then click one of the following options:.
   - **Bring to Front** to bring the shape to the front of a stack.
   - **Send to Back** to send the shape to the back of a stack.
   - **Bring Forward** to move the shape one layer up in a stack.
   - **Send Backward** to move the shape one layer down in a stack.

Changing appearance of shape compartments

You can show or hide shape compartments, show or hide compartment titles, resize compartments, and collapse compartments in diagrams.

Showing and hiding compartments

You can show or hide compartments in shapes so that only the compartments that are important to you appear in diagrams.

To show or hide compartments:

1. Right-click a shape or a group of selected shapes.
2. Click **Filters** → **Show/Hide Compartment** and click a compartment to show or remove the check mark.

Tip: You can also use the Properties page for Information Engineering (IE) notation diagrams to show or hide all compartments in the diagram by type. For physical IE diagrams, you can show or hide by key, non-key, index, and trigger. For logical IE diagrams, you can show or hide by key and non-key.

Resizing compartments

You can resize shape compartments to make compartment contents more or less visible as you want.

To resize a compartment:

1. Select a shape.
2. Place the cursor over a line that separates two compartments until the four arrow sizing pointer appears.
3. Drag the division line to adjust the compartment ratios.

The compartments are resized. Scroll bars appear to accommodate the resizing, if necessary.

Collapsing compartments

You can collapse a compartment to a minimal size to allow more space for the other compartments in a shape.

To collapse a compartment:

1. Select a shape.
2. In the upper-left corner of the compartment, click the down arrow.

To restore the collapsed compartment to its original size, click the down arrow again.

Filtering compartment items

You can filter compartment items in Unified Modeling Language (UML) diagrams to reduce the complexity or increase the level of abstraction in diagrams. By default, compartment items are listed in the same order as they are in the Properties view and the Data Project Explorer or Database Explorer.

To filter compartment items by visibility style, selection, or by both visibility style and selection:

1. Right-click a shape.
2. Click **Filters** → **Sort/Filter Compartment Items**.
3. In the Sort/Filter Compartment Items window, select a compartment and complete one or both of the following steps:
   - To filter by visibility, under **Filter Criteria**, select the visibility style or styles of compartment items. Click **Move to the Left** (**<**).
   - To filter by selection, select or clear the check boxes for each compartment item that you want to make visible or not.
4. Click **Apply**.
5. Click **OK**.

Sorting compartment items

You can sort compartment items in Unified Modeling Language (UML) diagrams in ascending or descending alphabetical order to reduce the complexity or increase the level of abstraction in data diagrams. By default, compartment items are listed in the same order as they are in the Properties view and the Data Project Explorer or Database Explorer.

To filter compartment items by name, visibility style, or return type in an ascending or descending alphabetical order:

1. Right-click a shape.
2. Click **Filters** → **Sort/Filter Compartment Items**.
3. In the Sort/Filter Compartment Items window, click a compartment and complete one or more of the following steps:
   - To sort compartment items by visibility, in the table, click the **Visibility** column heading.
   - To sort compartment items by visibility while filtering by selection, select or clear the check boxes for each compartment item that you want to make visible or not.
   - To sort compartment items by name, in the table, click the **Name** column heading.
   -
4. Click **Apply**.
5. Click **OK**.

Sorting attributes

You can sort attributes in Unified Modeling Language (UML) diagrams by their types in ascending or descending alphabetical order.

You can sort attributes only when an attribute compartment has several items that can be sorted alphabetically.

To sort attributes by type:

1. Right-click a shape.
2. Click **Filters** → **Sort/Filter Compartment Items**.
3. In the Sort/Filter Compartment Items window, expand **All Compartments** and click **Attribute**.
4. In the table, click the **Type** column heading and Click **Apply**.
5. Click **OK**.

Changing the color of connectors

You can change the color of individual connectors to add visual interest or to convey conceptual groupings between relationships in a data diagram.

To change the color of a connector:

1. Select a connector or connectors.
2. Right-click **Format** → **Line Color** and click a color.

Tip: For Information Engineering (IE) notation diagrams, you can also use the Properties page to change the color of all implicit foreign key relationships.

Adding bend points to connectors

You can add bend points to connectors to make diagrams more space efficient and clear.

To add a bend point to a connector:

1. Select a connector.
2. Place the cursor at the point where you want to add a bend point until a plus sign (+) appears.
3. Drag the connector bend point to a new location.

Moving bend points on connectors

Diagrams are easier to read if connectors do not cross other connectors and shapes. You can use bend points to customize connector line routing to minimize crossed connectors in a diagram.

