BMC Workbench for DB2 User Guide

Supporting
Version 12.1.00 of BMC Workbench for DB2
Version 12.1.00 of BMC High Speed Utilities for DB2
Version 12.1.00 of BMC Object Administration for DB2
Version 12.1.00 of BMC Performance for DB2 Databases
Version 12.1.00 of BMC Performance for DB2 SQL
Version 12.1.00 of BMC Recovery for DB2
Version 12.1.00 of BMC Large Object Management for DB2
Version 12.1.00 of BMC Utility Management for DB2

December 2016
Contacting BMC Software

Several methods are available for contacting BMC Software.

You can access the BMC Software website at http://www.bmc.com. From this website, you can obtain information about the company, its products, corporate offices, special events, and career opportunities.

United States and Canada

Address  
BMC SOFTWARE INC  
2103 CITYWEST BLVD  
HOUSTON TX 77042-2827 USA

Telephone  1 713 918 8800  
or  
1 800 841 2031

Fax  1 713 918 8000

Outside United States and Canada

Telephone  +01 713 918 8800  
Fax  +01 713 918 8000

© Copyright 2013-2016 BMC Software, Inc.

BMC, BMC Software, the BMC logo, the BMC Software logo, and other BMC marks are the exclusive properties of BMC Software, Inc. and are registered or may be registered in the U.S. and in other countries.

BladeLogic and other BladeLogic marks are the exclusive properties of BladeLogic, Inc. and are registered or may be registered in the U.S. and in other countries. All other trademarks or registered trademarks are the property of their respective owners.

Explorer is a trademark or registered trademark of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

DB2, IBM, IBM z, MVS, RACF, and z/OS are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

The information included in this documentation is the proprietary and confidential information of BMC Software, Inc., its affiliates, or licensors. Your use of this information is subject to the terms and conditions of the applicable End User License agreement for the product and to the proprietary and restricted rights notices included in the product documentation.

Restricted rights legend

U.S. Government Restricted Rights to Computer Software. UNPUBLISHED—RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure of any data and computer software by the U.S. Government is subject to restrictions, as applicable, set forth in FAR Section 52.227-14, DFARS 252.227-7013, DFARS 252.227-7014, DFARS 252.227-7015, and DFARS 252.227-7025, as amended from time to time. Contractor/Manufacturer is BMC SOFTWARE INC, 2103 CITYWEST BLVD, HOUSTON TX 77042-2827, USA. Any contract notices should be sent to this address.
Customer support

Support website
You can obtain technical support from BMC 24 hours a day, 7 days a week at http://www.bmc.com/support. From this website, you can:

■ Read overviews about support services and programs that BMC offers
■ Find the most current information about BMC products
■ Search a database for problems similar to yours and possible solutions
■ Order or download product documentation
■ Download products and maintenance
■ Report a problem or ask a question
■ Subscribe to receive proactive e-mail alerts
■ Find worldwide BMC support center locations and contact information, including e-mail addresses, fax numbers, and telephone numbers

Support by telephone or e-mail
In the United States and Canada, if you need technical support and do not have access to the web, call 1 800 537 1813 or send an e-mail message to customer_support@bmc.com. (In the subject line, enter SupID:yourSupportContractID, such as SupID:12345). Outside the United States and Canada, contact your local support center for assistance.

Before contacting BMC
Have the following information available so that Customer Support can begin working on your issue immediately:

■ Product information
  — Product name
  — Product version (release number)
  — License number and password (trial or permanent)
■ Operating system and environment information
  — Machine type
  — Operating system type, version, and service pack or other maintenance level such as PUT or PTF
  — System hardware configuration
  — Serial numbers
  — Related software (database, application, and communication) including type, version, and service pack or maintenance level
■ Sequence of events leading to the problem
■ Commands and options that you used
■ Messages received (and the time and date that you received them)
  — Product error messages
  — Messages from the operating system
  — Messages from related software
License key and password information

If you have questions about your license key or password, contact Customer Support through one of the following methods:

- Send an e-mail message to customer_support@bmc.com. (In the Subject line, enter SupID:yourSupportContractID, such as SupID:12345.)

- In the United States and Canada, call 1 800 537 1813. Outside the United States and Canada, contact your local support center for assistance.

# Contents

## About this book
- Related publications ................................................................. 9
- Conventions ................................................................................ 10
- Summary of changes ..................................................................... 10
- Syntax statements ........................................................................ 13
- Syntax diagrams ........................................................................... 14

## Chapter 1  Overview of BMC Workbench for DB2
- Overview of workspaces and perspectives ....................................... 18
- Overview of the BMC Workbench console ..................................... 19

## Chapter 2  Getting started
- Prerequisites ................................................................................ 23
  - Licensing .............................................................................. 23
  - Supported browsers ............................................................... 24
  - DB2 Privileges ....................................................................... 24
  - Component configuration ......................................................... 25
  - BMC product installation requirements ................................ 25
- Logging on to BMC Workbench for DB2 ....................................... 25
- Managing user access .................................................................. 27
- Connecting to DB2 subsystems ..................................................... 29
- Selecting a subsystem for the BMC Workbench repository .......... 32
- Managing a workspace layout ..................................................... 32
- Viewing messages and UIM properties ..................................... 33
- Preparing for maintenance .......................................................... 34
- Setting BMC Workbench options .............................................. 35

## Chapter 3  Managing workspaces
- Creating a workspace .................................................................. 39
  - Creating an entirely new workspace ...................................... 40
  - Creating a workspace from a template .................................. 41
  - Saving a workspace ................................................................ 43
- Opening a workspace ................................................................. 44
- Closing a workspace .................................................................. 45
- Removing a workspace ............................................................. 45
- Editing workspace properties .................................................... 46
<table>
<thead>
<tr>
<th>Chapter 4</th>
<th>Discovering DB2 subsystems and browsing catalogs</th>
<th>53</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discovering DB2 subsystems</td>
<td>53</td>
<td></td>
</tr>
<tr>
<td>Creating a new DB2 object view</td>
<td>53</td>
<td></td>
</tr>
<tr>
<td>Navigating through the DB2 catalog</td>
<td>57</td>
<td></td>
</tr>
<tr>
<td>Creating a DB2 statement cache view</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>Removing a view</td>
<td>59</td>
<td></td>
</tr>
<tr>
<td>Copying and pasting a view</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>Browsing the catalog</td>
<td>61</td>
<td></td>
</tr>
<tr>
<td>Navigating to an object</td>
<td>61</td>
<td></td>
</tr>
<tr>
<td>Viewing related objects</td>
<td>62</td>
<td></td>
</tr>
<tr>
<td>Viewing object properties</td>
<td>63</td>
<td></td>
</tr>
<tr>
<td>Exporting data</td>
<td>63</td>
<td></td>
</tr>
<tr>
<td>Copying data</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>Setting DB2 Navigator options</td>
<td>65</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 5</th>
<th>Working with commands</th>
<th>67</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supported commands</td>
<td>67</td>
<td></td>
</tr>
<tr>
<td>Generating a command</td>
<td>68</td>
<td></td>
</tr>
<tr>
<td>Generating and running Alter DDL statements</td>
<td>69</td>
<td></td>
</tr>
<tr>
<td>Editing text files in Scratchpad</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>Creating commands</td>
<td>73</td>
<td></td>
</tr>
<tr>
<td>Running a command from Scratchpad</td>
<td>73</td>
<td></td>
</tr>
<tr>
<td>Setting Scratchpad options</td>
<td>75</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 6</th>
<th>Tuning SQL</th>
<th>77</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tuning SQL workflow</td>
<td>77</td>
<td></td>
</tr>
<tr>
<td>Explaining SQL statements</td>
<td>78</td>
<td></td>
</tr>
<tr>
<td>Using What-If scenarios</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>Creating a What-If scenario</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>Creating and editing a What-If index scenario</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td>Comparing statements</td>
<td>88</td>
<td></td>
</tr>
<tr>
<td>Selecting a statement for SQL Analysis</td>
<td>90</td>
<td></td>
</tr>
<tr>
<td>Analyzing a statement from the DB2 statement cache</td>
<td>90</td>
<td></td>
</tr>
<tr>
<td>Analyzing a statement from a DB2 package</td>
<td>91</td>
<td></td>
</tr>
<tr>
<td>Analyzing an ad hoc statement</td>
<td>92</td>
<td></td>
</tr>
</tbody>
</table>
Performance perspective .................................................................................................. 93
    SQL tab .................................................................................................................. 96
    Indexes tab ............................................................................................................ 97
    Explain tab ............................................................................................................ 98
    Cost tab ................................................................................................................ 100
    Predicates tab ....................................................................................................... 100
    Compare tab ........................................................................................................... 101
    Related Objects tab ............................................................................................. 101
    Output tab ............................................................................................................. 103
Setting SQL tuning options ......................................................................................... 104

Chapter 7 Migrating statistics .................................................................................... 107
Migrating access-path statistics from the DB2 Navigator ......................................... 107
Migrating access-path statistics from a SQL Tuning session ..................................... 110
Setting statistic migration options ............................................................................ 113

Chapter 8 Displaying BMC utilities ........................................................................ 115
Creating a Utility Status view .................................................................................. 115

Chapter 9 Setting up NGT Utility Management ....................................................... 119
Adding an NGT Utility Management view ............................................................... 120
Managing the NGT Utility Management criteria settings ....................................... 121
Managing the NGT Utility Management exceptions settings ................................ 123
Managing the NGT Utility Management schedule settings ..................................... 125
NGT SQL language .................................................................................................. 128
    NGT SQL language elements .............................................................................. 128
    Features and functions of the NGT SQL language ............................................. 131
    Special registers that the NGT SQL language supports ................................... 134
    Constants that NGT SQL supports .................................................................... 136
    Functions that NGT SQL supports ..................................................................... 139
    Date and time durations ...................................................................................... 177
    Comparisons ....................................................................................................... 179
    NGT SQL language keywords ......................................................................... 180

Chapter 10 Comparing DB2 schemas ..................................................................... 185
Comparing two schemas ......................................................................................... 185
    Schema-comparison reports .............................................................................. 189
        Overview by Statement Type report ............................................................. 190
        Overview by Object Type report ................................................................. 193
        Details report ............................................................................................... 195
Setting Schema Management options .................................................................... 196
Chapter 11       Managing Backup and Recovery          203
Viewing recovery progress                                      203
Verifying recoverability                                     205
    Refreshing the recoverability information                   208
Estimating backup or recovery time                           208
    Estimating backup time                                    209
    Estimating recovery time                                  212
Setting Recovery Management options                          215

Chapter 12       Managing object sets                    217
Browsing to an object set                                     218
Creating and editing object sets                             219
Deleting an object set                                        223
Generating a list of resolved objects                        224

Chapter 13       Managing JES jobs                       227
Viewing JES jobs                                             227

Chapter 14       Managing data sets                    229
Viewing and editing a data set                               229
About this book

This book contains detailed information about the associated product or products. This preface explains the special conventions that the book uses, and how to access related publications.

If applicable, the preface also summarizes the major changes included in the latest release of the product.

Related publications

From the BMC Support Central website, you can use the following methods to access related publications that support your product or solution:


- View Quick Course videos (short overviews of selected product concepts, tasks, or features), which are available from the following locations:
  - Documentation Center (primary center and secured center)
  - Support Central (at http://www.bmc.com/support/mainframe-demonstrations)
  - BMC Mainframe YouTube channel (https://www.youtube.com/user/BMCSoftwareMainframe)


Products with online interfaces also offer online Help via the F1 key or, for graphical user interfaces (GUIs), via a Help button.
Conventions

This document uses the following special conventions:

- All syntax, operating system terms, and literal examples are presented in this typeface.

- Variable text in path names, system messages, or syntax is displayed in italic text: `testsyst-instance/fileName`

- Menu sequences use a symbol to convey the sequence. For example, `Actions => Create Test` instructs you to choose the `Create Test` command from the `Actions` menu.

Summary of changes

This topic summarizes product changes and enhancements.

Version 12.1 December 2016

This version of BMC Workbench includes the following features:

- Lets you manage object sets, which are used by RECOVERY MANAGER, DASD MANAGER PLUS, and certain NGT utilities

  For more information, view the Quick Course "Workbench for DB2 - Managing object sets."

You can create, edit, and delete object sets.

- Lets you manage the NGT Utility Manager tables by using a browser-based interface
For more information, view the Quick Course "Workbench for DB2 - Setting up NGT Utility Manager."

You can view and edit records in the Criteria, Exceptions, and Schedule tables. For more information about these tables, see the BMC Next Generation Technology Utility Manager for DB2 for z/OS Reference Manual.

- Adds a new perspective, Product Tools, that you can use to manage object sets and set up the NGT Utility Manager

  **Note**
  This perspective is not included in the default layout. For more information, see “Managing a workspace layout” on page 32.

### Version 1.1.00 May 2015

Following this release in May 2015, some significant features were made available via PTFs accompanying small program enhancements (SPE).

This version of BMC Workbench includes the following features:

- When you define a What-If index scenario during the SQL tuning process, add an index based on an expression.

- Migrate the access-path statistics for a set of objects from one subsystem to another, or from one schema to another within the same subsystem.

- Edit and generate Alter DDL for the following objects:
  - Databases
  - Table spaces
  - Tables
  - Indexes
  - Storage groups
  - Native procedures

- List and view the status of BMC utility jobs that are running on an IBM DB2 subsystem. For any BMC utility job, you can view the utility status detail table that displays the objects that the utility accesses.

- Compare schemas. You can receive a count and list of objects to be created, altered, or dropped and their corresponding DDL statements. You can make the following comparisons:
  - Catalog to catalog
  - DDL to DDL
  - Catalog to DDL
— DDL to catalog

- Generate recovery progress reports and verify recoverability for an object set. You can also estimate backup and recovery time for an object set.

- Submit JCL to refresh the Recoverability report.

- View lists of the following IBM DB2 Analytics Accelerator for z/OS objects:
  - Accelerators associated with a DB2 subsystem or DB2 data sharing group
  - Tables defined to an accelerator
  - Packages that have been bound with an accelerator option

- Access a tablet-optimized version of BMC Workbench. This version lets you view a Recoverability report for an object set.

**Version 1.0.00 June 2013**

Following the original release in June 2013, some significant features were made available via PTFs accompanying small program enhancements (SPE).

This version of BMC Workbench includes the following features:

- Lets you browse the DB2 catalog

- Lets you tune a SQL statement by:
  - Viewing a dynamic graphical representation of an Explain plan
  - Creating alternative What-If options by changing the SQL statement
  - Comparing side-by-side alternative graphical Explain plans, with different What-If options
  - Evaluating the potential performance of a SQL statement by creating an index, dropping an index, or updating index statistics. This feature is available if you have a license for BMC Performance for DB2 SQL.

- Lets you execute DB2 commands

- Supports the following DML statements:
  - SELECT
  - INSERT
  - UPDATE
  - DELETE

- Lets you browse the IBM z/OS file system in a hierarchical format
Let you view jobs and job output

Let you use the Connection Manager to define a local or a remote DB2 connection via a single UIM server

Let you use the Template Manager to organize templates hierarchically in user-defined folders, making it easier to find the templates that you need

Let you use System Authorization Facility (SAF) resources to manage individual user-access to BMC Workbench functions. For more information, see “Managing user access” on page 27.

Syntax statements

This topic explains conventions for showing syntax statements.

A sample statement follows:

\[
\text{COMMAND \ \text{KEYWORD1 [KEYWORD2 | KEYWORD3] KEYWORD4=\{YES | NO\} \ fileName...}
\]

The following table explains conventions for syntax statements and provides examples:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Items in italic type represent variables that you must replace with a name or value. If a variable is represented by two or more words, initial capitals distinguish the second and subsequent words.</td>
<td>alias&lt;br&gt;databaseDirectory&lt;br&gt;serverHostName</td>
</tr>
<tr>
<td>Brackets indicate optional items. Do not type the brackets when you enter the option. A comma means that you can choose one or more of the listed options. You must use a comma to separate the options if you choose more than one option.</td>
<td>[tableName, columnName, field]&lt;br&gt;[-full, -incremental, -level]</td>
</tr>
<tr>
<td>Braces indicate that at least one of the enclosed items is required. Do not type the braces when you enter the item.</td>
<td>{DBDName</td>
</tr>
<tr>
<td>A vertical bar means that you can choose only one of the listed items. In the example, you would choose either commit or cancel.</td>
<td>{commit</td>
</tr>
</tbody>
</table>
**Syntax diagrams**

The following figure shows the standard format for syntax diagrams:

![Syntax Diagram]

The following example illustrates the syntax for a hypothetical DELETE statement. Because the FROM keyword, alias variable, and WHERE clause are optional, they appear below the main command line. In contrast, the tableName variable appears on the command line because the table name is required. If the statement includes a WHERE clause, the clause must contain a search condition or a CURRENT OF clause. (The searchCondition variable appears on the main line for the WHERE clause, indicating that this choice is required.)

![DELETE Syntax Example]

The following guidelines provide additional information about syntax diagrams:

- Read diagrams from left to right and from top to bottom.

---

**Convention**

An ellipsis indicates that you can repeat the previous item or items as many times as necessary.

<table>
<thead>
<tr>
<th>Convention</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>An ellipsis indicates that you can repeat the previous item or items as many times as necessary.</td>
<td>columnName...</td>
</tr>
</tbody>
</table>
- A recursive (left-pointing) arrow above a stack indicates that you may choose more than one item in the stack.

- An underlined item is a default option.

- If a diagram shows punctuation marks, parentheses, or similar symbols, you must enter them as part of the syntax.

- In general, IBM commands, keywords, clauses, and data types are displayed in uppercase letters. However, if an item can be shortened, the minimum required portion might be shown in uppercase letters, with the remainder in lowercase (for example, CANcel).

- The following conventions apply to variables in syntax diagrams:
  - Variables are typically displayed in lowercase letters and are always italicized.
  - If a variable is represented by two or more words, initial capitals distinguish the second and subsequent words (for example, databaseName).
Overview of BMC Workbench *for DB2*

The BMC Workbench *for DB2* product offers a web-based graphical user interface (GUI) to IBM DB2 application developers and DBAs. With BMC Workbench, you can use a web browser (Microsoft Internet Explorer, Google Chrome, or Mozilla Firefox) to perform common tasks on your IBM z/OS mainframe.

BMC Workbench includes the following features:

- **Browsing the DB2 catalog and data sets**
  You can browse a DB2 object or set of objects, or any DB2 statement in the DB2 statement cache or in a package. You can also browse data set members and use the Scratchpad text editor to edit them.

- **Tuning SQL statements**
  For a selected statement, you can create alternative What-If options, use graphical Explain plans to compare the alternatives, and then tune your statement accordingly. In the graphical Explain plan, nodes represent each step in the statement; clicking a node displays details for that step from the Explain plan tables.

- **Viewing jobs and job output**
  You can browse Job Entry Subsystem (JES) jobs running on the LPAR where the UIM server is active.

- **Executing DB2 commands**
  You can navigate to a DB2 object and display and edit the syntax of a command for any of the supported DB2 commands. For more information, see “Supported commands” on page 67.

- **Migrating object statistics**
  You can migrate the access path statistics for a set of objects from one subsystem to another, or from one schema to another within the same subsystem.
- Comparing DB2 schemas
  You can compare two sets of data structures and generate reports showing the extent of the required changes. You can also generate Change Definition Language (CDL) that you can save and execute with the ALTER or CHANGE MANAGER product to make the schema structures identical.

- Managing backup and recovery
  You can generate reports to determine whether your object sets are recoverable. You can also view the progress of an ongoing recovery, and create What-If scenarios to estimate how long it would take to back up or recover.

- Edit and generate ALTER DDL
  You can edit and generate ALTER DDL for databases, table spaces, tables, indexes, native procedures, and storage groups.

- Managing object sets
  You can create and edit object sets, and share object sets with other BMC products that use them.

- Setting up NGT Utility Manager
  You can set up the NGT Utility Manager tables, which are used by the NGT Utility Manager to identify which database objects are to be processed by NGT utilities.

Overview of workspaces and perspectives

All work in BMC Workbench is performed within workspaces. You can define as many workspaces as you need, and no other users can view or edit your workspaces. You can create a workspace template, which others can use to create workspaces with the same set of views. After you create a workspace, it is available in the classic BMC Workbench interface or in the mobile interface.

Each workspace has several perspectives. A perspective contains command options to perform a set of tasks. The perspective displays results relating to different objects or statements in user-defined views. For example, you might define a view to display a subset of DB2 objects in the DB2 Navigator perspective. Performing certain actions on an object can automatically switch the perspective and open the corresponding view to display the relevant content.

Note
In the mobile interface, you can only manage workspaces and access the Recovery Manager perspective.
During a session, you can save a workspace at any time and return to it later in that session or another session. Workspace filters preserve your working environment. Consequently, you can partially complete a task (such as tuning SQL), save your workspace, and address another task that requires immediate attention; when time permits, you can return to your saved tuning session and continue creating What-If scenarios.

Overview of the BMC Workbench console

The BMC Workbench for DB2 main console is displayed after you log on to BMC Workbench. The product’s web-based console has one pane for managing workspaces and another for working with your DB2 objects and data sets.

When you first launch the product, the console features a welcome screen. Creating your first workspace enables all of the tools in the Workspace Manager toolbar and switches the focus in the right pane to the DB2 Navigator perspective.

**Note**

The following figure shows all perspective tabs and displays the DB2 Navigator perspective. If you create a workspace from a template, your screen might show only certain perspectives or might list them in a different order, based on the template’s settings. For more information, see “Setting BMC Workbench options” on page 35.

Figure 1: Sample BMC Workbench console
### Legend

<table>
<thead>
<tr>
<th>#</th>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Workspace Manager pane</td>
<td>Lists all of your workspaces</td>
</tr>
<tr>
<td>2</td>
<td>Workspace Manager toolbar</td>
<td>Lets you create, open, and save workspaces</td>
</tr>
<tr>
<td>3</td>
<td>Active workspace name</td>
<td>Indicates which workspace is active</td>
</tr>
<tr>
<td></td>
<td></td>
<td>An asterisk (*) indicates that the workspace has unsaved changes.</td>
</tr>
<tr>
<td>4</td>
<td>DB2 navigation pane</td>
<td>Lists all of the views that are in the currently open perspective</td>
</tr>
<tr>
<td>5</td>
<td>Perspective tabs</td>
<td>Provides access to the specified perspective. The current open perspective is indicated in blue.</td>
</tr>
<tr>
<td>6</td>
<td>Navigate To toolbar (DB2 Navigator perspective only)</td>
<td>Lets you navigate to object types that are dependents of the selected object in the results list</td>
</tr>
<tr>
<td>7</td>
<td>Commands toolbar (DB2 Navigator perspective only)</td>
<td>Lets you execute commands on the selected object, display properties, and display explainable statements</td>
</tr>
<tr>
<td>8</td>
<td>View toolbar</td>
<td>Lets you perform actions on an object selected in the results pane. The available options vary according to the type of object selected.</td>
</tr>
<tr>
<td>9</td>
<td>Tasks toolbar</td>
<td>Via the Migrate Statistics button, lets you migrate access path statistics for objects selected in the results pane</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Via the Compare Schema button, lets you compare schemas for two selected objects</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Via the ALTER button, lets you edit the properties of the selected object and run the generated ALTER DDL to make your changes</td>
</tr>
<tr>
<td>10</td>
<td>What's New</td>
<td>Lets you view details and short videos describing the new features added to BMC Workbench</td>
</tr>
<tr>
<td>11</td>
<td>Tools</td>
<td>Lets you access system messages, set global options, and view details of the User Interface Middleware (UIM)</td>
</tr>
<tr>
<td>12</td>
<td>Logout</td>
<td>Lets you log out from the BMC Workbench console</td>
</tr>
<tr>
<td>13</td>
<td>Options button</td>
<td>Lets you set options for the currently displayed perspective</td>
</tr>
<tr>
<td>14</td>
<td>Close Workspace button</td>
<td>Closes the active workspace</td>
</tr>
<tr>
<td>15</td>
<td>Layout button</td>
<td>Lets you manage the layout for your workspace</td>
</tr>
<tr>
<td>16</td>
<td>Help button</td>
<td>Displays the online Help</td>
</tr>
<tr>
<td>17</td>
<td>Results pane</td>
<td>Displays the results that a perspective generates, based on the type of action performed</td>
</tr>
<tr>
<td>#</td>
<td>Description</td>
<td>Details</td>
</tr>
<tr>
<td>----</td>
<td>--------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>18</td>
<td>Results list <em>(DB2 Navigator perspective only)</em></td>
<td>Displays all objects that correspond to the filter definition (for example, all objects returned by a filter in DB2 Navigator)</td>
</tr>
</tbody>
</table>
Getting started

You install the BMC Workbench for DB2 product from the Installation System. Following the installation, you must perform these tasks in order to use the product:

1 “Logging on to BMC Workbench for DB2” on page 25
2 “Managing user access” on page 27
3 “Selecting a subsystem for the BMC Workbench repository” on page 32
4 “Setting BMC Workbench options” on page 35
5 “Creating an entirely new workspace” on page 40

Prerequisites

This topic specifies the licensing, browsers, DB2 privileges, and so on that BMC Workbench requires.

Note
Information provided in this section supplements the information in the **BMC Products and Solutions for DB2 Customization Guide**.

Licensing

To run BMC Workbench, you must have a license for any of the following simplified solutions:

- BMC High Speed Utilities for DB2
- BMC Object Administration for DB2
- BMC Recovery for DB2
- BMC Performance for DB2SQL
- BMC Performance for DB2 Databases
BMC Utility Management for DB2
BMC Large Object Management for DB2

If you have any of these solutions, you can use BMC Workbench to set up the NGT Utility Manager and manage object sets.

In addition, certain licenses enable premium features:

BMC Object Administration for DB2
This license enables the following premium features:
— Schema-comparison
— Hierarchical Data Definition Language (HDDL)
— Drop Recovery

BMC Performance for DB2SQL
This license enables the What-If index premium feature.

BMC Recovery for DB2
This license enables the Recovery Management premium feature.

BMC Performance for DB2 Databases
This license enables the ALTER DDL premium feature.

You can install BMC Workbench with one of these solutions or install it separately, at a later time.

Supported browsers

BMC Workbench supports the following browsers on Microsoft Windows 7. BMC has tested BMC Workbench on the specific version numbers cited:

- Microsoft Internet Explorer 8, 9, 10, or 11
- Mozilla Firefox 39 or later
- Google Chrome 23, 24, or later

Also, you must have Adobe Flash Player version 22.0.0.000 or later installed.

DB2 Privileges

You must have READ/WRITE privileges to the BMC Workbench repository tables. The default CREATOR for these tables is BMCGUD.
Component configuration

The following configuration requirements apply to the components that work with BMC Workbench.

User Interface Middleware (UIM)

- You must have READ/WRITE privileges for the UIM HFS data set.
- The following requirements apply if you have multiple BMC Workbench installations sharing a single repository:
  - You can have only one UIM HFS data set per repository. If multiple UIMs share a repository, they must also share the UIM HFS data set.
  - All BMC Workbench installations must have the same maintenance level (same PTFs).

DB2 Component Services (DBC)

BMC Workbench uses GUD agents that require the DB2 DSNLOAD library. Unless that library is already included in your LINKLIST, you must add the DB2 DSNLOAD library to the <LOADLIB> tags in the GUDINIT step of the $U20INIT job.

For more information, see Knowledge Base article KA412340. See KA412340. You can access the Knowledge Base directly at http://www.bmc.com/available/search-kb.html or from the BMC Support Central website (http://www.bmc.com/support).

BMC product installation requirements

BMC Workbench version 12.1 requires that you have version 12.1 of the CATALOG MANAGER for DB2 product installed.

Logging on to BMC Workbench for DB2

Use the following procedure to log on to BMC Workbench for DB2 from your web browser.

Before you begin

Verify that the following requirements are met:
The required User Interface Middleware (UIM) server is running.

You have a user ID and password with:
— Authorization to access the host where the UIM server is installed
— Suitable DB2 authorization for your requirements

You know the host name of the UIM server that is connected to your DB2 server, and the port number on which the UIM server is listening.

The DBC started task is running and the DB2 Product Configuration (LGC) agent is active.

If you are using a security package like IBM Resource Access Control Facility (RACF), or CA Technologies CA-ACF2 or CA-Top Secret, ensure that you know the ID and account details.

To log on

1 In a web browser, enter the URL http://host:port/workbench/index.html.

Replace the variables host and port with the host name and port number of the UIM server.

To simplify future access, save the URL as a favorite.

2 In the Logon dialog box, enter your TSO user ID and password.

3 (optional) If using a security package like RACF, CA-ACF2, or CA-Top Secret, enter your group ID and account number.

4 Click OK.

The Welcome screen is displayed.

If you are inactive for 30 minutes, the BMC Workbench console times out and you must reenter your password.

Where to go from here

If this is the first time that you logged on to BMC Workbench, you must create a workspace to proceed.
Managing user access

By default, licensed users have full access to all BMC Workbench functions. Using the following procedures and the System Authorization Facility (SAF), you can disable perspectives for one or more users, or assign superuser authorization to a user.

**Note**
You cannot disable access to the Workspace Manager or to the DB2 Navigator perspective.

For ACF2 users in these procedures, define the resource as TYPE(XFC) when the documentation refers to the RACF XFACILIT class.

**Before you begin**

You must have SAF authorization that enables you to create and assign the required resources.

**To create and assign authorizations**

**Note**
You need to create these resources *only* if you want to revoke access to a perspective or a functionality for one or more users. By default, all perspectives are available to all licensed users.

1. For each UIM installation, create any of the following SAF resources that you need as an XFACILIT class:

<table>
<thead>
<tr>
<th>For this perspective</th>
<th>Create this SAF resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Locator</td>
<td>BMCGUD.WBAC.system.port.FILE_LOCATOR</td>
</tr>
<tr>
<td>Job Browser</td>
<td>BMCGUD.WBAC.system.port.JOB_BROWSER</td>
</tr>
<tr>
<td>SQL Tuning</td>
<td>BMCGUD.WBAC.system.port.SQL_TUNING</td>
</tr>
<tr>
<td>Scratchpad</td>
<td>BMCGUD.WBAC.system.port.SCRATCHPAD</td>
</tr>
<tr>
<td>Schema Management</td>
<td>BMCGUD.WBAC.system.port.SCHEMA_MANAGEMENT</td>
</tr>
</tbody>
</table>
Within the Product Tools perspective you can create and assign authorization for specific features:

<table>
<thead>
<tr>
<th>Product Tools feature</th>
<th>Create this SAF resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object set management feature</td>
<td>BMCGUD.WBAC.system.port.TOOLS.OBJ_SET_MGR</td>
</tr>
<tr>
<td>NGT utility management feature</td>
<td>BMCGUD.WBAC.system.port.TOOLS.NGT_UTIL_MGR</td>
</tr>
</tbody>
</table>

**Note**
Replace the variables *system* and *port* with the system name and the port number of the UIM server.

**To assign superuser authorization**

**Note**
You must give superuser authorization to at least one Workbench user. A superuser authorization is required to setup the Workbench repository.

With superuser authorization, a user can edit and delete connections or templates owned by any user, and create or delete public connections. The user who sets up the BMC Workbench repository requires superuser authorization.

Superuser authorization is specific to each UIM. If your site stores public connections and templates on several sysplexes according to business function, you can provide superuser authorization to specific users based on their areas of responsibility.

**Note**
If you previously used (ACT.WBSU.host.port) to define a superuser, you do not need to change it.

1 For each UIM installation, create the following SAF resource as an XFACILIT class:

BMCGUD.WBSU.system.port
Note
Replace the variables system and port with the system name and port number on the UIM server.

2 Assign ALTER authority to the superuser resource for the user requiring superuser authorization on the specified UIM.

Related Information

- “Tuning SQL” on page 77
- “Viewing and editing a data set” on page 229
- “Managing Backup and Recovery” on page 203
- “Comparing DB2 schemas” on page 185
- “Working with commands” on page 67
- “Viewing JES jobs” on page 227

Connecting to DB2 subsystems

During initialization, BMC Workbench discovers all DB2 subsystems on the sysplex where a UIM server is installed. The DB2 connections menu lists the discovered subsystems. You can also add connections to remote DB2 subsystems.

For more information, view the Quick Course “Workbench for DB2 - Using the Connection Manager.”

When you initially open the Connection Manager, a table of discovered connections is displayed. An icon indicates the type of connection (Table 1 on page 29). Unless you are a superuser, you can see only your own connections, public connections, and the discovered connections. You can use the following procedure to add connections.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🗝️</td>
<td>DB2 connections that you own</td>
</tr>
<tr>
<td>Icon</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>![Icon]</td>
<td>DB2 connections that Connection Manager discovered and that no one (not even a superuser) can remove</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Public DB2 connections that a superuser created</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Non-public DB2 connections visible to superusers, but owned by other users</td>
</tr>
<tr>
<td>![Icon]</td>
<td>An unsupported version of a DB2 connection BMC Workbench supports DB2 Version 10.1 and later.</td>
</tr>
</tbody>
</table>

**To add a DB2 connection**

You might want to give an easily identifiable name to a local DB2 connection or create a remote DB2 connection. Only you and superusers can view, use, and edit connections that you have created.

1. On the browser's main menu bar, click **Tools => Manage Connections**.
2. Click ![Icon].
3. Complete the Add DB2 Connection dialog box:
   a. Enter a name that describes the connection.
   b. Select a local subsystem.
   c. If you want to connect to a remote subsystem, enable **Remote Subsystem** and select from the menu a remote subsystem that is connected to the selected local subsystem.
   d. *(superusers only)* Select **Public** if you want to enable creating public connections that all users can use.
   e. Click **OK**.

**To remove a DB2 connection**

*Note*

You cannot remove a DB2 connection if a workspace is using it. Also, unless you are a superuser, you can edit or remove only DB2 connections that you created. See **Table 1 on page 29**.

1. On the browser's main menu bar, click **Tools => Manage Connections**.
2. Select a DB2 connection that you can edit.
To edit the list of favorite DB2 connections

You can filter the list of favorite DB2 connections to display only those that are relevant to you. This list populates the menu of DB2 connections from which you select a connection. Until you make a selection, all available connections are dimmed.

**Tip**

In many dialog boxes where you must select a DB2 connection, you can also edit the list of favorites by clicking next to the **DB2 Connection** field.

1. On the browser's main menu bar, click **Tools => Manage Connections**.

   The Connection Manager displays all subsystems that are automatically discovered, public DB2 connections, and any DB2 connections that you create.

   ■ next to a DB2 connection indicates that the DB2 connection is included in the list of favorites. This icon is dimmed ( ) if the connection is excluded from the list.

   ■ indicates a connection to an unsupported DB2 version. (BMC Workbench supports DB2 Version 10.1 and later.)

   **Tip**

   Click to update the list of discovered subsystems. For example, you might update the list after restarting a DB2 server, or to see public connections that a superuser has added during your session.

2. Select a DB2 connection and use the command bar to edit the list of favorite connections:

   ■ Click to include a connection in the list.

   ■ Click to exclude a connection from the list.

3. Continue adding or excluding connections in your favorites list, and click **OK** when finished.
Selecting a subsystem for the BMC Workbench repository

Product information, such as user preferences and workspaces, is saved in a repository on a DB2 subsystem.

Only BMC Workbench superusers (see “Managing user access” on page 27) can select the DB2 subsystem where the repository is installed.

**Tip**
After you have stored user preferences and workspaces, *BMC recommends that you do not change where the repository is installed*. If you have to change your repository, contact BMC Support for advice.

**Note**
Consider the following guidelines if you have BMC Workbench installations sharing a single repository:

- You should have only one UIM HFS data set per repository. If multiple UIMs share a repository, they must also share the UIM HFS data set.
- All BMC Workbench installations must have the same maintenance level (that is, the same PTFs applied).

**To select the subsystem**

1. On the browser's main menu bar, click **Tools => Server Setup**.
2. Select **Workbench Repository**.
3. Select a DB2 subsystem and click **OK**.

Managing a workspace layout

Use the following procedure to define the layout of perspectives for each workspace and template. You define the order and specify the perspectives displayed in the layout. BMC Workbench provides two predefined layouts, and you can create more according to your own requirements.
Customizing the layout of one workspace does not affect any other workspaces. You can save the workspace layout with a name to use in other workspaces. For more information, see “Setting BMC Workbench options” on page 35.

For more information, view the Quick Course “Workbench for DB2 - Managing Layouts.”

To manage a workspace layout

1 In the Workspace Manager, open the workspace for which you want to customize the layout.

2 Near the top-right corner of the main console, click the Layout button.

3 In the Layout field, either accept the default, select a saved layout, or select one of the following predefined layouts to use as a starting point:

   - Workbench Basic includes the DB2 Navigator, File Locator, Job Browser, Performance, and Scratchpad perspectives.

   - Workbench Advanced includes all perspectives.

4 Select or clear check boxes to add or remove the corresponding perspectives from the layout.

5 (optional) To change the order in which a perspective is displayed, select the perspective and click the Up or Down arrow (↑↓) to move it.

6 Click OK.

Viewing messages and UIM properties

Use the following procedure to view the system information, warning, error, and debug messages. You can also view the properties of the User Interface Middleware (UIM) server that is connected to your DB2 server.

To view messages

1 On the BMC Workbench console's main menu, click Tools => View Messages.

   The View Messages screen displays messages chronologically. By default, only system messages are displayed.
2 Perform any of the following tasks:

- To view debug messages, select **Show debug messages**.

- To refresh the display, click **Refresh**.

- To change the size of the View Messages screen, click and drag the screen's bottom right corner.

- To save the messages, click **Save Messages**. Only the currently displayed messages are saved. Debug messages are saved only if **Show debug messages** is selected.

- To clear the message display, click **Clear Messages**.

**To view UIM details**

1 On the BMC Workbench console's main menu, click **Tools => Server Setup**.

2 Select **Server Information**. The properties of the UIM server are displayed.

**Preparing for maintenance**

When the DB2 subsystem that hosts the repository is offline for maintenance, the repository is unavailable. Use the following procedure to maintain access to repository information, such as preferences, templates, and saved workspaces.

---

**Note**

You can skip this procedure if you do not need access to the repository. For example, you can skip it but continue using BMC Workbench if you do not need to access information, save workspaces, or create connections through Connection Manager.

---

**To prepare for DB2 maintenance**

1 Migrate the data by unloading it from the current subsystem and loading it on the fallback subsystem.

2 Run Runstats on the fallback subsystem.

3 Log on to BMC Workbench as a superuser and select the fallback DB2 subsystem as the repository.
See “Selecting a subsystem for the BMC Workbench repository” on page 32.

**Note**
Consider the following guidelines if you have BMC Workbench installations sharing a single repository:

- You should have only one UIM HFS data set per repository. If multiple UIMs share a repository, they must also share the UIM HFS data set.

- All BMC Workbench installations must have the same maintenance level (that is, the same PTFs applied).

After the maintenance is complete, repeat the procedure with the original subsystem as the recipient.

### Setting BMC Workbench options

Use the following procedure to enable certain settings and troubleshooting tools that affect all perspectives.

**Note**
To set options for a particular perspective only, use the Options button. The Support tab provides access to product trace options. Use these options only when requested to do so by BMC Support.

#### To change the filter options

1. On the browser's main menu bar, click Tools => Options.
2. Select the General tab.
3. Select Filter Options.

   In all perspectives, this option converts all entered filter values to uppercase.

   **Note**

   After selecting (enabling) this option, you can override it for a specific filter by beginning the filter value with a quotation mark ("').

   - Leave the check box selected if you want BMC Workbench to convert all filter values that you enter to uppercase (the default).
- Clear the check box if you want to preserve the case that you use when typing filter values.

**To save a layout**

You can save a workspace layout for use on other workspaces.

1. On the browser's main menu bar, click **Tools => Options**.

2. On the Layouts tab, next to the layout you want, click **Save the current workspace layout as**.

3. Enter a name for the layout.

4. Click **OK**.

**To copy a layout**

Use the following procedure to create a new layout based on an existing one. You will copy the layout, customize it, and save it under a new name.

---

**Tip**

You cannot edit the BMC predefined layouts, but you can copy and customize a predefined layout.

---

1. On the browser's main menu bar, click **Tools => Options**.

2. On the Layouts tab, select a layout.

3. Click **Copy the selected layout**.

4. Enter a name for the new layout.

5. In the Copy Layout dialog box, select or clear the checkbox to add or remove perspectives.

6. *(optional)* To change the order in which a perspective is displayed, select the perspective and click the Up or Down arrow to move it.

7. Click **OK**.
To set the default layout

The default layout is used when you create a new workspace, or when you open a workspace or template that was created by a version of BMC Workbench that did not support layouts.

1  On the browser's main menu bar, click **Tools => Options**.

2  On the Layouts tab, click **Select as your default layout** next to the layout that you want.

   The default layout is indicated with the icon.

3  Click **OK**.
Managing workspaces

In BMC Workbench for DB2, you perform all of your activities in workspaces. Use the Workspace Manager to create and manage your workspaces.

After installing and logging on to BMC Workbench, you must create a workspace before you can start using the product. Each workspace has a customizable set of perspectives, and each perspective can contain one or more views.

Creating a workspace

In BMC Workbench for DB2, you can create an entirely new workspace, create a workspace from a template, or copy an existing workspace.

For more information, view the Quick Course "Workbench for DB2 - Creating a Workspace."

The Workspace icon indicates the current status of the workspace:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Name</th>
<th>Workspace status</th>
</tr>
</thead>
<tbody>
<tr>
<td>![New workspace icon]</td>
<td>New workspace</td>
<td>The workspace has been created and is unsaved.</td>
</tr>
<tr>
<td>![Open workspace icon]</td>
<td>Open workspace</td>
<td>The workspace has been opened in this session.</td>
</tr>
<tr>
<td>![Edited workspace icon]</td>
<td>Edited workspace</td>
<td>Unsaved changes have been made to the workspace.</td>
</tr>
<tr>
<td>![Inactive workspace icon]</td>
<td>Inactive workspace</td>
<td>The workspace has not been opened in this session.</td>
</tr>
</tbody>
</table>
Creating an entirely new workspace

Use the following procedure to create a new workspace that is not based on a template or an existing workspace.

To create an entirely new workspace

1. In the Workspace Manager, click Add Workspace.

2. In the Add Workspace dialog box, enter a workspace name and (optionally) a description for the workspace.

   The name of the workspace must be unique for the user ID.

   **Note**

   In the Workspace Manager, the asterisk beside the workspace name indicates that the workspace is new or contains unsaved changes. The Workspace icon is displayed as.

3. In the Layout field, accept the default layout or select the initial layout for this workspace.

   You can select a saved layout or one of the following predefined layouts:

   - **Workbench Basic** includes: DB2 Navigator, File Locator, Job Browser, Performance, and Scratchpad.
   - **Workbench Advanced** includes all the perspectives.

   You can edit the layout later. For details, see “Managing a workspace layout” on page 32.

4. Click Ok.
Click **Save Workspace** to save the workspace.

The Workspace icon is displayed as 📀.

**Where to go from here**

After you create a workspace, you can perform the following tasks:

- “Discovering DB2 subsystems” on page 53
- “Tuning SQL” on page 77
- “Working with commands” on page 67
- “Managing JES jobs” on page 227
- “Managing data sets” on page 229

**Related Information**

- “Creating a workspace from a template” on page 41
- “Overview of workspaces and perspectives” on page 18
- “Creating a new DB2 object view” on page 53
- “Managing workspaces” on page 39

---

**Creating a workspace from a template**

Use the following procedure to create a new workspace from any template that is stored in the template repository.