To move a bend point on a connector:

1. Select a connector.
2. Place the cursor over the bend point that you want to move until the moving pointer appears.
3. Drag the bend point to a new location on the connector.

Showing and hiding connector labels

You can show or hide labels on all connectors or selected connectors in a diagram that includes related elements and connectors.

To show or hide connectors labels, complete one of the following steps:

- To show all connector labels, right-click an empty space; then click **Filters** → **Show Connector Labels**.
- To hide all connector labels, right-click an empty space; then click **Filters** → **Hide Connector Labels**.
- To show connector labels on selected connectors, right-click a connector or selected connectors; then click **Filters** → **Show Connector Labels**.
- To hide connector labels on selected connectors, right-click a connector or selected connectors; then click **Filters** → **Hide Connector Labels**.

Tip: For Information Engineering (IE) notation diagrams, you can also use the Properties page to show or hide name, label, and referential integrity for foreign key relationships.

Selecting diagram elements

You can select all or a group of elements, shapes, or connectors in a diagram.

To select diagram elements:

1. Right-click an empty space in the diagram.
2. Click **Select** and complete one of the following steps:
   - To select all the shapes and connectors, click **All**.
   - To select all the shapes, click **All Shapes**.
   - To select all the connectors, click **All Connectors**.

Tip: To select a group of elements, hold the shift key and click on the elements to be selected.

Finding diagram elements

You can use the find feature to locate diagram elements by searching for specific text in a diagram.

To find a diagram element:

1. Click **Edit** → **Find**.
2. Enter the text to search for in the Find field.
3. Select search options and click **Find Next**.

The next occurrence is highlighted. You can click **Find Next** again until all occurrences have been found.

Viewing diagrams

You can use the grid and rulers, zoom, outline view, and page breaks to better view diagrams.

Showing and hiding the rulers and grid

You can show or hide the rulers and the grid in diagrams.

To show or hide the rulers or the grid:

1. Right-click an empty space in the diagram.
2. Click **View** and then complete one of the following steps:
   - To show or hide the rulers, click **Rulers**.
   - To show or hide the grid, click **Grid**.

Snapping diagram elements to the grid

You can snap diagram elements to the grid in diagrams.

To snap diagram elements to the grid:

1. Right-click an empty space in the diagram.
2. Click **View** → **Snap to Grid**.

Changing zoom levels of diagrams

You can change the zoom level of a diagram to better view the diagram.

To change the zoom level of a diagram:

1. Right-click an empty space in the diagram.
2. Click **Zoom** and click a zoom option.

The diagram is enlarged or reduced according to the zoom option that you select. Zoom settings apply only to a specific diagram.

Navigating large diagrams

You can use the outline view to quickly find your location in a large diagram. The outline view shows the entire diagram with a small gray box that represents the viewing area.

To navigate a large diagram in the outline view:

In the outline view, complete one of these steps:

- Click a location on the diagram. The viewing area is changed to this new location.
- Click and drag the gray box to move the viewing area to a new location.

Showing and hiding page breaks

You can show or hide page breaks in diagrams. You can use page breaks to center a diagram on the minimum number of pages before printing the diagram.

To use page breaks, an open diagram must have shapes and connectors spanning over more than one default printer page. You must also have a printer installed on your system so that information about the printer and printer settings are available to the application for calculating page breaks.

To show or hide page breaks:

1. Right-click an empty space in the diagram.
2. Click **View** → **Page Breaks**.

Recalculating page breaks

You can recalculate page breaks after you make changes to data diagrams with page breaks.

To recalculate page breaks:

1. Right-click an empty space in the diagram.
2. Click **View** → **Recalculate Page Breaks**.

Chapter 12. Glossary

A

auditing
The process of keeping track of database changes to provide a record of the evolution of the database. Auditing provides a mechanism to track who performed what actions and helps prevent unknown or unanticipated access to the data.

apply The process of integrating change commands or changes into a model.

archive
The process of capturing data, definitions, and environment information from the archive source (expressed in SQL) and storing this information in an archive target; as well as deleting some or all of the source data. This process can be performed by running an unload task.

B

base model
A model that is a representation of a database before changes are applied and upon which the DB2 Change Management Expert deployment script is based. See also *model*, *model version*, and *target model*.

C

change commands
An ordered collection of operations such as DDL, DML, DCL, running utilities, bind, and rebind.

change specification
The process of defining a set of changes that you want to apply to a database. The process involves making changes to models of the database and analyzing the impact of those changes. After the changes are defined, they can be deployed to the actual database

comparison
The process of analyzing the differences between two models. Often, a model of a database that contains the proposed changes is compared to an earlier model of the database to understand the effects of the changes. Or, when the schema of one database is being changed to match another database, the model of the source database can be compared to a model of target database to determine the differences between the databases.

copying
Moving an object or objects where the entire internal structure is copied from the source database to the target database.