**To create a workspace from a template**

1. From the Workspace Manager, click **Create Workspace from Template**.

2. In the Create Workspace from Template dialog box, search for the required template:
- From the **Filter by owner** menu, select a user name.
  Only the templates belonging to the selected user are displayed. You can use this filter in combination with the search string.

  **Note**
  
  identifies templates that you created

- Enter a search string into the search field.
  As you enter the search string, matching folders and templates are displayed in a search list.

- Use the vertical scroll bar or the mouse to navigate to and select the template that you want from the list.

- Select the folder that contains the required template, and expand the folder to search within it.

  **Tip**
  
  Hovering over the template's name lets you see the template's owner and description, and when it was last modified.

  **Note**
  
  In this dialog box, you can also create new folders, or move and edit folders. For more information, see “Managing folders” on page 47

3 Enter a workspace name and (optionally) a description.

  The workspace has the same layout as the template. You can edit the layout later. For details, see “Managing a workspace layout” on page 32

4 Click **Create Workspace**.

  **Note**
  
  In the Workspace Manager, the asterisk beside the workspace name indicates that the workspace is unsaved or contains unsaved changes.

  The Workspace icon is displayed as 

5 Click **Save Workspace**.

  The Workspace icon is displayed as 

---

Creating a workspace

42  BMC Workbench for DB2 User Guide
Where to go from here

After you create a workspace, you can perform the following tasks:

- “Discovering DB2 subsystems” on page 53
- “Tuning SQL” on page 77
- “Working with commands” on page 67
- “Managing JES jobs” on page 227
- “Managing data sets” on page 229
- “Managing a workspace layout” on page 32

Related Information

- “Creating a workspace from a template” on page 41
- “Overview of workspaces and perspectives” on page 18
- “Creating a new DB2 object view” on page 53
- “Managing workspaces” on page 39

Saving a workspace

Saving a workspace enables you to close a workspace or log out from BMC Workbench and return later to use the same defined definitions. If you do not save the workspace, any changes or additions that you made will be unavailable next time you log on to the console.

Note

All unsaved workspaces are indicated by an asterisk beside the workspace name, and the Workspace icon is displayed as a new workspace or an edited workspace.

To save a workspace

1 Select a workspace.
2 Select one of the following options:

- **Save a Workspace**
- **Save All Workspaces**
- **Save Workspace As** to save under a different name
- **Create Template** to save the workspace as a template that others can find and copy (as explained in “Creating a template” on page 48)

This option does not save the workspace itself. If you have not saved changes to the workspace, or remains .

### Opening a workspace

You can open any workspace from the Workspace Manager.

**Note**
The last active workspace is displayed when you reopen BMC Workbench.

### To open a workspace

1 In the Workspace Manager, perform one of the following actions:

- Double-click the workspace.
- Select the workspace from the list, and click **Open Workspace**.

The selected workspace opens in the DB2 Navigator perspective, and workspaces that were previously saved are opened in the perspective that was active when saved.

**Tip**
In the Workspace Manager, hovering over a workspace name lets you see the workspace’s owner and description, and when it was last updated.

### Where to go from here

You can add or remove views from the workspace.
Closing a workspace

Use the following procedure to close a workspace.

**Note**
Next time you open the workspace, BMC Workbench will display the perspective that was active when the workspace was last saved.

**To close the workspace**

1. Click **Close workspace** (positioned near the console's top-right corner).

   **Note**
   If the workspace has unsaved work, you are asked *if you want to close the workspace and discard your changes.*

Removing a workspace

Use the following procedure to remove or delete a workspace that you no longer need.

**To remove a workspace**

1. In the Workspace Manager, select the workspace that you no longer need.

2. Click **Remove Workspace**.

3. When asked to confirm, click **Yes**.
Editing workspace properties

Use the following procedure to change a workspace's name or description.

To edit workspace properties

1. In the Workspace Manager, select the workspace.

2. Click Edit Workspace Info.

   The Edit Workspace Info dialog box is displayed.

3. Make changes to the name, description, or both.

4. Click OK.

Related Information

- “Saving a workspace” on page 43

Working with templates

You can use any workspace as a template, and you can create new workspaces from templates that you or other users have created. Each template contains a set of views.

All templates are saved to a repository database and are available to all BMC Workbench users who have suitable DB2 authorization to access the repository.

For more information, view the Quick Course "Workbench for DB2 - Using the Template Manager."

Example

A DBA creates a workspace that includes a set of filters for DB2 Navigation; the filters specify a specific set of tables for accounting software that runs on a specific DB2 subsystem. The DBA saves the workspace as a template. The template is then available to all users, which saves the time and effort that would be required to create the workspace again from scratch.
Managing folders

Use the following procedures to create, delete, move, and rename folders.

Templates are saved to public folders, which can contain templates of one or more users. You can also add, delete, and edit folders when you create a workspace from a template or create a template.

**To create a folder**

Any user can create a folder, and other users can access and use the created folder.

1. On the browser's main menu bar, click **Tools => Manage Templates**.

   A folder tree displays all templates stored in the repository.

2. Navigate to and select the folder that will contain the new folder, or select **Templates** to create the new folder at the root level.

3. Click ![Create](image).

4. Enter a valid folder name (up to 50 alphanumeric characters).

   The name is not case sensitive, and it can include spaces, dashes, or underscores. Any leading or trailing spaces will be removed.

5. Click **OK**.
To delete a folder

You can delete any empty folder.

1. On the browser's main menu bar, click Tools => Manage Templates.
2. In the folder tree, navigate to and select the folder to be deleted.
3. Click .
4. When asked to confirm, click Yes.

To move a folder

1. On the browser's main menu bar, click Tools => Manage Templates.
2. In the folder tree, select the folder that you want to move.
3. Click .
4. Select a new parent folder.
5. Click OK.

To change a folder name

1. On the browser's main menu bar, click Tools => Manage Templates.
2. In the folder tree, navigate to and select the folder name that you want to change.
3. Click .
4. Enter a new valid name.
   See “To create a folder” on page 47 for permitted characters.
5. Click OK.

Creating a template

Use the following procedure to create a template from any workspace. You or your team can make a set of templates for different purposes.
To create a template

1 In the Workspace Manager, open the workspace that you want to use as a template.

2 Click Create Template.

*Note*
In the Create Template dialog box, you can also create new folders and edit your folders. For more information, see “Managing folders” on page 47.

3 In the Create Template dialog box, enter a template name (up to 50 alphanumeric characters) and optionally a template description.

   The name is not case sensitive, and it can include spaces, dashes, or underscores. Any leading or trailing spaces will be removed.

   Entering a description can help other users determine whether the template suits their purposes.

4 Either navigate to and select a folder to contain the template, or click and add a new folder as described in “Managing folders” on page 47.

5 Click Create Template.

*Note*
The new template inherits the layout of the workspace.

Related Information
- “Creating a workspace from a template” on page 41

Removing a template

Unless you are a superuser, you can remove only templates that you own.

To remove a template

1 On the browser's main menu bar, click Tools => Manage Templates.

   A folder tree is displayed.

2 In the navigation tree, navigate to and select the template.
You can use **Ctrl+Click** to select more than one template.

3 Click 

4 When asked to confirm, click **Yes**.

### Moving and editing templates

Use these procedures to move your templates between folders and edit template information.

Unless you are a superuser, you can move and edit only templates that you created.

#### To move a template to a different folder

1 On the browser's main menu bar, click **Tools => Manage Templates**.

   **Tip**
   You can also access the Manage Templates dialog box from the Create Template command.

2 Select your user name from the **Filter by owner** menu, and filter the templates so that you can see only templates that you created.

   **Note**
   Skip this step if you are a superuser and intend to move another user's templates.

3 If you want to move the template to a new folder, create the new folder as described in “To create a folder” on page 47.

4 Navigate to and select the template to be moved.

5 Click 

6 Navigate to and select the target folder.

7 Click **OK**.

#### To edit a template's information

1 On the browser's main menu bar, click **Tools => Manage Templates**.
2 Select your user name from the Filter by owner menu, and filter the templates so that you can see only templates that you created.

**Note**
Skip this step if you are a superuser and intend to move another user's templates.

3 Navigate to and select the template.

4 Click 🖊.

5 Edit the template's information.

6 Click OK.
Discovering DB2 subsystems and browsing catalogs

BMC Workbench for DB2 enables you to discover DB2 subsystems, and then browse through the DB2 catalog tables where you can select DB2 objects, view their properties, and perform actions on them.

Discovering DB2 subsystems

Sysplex discovery occurs during BMC Workbench for DB2 initialization.

BMC Workbench discovers all DB2 subsystems on the sysplex where a UIM server is installed; however, the filtering mechanism prevents you from viewing and connecting to subsystems running on DB2 versions that are not supported by BMC Workbench. If you have multiple sysplexes, you must install a UIM on each sysplex. For more information, see the Installation System documentation.

**Note**

BMC Workbench strictly maintains DB2 authorization rules; you can access DB2 objects on a subsystem only if you have the appropriate authorization in DB2.

Creating a new DB2 object view

Use the following procedure to create a view in a workspace. Each view contains a subset of DB2 objects retrieved from a DB2 subsystem. This topic also explains how to build an advanced search to use as the basis for the new view.

**To create a DB2 object view**

1. Open or create a workspace.
2 In the DB2 Navigator perspective, click the Add View menu arrow, and then click **Add DB2 Object View**.

*Tip*
You can also right-click in the DB2 navigation pane to select this action.
The default value of the Add View menu is always the last option that you created. If you previously created, in this or a previous saved session, a DB2 Statement Cache view then clicking creates a new DB2 Statement Cache view.

3 In the Add DB2 Object View dialog box, complete the following fields or accept the displayed default values.

<table>
<thead>
<tr>
<th>Field</th>
<th>Action</th>
</tr>
</thead>
</table>
| DB2 Connection | Use either of the following methods:  
  ■ Type the full or partial name of a DB2 connection to select the first subsystem that matches this value in the list of favorite connections.  
  ■ Use the menu arrow to select from your list of favorite connections.  
  *Tip:* You can add DB2 connections to your list of favorite connections by clicking and specifying which connections to include. |
| Object Type | Select an object type. |

4 Create either a basic search or an advanced search to use as the basis for the new view:

  ■ For a basic search, enter a filter pattern for the selected object.
  
  ■ For an advanced search, click **Advanced Search** and complete the following fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Label (optional)</td>
<td>Enter a label for this view. The default name is the first 30 characters of the search clause.</td>
</tr>
<tr>
<td>String Delimiter</td>
<td>From the menu, select <strong>Apostrophe</strong>, <strong>Quotation mark</strong>, or <strong>None</strong> as the string delimiter for character-type columns.</td>
</tr>
<tr>
<td>Search clause builder</td>
<td>Build an advanced search clause as instructed in “To build an advanced search clause for a view” on page 55.</td>
</tr>
</tbody>
</table>
5 Click **OK**.

The view is now included in the DB2 navigation pane, in a hierarchical display organized by subsystem and type.

**Note**

![icon] identifies a view created from a basic search, and ![icon] identifies one created from an advanced search.

6 When finished, save the workspace.

**To build an advanced search clause for a view**

The search clause is built from rows of search criteria in a grid. You can add, remove, and move rows to fine-tune the search clause.

As you enter the search criteria and build the search clause, the clause is displayed in the **Search Clause** text area below.

**Note**

By default, column names are displayed in catalog order. You can change to alphabetical order by clicking **Options** and changing the setting.

1 In the **Column** column, click the empty cell and either select a column from the displayed list or type a column name.

   BMC Workbench selects a matching column name in the list based on your entry; the list includes all column names for the DB2 connection and object type.

2 In the **Operator** column, click the cell and select an operator.

   **Note**
   
   The LIKE operator accepts only DB2 wildcards (% and _).

3 Click in the **Value** column and add a value.

   The value is case sensitive and is automatically enclosed in the delimiter that you selected.
BMC Workbench requires a nonblank value but does not validate this value.

4 Click in the **And/Or** column and select one of the following options:

- **AND**
- **OR**
- **-** (which closes the search clause)

If you select **AND** or **OR**, you are prompted to enter another column name, operator, and value in the row below.

5 Continue adding rows until you have built the search clause, ending the clause with **-**.

6 *(optional)* If you want to make any changes, edit the search criteria:

Use the following buttons to add, remove, or move rows. (You can perform the same operations by right-clicking in a row and selecting the appropriate command.)

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
</table>
| ![Plus](image) | Adds a row to the **Search Clause** criteria grid  
In the grid, selecting a row and clicking ![Plus](image) adds a row above the one that you selected. |
| ![Minus](image) | Removes the selected row from the grid |
| ![Up](image) | Moves the selected row up |
| ![Down](image) | Moves the selected row down |

**Note:**
Changes that you type into the text area are not synchronized with the search criteria in the grid. The grid is disabled.

You can also edit the search clause directly by typing in the search-clause text area. You can use the following buttons or shortcut keys:

<table>
<thead>
<tr>
<th>Button or shortcut key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Column</strong> or <strong>Alt+Ctrl+Space key</strong></td>
<td>Click in the clause and, from the menu, select a column name to add to the search clause.</td>
</tr>
<tr>
<td><strong>Operator</strong> or <strong>Alt+Ctrl+o</strong></td>
<td>Click and, from the menu, select an operator to add to the search clause.</td>
</tr>
<tr>
<td>Button or shortcut key</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>Click to discard any changes that you made in the text area and to re-synchronize the search clause with the search criteria in the grid. (Clicking this button <em>again</em> would restore your unsaved changes and disable the grid.)</td>
</tr>
</tbody>
</table>

**Note:** This button is available only after you begin making changes in the text area. Making any change in the search criteria grid disables this button.

7 When finished building the search clause, click **OK**.

---

**Related Information**

- “Overview of workspaces and perspectives” on page 18
- “Opening a workspace” on page 44
- “Creating an entirely new workspace” on page 40
- “Navigating to an object” on page 61
- “Viewing related objects” on page 62

---

**Navigating through the DB2 catalog**

Use the following procedure to select an object, and then create new views of related DB2 objects. For example, you can select a table and then create an object view of its indexes or of the database that contains the table.

**To navigate through the DB2 catalog**

1 Select or create a DB2 object view that contains the object.

   For details, see “Creating a new DB2 object view” on page 53.

2 Navigate to and select your source object.

   For details, see “Navigating to an object” on page 61.

3 From the **Navigate To** toolbar, click the required object button.

   Only object buttons related to the selected object are available.
Note
Only five object buttons can be displayed concurrently in the toolbar. Click the More arrow to access any additional object buttons.

The newly created DB2 object view is displayed in the DB2 navigation pane. The source object is indicated in the view label.

Creating a DB2 statement cache view

You can create a view of DB2 statements from the statement cache. Then, you can select statements to view, Explain, and tune.

To create a DB2 statement cache view

1. Open or create a workspace.

2. In the DB2 Navigator perspective, click the Add View menu arrow, and then click Add DB2 Statement Cache View.

   Tip
   You can also right-click in the DB2 navigation pane to select this option.
   The default value of the Add View menu is always the last option that you created. If you previously created, in this or a previous saved session, a DB2 Statement Cache view then clicking creates a new DB2 Statement Cache view.

3. Complete the Add DB2 Statement Cache View dialog box:
   a. Select a DB2 connection:
      - Type the full or partial name of a DB2 connection to select the first subsystem that matches this value in the list of favorite connections.
      - Use the menu arrow to select from your list of favorite connections.

         Tip
         Clicking lets you specify which DB2 connections to include in your list of favorite connections.

   b. Enter a filter pattern for the selected program name or package name.

4. Click OK.
The view is added into the DB2 navigation pane in a hierarchical display organized by subsystem and type.

5  *(optional)* To add additional filters to the same subsystem, return to Step 3.b on page 58 and specify a filter.

---

**Related Information**

- “Overview of workspaces and perspectives” on page 18
- “Opening a workspace” on page 44
- “Creating an entirely new workspace” on page 40
- “Creating a new DB2 object view” on page 53
- “Viewing related objects” on page 62
- “Navigating through the DB2 catalog” on page 57
- “Tuning SQL” on page 77
- “Explaining SQL statements” on page 78

---

**Removing a view**

Use the following procedure to remove a DB2 statement cache view or DB2 object view.

**To remove a view**

1. Open the workspace containing the DB2 view.

2. From the DB2 navigation pane, select the DB2 view that you want to remove.

   **WARNING**

   The DB2 navigation pane displays the views in a hierarchy. Selecting a subsystem node or type node will delete all of the views for that node.

3. Click **Remove View**.

---

**Chapter 4  Discovering DB2 subsystems and browsing catalogs**  59
Tip
You can also right-click in the navigation pane and select the Remove View menu option.

4 When asked to confirm, click Yes.

Copying and pasting a view

Use the following procedures to copy and paste a DB2 statement cache view or DB2 object view.

To copy a view

1 Open the workspace containing the DB2 view.

2 From the DB2 navigation pane, select the DB2 view that you want to copy.

3 Right-click and select Copy View.

You can now paste this view.

To paste a view

1 Open the workspace where you want to add the view.

2 On the DB2 navigation pane, select and right-click where you want to paste the view.

3 Click Paste View.

The Add Object View dialog box is displayed.

4 In the Add Object View dialog box, edit the view settings.

For more information, see “Creating a new DB2 object view” on page 53.

Note
By default, the value of the DB2 connection field is automatically updated to match the DB2 connection under which the view is pasted.

5 Click OK.
Browsing the catalog

You can browse the DB2 catalog, filter results, and view DB2 objects.

Tip
Click the column header to sort the contents of a table according to that column. Click again to toggle between ascending and descending order.

Navigating to an object

Use the following procedure to search DB2 catalogs and navigate to objects that you have selected.

Note
To navigate to a statement, see “Selecting a statement for SQL Analysis” on page 90.

To navigate to an object

1. Open a workspace as described in “Opening a workspace” on page 44.

2. Use one of the following methods to navigate to the required object:

   - Add a DB2 object view containing the required object.
   - From the Navigate To toolbar, click an object icon to open a DB2 object view containing the required object.
   - In a view, right-click an object row, select Navigate To, and select the required object type.
   - In a view, click Related Objects to navigate to the required object type.

3. Select one or more objects from the object list.

Where to go from here

You can now view the object properties, view related objects, create a new DB2 object view using the selected object as a source, or perform commands on the selected object or objects.
Viewing related objects

Use the following procedure to select an object and view lists of objects that are related to it. This procedure creates a related objects view.

**To view related objects**

1. From the DB2 navigation pane, select a view containing the source object that interests you.

2. In the results pane, select the source object.

3. From the Commands toolbar, click **Related Objects**.

   **Tip**
   You can also right-click and select **View => Related Objects**.

   **Note**
   Only buttons for valid related objects are displayed.

4. From the Related Objects toolbar, click the button of the required object.

   A list of corresponding objects is displayed in the results pane. You can select an object and use it as a source for navigation, or you can perform commands on the object.

---

**Related Information**

- “Viewing related objects” on page 62
- “Viewing object properties” on page 63
- “Navigating through the DB2 catalog” on page 57
- “Generating a command” on page 68

---

BMC Workbench for DB2 User Guide
Viewing object properties

Use the following procedure to view the properties of any object in the catalog for which you have authorization.

To view object properties

1. Select or create a workspace that contains a DB2 object view containing the object.

   **Note**
   If required, you can add a new DB2 object view.

2. Navigate to the object.

3. Perform one of the following actions:
   - In the Commands toolbar, click **Properties**.
   - In the results list, select and double-click the object row.
   - In the results list, select an object row, right-click and select **View => Properties**.

   The properties box of the selected object is displayed.

4. Click **OK** to close the properties box.

Exporting data

Use the following procedure to export data in .csv format to the location that you chose. The output includes the column headings.

To export data

1. Select or create a DB2 object view that contains the object or objects that you want to export.

   For details, see “Creating a new DB2 object view” on page 53.

2. Navigate to and select your source object or objects.

   For details, see “Navigating to an object” on page 61.
3 Perform one of the following actions:

- To export all rows in the view, right-click anywhere in the results pane and select **Select All**; then right-click again and select **Export data**.

- To export a specific object, select one or more object entries, right-click and select **Export Data**.

4 When prompted, select a location for the exported data.

   The default file name is **export_data.csv**.

5 Save the file.

---

**Related Information**

- “Copying data” on page 64

---

**Copying data**

Use the following procedure to copy data from the results pane to the clipboard.

**To copy data**

1 Select or create a DB2 object view that contains the object or objects that you want to export.

   For details, see “Creating a new DB2 object view” on page 53.

2 Navigate to and select your source object or objects.

   For details, see “Navigating to an object” on page 61.

3 Perform one of the following tasks:

   - To copy all rows in the view, right-click anywhere in the results pane and select **Select All**; then right-click and select **Copy**.

   - To copy a specific object, select one or more object entries; right-click, and select **Copy**.

You can now paste the data directly into a file or a spreadsheet application such as Microsoft Excel. The data automatically contains the column names.
Setting DB2 Navigator options

Use the following procedure to set the behavior of certain DB2 Navigator options.

1. Select the Options button.
2. Select the Commands tab.
3. Specify your preferences, and click OK.

Table 2 on page 65 shows the options that you can set.

Table 2: DB2 Navigator options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decimal Point</td>
<td>Select the check box if you want to use a period (.) for decimal points. Clear this check box if you want to use a comma.</td>
<td>Selected (period)</td>
</tr>
<tr>
<td>Verbose Output</td>
<td>Select this check box to run the command with output level VERBOSE. Clear this check box to use output level TERSE.</td>
<td>Selected (VERBOSE output)</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
<td>Default</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------------------------------------</td>
</tr>
</tbody>
</table>
| Object types                  | Select the object types that you want to include in the HDDL output. Clear the check box for any object type that you want to omit. You can select the following object types:  
  - Table spaces  
  - Tables  
  - Foreign keys  
  - Views  
  - Indexes  
  - Synonyms  
  - Aliases  
  - Plans  
  - Triggers  
  - Auths                                                                                                                                  | All selected except Auths                    |
| Define (HDDL and DDL only)    | Select this check box if you want to include the DEFINE parameter in DDL or HDDL for a table space or index. Clear the check box to omit the parameter.                                                        | Not selected                                 |
| SQLID before Grant (DCL only) | Select this check box to generate a SET CURRENT SQLID= grantor statement before each GRANT statement. This option is used for processing HDDL with AUTHs, HGRANT, and CASCADE REVOKE REASSIGN. Clear the check box to omit the statement. | Not selected                                 |
Working with commands

You can edit and run many common commands directly from the BMC Workbench product's Scratchpad editor, a full-featured editing tool.

The typical workflow involves completing these tasks:

1. “Navigating to an object” on page 61
2. “Generating a command” on page 68
3. “Editing text files in Scratchpad” on page 70
4. “Running a command from Scratchpad” on page 73

Supported commands

This topic describes the supported commands that you can edit and run directly from BMC Workbench for DB2.

Note

The HDDL command is available if you have a valid license for the BMC Object Administration for DB2 solution.

<table>
<thead>
<tr>
<th>Command</th>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alter</td>
<td>![icon]</td>
<td>Opens an ALTER editor view for the selected database, table space, table, index, native procedures, and storage groups. For more information, see “Generating and running Alter DDL statements” on page 69.</td>
</tr>
<tr>
<td>DDL</td>
<td>![icon]</td>
<td>Displays the Data Definition Language (DDL) for a selected object</td>
</tr>
<tr>
<td>HDDL</td>
<td>![icon]</td>
<td>Displays the Hierarchical Data Definition Language (HDDL) for the selected object and its dependents</td>
</tr>
</tbody>
</table>
### Generating a command

Use the following procedure to generate a command for a selected object. The generated command is displayed in Scratchpad.

For more information, view the Quick Course "Workbench for DB2 - Generating DML."

<table>
<thead>
<tr>
<th>Command</th>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCL</td>
<td><img src="image" alt="dcl" /></td>
<td>Displays the Data Control Language (DCL) for the selected object</td>
</tr>
<tr>
<td>DROP</td>
<td><img src="image" alt="x" /></td>
<td>Displays the DROP command for the selected object</td>
</tr>
<tr>
<td>BIND</td>
<td><img src="image" alt="folder" /></td>
<td>Displays the BIND command for the selected package, plan, or collection ID</td>
</tr>
<tr>
<td>REBIND</td>
<td><img src="image" alt="folder" /></td>
<td>Displays the REBIND command for the selected package, plan, or collection ID</td>
</tr>
<tr>
<td>FREE</td>
<td><img src="image" alt="x" /></td>
<td>Displays the FREE command for the selected package or plan</td>
</tr>
<tr>
<td>DISPLAY</td>
<td><img src="image" alt="book" /></td>
<td>Displays the status of the selected object</td>
</tr>
<tr>
<td>START</td>
<td><img src="image" alt="play" /></td>
<td>Displays the START command for a database, table space, table, index, view, or stored procedure</td>
</tr>
<tr>
<td>STOP</td>
<td><img src="image" alt="stop" /></td>
<td>Displays the STOP command for a database, table space, table, index, view, or stored procedure</td>
</tr>
<tr>
<td>SELECT</td>
<td><img src="image" alt="table" /></td>
<td>Displays the SELECT command for a table, alias, or view</td>
</tr>
<tr>
<td>INSERT</td>
<td><img src="image" alt="insert" /></td>
<td>Displays the INSERT command for a table, alias, or view</td>
</tr>
<tr>
<td>UPDATE</td>
<td><img src="image" alt="update" /></td>
<td>Displays the UPDATE command for a table, alias, or view</td>
</tr>
<tr>
<td>DELETE</td>
<td><img src="image" alt="delete" /></td>
<td>Displays the DELETE command for a table, alias, or view</td>
</tr>
</tbody>
</table>

To generate commands

1. In the DB2 Navigator perspective, select the required object or objects (up to five).
For more information, see “Navigating to an object” on page 61.

The command tool bar displays valid command buttons for the selected objects. See “Supported commands” on page 67.

2 Click the relevant command button.

The Scratchpad perspective opens to display the SQL statement in the Command Text tab. If you selected multiple objects, the SQL statements or commands are displayed consecutively in the Scratchpad.

3 You can now edit the SQL statement or command, as described in “Editing text files in Scratchpad” on page 70.

--- Related Information ---

- “Supported commands” on page 67

--- Generating and running Alter DDL statements ---

BMC Workbench lets you generate and run an Alter DDL statement for storage groups, native procedures, databases, table spaces, tables, and indexes. You must have the required DB2 system privileges to make these changes.

--- To generate and run an Alter DDL statement ---

1 In the DB2 Navigator perspective, select one of the supported objects:
   - Database
   - Index
   - Native Procedure
   - Storage group
   - Table space
   - Table

   For more information, see “Navigating to an object” on page 61.
2. In the Tasks tool bar, click Alter.

*Tip*

You can also right-click on the object and select from the menu.

An Alter editor view is displayed in the Schema Management perspective.

3. Make any needed changes to the editable properties.

*Note*

Additional displayed panels also let you change the properties of the object’s subelements, such as table columns or index partitions.

When you make changes to a property, the name of the changed property field is italicized.

4. Click Generate Alter DDL.

The DDL is displayed in the Alter DDL tab.

5. Click to run the DDL statement.

*Tip*

The editable properties are disabled when you return to the editor tabs. If you want to make more changes, click to reload the object definition and continue.

---

**Editing text files in Scratchpad**

Use the following procedure to edit a text file (such as a command statement) that is opened in Scratchpad.

**To edit a file**

1. To open the text file in the Scratchpad perspective, perform one of the following actions:

   - In the Scratchpad perspective, open an existing file displayed under Files.
To create a new command file, add a new file view by clicking 📊. Then, in the Command Text tab in Scratchpad, enter the command text (either by typing it or pasting copied text).

- Select an object or group of objects in the DB2 navigation perspective, and click one of the command options such as DDL.

- Double-click a data set in the File Locator perspective.

2 In the Scratchpad's Command Text tab, edit the command as needed:

Table 3: Scratchpad commands

<table>
<thead>
<tr>
<th>To</th>
<th>Do this</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find a search string</td>
<td>Enter the search string in the <strong>Search</strong> box and click <strong>Find</strong>.</td>
</tr>
<tr>
<td>Find and replace a search string</td>
<td>1  Double-click a data set in the File Locator</td>
</tr>
<tr>
<td></td>
<td>2  Enter the search string in the <strong>Search</strong> box.</td>
</tr>
<tr>
<td></td>
<td>3  Enter the replacement string in the <strong>Replace</strong> box.</td>
</tr>
<tr>
<td></td>
<td>4  Click <strong>Find</strong>.</td>
</tr>
<tr>
<td></td>
<td>5  Click <strong>Replace</strong>.</td>
</tr>
<tr>
<td>Find and replace <em>all instances</em> of a search string</td>
<td>1  Enter the search string in the <strong>Search</strong> box.</td>
</tr>
<tr>
<td></td>
<td>2  Enter the replacement string in the <strong>Replace</strong> box.</td>
</tr>
<tr>
<td></td>
<td>3  Enable the <strong>All</strong> checkbox.</td>
</tr>
<tr>
<td></td>
<td>4  Click <strong>Replace</strong>.</td>
</tr>
<tr>
<td>Make a search case sensitive</td>
<td>Select the <strong>A/a</strong> checkbox.</td>
</tr>
<tr>
<td>Undo an action</td>
<td>Click 🔄 to undo the last action.</td>
</tr>
<tr>
<td>Redo an action</td>
<td>Click 🔄 to cancel undoing the last action.</td>
</tr>
<tr>
<td>Enable the editor to accept Regular expressions</td>
<td>Select the <strong>RegEx</strong> checkbox.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: Scratchpad supports all standard Regular expressions.</td>
</tr>
<tr>
<td>To</td>
<td>Do this</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Copy an existing file</td>
<td>1  Browse to a data set and open an existing file.</td>
</tr>
<tr>
<td></td>
<td>2  Click <strong>Copy File</strong>.</td>
</tr>
<tr>
<td></td>
<td>3  Save the file. Follow the instructions in <strong>Step 4 on page 72</strong>.</td>
</tr>
</tbody>
</table>

---

**Note**

To remove a file view from the Scratchpad navigator pane, select the file view and click **Remove File View**.  

---

3  Save your changes:

- To save the active edited command output, click **Save File** in the toolbar.
- To save multiple open files (for example, you selected and created DDL for several objects simultaneously), click **Save all files** (in the navigation pane) to open a Save dialog box for each file.

---

**Note**

For each file, you can save to a separate data set or select **Don't Save**.

---

Under **Files** in the navigation pane, Scratchpad automatically displays all files that are saved to the mainframe.

4  Select a z/OS data set in which to save the file.

   a  Click **Add Data Set Filter**.

   b  In the **Filter Pattern** box, enter a filter (the wildcard * is supported) to display a filtered list of data sets that includes the one you want.

   c  In the displayed hierarchical tree, navigate to and select the data set that you want.

      If you select an archived data set, you will be asked to confirm that you want to restore that data set. If the data set has been archived, you are asked if you want to restore it.

5  Enter a file name.
Creating commands

Use the following procedure to create a command.

1. Click the **Scratchpad** perspective tab.

2. Click **Add File View**.

3. In the Command Text tab in Scratchpad, enter the command text (either by typing or by pasting copied text).

4. Edit the command, if needed, as explained in “Editing text files in Scratchpad” on page 70.

5. Run the command.

   For more information, see “Running a command from Scratchpad” on page 73.

   All files containing commands can be saved to the mainframe and are displayed in the Scratchpad navigation pane under **Files**.

---

Running a command from Scratchpad

Use the following procedure to run one or more commands directly from the Scratchpad perspective. If Scratchpad is displaying multiple commands, you can run all of them, or select and run a single command.
To run a command

1. Enter the command or commands into Scratchpad.

   The source can be either:
   - A command that you pasted or typed into Scratchpad and then edited.
   - A command that you generated from DB2 objects.

2. Perform one of the following tasks:
   - To run one command from a group displayed in Scratchpad, select the command and click Run Selected.
   - To run all commands displayed in Scratchpad, click Run.

   **Note**
   Any existing output is discarded when you run the commands again.

3. Select the DB2 subsystem on which to run the command. Guidelines are as follows:
   - By default, the last-executed DB2 connection is pre-selected.
   - Type the full or partial name of a DB2 connection to select the first subsystem that matches this value in the list of favorite connections.
   - Use the menu arrow to select from your list of favorite connections.

   **Tip**
   Click and specify which DB2 connections to include in your list of favorite connections.

   The status bar at the bottom of the screen indicates whether the command runs successfully. The Output tab displays a summary for each command that you ran.

4. To view results, perform one of the following tasks:
   - To view all results, click View All Result Sets.
To view the result set for a specific command, on the Output tab, click the relevant **View Result Set**.

---

**Tip**

If the command failed to run successfully, click **View Messages** to see all error messages, or click on the Output tab to view error messages for a specific command.

Also, note that **View All Result Sets** and **View Output Text** are enabled only when the Output tab is active.

---

5 Save the results as a text file on your local file system:

a. Click the Output tab and in the tool bar, then click **View Output Text** in the tool bar.

b. Click .

---

**Related Information**

- “Supported commands” on page 67

---

### Setting Scratchpad options

When the Scratchpad perspective is open, clicking the Options button lets you set the Scratchpad options.

The following options are available:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drop Recovery</td>
<td>This option lets you restore dropped objects from the CATALOG MANAGER product via the Drop Recovery command. Note: This option is available only if you have a valid license for the BMC Object Administration for DB2 solution.</td>
<td>No</td>
</tr>
</tbody>
</table>
### Setting Scratchpad options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select Options</td>
<td>For SELECT statements, this option lets you set maximum values for the following items (with defaults shown in parentheses):</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>■ Maximum numeric length (10)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ Maximum CHAR length (64)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ Maximum VARCHAR length (64)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ Maximum select rows (300), where 0 returns all rows in the result set.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The total amount of returned data in the result set cannot exceed 50 MB; the Select Options settings do not affect that limit. If the limit is reached, the product alerts you with a message.</td>
<td></td>
</tr>
</tbody>
</table>

#### Related Information

- “Supported commands” on page 67
Tuning SQL

You can create one or more What-If statements and use them to progressively tune a selected baseline statement.

By default, BMC Workbench for DB2 honors the DB2 privileges to generate Explain plans. To configure BMC Workbench to use Install SYSADM for executing Explains, see information about the authexpl option in the BMC Products and Solutions for DB2 Customization Guide.

Tuning SQL workflow

The workflow for tuning a SQL statement typically includes the following steps:

To tune a SQL statement

1. Select a SQL statement to be used as the baseline statement (“Selecting a statement for SQL Analysis” on page 90).
2. Perform an Explain on the statement (“Explaining SQL statements” on page 78).
3. Create a What-If scenario based on the baseline statement (“Creating a What-If scenario” on page 79).
4. Compare the cost of the statements (“Comparing statements” on page 88).
5. Create a new What-If scenario.
6. Continue the process until you have optimized the statement.

For more information, view the Quick Course “Workbench for DB2 - Analyzing SQL Performance.”
Explaining SQL statements

The Explain command explains the steps that the DB2 optimizer must take to execute the selected SQL statement.

SQL Explain displays the cost for each step of the SQL statement and the total cost of the entire statement. This data enables you to determine which part of the statement is estimated to consume the most resources.

An Explain takes one of the following forms:

- A dynamic Explain that asks the DB2 optimizer for an explanation of the access path. This process evaluates access paths, SQL text, and key catalog statistics in real time.

- A static Explain from a bind with EXPLAIN(YES). This process reports the access path information that was derived when you performed a bind with EXPLAIN(YES).

- An Explain Statement Cache on a dynamic SQL statement. This process performs an EXPLAIN STMTCACHE STMTID on a dynamic SQL statement that was executed and is still in the statement cache. The access path used at the time that the statement was executed is retrieved from the statement cache and reported on.

- An Explain Package, in which the current access path for a static SQL statement is retrieved and reported on even if a BIND with EXPLAIN(YES) was not previously performed.

Note
By default, BMC Workbench for DB2 honors the DB2 privileges to generate Explain plans. To configure BMC Workbench to use Install SYSADM for executing Explains, see information about the authexpl option in the BMC Products and Solutions for DB2 Customization Guide.

The resulting Explain plan is the hierarchical representation of rows extracted from the DB2 plan table.

To Explain a SQL statement

1. Use any of the following methods to select a SQL statement as the baseline statement.
   - “Analyzing a statement from the DB2 statement cache” on page 90
   - “Analyzing a statement from a DB2 package” on page 91
   - “Analyzing an ad hoc statement” on page 92
The Performance perspective displays the SQL statement and its parameters. For more information, see “Explain tab” on page 98.

2 Click Generate explain plan.

The Explain results are displayed along with the following tabs:

- SQL (see “SQL tab” on page 96)
- Indexes (see “Indexes tab” on page 97)
- Explain (see “Explain tab” on page 98)
- Cost (see “Cost tab” on page 100)
- Predicates (see “Predicates tab” on page 100)
- Compare (see “Compare tab” on page 101)

**Note**
The Compare tab is dimmed until you generate the first What-If scenario.

- Catalog Objects (see “Related Objects tab” on page 101)

## Using What-If scenarios

The What-If scenarios enable you to edit a statement, add or drop indexes, execute Explain, and compare the results of the statement to another statement.

### Creating a What-If scenario

Use the following procedure to create a What-If scenario from a source statement and then compare it to Explain results.

**To create a What-If scenario**

1 In the Performance perspective, select the source statement from a Tuning session.
You can use a baseline statement (identified by 🔄️) or one of the What-If scenarios that you have previously created (identified by 🤔).

The SQL tab displays the details of the source SQL statement.

2

Click What-If 🤔.

3 (optional) On the SQL tab, enter a unique name for the What-If scenario.

If you do not give the scenario a new name, by default an incremental number is added as a suffix to the source statement name.

a Edit the details of the scenario.

You can change the DB2 connection, table qualifier, Explain type and degree.

Click 🔄️ and specify which DB2 connections to include in your list of favorite connections.

b Edit the SQL statement.

4 (optional) On the Indexes tab, modify the indexes.

For more information, see “Creating and editing a What-If index scenario” on page 81.

5 When finished, click Generate explain plan 🛠️.

**Where to go from here**

You can now compare this What-If scenario with another scenario or baseline statement.

---

**Related Information**

- “Comparing statements” on page 88
Creating and editing a What-If index scenario

Use the following procedure to evaluate how adding an index, dropping an index, or updating index statistics would affect a statement's performance.

Before you begin

The What-If index feature is available only if you have a valid license for the BMC Performance for DB2SQL solution.

To create a What-If index

1. In the Performance perspective, select the source statement.

   You can use a baseline statement (identified by ) or a previously created What-If scenario (identified by ).

   The SQL tab displays the details of the source SQL statement.

2. If the selected statement has not been explained, click to Explain it.

3. Click What-If .

4. (optional) Enter a unique name for the new scenario.

   If you do not enter a name, the scenario uses the source statement's name with an incremental number added as a suffix.

5. Click the Indexes tab.

   For more information, see “Indexes tab” on page 97. The Indexes tab is available only if the baseline statement was successfully explained.

The Indexes tab includes these items:
Command buttons enable you to add, drop, edit, or copy indexes from the What-If statements.

The Tables pane lists all tables that are referenced in the SQL statement.

The Indexes pane lists the indexes of the selected table (initially the top entry in the list).

**Note**

✔ identifies indexes that the Explain plan uses.

6 From the Tables pane, select the table that contains the index that you want to edit.

7 From the Indexes pane, select the index.

Perform any of the following actions to set up the scenario that you want to evaluate:

- Add a new index for the selected table (as explained in “Adding an index to a What-If scenario” on page 83).
- Copy an existing index and modify the copy (“Copying an index for a What-If scenario” on page 86).
- Edit an existing index (“Editing an index for a What-if scenario” on page 87).
- Drop an index (“Dropping an index from a What-If scenario” on page 87).
- Revert any changes made to the catalog by clicking **Revert**.
- Generate statements to reflect changes for this What-If index scenario (“Generating a tuned statement” on page 88).

**Note**

You cannot make changes to the baseline statement.

8 

Click ![lightning bolt] to explain the edited What-If scenario.

**Note**

You cannot edit the What-If scenario after it is explained.

9 Compare the resulting scenario with the baseline or with another What-If scenario as explained in “Comparing statements” on page 88.
10 Continue from Step 7 on page 82 until you have fully tuned the statement.

11 If you want to generate the tuned statement, click Generate to copy the DDL.

For more information, see “Generating a tuned statement” on page 88.

--- Related Information ---

- “Working with commands” on page 67

--- Adding an index to a What-If scenario ---

Use the following procedure to add a What-If index to a selected table and evaluate how the change would affect the SQL statement's performance.

--- Note ---

You cannot add an index to the baseline statement (identified by ).

--- To add a new index ---

1 Select the table as described in “Creating a What-If scenario” on page 79.

2 Click Add.

3 In the Index Attributes dialog box, define the new index:
   a At Index name, enter the name that you want to use.
   b At Unique rule, accept the default (Duplicate) or select Unique or Unique unless Null.
   c Select the corresponding check boxes if you want the index to be clustering, partitioned, or padded.
   d Click Next.

4 In the Key Columns dialog box, define the key columns for the index:
   a Add columns or remove available columns from the Selected Columns pane.
   b If you want to reorder a column, select it and use the Up or Down button to change the column's position.
c  If you want to add an index on expression, click \( \text{+} \) and enter an expression. Then, click \( \text{OK} \).

You can enter any valid standard expression that SQL supports.

---

**Tip**

To remove an index on expression, click \( \text{-} \).

---

d  Click the arrow that indicates to sort the columns in ascending, descending, or random order.

e  Click Next.

5  In the **Index Statistics** dialog box, define the statistics for this index:

a  At **First key cardinality**, enter a positive integer (default 0) or -1 (indicating that no statistics have been gathered).

b  At **Full key cardinality**, enter a positive integer (default 0) or -1 (indicating that no statistics have been gathered).

c  At **Cluster ratio**, enter a number between 0 and 1 (default 0.8) or -2 (indicating that no statistics have been gathered).

d  At **Leaf pages**, enter a positive integer or -1 (the default, indicating that no statistics have been gathered).

e  At **Number of Levels**, enter the number of required levels (default -1).

f  Click **Finish**.

The wizard closes, and the results pane shows the new index with \( \text{+} \) beside it.

**Where to go from here**

You can now Explain and compare the new scenario.
Dropping an index from a What-If scenario

Use the following procedure to evaluate how dropping an index would affect the performance of a SQL statement.

**Note**

You cannot add an index to the baseline statement (identified by 💡).

**To drop an index**

1. Create a What-If scenario and select an index.
   
   For more information, see “Creating a What-If scenario” on page 79.

2. Click **Drop**.
   
   The **Indexes** pane displays 💡 next to the dropped index.

3. Click 💡 to Explain the statement.

4. (optional) Select an index and click **Revert** to return indexes that you have dropped.

**Where to go from here**

Explain and compare the scenario after you have completed editing.
Related Information

- “Comparing statements” on page 88
- “Indexes tab” on page 97
- “Explaining SQL statements” on page 78

Copying an index for a What-If scenario

Use the following procedure to copy an index, edit it, and save it as a What-If index in a What-If scenario.

Note

You cannot add an index to the baseline statement (identified by ).

To copy an index

1. Create a What-If scenario and select an index.
   
   For more information, see “Creating a What-If scenario” on page 79.

2. Click Copy.

3. In the Index Attributes dialog box, change the name and edit any other index attributes.
   
   For more information, see “Adding an index to a What-If scenario” on page 83.

4. Click Finish.

   The wizard closes, and the results pane shows the new index with beside it.

Where to go from here

You can now Explain and compare the scenario after you have completed editing.
Editing an index for a What-if scenario

Use the following procedure to edit an index and evaluate the effect on the SQL statement's performance.