D

data design project
A container for storing the resources that are needed to plan, analyze, and make changes to a database. Projects map to directories in the file system.

data preservation
The process of specifying how to control data in the target database when there are changes to database structure or metadata.

database timeline
The state of a database at various points of time. Models of the database timeline can be compared by using the Compare With selection.

deploy
The process of issuing change commands against DB2 objects. For example, tables, table spaces, and triggers. Deploy is a step in the deployment process.

deployment
The process of changing the database to reflect the conditions that were specified in the change specification phase. Deployment encompasses the process of naming the target database, declaring the change commands that will be deployed against that database, and the steps to implement the change, deploy the change, and manage the undo process.

deployment script
The set of changes that you want to make to your database. The deployment script binds the change management resources together and organizes those resources into a coherent form. It describes the specification resources and metadata about how the change will be issued. You use the deployment script to deploy the changes to your DB2 databases.

E

Environment.
See *tier*.

G

generate
The process of creating change commands .

I

impact analysis
The process of finding the dependencies that exist for an object. As changes are made to an object, a list of the related objects that might be impacted by the changes can be generated.

inoperative object
A DB2 object that is valid, but must be recreated to be used.

L

load providers
A data load provider generates a particular type of load command during data preservation. The 'IMPORT' provider generates the DB2 Import command and the 'LOAD' provider generates the DB2 Load command. You can specify the default Data Load provider in your preferences. You can override the default setting for any particular data load operation by choosing the load provider in the Generate Change Commands wizard.

M

migration
Moving an object or objects where only part of the internal structure is copied from source to target. The process of changing the catalog schema of one database to match another database. Migration can be initiated from

several different starting points. For example, source and target objects are compared and DDL generated to bring the target in line with the source. Database objects, attributes, or dependencies can be migrated within the same database or between database tiers.

migration analysis
Comparing and moving the differences between the model of one database to the model of another database when changing the catalog schema of one database to match another database.

model A representation of the database catalog (referring to database schema objects). A model can be stored in memory or stored on the local file system. See also *base model*, *model version*, *source model*, and *target model*.

model analysis
The process of analyzing a model to verify that it is compliant with the defined constraints.

model version
A type of model that provides a representation of the database catalog at a particular point in time. As a snapshot of the database at a particular point in time, a model version provides important auditing information.

multiple provisioning
The process of deploying changes to multiple databases at once.

P

project
See *data design project*.

R

refresh
The process of ensuring that the base model matches the most current version of the DB2 database catalog. It is possible that changes occurred to the database between the time the change commands were created and the time the change commands are to be deployed. Refreshing the base model provides the opportunity to discover any changes and, if necessary, re-create the change commands.

resources
The models and scripts that make up a change. The resources for a change reside in folders in the project for the change.

S

source model
A model that serves as the basis, or source, of change for another model.

T

target model
A model that is a representation of a database after changes are applied and upon which the DB2 Change Management Expert deployment script is based. See also *model* and *base model*.

tier Describes the database's purpose in the IT infrastructure. For example, the development rack, the test rack, or the production rack are all different database tiers.

U

undefined object
A DB2 object that is valid and operative, but is undefined at runtime.

unload providers
An unload provider generates a particular type of unload command during data preservation. The 'EXPORT_DEL' unload provider generates EXPORT commands with the DEL data file format. The 'EXPORT_IXF' unload provider generates EXPORT commands with IXF data file format. The HPU provider generates the High Performance Unload command. For the HPU commands to be deployed successfully, DB2 High Performance Unload for Multiplatforms product must be installed. You can specify the default Data Unload Provider to be used when you generate your Data preservation commands. You can override the default for any data preservation entry in the Generate Change Commands wizard.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program or service may be used. Any functionally equivalent product, program or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not 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 or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country 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.

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 Corporation
J46A/G4

555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

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 information 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.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify and distribute these sample programs in any form without payment to IBM, for 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.

Programming interface information

This book is intended to help you use DB2 Change Management Expert Version 1.

This book also documents diagnosis, modification or tuning information, which is provided to help you customize DB2 Change Management Expert.

Warning: Do not use this diagnosis, modification or tuning information as a programming interface.

Diagnosis, modification or tuning information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

Trademarks and service marks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

AIX
DB2
DB2 Universal Database
IBM
pSeries
xSeries
zSeries

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

Index

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

X

Program Number: 5724-L29

Printed in USA

SC18-9958-01