**Note**
You cannot edit the columns of an index in the baseline statement (indicated by [ ]), but you can make changes to the index attributes and statistics.

**To edit an index**

1. Select an index.
   
   For more information, see “Adding an index to a What-If scenario” on page 83.

2. Click **Edit**.

3. Complete the Index Attributes dialog box.
   
   For more information, see “Adding an index to a What-If scenario” on page 83.

**Where to go from here**

Explain and compare the statement after you have completed editing.
Generating a tuned statement

After evaluating the effects of What-If indexes on statement performance, you can generate the SQL statements required to create indexes, drop indexes, and update index statistics.

To generate a tuned statement

1. Create a What-If scenario.
   For more information, see “Creating and editing a What-If index scenario” on page 81.

2. Make changes as needed to the indexes:
   - “Adding an index to a What-If scenario” on page 83
   - “Dropping an index from a What-If scenario” on page 85
   - “Editing an index for a What-if scenario” on page 87
   - “Copying an index for a What-If scenario” on page 86

3. Explain the What-If scenario.
   For more information, see “Explaining SQL statements” on page 78.

4. Click Generate.
   The Generated Statements pane displays the statement required to make the index changes.

5. Click Save.

6. Save the file to the required data set.
   For more information, see “Editing text files in Scratchpad” on page 70.

Comparing statements

Use the following procedure to compare Explain results within the same tuning session.
Before you begin

Create one or more What-If statements, as instructed in “Creating a What-If scenario” on page 79.

Note

The Compare tab is not available until you create at least one What-If scenario.

To compare statements

1 In the Performance perspective, open the Compare tab.

2 From the tuning session pane, in the comparison pane on the left, select the first statement for comparison.

Note

The first statement is always displayed in the comparison panel on the left, and the Performance perspective’s tabs show information about that statement.

3 From the list in the comparison pane on the right, select a second statement for comparison.

4 Click any step node of the Explain tree to compare the attributes of the selected node.

The results pane shows detailed information for the selected nodes in the trees.

Explain trees of both statements are displayed side-by-side, and in the results pane, a table compares attributes of the two statements. In each row, the first column displays the attribute name, the second column displays the value of the node in the first statement (left tree), and the third column shows the value of the second statement (right tree).

Note

* indicates that the attribute has different values in the two statements being compared.

Related Information

- “Compare tab” on page 101
Selecting a statement for SQL Analysis

You can perform SQL analysis on a statement selected from the DB2 statement cache, on a statement selected from a DB2 package, or on an ad hoc statement.

1. Perform one of the following tasks:
   - “Analyzing a statement from the DB2 statement cache” on page 90
   - “Analyzing a statement from a DB2 package” on page 91
   - “Analyzing an ad hoc statement” on page 92

Analyzing a statement from the DB2 statement cache

Use the following procedure to select and tune any statement in the DB2 statement cache. You must ensure that the workspace contains a DB2 object view that contains the statement.

To analyze a statement from the DB2 statement cache

1. Click the DB2 Navigator tab.
2. Click Add DB2 statement cache view.
3. In the DB2 Statement Cache for Program Filter dialog box, select a DB2 subsystem:
   - Type the full or partial name of a DB2 connection to select the first subsystem that matches this value in the list of favorite connections.
   - Use the menu arrow to select from your list of favorite connections.

   Note

   Click ➔ and specify which DB2 connections to include in your list of favorite connections.

4. In the filter box, enter the name of the program that contains the statement.

   You can use wildcards to return a list of statements contained by programs that match the pattern.
5 Click **OK**.

In the results pane, a list of statements is displayed.

---
**Note**
Clicking the column headers sorts the results.
---

6 Select the desired statement and click **Explain**.

The Performance perspective displays the graphical Explain for the selected statement in the Explain screen.

---
**Related Information**

- “Selecting a statement for SQL Analysis” on page 90
---

## Analyzing a statement from a DB2 package

Use the following procedure to select and tune a statement from any package.

**To analyze a statement from a package**

1 Click the DB2 Navigator tab.

2 Open or create a package view that contains the required statement.

3 Select the package containing the required statement.

4 From the command toolbar, click **Show Explainable Statements**.

The results pane displays all Explainable statements for that package.

5 Search for and select the required statement.

6 Perform one of the following actions:

- Click **Properties** to view the statement properties.
Click **Explain** ![Explain icon] to display the graphical Explain of the selected statement in the Explain tab. You can now perform SQL tuning tasks.

**Where to go from here**

You can return to the results pane in the DB2 Navigator tab to select other statements.

---

**Related Information**

- “Selecting a statement for SQL Analysis” on page 90

---

**Analyzing an ad hoc statement**

Use the following procedure to browse and copy a statement stored in a data set, or to write a statement into the statement box of the Performance perspective.

**Note**

You can set values that are persistent for all ad hoc statements that you create in the current session in the current workspace. For details, see “Setting SQL tuning options” on page 104.

**To analyze an ad hoc statement**

1. Click the Performance perspective.

2. Click **Add SQL Statement View** ![Add SQL Statement View icon].

3. Complete the SQL tab:
   a. Enter a name for the statement.
   b. Perform one of the following tasks:
   c. Select a DB2 subsystem by using either of the following methods:
      - Type the full or partial name of a DB2 connection to select the first subsystem that matches this value in the list of favorite connections.
      - Use the menu arrow to display and choose from a list of connected subsystems.
Tip

Click and specify which DB2 connections to include in your list of favorite connections.

d (optional) Enter the table qualifier.

BMC Workbench for DB2 uses this qualifier if the table in the SQL is unqualified.

e For the Explain Type, select Dynamic.

f At Degree, select the degree of parallel processing to allow:

- Select Any to allow parallel processing.
- Select 1 to prohibit parallel processing.

g Type or paste the SQL statement into the SQL statement box.

4 Click Generate explain plan .

The graphical Explain diagram of the SQL statement is displayed.

Related Information

- “Selecting a statement for SQL Analysis” on page 90

Performance perspective

The commands in the Performance perspective enable you to analyze SQL statements, perform What-If comparisons, and migrate access path object statistics.

Selecting a SQL statement for analysis (see “Selecting a statement for SQL Analysis” on page 90), displays the Performance perspective.

The Performance perspective provides access to the result view tabs, which enable you to perform the tuning process. Depending on the activities being performed, the following tabs can be displayed:

- SQL tab
- Indexes tab
- Explain tab
- Cost tab
- Predicates tab
- Compare tab (shown in Figure 2 on page 94)
- Related Objects tab
- Output tab

Figure 2: Performance perspective (Compare tab displayed)

Legend

<table>
<thead>
<tr>
<th>#</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Tuning session pane</td>
</tr>
<tr>
<td>2</td>
<td>Buttons for adding SQL statement views (+) and removing (-) views</td>
</tr>
<tr>
<td>#</td>
<td>Description</td>
</tr>
<tr>
<td>----</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| 3  | Active workspace name  
An asterisk (*) indicates that the workspace has unsaved changes. |
| 4  | Perspective tabs  
The Performance tab is blue, indicating that it is the active tab. |
| 5  | Generate Explain plan button  
The button is dimmed after the statement is successfully explained. |
| 6  | What-If button  
The What-If button is disabled (dimmed) on a selected statement if the statement has not been successfully explained. Unless the statement is successfully explained, you cannot create a What-If scenario based on that statement. |
| 7  | Run button |
| 8  | Migrate statistics button  
This button lets you migrate access path statistics for one or more objects. |
| 9  | Name and ID of the active statement |
| 10 | (Compare tab only) Explain tree of the first statement  
**Note:** 🌟 indicates that at least one attribute of an object has different values in the two statements being compared. |
| 11 | The Result view tabs that show the result of explaining a statement, a compare, and cataloged objects. |
| 12 | (Compare tab only) Explain tree of the second statement |
| 13 | Explain timestamp and DB2 subsystem name |
| 14 | Options button |
| 15 | (Compare tab only) Attribute details of the second statement |
| 16 | (Compare tab only) Attribute details of the first statement |
| 17 | (Compare tab only) Attribute names  
**Note:** 🌟 indicates that the attribute has different values in the two statements being compared. |
The SQL tab opens when you select a statement for Explain in the Performance perspective, or when you create a tuning session for an ad hoc SQL statement.

If you select a statement from a package or from the statement cache, the statement is displayed in the SQL tab.

**Tip**
If you want to edit a statement selected from the statement cache or from a package, you must create a What-If scenario.

If you are creating an ad hoc tuning session, you must enter the statement into this pane. You must also populate the dialog box. For more information, see “Analyzing an ad hoc statement” on page 92.

**Note**
After the statement has been explained, click What-If to modify the statement, or enter new parameters to run a subsequent Explain.
Indexes tab

The Indexes tab lists indexes for all tables participating in the Explain plan. The tab is displayed only after the baseline statement is Explained.

Buttons on the Indexes tab

The following buttons are available:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Adds an index to a What-If scenario</td>
</tr>
<tr>
<td>Copy</td>
<td>Build a copy of an existing index</td>
</tr>
</tbody>
</table>
| Edit   | Edits the attributes of an index  
  ■ For an existing index, you can only edit the statistics.  
  ■ For an added or copied index, you can edit all fields. |
| Drop   | Drops an index |
| Revert | Reverts changes made for the selected index  
  ■ If the selected index was added, the index is dropped.  
  ■ If the selected index was dropped, the index is reinstated. |
| Generate | Generates statements reflecting changes that you made to a What-If scenario  
  This command button is dimmed unless you made changes to the SQL statement. |

Note

The command buttons are available only:

- For What-If scenarios
  You can upgrade statistics only for existing indexes, but you can copy indexes from existing indexes.

- For unexplained statements
  After you have Explained a statement, the command buttons are dimmed.

- If you have a license for BMC Performance for DB2SQL

Panes

The Indexes tab has a Tables pane and an Indexes pane:

- The Tables pane lists all tables that are referenced in the SQL statement.
The Indexes pane lists the indexes of the selected table (by default, the table shown at the top of the Tables pane).

**Note**

- ✓ indicates an index that participates in the Explain plan.
- ✖ indicates an index that has been dropped.
- + indicates an index that has been added.
- ★ indicates an index that has been edited

**Related Information**

- “Performance perspective” on page 93

---

**Explain tab**

The Explain tab displays a visual representation of the DB2 Explain plan.

You can view the DB2 statement in the format of an Explain diagram (default) or an Explain tree:

- The *Explain diagram* displays each step as a node and indicates the step's cost, type, and message severity level. The following icons indicate the step's type:
  
  - Operator
  - Query block
  - Table
  - Index

The node displays one of the following icons to indicate the highest severity of the message rules associated with that step (if a message rule exists):

- Severe
To focus on specific steps, you can move the nodes around the screen and zoom in and out.

- The *Explain tree* lets you expand or hide steps in order to focus on specific steps.

Graphical Explain enables you to perform the following tasks:

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View step details</td>
<td>1. Perform one of the following actions:</td>
</tr>
<tr>
<td></td>
<td>■ In the Explain diagram, click the linked step name of a node.</td>
</tr>
<tr>
<td></td>
<td>■ In the Explain tree, click the node.</td>
</tr>
<tr>
<td></td>
<td>2. In the Explain Detail property box, select the Step Details tab to</td>
</tr>
<tr>
<td></td>
<td>view the step information from the plan table.</td>
</tr>
<tr>
<td>View message rules</td>
<td>1. Perform one of the following actions:</td>
</tr>
<tr>
<td></td>
<td>■ In the Explain diagram, click the linked step name of a node.</td>
</tr>
<tr>
<td></td>
<td>■ In the Explain tree, click the node.</td>
</tr>
<tr>
<td></td>
<td>2. In the Explain Detail property box, select the message rules tab to</td>
</tr>
<tr>
<td></td>
<td>view the message rules (those delivered by BMC and user-defined message</td>
</tr>
<tr>
<td></td>
<td>rules).</td>
</tr>
<tr>
<td>Toggle between the Explain diagram and Explain tree formats</td>
<td>Click the <strong>Explain Display</strong> toggle.</td>
</tr>
<tr>
<td>Move the node</td>
<td>Click the step node and hold the mouse button while moving the mouse.</td>
</tr>
<tr>
<td>Zoom in and out</td>
<td>Move the Zoom slider, or repeatedly click <strong>Zoom in</strong> or <strong>Zoom out</strong>.</td>
</tr>
<tr>
<td>Move a node in the diagram to the top of the screen</td>
<td>To move a node in a diagram to the top of the screen, double-click the</td>
</tr>
<tr>
<td></td>
<td>node.</td>
</tr>
<tr>
<td><strong>Tip:</strong> Clicking <strong>Reset Diagram</strong> reverts to the initial state of the graph.</td>
<td></td>
</tr>
</tbody>
</table>
### Task

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revert the positioning of the Explain diagram to the initial positions</td>
<td>Click <strong>Reset Diagram</strong>.</td>
</tr>
<tr>
<td>Expand or hide steps in the Explain tree</td>
<td>Click the expansion arrow next to the step node.</td>
</tr>
</tbody>
</table>

---

**Related Information**

- “Performance perspective” on page 93

---

**Cost tab**

The Cost tab displays the total cost of the Explain plan and other details about the Explain plan.

The results pane is divided into an upper and lower pane:

- The upper pane displays the cost and other details that are extracted from the DB2 plan table.
- The lower pane lists BMC generated message rules and any user-defined message rules.

The associated icon identifies the rule as Severe ![Severe](image), Warning ![Warning](image), or Informational ![Informational](image).

---

**Related Information**

- “Performance perspective” on page 93

---

**Predicates tab**

The Predicates tab displays the filter factors and indexability of each predicate.

The predicates are listed in a table in the order in which they are extracted from the Explain plan tables.
**Compare tab**

The Compare tab displays side-by-side Explain trees representing two statements that are being compared.

You can create a What-If scenario and compare it to the source statement or to another What-If scenario. For more information, see “Comparing statements” on page 88.

**Note**
The Compare tab is available only if the tuning session contains two or more statements.

See “Performance perspective” on page 93 for an example of the Compare tab.

**Related Objects tab**

The Related Objects tab enables you to view objects related to the tables that are referenced in the statement.

The tab contains two panes:

- The Tables pane lists all of the tables referenced in the SQL statement.
- The other pane contains the Related Objects toolbar and a results table. The results table lists all of the related objects of the selected object type.

You can perform the following tasks:

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select a table</td>
<td>On the Tables pane, select the required table. When you change the selected table, the results table is updated.</td>
</tr>
</tbody>
</table>
Select a related object

From the Related Objects toolbar, click an object-type button. The results table is populated with all objects of this type that are related to the selected table.

**Note:** You are notified if no related objects of that type exist.

### Table 4: Object-type buttons

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Accelerators" /></td>
<td>Accelerators</td>
</tr>
<tr>
<td><img src="image2.png" alt="Accelerator packages" /></td>
<td>Accelerator packages</td>
</tr>
<tr>
<td><img src="image3.png" alt="Accelerator tables" /></td>
<td>Accelerator tables</td>
</tr>
<tr>
<td><img src="image4.png" alt="Aliases" /></td>
<td>Aliases</td>
</tr>
<tr>
<td><img src="image5.png" alt="Data types" /></td>
<td>Data types</td>
</tr>
<tr>
<td><img src="image6.png" alt="Databases" /></td>
<td>Databases</td>
</tr>
<tr>
<td><img src="image7.png" alt="Data sets" /></td>
<td>Data sets</td>
</tr>
<tr>
<td><img src="image8.png" alt="Key columns" /></td>
<td>Key columns</td>
</tr>
<tr>
<td><img src="image9.png" alt="Image copy" /></td>
<td>Image copy</td>
</tr>
<tr>
<td><img src="image10.png" alt="Indexes" /></td>
<td>Indexes</td>
</tr>
<tr>
<td><img src="image11.png" alt="Packages" /></td>
<td>Packages</td>
</tr>
<tr>
<td><img src="image12.png" alt="Plans" /></td>
<td>Plans</td>
</tr>
<tr>
<td><img src="image13.png" alt="Procedures" /></td>
<td>Procedures</td>
</tr>
<tr>
<td><img src="image14.png" alt="Sequences" /></td>
<td>Sequences</td>
</tr>
<tr>
<td>Button</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>![Storage groups icon]</td>
<td>Storage groups</td>
</tr>
<tr>
<td>![Synonyms icon]</td>
<td>Synonyms</td>
</tr>
<tr>
<td>![Tables icon]</td>
<td>Tables</td>
</tr>
<tr>
<td>![Table columns icon]</td>
<td>Table columns</td>
</tr>
<tr>
<td>![Table constraint icon]</td>
<td>Table constraint</td>
</tr>
<tr>
<td>![Table spaces icon]</td>
<td>Table spaces</td>
</tr>
<tr>
<td>![Table space partitions icon]</td>
<td>Table space partitions</td>
</tr>
<tr>
<td>![Triggers icon]</td>
<td>Triggers</td>
</tr>
<tr>
<td>![Views icon]</td>
<td>Views</td>
</tr>
</tbody>
</table>

**Related Information**

- “Performance perspective” on page 93

---

**Output tab**

The Output tab displays the results of an executed SQL statement. The Output tab is displayed when you click **Run** in the Performance perspective.

**Note**

The Run button is dimmed and unavailable when the Output tab is selected.

---

**Related Information**

- “Performance perspective” on page 93
Setting SQL tuning options

You can use the **Options** button to set optional behavior of the SQL tuning feature.

The following tabs are displayed:

- The Tuning tab sets the options for defining SQL tuning settings for the current session.
- The Execution tab sets the options for executing SELECT statements.
- The Statistics Migration tab sets the options for statistics migration. For more details, see “Setting statistic migration options” on page 113.

### Tuning tab

The Tuning tab lets you set the following types of options: dynamic SQL, ad hoc SQL, and general.

#### Table 5: Dynamic SQL options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plan Table Owner</td>
<td>Specifies the name of the owner for the DB2 plan table used</td>
<td>BMC</td>
</tr>
</tbody>
</table>

**Note:** If you want the owner to use lowercase characters, verify that you have not enabled Convert filter values to upper case. See “Setting BMC Workbench options” on page 35.

#### Table 6: Ad Hoc SQL options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Qualifier</td>
<td>Specifies a table qualifier to qualify DB2 objects that are not qualified in the SQL statement</td>
<td>No default value</td>
</tr>
<tr>
<td>Degree</td>
<td>Specifies whether to consider parallel processing during an Explain:</td>
<td>Default = Any</td>
</tr>
<tr>
<td></td>
<td>- Any considers parallel processing.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- 1 does not consider parallel processing.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Blank uses the value at bind time.</td>
<td></td>
</tr>
</tbody>
</table>
**Table 7: General options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule set</td>
<td>Defines the message rules:</td>
<td>DEFAULT</td>
</tr>
<tr>
<td></td>
<td>■ DEFAULT displays all rules that are related to performance issues and are primarily relevant to DBAs.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ APPDEV displays rules that are primarily relevant to application developers.</td>
<td></td>
</tr>
<tr>
<td>Online Application</td>
<td>Specifies whether to trigger specific rules for table space scan, list and sequential prefetch, and multiple index access paths (MIAP) considerations:</td>
<td>Unchecked</td>
</tr>
<tr>
<td></td>
<td>■ Select to trigger the rules</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ Clear to not trigger the rules</td>
<td></td>
</tr>
<tr>
<td>Cost Translation Rate per Timeron</td>
<td>Specifies the rate used to translate the timeron cost into a monetary unit</td>
<td>1.0</td>
</tr>
<tr>
<td>Cost Translation Rate per Service Unit</td>
<td>Specifies the rate used to translate the service unit cost into a monetary unit</td>
<td>1.0</td>
</tr>
</tbody>
</table>

**Execution tab**

The Execution tab lets you set the select options.

**Table 8: Select options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select Options</td>
<td>For SELECT statements, this option lets you set maximum values for the following items (with defaults shown in parentheses):</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>■ Maximum numeric length (10)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ Maximum CHAR length (64)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ Maximum VARCHAR length (64)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ Maximum select rows (300), where 0 returns all rows in the result set.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The total amount of returned data in the result set cannot exceed 50 MB; the Select Options settings do not affect that limit. If the limit is reached, the product alerts you with a message.</td>
<td></td>
</tr>
</tbody>
</table>
Related Information

- “Analyzing an ad hoc statement” on page 92
Migrating statistics

BMC Workbench enables you to migrate the access path statistics for a set of objects from one subsystem to another, or from one schema to another within the same subsystem.

For the selected objects, BMC Workbench migrates the table, table space, and index statistics that affect the access-path.

For example, you can migrate production statistics to a test environment to view the actual access paths that the DB2 optimizer selects, without the cost of replicating the production environment.

You can select objects from a Results list in the DB2 Navigator perspective, or select objects included in a SQL Tuning session in the Performance perspective.

You can reuse a saved Migrate Statistics session. In the reused session, you can update objects with current access-path statistics, or change the source or target of the statistics.

You can use the Options button to set optional behaviors of the statistics migration feature.

For more information, view the Quick Course "Workbench for DB2 - Migrating Statistics."

Migrating access-path statistics from the DB2 Navigator

To update objects with migrated access-path statistics, use the following procedure to select objects for statistics migration from the DB2 Navigator.
To migrate access-path statistics from the DB2 Navigator

1. Click the DB2 Navigator perspective.

2. Open or create an object view.

3. Select one or multiple objects from the database, table space, table, or index list.

   **Note**
   The table, table space, and index statistics that affect the access path are migrated for the selected objects.

4. Click Migrate Statistics, and then click either Migrate Statistics From or Migrate Statistics To:

   - Select Migrate Statistics From to migrate statistics from the objects selected from the Results list. The object or objects that you want to update must already exist on the target subsystem.
   - Select Migrate Statistics To to migrate statistics to the objects selected from the Results list.

   **Tip**
   Alternatively, you can right-click the selected objects and select Tasks => Migrate Statistics => Migrate Statistics From or Migrate Statistics To.

5. Complete the following fields of the Parameters of the Migrate Statistics panel, and then click Next:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a name for the Migrate Statistics session.</td>
</tr>
<tr>
<td>Include Table Statistics</td>
<td>Select this box to migrate the table statistics for the selected object.</td>
</tr>
<tr>
<td></td>
<td>This box is initially selected by default.</td>
</tr>
<tr>
<td>Include Index Statistics</td>
<td>Select this box to migrate the statistics of all indexes for the selected object.</td>
</tr>
<tr>
<td></td>
<td>This box is initially selected by default.</td>
</tr>
<tr>
<td>DB2 Connections</td>
<td>Select the DB2 subsystem from the list of subsystems that are connected to the source DB2 subsystem.</td>
</tr>
<tr>
<td></td>
<td>If you selected Migrate Statistics From, you are selecting the target DB2 subsystem.</td>
</tr>
<tr>
<td></td>
<td>If you selected Migrate Statistics To, you are selecting the source DB2 subsystem.</td>
</tr>
</tbody>
</table>
Define relevant translation rules for source objects that you are migrating, so that they match the target objects. You can replace all occurrences of a specified string with another specified string. Do not use wildcard characters (including % or *).

Clicking **Next** displays the Object List of the Migrate Statistics panel, which lists the tables, table spaces, and indexes to be migrated. The panel also indicates the source and target direction of the migration and displays a status icon beside each object. The status icon indicates all valid objects with ✅, and invalid objects with ⚠.

6 For each invalid object, take one of the following actions:

- Click **Back**, and update the translation rules. Then click **Next**.

- Select and update the invalid name. Then click Verify ✅ or Verify all ✅.

7 Clear any objects that you do not want to migrate.

**Tip**

Toggle ✅ to select or clear all objects.

You can use the filter buttons to view specific object types, or to view only the valid or invalid objects.

8 *(optional)* Click **Next** to view the Summary.

9 To start the migration, click **Finish**.

The results are displayed in a results table in the Results tab in the Performance perspective:

- ✅ identifies objects that migrated successfully.

- ⚠ identifies objects that failed to migrate, and a message explains the failure. You can hover over the message to view details.

The Migrate Statistics session is listed under **Stats Migration** in the Navigation pane in the Performance perspective.

10 For any objects that failed to migrate, complete this step:

a Hover the cursor over the explanatory message to see details.
b Click the Parameters tab.

c Make the required changes to the Translation parameters.

d When the following message is displayed at the bottom of the panel, click to refresh the objects list.

Input parameters have been updated. The object list is stale.

e On the toolbar, click to repeat the migration.

A new Results tab displays updated results.

Statistics migration updates the following DB2 statistics tables:

SYSTABLESPACE
SYSTABLES
SYSCOLUMNS
SYSCOLSTATS
SYSCOLDIST
SYSTABSTATS
SYSINDEXES
SYSKEYTARGETS
SYSKEYTARGETSTATS
SYSKEYTGTDIST
SYSKEYTGTDISTSTATS

---Related Information---

■ “Migrating access-path statistics from a SQL Tuning session” on page 110

■ “Setting statistic migration options” on page 113

---

Migrating access-path statistics from a SQL Tuning session

To update objects with migrated access-path statistics, use the following procedure to select objects for statistics migration from a SQL Tuning session.
To migrate access-path statistics from a SQL Tuning session

1. Select the Performance perspective.

2. From a tuning session, either select a statement that has been explained or perform Explain on the statement.

   See “Explaining SQL statements” on page 78.

3. Click Migrate Statistics, and then click either Migrate Statistics From or Migrate Statistics To:

   - Select Migrate Statistics From to migrate statistics from the objects included in the SQL Tuning session. The object or objects that you want to update must already exist on the target subsystem.

   - Select Migrate Statistics To to migrate statistics to the objects included in the SQL Tuning session.

4. Complete the following fields of the Parameters of the Migrate Statistics panel, and then click Next:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a name for the Migrate Statistics session.</td>
</tr>
<tr>
<td>Include Table Statistics</td>
<td>Select this box to migrate the table statistics for the selected object. This box is initially selected by default.</td>
</tr>
<tr>
<td>Include Index Statistics</td>
<td>Select this box to migrate the statistics of all indexes for the selected object. This box is initially selected by default.</td>
</tr>
<tr>
<td>DB2 Connections</td>
<td>Select the DB2 subsystem from the list of subsystems that are connected to the source DB2 subsystem.</td>
</tr>
<tr>
<td></td>
<td>If you selected Migrate Statistics From, you are selecting the target DB2 subsystem.</td>
</tr>
<tr>
<td></td>
<td>If you selected Migrate Statistics To, you are selecting the source DB2 subsystem.</td>
</tr>
<tr>
<td>Translation</td>
<td>Define relevant translation rules for source objects that you are migrating, so that they match the target objects. You can replace all occurrences of a specified string with another specified string. Do not use wildcard characters (including % or *).</td>
</tr>
</tbody>
</table>

Clicking Next displays the Object List of the Migrate Statistics panel, which lists the tables, table spaces, and indexes to be migrated. The panel also indicates the source and target direction of the migration and displays a status icon beside each.
object. The status icon indicates all valid objects with ✔️, and invalid objects with 🔄.

5 For each invalid object, take one of the following actions:

- Click Back, and update the translation rules. Then click Next.

- Select and update the invalid name. Then click Verify 🔄 or Verify all 🔄.

6 Clear any objects that you do not want to migrate.

**Tip**

Toggle 🔄 to select or clear all the objects.
You can use the filter buttons to view specific object types, or to view only the valid or invalid objects.

7 (optional) Click Next to view the Summary.

8 To start the migration, click Finish.

The results are displayed in a results table in the Results tab in the Performance perspective:

- ✔️ identifies objects that migrated successfully.

- 🔄 identifies objects that failed to migrate, and a message explains the failure.
  You can hover over the message to view details.

The Migrate Statistics session is listed under Stats Migration in the Navigation pane in the Performance perspective.

9 For any objects that failed to migrate, complete this step:

a Hover the cursor over the explanatory message to see details.

b Click the Parameters tab.

c Make the required changes to the Translation parameters.

d When the following message is displayed at the bottom of the panel, click 🔄 to refresh the objects list.

  Input parameters have been updated. The object list is stale 🔄.
On the toolbar, click ⏳ to repeat the migration.

A new Results tab is displayed, which displays updated results.

Statistics migration updates the following DB2 statistics tables:

- SYSTABLESPACE
- SYSTABLES
- SYSCOLUMNS
- SYSCOLSTATS
- SYSCOLDIST
- SYSTABSTATS
- SYSINDEXES
- SYSKEYTARGETS
- SYSKEYTARGETSTATS
- SYSKEYTGTDIST
- SYSKEYTGTDISTSTATS

**Related Information**

- “Migrating access-path statistics from the DB2 Navigator” on page 107
- “Setting statistic migration options” on page 113

---

**Setting statistic migration options**

Use the following procedure to set options for migrating access-path statistics. You can set these options from the Performance perspective, the DB2 Navigation perspective, or the Migrate Statistics panel.

1. Select the **Options 👉** button.

2. On the Statistics Migration tab, set the following options and click **OK**:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp to use for STATSTIME</td>
<td>Specifies the timestamp used to update STATSTIME on the target object:</td>
</tr>
<tr>
<td></td>
<td>- Select <strong>Current</strong> to use the machine time.</td>
</tr>
<tr>
<td></td>
<td>- Select <strong>Source</strong> to use the source object timestamp.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Continue on SQL Error         | Tells BMC Workbench what to do if the migration process encounters an object containing a SQL error:  
|                               | ■ Selecting this option tells BMC Workbench to continue the migration.  
|                               | ■ Clearing this option tells BMC Workbench to exit the migration.         |
| Delete statistics before migration | Specifies whether to delete all previous instances of the target table spaces' SYSCOLDIST, SYSTABSTATS, SYSKEYTGTDIST, and SYSKEYTGTDISTSTATS tables before inserting the migrated statistics row or rows |

**Related Information**

- “Migrating access-path statistics from the DB2 Navigator” on page 107
- “Migrating access-path statistics from a SQL Tuning session” on page 110
Displaying BMC utilities

You can view a list of all BMC Utilities that are running on a specified IBM DB2 subsystem. From this list, you can drill down and view the objects that each utility is accessing.

Creating a Utility Status view

Use the following procedure to create a new Utility Status view.

1. Open or create a workspace.

2. In the DB2 Navigator perspective, click the Add View menu arrow, and then click Add Utility Status View.

   Tip
   You can also right-click in the DB2 navigation pane to select this action.
   The default value of the Add View menu is always the last option that you created. For example, if the last view you created was a new DB2 Statement Cache view, then clicking creates a new DB2 Statement Cache view.

3. In the Add Utility Status View dialog box, complete the following fields or accept the displayed default values:
### Field | Action
--- | ---
DB2 Connection | Use either of the following methods:
  - Type the full or partial name of a DB2 connection to select the first subsystem that matches this value in the list of favorite connections.
  - Use the menu arrow to select from your list of favorite connections.
  **Tip**: You can add DB2 connections to your list of favorite connections by clicking and specifying which connections to include.
Filter | Enter a filter pattern for the Utility ID.

4. Click **OK**.

The navigation pane lists the matching utilities and displays the following information about each one:

  - Utility ID
  - User ID
  - Utility name
  - Status
  - Phase
  - Restart
  - Database
  - Space name
  - Space status
  - Start timestamp
  - Command
  - NOTEID

5. *(optional)* Select a utility and click **Utility Status Details**.

  **Note**
  You can also right-click on a utility to select this action.

The Utility Status Details screen shows the objects that the utility accesses and includes the following information about each object:

  - Fully qualified object name
  - Partition
  - Blocks
- BMCID
- DDNAME
- EXTRABA
- KIND
- ORIG_STATUS
- SHRLEVEL
- XCOUNT

6 When finished, save the workspace.
Setting up NGT Utility Management

This topic describes how to set up the BMC Next Generation Technology Utility Manager for DB2 for z/OS product. The NGT Utility Manager uses the IBM DB2 Real Time Statistics (RTS) facility and user-defined criteria to identify which database objects need to be processed by the NGT utility products, and under what conditions.

NGT Utility Manager processes table spaces and indexes according to values contained in the NGT Utility Manager tables, which are installed during the installation of any NGT utility:

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Criteria table</td>
<td>Defines the criteria for selecting DB2 objects for processing</td>
</tr>
<tr>
<td>Exceptions table</td>
<td>Defines deviations from regular processing (for example, preventing reorganization during ad hoc maintenance)</td>
</tr>
<tr>
<td>Schedules table</td>
<td>Defines periodic processing conditions (for example, preventing reorganization on the specified day of the month)</td>
</tr>
</tbody>
</table>

Using BMC Workbench, you can specify values in each of these tables for groups of indexes, table spaces, and applications.

After setting up NGT Utility Manager, you activate it by including the RTS keyword in the SYSIN command of an NGT utility product.

**Example**

```
//SYSIN DD *
COPY TABLESPACE CENTRAL.ACCTGREC RTS
```

For more information, view the Quick Course "Workbench for DB2 - Setting up NGT Utility Manager."

For more information, see *BMC Next Generation Technology Utility Manager for DB2 for z/OS Reference Manual*. 
Adding an NGT Utility Management view

You set up and edit the NGT Utility Manager tables in an NGT Utility Management view. Use the following procedures to add a new NGT Utility Management view.

**To add an NGT Utility Management view**

1. Open or create a workspace.

2. In the Product Tools perspective, click the Add View menu arrow, and then click **NGT Utility Management View**.
   
   *Tip*  
   You can also right-click in the navigation pane to select this action. The default value of the Add View menu is always the last option that you created.

3. Select a DB2 connection using either of the following methods:
   - Type the full or partial name of a DB2 connection to select the first subsystem that matches this value in the list of favorite connections.
   - Use the menu arrow to select from your list of favorite connections.
   
   *Tip*  
   You can add DB2 connections to your list of favorite connections by clicking and specifying which connections to include.

4. Click **OK**.

   The new NGT Utility Management view is listed in the Navigation pane, and the Criteria, Exceptions, and Schedules tables for that subsystem are listed in the Results pane.

5. Click **** to save the workspace.
Managing the NGT Utility Management criteria settings

You can set utility processing criteria for groups of table spaces, indexes, and applications. Use the following procedures to add to, edit, or delete from the criteria table.

To add criteria

1. In the Product Tools perspective, select or add an NGT Utility Management view. The Results pane is displayed, which contains the Criteria table and the Exceptions table.

   **Note**

   Whenever you want, you can click to refresh the criteria.

2. In the Criteria table, click Add Criteria.

   You can also right-click in the Results pane to select this action.

3. In the Add NGT Management Criteria dialog box, complete the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Select one of the following types:</td>
</tr>
<tr>
<td></td>
<td>■ Applications</td>
</tr>
<tr>
<td></td>
<td>■ Table spaces</td>
</tr>
<tr>
<td></td>
<td>■ Indexes</td>
</tr>
<tr>
<td>Name pattern</td>
<td>Enter a name pattern that matches the object type that you selected. You can use both the * and the % wildcards.</td>
</tr>
<tr>
<td>Partition</td>
<td>Enter a partition number or specify 0 for all partitions.</td>
</tr>
<tr>
<td>Comment</td>
<td>Enter a comment describing the criteria.</td>
</tr>
<tr>
<td>Last update</td>
<td>This read-only field (displayed only for an edited criterion) shows the user ID and timestamp of the last update to this criterion</td>
</tr>
</tbody>
</table>

4. According to the object type that you want to define, select the **Table Space Criteria** tab or the **Index criteria** tab.

   The tab lists trigger criteria, grouped by utility (NGT Reorg triggers, NGT Copy triggers, and NGT Stats triggers).
5 Edit any trigger criteria values that you want to change.

Tip
Hovering over a trigger criteria name lets you view a description.

6 (optional) If you want to add or edit a trigger on a WHERE clause, enter a valid WHERE clause.

NGT Utility Manager uses the NGT SQL language for WHERE clauses. For more information, see “NGT SQL language” on page 128.

Note
BMC Workbench does not validate the WHERE clause.

7 Click OK.

To edit a criterion

1 In the Product Tools perspective, select a NGT Utility Management view.

2 In the Criteria table, select a criterion, then click Edit criteria .

3 In the Edit Utility Management Criteria dialog box, edit the General settings.

   The criterion that you are editing is displayed at the top of the dialog box, click the arrow to see and select another criterion. You can also use the Previous criteria and Next criteria arrows to navigate to another criterion.

4 According to the object type that you want to define, select the Table Space Criteria tab or the Index criteria tab.

5 Edit any trigger criteria values that you want to change.

Tip
Hovering over the trigger criterion name lets you view a description.

6 Click OK.

To delete criteria

1 In the Product Tools perspective, select a NGT Utility Management view.

2 In the Criteria table, select one or more criteria for deletion.
Managing the NGT Utility Management exceptions settings

You can set exceptions to the utility processing criteria that you defined in the Criteria table. You can specify that the process is performed on specific groups of indexes, table spaces, and applications; or is not performed when certain conditions exist. Use the following procedures to add to, edit, or delete from the Exceptions table.

**To add an exception**

1. In the Product Tools perspective, select or add a NGT Utility Management view.

   The Results pane is displayed, which contains the Criteria table and the Exceptions table.

2. In the Exceptions table, click Add exception .

   You can also right-click in the Results pane to select this action.
3 in the NGT Utility Management Exception dialog box, complete the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Type**    | Select one of the following types:  
  ■ Applications  
  ■ Table spaces  
  ■ Indexes                                                                                           |
| **Name pattern** | Enter a name pattern for the group of objects or applications for which you want to create an exception. You can use both the * or the % wildcards. |
| **Partition** | Enter a partition number or specify 0 for all partitions.                                                                                  |
| **Comment** | Enter a comment describing the exception.                                                                                                    |
| **Last update** | This read-only field (displayed only for an edited exception) shows the user ID and timestamp of the last update to this exception.          |

4 Select the Exceptions tab.

5 Select the rule type for this exception for a utility or utilities.

You can tell NGT Utility Manager to force the process for the designated utilities, or to exclude the designated utilities from the process, according to the conditions in the WHERE clause.

6 Select the appropriate check box to instruct NGT Utility Manager to apply or ignore the exception for that utility.

7 Enter a valid WHERE clause for the exception.

NGT Utility Manager uses the NGT SQL language for WHERE clauses. For more information, see “NGT SQL language” on page 128.

**Note**

BMC Workbench does not validate the WHERE clause.

8 Click OK.

**To edit an exception**

1 In the Product Tools perspective, select a NGT Utility Management view.

2 In the Exceptions table, select an exception, then click **Edit exception**.
3 In the Edit Utility Management Exception dialog box, edit the General settings.

The exception that you are editing is displayed at the top of the dialog box, click the arrow to see and select another exception. You can also use the Previous exception and Next exception arrows to navigate to another exception.

4 Select the Exceptions tab.

5 Edit the exception that you want changed.

6 Click OK.

--- Related Information ---

- “NGT SQL language elements” on page 128
- “Features and functions of the NGT SQL language” on page 131
- “Special registers that the NGT SQL language supports” on page 134
- “Constants that NGT SQL supports” on page 136
- “Functions that NGT SQL supports” on page 139
- “Date and time durations” on page 177
- “Comparisons” on page 179
- “NGT SQL language keywords” on page 180

---

Managing the NGT Utility Management schedule settings

You can schedule the utility processing criteria for groups of table spaces, indexes, and applications. Use the following procedures to add to, edit, or delete from the schedule settings table.

To add a schedule

1 In the Product Tools perspective, select or add a NGT Utility Management view.

   The Results pane is displayed.
2 In the Results pane, click the **Schedules** tab.

3 In the Schedules table, click Add schedule <button>.

You can also right-click in the Results pane to select this action.

4 In the General tab, complete the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td>Select one of the following types:</td>
</tr>
<tr>
<td></td>
<td>■ Applications</td>
</tr>
<tr>
<td></td>
<td>■ Table spaces</td>
</tr>
<tr>
<td></td>
<td>■ Indexes</td>
</tr>
<tr>
<td><strong>Name pattern</strong></td>
<td>Enter a name pattern for the group of objects or applications for which you want to create a schedule. You can use both the * or the % wildcards.</td>
</tr>
<tr>
<td><strong>Partition</strong></td>
<td>Enter a partition number or specify 0 for all partitions.</td>
</tr>
<tr>
<td><strong>Comment</strong></td>
<td><em>(optional)</em> Enter a comment describing the schedule.</td>
</tr>
<tr>
<td><strong>Last update</strong></td>
<td>This read-only field <em>(displayed only for an edited schedule)</em> shows the user ID and timestamp of the last update to this schedule.</td>
</tr>
</tbody>
</table>

5 In the Schedule tab, complete the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Utility ID</strong></td>
<td>Leave blank to include all Utility IDs in this schedule or enter a Utility ID to limit this schedule to a specific utility. For more information, see “Creating a Utility Status view” on page 115.</td>
</tr>
<tr>
<td><strong>Apply to NGT Reorg</strong></td>
<td>Select one or more utilities that you want to schedule.</td>
</tr>
<tr>
<td><strong>Apply to NGT Copy</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Apply to NGT Stats</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Include or Exclude</strong></td>
<td>Indicate whether to include or exclude the group of objects from processing according to the schedule.</td>
</tr>
<tr>
<td><strong>Monthly</strong></td>
<td>Select when to perform the processing:</td>
</tr>
<tr>
<td><strong>Weekly</strong></td>
<td>■ On a specific day of each month, or a specific day of a specific month</td>
</tr>
<tr>
<td></td>
<td>■ Every day of the month</td>
</tr>
<tr>
<td></td>
<td>■ On a specific day each week, or every day of the week</td>
</tr>
<tr>
<td><strong>Start time</strong></td>
<td>Select a start time for the process.</td>
</tr>
</tbody>
</table>

**Note:** The product does not support starting a process before midnight and completing it the following day. If you want to do this, create two schedules (one ending at 23:59:59 and one beginning at 00:00:00).
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>End time</td>
<td>Select an end time for the process.</td>
</tr>
</tbody>
</table>

6 Click **OK**.

**To edit a schedule**

1 In the Product Tools perspective, select a NGT Utility Management view.

2 In the Schedules table, select a schedule, then click **Edit schedule**.

3 In the Edit Utility Manager Schedule dialog box, edit the General settings.
   
   The schedule that you are editing is displayed at the top of the dialog box, click the arrow to see and select another schedule. You can also use the Previous schedule and Next schedule arrows to navigate to another schedule.

4 Select the **Schedule** tab.

5 Edit the schedule that you want changed.

6 Click **OK**.

---

**Related Information**

- “NGT SQL language elements” on page 128
- “Features and functions of the NGT SQL language” on page 131
- “Special registers that the NGT SQL language supports” on page 134
- “Constants that NGT SQL supports” on page 136
- “Functions that NGT SQL supports” on page 139
- “Date and time durations” on page 177
- “Comparisons” on page 179
- “NGT SQL language keywords” on page 180
NGT SQL language

NGT utility products provide a SQL-like language known as the NGT SQL language (NGT SQL). The language provides constructs and built-in functions similar to the DB2 SQL language.

Note

For column names that you can include in the WHERE clause, see BMC Next Generation Technology Utility Manager for DB2 for z/OS Reference Manual.

NGT SQL language elements

The basic elements of the NGT SQL language are expressions and predicates. When an expression is evaluated, it normally returns a value of a known data type defined in the language. You can use expressions to form predicates, or to manipulate data elements. When a predicate is evaluated, it normally results in a true/false condition. You can use predicates to define search conditions.

An expression in its basic form can be one of the following items:

- Function
- Case expression
- Constant
- Column name
- Special register
- Labeled duration

An expression can also include other expressions. The following diagram shows the syntax of an expression.

Figure 3: NGT SQL syntax for expressions
Figure 4: NGT SQL syntax for expressions (labeled-duration syntax)

function
  (expression)
  constant
  columnName
  YEAR
  YEARS
  MONTH
  MONTHS
  DAY
  DAYS
  HOUR
  HOURS
  MINUTE
  MINUTES
  SECOND
  SECONDS
  MICROSECOND
  MICROSECONDS

Figure 5: NGT SQL syntax for expressions (CASE statement)

CASE searched WHEN clause
  simple WHEN clause
  ELSE NULL
  ELSE resultExpression
  END

Figure 6: searched WHEN clause

WHEN searchCondition
  resultExpression
  NULL
Figure 8: NGT SQL syntax for predicates (continued)

![NGT SQL syntax for predicates](image)

Features and functions of the NGT SQL language

The following table lists the features and functions that NGT SQL supports.

Table 9: Features and functions of the NGT SQL language

<table>
<thead>
<tr>
<th>Feature/Function</th>
<th>Supported by NGT SQL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WHERE clause elements</strong></td>
<td></td>
</tr>
<tr>
<td>Column-to-column compares</td>
<td>Yes</td>
</tr>
<tr>
<td>Constant-to-Constant compares</td>
<td>No</td>
</tr>
<tr>
<td>Constant-to-Column compares</td>
<td>Yes</td>
</tr>
<tr>
<td>Host variables</td>
<td>No</td>
</tr>
<tr>
<td><strong>WHERE clause keywords</strong></td>
<td></td>
</tr>
<tr>
<td>BETWEEN, IN, LIKE keywords</td>
<td>Yes</td>
</tr>
<tr>
<td>NOT BETWEEN, NOT IN, NOT LIKE keywords</td>
<td>Yes</td>
</tr>
<tr>
<td>IS NULL, IS NOT NULL keywords</td>
<td>Yes</td>
</tr>
<tr>
<td>CONCAT keyword</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Data manipulation</strong></td>
<td></td>
</tr>
<tr>
<td>Concatenation</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>CONSTANT data formats</strong></td>
<td></td>
</tr>
<tr>
<td>CHAR (character string), HEX (hexadecimal string), GRAPHIC (graphic string)</td>
<td>Yes</td>
</tr>
<tr>
<td>INTEGER, DECIMAL, FLOAT (floating point), DECFLOAT (decimal floating point)</td>
<td>Yes</td>
</tr>
<tr>
<td>Feature/Function</td>
<td>Supported by NGT SQL</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>Mixed data</td>
<td>Yes</td>
</tr>
<tr>
<td>Date formats: EUR/ISO/JIS/USA</td>
<td>Yes</td>
</tr>
<tr>
<td>Time formats: EUR/ISO/JIS/USA</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Date/Time elements</strong></td>
<td></td>
</tr>
<tr>
<td>Date/time durations</td>
<td>Yes</td>
</tr>
<tr>
<td>Day of week</td>
<td>Yes</td>
</tr>
<tr>
<td>Day of month</td>
<td>Yes</td>
</tr>
<tr>
<td>Day of year</td>
<td>Yes</td>
</tr>
<tr>
<td>Timestamp duration</td>
<td>Yes</td>
</tr>
<tr>
<td>Labeled durations</td>
<td>Yes</td>
</tr>
<tr>
<td>Week</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Arithmetic</strong></td>
<td></td>
</tr>
<tr>
<td>Date/time arithmetic</td>
<td>Yes</td>
</tr>
<tr>
<td>Decimal arithmetic</td>
<td>Yes</td>
</tr>
<tr>
<td>Float/decimal float arithmetic</td>
<td>Yes</td>
</tr>
<tr>
<td>Integer arithmetic</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Functions</strong></td>
<td></td>
</tr>
<tr>
<td><code>DATE()</code>, <code>TIME()</code>, <code>TIMESTAMP()</code></td>
<td>Yes</td>
</tr>
<tr>
<td><code>YEAR()</code>, <code>MONTH()</code>, <code>SECOND()</code>, <code>MICROSECOND()</code></td>
<td>Yes</td>
</tr>
<tr>
<td><code>DAYOFWEEK()</code>, <code>DAYOFMONTH()</code>, <code>DAYOFYEAR()</code></td>
<td>Yes</td>
</tr>
<tr>
<td><code>INTEGER()</code>, <code>DECIMAL()</code>, <code>FLOAT()</code>, <code>REAL()</code>, <code>SMALLINT()</code></td>
<td>Yes</td>
</tr>
<tr>
<td><code>BIGINT()</code>, <code>DECFLOAT()</code></td>
<td>Yes</td>
</tr>
<tr>
<td><code>VARCHAR()</code></td>
<td>Yes</td>
</tr>
<tr>
<td><code>VALUE()</code></td>
<td>Yes</td>
</tr>
<tr>
<td><code>LENGTH()</code></td>
<td>Yes</td>
</tr>
<tr>
<td><code>SOUNDEX()</code></td>
<td>Yes</td>
</tr>
<tr>
<td><code>SUBSTR()</code>, <code>STRIP()</code></td>
<td>Yes</td>
</tr>
<tr>
<td><code>HEX()</code>, <code>CHAR()</code></td>
<td>Yes</td>
</tr>
<tr>
<td><em>(CHAR(numeric argument) is not supported for Index on Expression)</em></td>
<td></td>
</tr>
<tr>
<td><code>UPPER()</code>, <code>LOWER()</code></td>
<td>Yes</td>
</tr>
<tr>
<td><code>UCASE()</code>, <code>LCASE()</code></td>
<td>Yes</td>
</tr>
<tr>
<td>Feature/Function</td>
<td>Supported by NGT SQL</td>
</tr>
<tr>
<td>-------------------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>AVG(), MIN(), MAX(), SUM(), COUNT()</td>
<td>No</td>
</tr>
<tr>
<td>DIGITS()</td>
<td>Yes</td>
</tr>
<tr>
<td>VARGRAPHIC()</td>
<td>No</td>
</tr>
<tr>
<td>REPEAT()</td>
<td>Yes</td>
</tr>
<tr>
<td>LOCATE()</td>
<td>Yes</td>
</tr>
<tr>
<td>TRANSLATE()</td>
<td>Yes</td>
</tr>
<tr>
<td><em>(not supported for Index on Expression)</em></td>
<td></td>
</tr>
<tr>
<td>IFNULL()</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Operators</strong></td>
<td></td>
</tr>
<tr>
<td>AND, OR</td>
<td>Yes</td>
</tr>
<tr>
<td>NOT</td>
<td>No</td>
</tr>
<tr>
<td>+, -, *, /</td>
<td>Yes</td>
</tr>
<tr>
<td>Remainder divide (/)</td>
<td>Yes</td>
</tr>
<tr>
<td>Concatenation (</td>
<td></td>
</tr>
<tr>
<td><strong>Special registers</strong></td>
<td></td>
</tr>
<tr>
<td>CURRENT DATE</td>
<td>Yes</td>
</tr>
<tr>
<td>CURRENT TIME</td>
<td>Yes</td>
</tr>
<tr>
<td>CURRENT TIMESTAMP WITHOUT TIMEZONE</td>
<td>Yes</td>
</tr>
<tr>
<td>CURRENT TIMEZONE, CURRENT TIMESTAMP WITH TIMEZONE</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Other clauses</strong></td>
<td></td>
</tr>
<tr>
<td>GROUP BY, HAVING</td>
<td>No</td>
</tr>
<tr>
<td>JOINS, UNIONS</td>
<td>No</td>
</tr>
<tr>
<td>Correlated queries</td>
<td>No</td>
</tr>
<tr>
<td><strong>Miscellaneous</strong></td>
<td></td>
</tr>
<tr>
<td>Aliases</td>
<td>Yes</td>
</tr>
<tr>
<td>Data conversion for numeric comparisons</td>
<td>Yes</td>
</tr>
<tr>
<td>Decimal point as comma</td>
<td>Yes</td>
</tr>
<tr>
<td>Expressions</td>
<td>Yes</td>
</tr>
<tr>
<td>Parenthesized expressions</td>
<td>Yes</td>
</tr>
<tr>
<td>Delimited column names</td>
<td>Yes</td>
</tr>
<tr>
<td>Qualified column names</td>
<td>No</td>
</tr>
</tbody>
</table>
### Special registers that the NGT SQL language supports

This topic lists and describes the special registers that NGT SQL language supports.

#### CURRENT DATE

The CURRENT DATE special register specifies a date that is based on a reading of the extended time-of-day (TOD) clock when the SQL statement is executed for a particular row of a table. The TOD clock value is then converted to local time, using the current time zone defined to the IBM z/OS system.

If, *in a single statement*, you use CURRENT DATE more than once or use it with CURRENT TIME or CURRENT TIMESTAMP, all corresponding values are based on a single clock reading.

Specifying CURRENT_DATE is equivalent to specifying CURRENT DATE.

#### CURRENT TIME

The CURRENT TIME special register specifies a time that is based on a reading of the extended time-of-day (TOD) clock when the SQL statement is executed for a particular row of a table. The TOD clock value is then converted to local time, using the current time zone defined to the IBM z/OS system.

If, *in a single statement*, you use CURRENT DATE more than once or use it with CURRENT TIME or CURRENT TIMESTAMP, all corresponding values are based on a single clock reading.

Specifying CURRENT_TIME is equivalent to specifying CURRENT TIME.
**CURRENT_TIMESTAMP(n)**

The CURRENT_TIMESTAMP special register specifies a timestamp that is based on a reading of the extended time-of-day (TOD) clock when the SQL statement is executed for a particular row of a table. The TOD clock value is then converted to local time, using the current time defined to the IBM z/OS system.

Specifying CURRENT_TIMESTAMP is equivalent to specifying CURRENT_TIMESTAMP.

Specifying (n) generates a timestamp with n precision. Valid values for n are 0 through 12. The default is 6.

**CURRENT TIMEZONE**

The CURRENT TIME ZONE special register specifies a value that contains the difference between UTC and local time. The data type is DECIMAL(6,0).

The difference between UTC and local time is represented by a time duration.

Subtracting CURRENT TIME ZONE from a local time converts the local time to UTC.

**CURRENT_TIMESTAMP(n) WITH TIME ZONE**

The CURRENT_TIMESTAMP WITH TIME ZONE special register specifies a timestamp that is based on a reading of the extended time-of-day (TOD) clock when the SQL statement is executed for a particular row of a table. The TOD clock value is then converted to local time using the implicit time zone defined in DSNHDECP; if DSNHDECP does not include a time-zone definition, the utility uses the current time zone defined to the IBM z/OS system.

Specifying (n) generates a timestamp with n precision. Valid values for n are 0 through 12. The default is 6.

---

**Note**

CURRENT_TIMESTAMP might be referenced in a timestamp with time zone context (for example, when compared with a timestamp with a time zone column). In that case, the utility bases the implicit time zone for CURRENT_TIMESTAMP on the implicit time zone system parameter, which could be a different value from the CURRENT TIME ZONE special register.
Constants that NGT SQL supports

This topic lists and describes the constants that NGT SQL language supports.

**Note**

All constants have the attribute NOT NULL.

**Integer constants**

An integer constant specifies an integer as a signed number with a maximum of 19 digits and without a decimal point.

The data type of an integer constant is:

- A large integer if the value is within the range of a large integer
- A big integer if the value is outside the range of a large integer, but within the range of a big integer
- A long decfloat constant if the value is defined outside the range of big integer values

**Floating-point constants**

A floating-point constant specifies a double-precision floating-point number as two numbers separated by an E:

- The first number can include a sign and a decimal point.
- The second number can include a sign but not a decimal point.

The constant's value is the product of the first number and the power of 10 specified by the second number. If the value is outside the range of floating-point numbers, the utility converts the value to a long decimal-floating point constant, as explained in Decimal floating-point constants on page 137.

**Decimal constants**

A decimal constant is a signed or unsigned number of no more than 31 digits that includes a decimal point.

The precision is the total number of digits, including those (if any) to the right of the decimal point. The total includes all leading and trailing zeros. The scale is the number of digits to the right of the decimal point, including trailing zeros.
Decimal floating-point constants

A decimal floating-point constant specifies a decimal floating-point number as two numbers separated by an E:

■ The first number can include a sign and a decimal point.
■ The second number can include a sign but not a decimal point.

The value of the constant is the product of the first number and the power of 10 specified by the second number. The value must be within the range of \text{DECFLOAT}(34).

Basically, a constant is a decimal floating-point constant only if one of the following conditions exists:

■ The value is specified as two numbers separated by E, and the value is outside the range of a floating-point constant.
■ The value is specified as a number that does not contain E, and the number has more than 31 digits.

In addition to numeric constants, you can use the following special values to specify decimal floating-point special values:

■ INF or INFINITY—represents infinity
■ NAN—represents quiet not-a-number
■ SNAN—represents signaling not-a-number

You can precede the special values by an operational sign (+ or -).

When you use special values in a predicate, the following order of precedence applies:

\[-\text{NAN} < -\text{SNAN} < -\text{INFINITY} < -0 < 0 < \text{INFINITY} < \text{SNAN} < \text{NAN}\]

Character string constants

A character string constant specifies a fixed-length character string that does not exceed 255 characters.

The string is a sequence of characters enclosed by a pair of apostrophes ('). The string length does not include the delimiters. If the string itself contains a string delimiter, then it should be duplicated. Blanks within the string are preserved.
Hexadecimal constant

A hexadecimal constant is a string representation of hexadecimal digits in the form of X'HH…HH'.

The number of hexadecimal digits must not exceed 510. A hexadecimal digit can be a numeric digit, any letter from A through F, or a combination. Each pair of hexadecimal digits represents a character.

Graphic string constant

A graphic string constant specifies a fixed-length graphic string that does not exceed 128 characters. The string is in the following format:

G'<gg…gg>'

gg is a graphic character (double byte) enclosed by a shift-in '<' and a shift out '>' characters.

DATE string constant

A DATE string constant contains a value that conforms to one of the valid formats for string representations of dates, subject to the following rules:

- Leading blanks are not allowed.
- You can omit leading zeros from the month and day elements of the date. An implicit specification of 0 is assumed for any omitted digit.
- You must include leading zeros for the year element of the date.
- You can include trailing blanks.

TIME string constant

A TIME string constant must contain a value that conforms to one of the valid formats for string representations of times, subject to the following rules:

- Leading blanks are not allowed.
- You can omit the seconds element of the time.
- You can include trailing blanks.
- If the USA format is not used and the minutes and seconds are all zeros, the hour can be 24. If the format is USA, mm may be omitted but hh must be specified.
TIMESTAMP string constant or TIMESTAMP WITH TIMEZONE constant

A string representation of a timestamp is a character string whose value conforms to a date string constant concatenated with a time string constant and separated by a hyphen (-).

A TIMESTAMP string constant must contain a value that conforms to a date string constant concatenated with a time string constant, and separated by a hyphen. It is subject to the following rules:

- The string is at least 16 characters.
- Imbedded blanks are not allowed.
- Trailing blanks are allowed.
- Optionally the constant can be concatenated with a maximum of 12 digits, representing fractions of a second, called precision.

Also, you can optionally append a time zone to the string according to the following syntax:

- +HH:MM
- -HH:MM

The variables are defined as follows:

- You can specify HH as a single H.
- You can specify MM as a single M. If you include MM or M, it must follow the colon; if you omit MM or M, omit the colon.

Example

A timestamp string of precision 6 with a time zone:
2010-12-31-12.00.50.123456-6:00

Functions that NGT SQL supports

This topic lists and describes the functions that NGT SQL language supports.

BIGINT function

The BIGINT function returns a very large integer representation of a number.
The numericExpression is converted to a big integer data type from one of the following numerical data types:
- SMALLINT
- INTEGER
- BIGINT
- FLOAT
- REAL
- DECIMAL
- DECFLOAT

The numerical expression is not rounded before conversion and any fractional part is truncated. If the result exceeds the allowable values for a BIGINT, an overflow condition occurs. The valid range is -9223372036854775808 through 9223372036854775807.

The stringExpression can be in any external numerical format. It is first converted to its corresponding format, and then converted to BIGINT. If the CCSID of the string is not in the EBCDIC encoding scheme, the CCSID is cast to EBCDIC CCSID(37) before conversion takes place.

Note
If a constant, then the argument must adhere to integer value rules. Specifying BIGINT(1.5) causes an error since 1.5 is not an integer.
If the argument can be null, the result can be null; if the argument's value is null, the result is the null value.

CHAR function

The CHAR function returns a fixed-length character string representation of one of the following values:
- An integer number if the first argument is a small, large, or big integer
- A decimal number if the first argument is a decimal number
- A floating-point number if the first argument is a single or double precision floating-point number
A decimal floating-point number if the first argument is a decimal floating-point number

A character string value if the first argument is any type of string

A date/time value if the first argument is a date, time, timestamp, or timestamp with time zone

The arguments are defined as follows:

**integerExpression**

*integerExpression* returns a value that is a built-in integer data type (SMALLINT, INTEGER, or BIGINT).

The result is the fixed-length character string representation of the argument in the form of an integer constant. The result consists of *n* characters that are the significant digits that represent the value of the argument. If the argument is negative, the result has a preceding minus sign.

The result is left justified. The result length must be 6 for a small integer, 11 for a large integer, and 20 for a big integer. Shorter results are padded on the right with blanks to achieve the required length.

A positive value always includes one leading blank.

**decimalExpression**

*decimalExpression* returns a value that is a built-in decimal data type. The result is the fixed-length character string representation of the argument. The result includes a decimal point and up to *p* digits; *p* is the precision of the *decimalExpression* with the preceding minus sign if the argument is negative. If the number of bytes in the result is less than the defined length of the result, the result is padded on the right with blanks. The decimal point character is a period (.) or a comma (,) based on DSNHDECP specifications.
floatingExpression

floatingExpression returns a value that is a built-in floating-point data type (DOUBLE or REAL).

The result is the fixed-length character string representation of the argument in the form of a floating-point constant. The length of the result is 15 bytes for a short floating point and 23 for a long floating point in the following format:

+/-.mE+/-.nn

/ denotes an OR condition, \( m \) represents mantissa, and \( n \) represents an exponent.

decfloatExpression

decfloatExpression returns a value that is a built-in decimal floating data type (long or short).

The result is the fixed-length character string representation of the argument in the form of a floating-point constant. The length of the result is 34 bytes for a short decfloat and 42 for a long decfloat in the form of: +/-mE+/−nnn and 42 bytes for a long floating point number in the following form:

+/-.mE+/−nnnn

/ denotes an OR condition, \( m \) represents mantissa, and \( n \) represents an exponent.

If the DECFLOAT value is one of the special values INFINITY, SNAN, or NAN, the strings INFINITY, SNAN, or NAN, respectively, are returned. If the special value is negative, a minus sign is the first character in the returned string.

stringExpression, length

stringExpression must be an expression that returns a value of a built-in character string whose length equals the value of the length argument.

The length value must be an integer constant not exceeding 255.

datetimeExpression, format
datetimeExpression, string

datetimeExpression must be an expression that is one of the following built-in data types:

- DATE
- TIME
- TIMESTAMP
- **TIMESTAMP WITH TIMEZONE**

  The expression is converted to an external format using the format template, if applicable.

  The format must be defined as follows:

<table>
<thead>
<tr>
<th>Expression type</th>
<th>Format type</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>ISO, USA, EUR, JIS, LOCAL</td>
</tr>
<tr>
<td></td>
<td>If the format is not specified, it defaults to ISO. Additional non-conventional formats can be specified in the form of a string of date format elements. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>- MM for month</td>
</tr>
<tr>
<td></td>
<td>- DD for day</td>
</tr>
<tr>
<td></td>
<td>- YY or YYYY for year</td>
</tr>
<tr>
<td></td>
<td>You can specify these elements in any order and can delimit them with any character, number, or series of characters or numbers. Although the process does not require a delimiter, it treats any other characters as constants in their specified position.</td>
</tr>
<tr>
<td>TIME</td>
<td>ISO, USA, EUR, JIS, LOCAL</td>
</tr>
<tr>
<td></td>
<td>If the format is not specified, it defaults to ISO. Additional non-conventional formats can be specified in the form of a string of time format elements. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>- HH for hour</td>
</tr>
<tr>
<td></td>
<td>- MM for minute</td>
</tr>
<tr>
<td></td>
<td>- SS for second</td>
</tr>
<tr>
<td></td>
<td>- XM for AM or PM</td>
</tr>
<tr>
<td></td>
<td>You can specify these elements in any order and can delimit them with any character, number, or series of characters or numbers. Although the process does not require a delimiter, it treats any other characters as constants in their specified position.</td>
</tr>
<tr>
<td>Expression type</td>
<td>Format type</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TIMESTAMP WITHOUT TIMEZONE</td>
<td>The result is the character string representation of the timestamp. If <code>datetimeExpression</code> is a TIMESTAMP(0) value, the length of the result is 19. If <code>datetimeExpression</code> is a TIMESTAMP (integer) value, the length of the result is 20+ integer. Otherwise, the length of the result is 26. The second argument of string may be specified with <code>timestamp(6)</code> as a formatting template that has date/time/microseconds elements. Valid values are:</td>
</tr>
</tbody>
</table>
|                                     |  ■ MM for month  
|                                     |  ■ DD for day  
|                                     |  ■ YY or YYYY for year  
|                                     |  ■ HH for hour  
|                                     |  ■ MM for minute  
|                                     |  ■ SS for second  
|                                     |  ■ XM for AM or PM  
|                                     |  ■ NN for hundredths of a second  
|                                     |  ■ NNNNN for milli-second  
|                                     |  ■ NNNNNNN for a microsecond.                                                                                                                                                                           |
| TIMEZONE WITH TIMEZONE              | The result is the character string representation of the timestamp with time zone, formatted as `yyyyMMdd.hh:mm:ss.nnnnn±th:tm` with the appropriate number of `n` characters for the precision of the timestamp. If `datetimeExpression` is a TIMESTAMP (0) with timezone, the length of the result is 25. If `datetimeExpression` is a TIMESTAMP (integer) with timezone, the length of the result is 26+ integer. The second argument must not be specified. |

**Note**

If the argument can be null, the result can be null; if the argument’s value is null, the result is the null value.

**DATE function**

The DATE function returns an internal representation of one of the following date data types:

- DATE
- TIMESTAMP
- TIMESTAMP WITH TIMEZONE
- CHAR
- INTEGER
stringExpression

stringExpression must be a valid string representation of a date or timestamp with or without a timezone. If the encoding of the character string is not EBCDIC, the character string is cast to an EBCDIC CCSID(37) before conversion takes place.

integerExpression

integerExpression must be a positive integer not exceeding 3652059. This value represents the number of days since Jan 1,0001 plus 1.

dateExpression

The result is the date returned by the expression.

timestampExpression

The result is the date part of the timestamp.

In the case of a timestamp with a time zone, the expression is cast to TIMESTAMP without timezone, and then the date portion is extracted.

Example

Date('1991-12-31')

Date(COL01+100 days) Where COL01 is either a date or timestamp with/out time zone

Date('1991'||'-12-'||SUBSTR(COL02,1,2)) Where COL02 is a var/character field having a valid day number.

Date(22) returns 0001-01-22

Note

If the argument can be null, the result can be null; if the argument's value is null, the result is the null value.

DAY function

The DAY function returns an integer data type.
The expression must return one of the following built-in data types:

- DATE
- TIMESTAMP
- TIMESTAMP WITH TIMEZONE
- Date duration
- Timestamp duration

**Note**

A CHAR/VARCHAR argument is not accepted unless converted by a DATE, TIMESTAMP, or TIMESTAMP_TZ function.

A date duration occurs as the result of subtracting two dates; a timestamp duration occurs as the result of subtracting two timestamps with precision=6. The result is the day portion of the derived date or date/timestamp duration.

If the argument can be null, the result can be null; if the argument’s value is null, the result is the null value.

**Example**

```
DAY (DATE ('1991-12-31'))
```

Will result in an integer of value = 31

It is invalid to specify `DAY ( DATE ('1991-12-31') )` since the argument is of type CHAR.

**Example**

```
DAY (DATE ('1991-12-31')-DATE ('1995-12-31'))
```

Will result in an integer of value = 0

**DAYS function**

The DAYS function returns an integer data type.

The expression must return one of the following built-in data types:

- DATE
- TIMESTAMP
- TIMESTAMP WITH TIMEZONE
- CHAR
The result is 1 more than the number of days from January 1, 0001 to the date represented by the argument after conversion to a date data type.

Note

If the argument can be null, the result can be null; if the argument's value is null, the result is the null value.

**DAYOFMONTH function**

The DAYOFMONTH function returns the day part of its argument. The function is similar to the DAY function, except DAYOFMONTH does not support a date or timestamp duration as an argument.

```
DAYOFMONTH( expression )
```

The expression must return one of the following built-in data types:

- DATE
- TIMESTAMP
- TIMESTAMP WITH TIMEZONE
- Valid string representation of a date or timestamp with or without a time zone

Note

The expression cannot be specified as a constant.

If the argument can be null, the result can be null; if the argument's value is null, the result is the null value.

**Example**

```
UNLOAD FORMAT UNLOAD
SELECT CUSTOMER_NUM, DAYOFMONTH(BILLING_DATE)
FROM NWREGION.ACCTS_RECEIVABLE
REORG TABLESPACE NWREGION.ARTABLES
DISCARD FROM TABLE NWREGION.ACCTS_RECEIVABLE
WHERE (DAYOFMONTH(BILLING_DATE) = 15)
```

Note

Parentheses are not required when specifying a WHERE clause condition.

**DAYOFWEEK function**

The DAYOFWEEK function returns an integer in a range of 1 through 7 that represents the day of the week, where 1 is Sunday and 7 is Saturday.

```
DAYOFWEEK( expression )
```
The expression must return one of the following built-in data types:

- DATE
- TIMESTAMP
- TIMESTAMP WITH TIMEZONE
- Valid string representation of a date or timestamp with or without a time zone

**Note**
The expression cannot be specified as a constant.
If the argument can be null, the result can be null; if the argument's value is null, the result is the null value.

**Example**

```sqlc
UNLOAD FORMAT CSV
SELECT EMPLOYEE_NUM, SALARY, DAYOFWEEK(PAYDATE)
FROM SWREGION.ACCTS_PAYABLE
REORG TABLESPACE SWREGION.APTABLES
DISCARD FROM TABLE SWREGION.ACCTS_PAYABLE
WHERE (DAYOFWEEK(PAYDATE) = 3)
```

**Note**
Parentheses are not required when specifying a WHERE clause condition.

**DAYOFYEAR function**

The DAYOFYEAR function returns a large integer in a range of 1 to 366 that represents the day of the year, where 1 is January 1.

```sql
DAYOFYEAR(expression)
```

The expression must return one of the following built-in data types:

- DATE
- TIMESTAMP
- TIMESTAMP WITH TIMEZONE
- Valid string representation of a date or timestamp with or without a time zone

**Note**
The expression cannot be specified as a constant.
If the argument can be null, the result can be null; if the argument's value is null, the result is the null value.
Example

```
UNLOAD FORMAT UNLOAD
SELECT MEMBERSHIP_NUM, DAYOFYEAR(ANNIVERSARY_DATE)
FROM REGION49.MEMBERSHIP_RECS
REORG TABLESPACE REGION49.MEMBERS7
DISCARD FROM TABLE REGION49.MEMBERSHIP_RECS
WHERE (DAYOFYEAR(ANNIVERSARY_DATE) > 360)
```

Note

Parentheses are not required when specifying a WHERE clause condition.

DECFLOAT function

The DECFLOAT function returns a DECFLOAT representation of a number using the precision specified in the second argument.

```
DECFLOAT( numericExpression ,precision )
```

Note

If the argument can be null, the result can be null; if the argument’s value is null, the result is the null value.

The arguments are defined as follows:

numericExpression

```
numericExpression
```

numericExpression is converted to a DECFLOAT from one of the following numerical data types:

- SMALLINT
- INTEGER
- BIGINT
- FLOAT
- REAL
- DECIMAL
- DECFLOAT

stringExpression

If the argument is a string expression, it can be in any external numerical format. It is converted directly to DECFLOAT with the specified precision. If the CCSID of the string is not in EBCDIC encoding scheme, it is cast to EBCDIC CCSID(37) before conversion takes place.
Other special strings are accepted and converted to their internal DECFLOAT values. These include:

- NAN
- SNAN
- INF
- INFINITY

**precision**

The precision can be either short (16) or long (34).

### DECIMAL or DEC function

The DECIMAL function returns a decimal representation of a number.

```sql
{DECIMAL|DEC}( numericExpression, precision, scale )
```

The arguments are defined as follows:

**numericExpression**

- `numericExpression` is converted to a decimal data type with the specified precision and scale. The data type of the numeric expression can be:
  - SMALLINT
  - INTEGER
  - BIGINT
  - FLOAT
  - REAL
  - DECIMAL
  - DECFLOAT

**stringExpression**

If the argument is a string expression, it can be in any external numerical format. It is converted directly to decimal data type. If the string expression has a decimal point, the number is converted to the implied precision and scale, then rescaled to the specified precision and scale. No rounding takes place. Truncation of the fractional part may take place. A decimal point conversion overflow will be declared if a loss of precision is detected.

If the CCSID of the string is not in EBCDIC encoding scheme, it is cast to EBCDIC CCSID (37) before conversion takes place.
Other special strings are accepted and converted to their internal DECFLOAT values. These include:

- NAN
- SNAN
- INF
- INFINITY

### precision

The precision argument must be an integer constant with a value in the range of 1 through 31. The value of this second argument specifies the precision of the result.

The default value depends on the data type of the first argument, as follows:

<table>
<thead>
<tr>
<th>Data type</th>
<th>Default precision value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMALLINT</td>
<td>5</td>
</tr>
<tr>
<td>INTEGER</td>
<td>11</td>
</tr>
<tr>
<td>BIGINT</td>
<td>19</td>
</tr>
<tr>
<td>DECFLOAT</td>
<td>31</td>
</tr>
<tr>
<td>All other cases</td>
<td>15</td>
</tr>
</tbody>
</table>

### scale

The scale argument must be an integer constant that is greater than or equal to zero and less than or equal to `precision`. The value specifies the scale of the result. The default value is 0.

#### Note

If the argument can be null, the result can be null; if the argument's value is null, the result is the null value.

### DIGITS function

The DIGITS function returns a character string representation of the absolute value of a number.

```
DIGITS( numericExpression )
```

`numericExpression` must return a value that is a SMALLINT, BIGINT, INTEGER, or DECIMAL built-in numeric data type.
Table 10: Output lengths for the BIGINT function

<table>
<thead>
<tr>
<th>Data type</th>
<th>Output length</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMALLINT</td>
<td>5</td>
</tr>
<tr>
<td>BIGINT</td>
<td>19</td>
</tr>
<tr>
<td>INTEGER</td>
<td>10</td>
</tr>
<tr>
<td>DECIMAL</td>
<td>(p) if the argument is a decimal with a precision of (p)</td>
</tr>
</tbody>
</table>

**Note**
If the argument can be null, the result can be null; if the argument's value is null, the result is the null value.

**FLOAT function**

The FLOAT function returns a floating-point representation of a number.

\[
\text{FLOAT(} \quad \text{numericExpression} \quad ,\text{precision} \quad \text{stringExpression} \quad \text{)}
\]

The arguments are defined as follows:

**numericExpression**

The numeric expression is converted to a floating point number of length (8) from one of the following numerical data types:
- SMALLINT
- INTEGER
- BIGINT
- FLOAT
- REAL
- DECIMAL
- DECFLOAT
- @ROWID

Decimal precision might be lost because a floating-point number is not an exact representation of a decimal field.

**stringExpression**

If the argument is a string expression, it can be in any external numerical format. It is first converted to its corresponding format, and then converted
to a float. If the CCSID of the string is not in EBCDIC encoding scheme, the
CCSID is cast to EBCDIC CCSID(37) before conversion takes place.

Float is capable of converting the value returned by the internal @ROWID
function into a floating-point value between 0 and 1. This value can be used
to generate random numbers.

**precision**

The precision can be either short (16) or long (34).

**Note**

If the argument can be null, the result can be null; if the argument’s value is null, the
result is the null value.

**HEX function**

The HEX function returns a hexadecimal representation of a value as a fixed or
variable character string.

**expression** must be an expression that returns a value of any of the following data
types:

- SMALLINT
- INTEGER
- BIGINT
- FLOAT
- REAL
- DECIMAL
- DECIMAL
- DECIMAL
- DECIMAL
- DATE
- TIME
- TIMESTAMPT
- TIMESTAMPT WITH TIME ZONE
- CHAR
- VARCHAR
- GRAPHIC
- VARGRAPHIC
**Note**
A character string must not have a maximum length greater than 4000. A graphic string must not have a maximum length greater than 2000.
The HEX function is currently restricted to EBCDIC tables.
If the argument can be null, the result can be null; if argument's value is null, the result is the null value.

The result is a string of hexadecimal digits. The first two represent the first byte of the argument, the next two represent the second byte of the argument, and so on. If the argument is a datetime value, the result is the hexadecimal representation of the internal form of the argument. If the argument is a numeric value, the result is the hexadecimal representation of the decoded form of the argument.

If the argument is a fixed-length string and the length of the result is less than 255, the result is a fixed-length string. Otherwise, the result is a variable-length string with a length attribute that depends on the following rules:

- If the argument is not a variable-length string, the length attribute of the result string is the same as the length of the result.
- If the argument is a variable character or binary string, the length attribute of the result string is twice the length attribute of the argument.
- If the argument is a variable graphic string, the length attribute of the result string is four times the length attribute of the argument.

**HOUR function**

The HOUR function returns an integer data type.

```
HOUR( expression )
```

The *expression* must return one of the following built-in data types:

- TIME
- TIMESTAMP
- TIMESTAMP WITH TIME ZONE
- TIME DURATION
- TIMESTAMP DURATION
**Note**

A time duration occurs as the result of subtracting two times; a timestamp duration occurs as the result of subtracting two timestamps of precision=6. The result is the hour portion of the derived time or time/timestamp duration.

A CHAR/VARCHAR argument is not accepted unless converted by a TIME, TIMESTAMP, or TIMESTAMP_TZ function.

**Example**

HOUR (TIME ('13.30.05'))

Will result in an integer of value = 13

It is invalid to specify HOUR ('13.30.05') since the argument is of type CHAR.

**Example**

HOUR (TIME ('13.30.05')-TIME ('12.02.01'))

Will result in an integer of value = 1

**INTEGER or INT function**

The INTEGER function returns an integer representation of a number. The argument can be either a string expression or a numeric expression.

\[
\text{INTEGER} | \text{INT} (\text{numericExpression}, \text{stringExpression})
\]

The arguments are defined as follows:

**numericExpression**

*numericExpression* is converted to an integer from one of the following numerical data types:
- SMALLINT
- INTEGER
- BIGINT
- FLOAT
- REAL
- DECIMAL
- DECFLOAT

The numeric expression is not rounded before conversion and any fractional part is truncated. If the result exceeds the allowable values for an integer, an
overflow condition will occur. The valid range is -2147483648 through 2147483647.

---

**Note**

If the argument can be null, the result can be null; if the argument's value is null, the result is the null value.

If the argument is a constant, it must adhere to integer value rules. Specifying INT(1.5) causes an error because 1.5 is not an integer.

---

**stringExpression**

If the argument is a string expression, it can be in any external numerical format. It is first converted to its corresponding format, and then converted to an integer. If the CCSID of the string is not in EBCDIC encoding scheme, CCSID is cast to EBCDIC before conversion takes place.

---

**LENGTH function**

The LENGTH function returns the length of the argument's value as a large integer.

```sql
LENGTH(expression)
```

The result is the length of the argument. The length does not include the null indicator byte of column arguments that allow null values. The length of strings includes blanks. The length of a varying-length string is the actual length, not the maximum length.

The length of a graphic string including UTF-16 is the number of double-byte characters.

The length of all other values is the number of bytes used to represent the value:

<table>
<thead>
<tr>
<th>Data type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMALLINT</td>
<td>2</td>
</tr>
<tr>
<td>INTEGER</td>
<td>4</td>
</tr>
<tr>
<td>BIGINT</td>
<td>8</td>
</tr>
<tr>
<td>DECIMAL with precision ( p )</td>
<td>The integer part of ( \lfloor p / 2 \rfloor + 1 )</td>
</tr>
<tr>
<td>DECFLOAT (34)</td>
<td>16</td>
</tr>
<tr>
<td>DECFLOAT (16)</td>
<td>8</td>
</tr>
<tr>
<td>Single precision floating point</td>
<td>4</td>
</tr>
<tr>
<td>Double precision floating point</td>
<td>8</td>
</tr>
<tr>
<td>Strings</td>
<td>The length of the string</td>
</tr>
<tr>
<td>Data type</td>
<td>Length</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>DATE</td>
<td>4</td>
</tr>
<tr>
<td>TIME</td>
<td>3</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>10</td>
</tr>
<tr>
<td>TIMESTAMP WITH TIMEZONE</td>
<td>12</td>
</tr>
<tr>
<td>TIMESTAMP (p)</td>
<td>$7 + \left(\frac{p + 1}{2}\right)$</td>
</tr>
<tr>
<td>TIMESTAMP(p) WITH TIMEZONE</td>
<td>$9 + \left(\frac{p + 1}{2}\right)$</td>
</tr>
</tbody>
</table>

**Note**

If the argument can be null, the result can be null; if the argument’s value is null, the result is the null value.

**LOCATE function**

The LOCATE function returns a large integer indicating the position of one string, `searchString`, in another, `sourceString`. The function returns 0 if `searchString` is a zero-length string or is not found, or if `start` is greater than the length of `sourceString`. By default, the search starts at the first character of the receiving string (that is, the value of `start` is 1), and continues to the end of the string.

\[
\text{LOCATE}(\text{searchString}, \text{sourceString}, \text{start})
\]

The arguments are defined as follows:

**searchString**

`searchString` is an expression that specifies the string for which you are searching for. `searchString` must return a value that is a built-in single-byte character/varying string data type.

**sourceString**

`sourceString` must return a value that is a built-in single-byte character/varying string data type.

**start**

`start` is an expression that specifies the position within the `searchString` where the search is to start. `start` is expressed in bytes and must return an integer value that is greater than or equal to 1.
If any argument can be null, the result can be null; if any argument's value is null, the result is the null value.

**LOWER or LCASE function**

The **LOWER** function returns a string in which all characters have been converted to lowercase characters.

\[
\text{LOWER}(\text{stringExpression}, \text{localeNameString})
\]

\[
\text{LOWER}(\text{stringExpression}, \text{UNI}, \text{length})
\]

If the argument can be null, the result can be null; if argument's value is null, the result is the null value.

**LOWER or LCASE without UNI locale**

The arguments are defined as follows:

**stringExpression**

The **stringExpression** can be one of the following:

- CHAR SBCS in EBCDIC encoding scheme
- VARCHAR SBCS in EBCDIC encoding scheme

**localeNameString**

**localeNameString** is a string constant in EBCDIC that specifies a valid locale name. If you do not specify this argument, the utility treats the locale as blank.

The only accepted **localeNameString** types are:

- Blank—SBCS uppercase characters A-Z are converted to SBCS lowercase characters.
- EN_US—SBCS uppercase characters A-Z are converted to SBCS lowercase characters.
- DE_DE—SBCS Deutsch uppercase characters are converted to Deutsch SBCs lowercase characters.
**LOWER or LCASE with UNI locale**

The arguments are defined as follows:

`stringExpression`

`stringExpression` can be defined in the Unicode encoding scheme, and in one of the following data types:

- CHAR SBCS or mixed
- VARCHAR SBCS or mixed
- GRAPHIC or VARGRAPHIC

**UNI**

Currently the NGT SQL language supports only the UNI locale.

**length**

The `length` argument defines the length of the output expressed in bytes. The result string is padded when there is a varying source string.

---

**Note**

- `length` is supported only for the UNI locale.
- The value of `length` cannot exceed 255 for `stringExpression` when defined as a CHAR value, and cannot exceed 32704 for a varying string.
- The `length` argument must be a positive integer constant.
- The value of `length` must be bigger than the length of the source string. If less, the function returns an error.

---

**MICROSECOND function**

The MICROSECOND function accepts a single argument and returns an integer data type.

```
MICROSECOND( expression )
```

`expression` must return one of the following built-in data types:

- TIMESTAMP
TIMESTAMP WITH TIMEZONE

TIMESTAMP DURATION

A timestamp duration occurs as the result of subtracting two timestamps of precision=6. The result is the microsecond portion of the derived timestamp or timestamp duration.

**Note**

A CHAR/VARCHAR argument is not accepted unless converted by a TIMESTAMP or TIMESTAMP_TZ function.

If the argument can be null, the result can be null; if the argument’s value is null, the result is the null value.

**Example**

MICROSECOND (TIMESTAMP ('1991-12-31-13.30.05.123456'))

Will result in an integer of value = 123456

It is invalid to specify MICROSECOND ('1991-12-31-13.30.05.123456') since the argument is of type CHAR.

**Example**

MICROSECOND (TIMESTAMP ('1991-12-31-13.30.05.123456') - TIMESTAMP ('1991-12-31-13.30.05.123456'))

Will result in an integer of value = 0

**MINUTE function**

The MINUTE function accepts a single argument and returns an integer data type.

`MINUTE(expression)`

*expression* must return one of the following built-in data types:

- TIMESTAMP
- TIMESTAMP WITH TIMEZONE
- TIMESTAMP DURATION

A timestamp duration occurs as the result of subtracting two timestamps of precision=6. The result is the minute portion of the derived timestamp or timestamp duration.
Note
A CHAR/VARCHAR argument is not accepted unless converted by a TIMESTAMP or TIMESTAMP_TZ functions.
If the argument can be null, the result can be null; if the argument's value is null, the result is the null value.

Example
MINUTE (TIME ('13.30.05'))
Will result in an integer of value = 30
It is invalid to specify MINUTE ('13.30.05') since the argument is of type CHAR

Example
MINUTE (TIME ('13.30.05')-TIME ('12.02.01'))
Will result in an integer of value = 28

MONTH function

The MONTH function returns an integer data type.

MONTH(expression)

expression must return one of the following built-in data types:

- TIMESTAMP
- TIMESTAMP WITH TIMEZONE
- TIMESTAMP DURATION

A timestamp duration occurs as the result of subtracting two timestamps of precision=6. The result is the month portion of the derived timestamp or timestamp duration.

Note
A CHAR/VARCHAR argument is not accepted unless converted by a TIMESTAMP or TIMESTAMP_TZ functions.
If the argument can be null, the result can be null; if the argument's value is null, the result is the null value.
**Example**

MONTH (DATE ('1991-12-31'))

Will result in an integer of value = 12

It is invalid to specify MONTH ('1991-12-31') since the argument is of type CHAR.

**Example**

MONTH (DATE ('1991-12-31')-DATE ('1995-12-31'))

Will result in an integer of value = 0

---

**REAL function**

The REAL function returns a short floating point representation of a long floating-point number.

\[
\text{REAL}( \text{numericExpression} \text{, stringExpression} )
\]

The arguments are defined as follows:

* **numericExpression**

  *numericExpression* must be of type float. Other data types are not allowed but can be converted to float using the FLOAT() function.

* **stringExpression**

  *stringExpression* can be in any external numerical format. It is first converted to its corresponding format, and then converted to a short floating point. If the CCSID is not in EBCDIC encoding scheme, CCSID is cast to EBCDIC CCSID(37) before conversion takes place.

---

**REPEAT function**

The REPEAT function returns a varying length character string that is composed of an argument concatenated to itself the specified number of times. The maximum length of the result is 32000 bytes.

\[
\text{REPEAT}( \text{stringExpression}, \text{integer} )
\]
The arguments are defined as follows:

**stringExpression**

*stringExpression* is an expression that specifies the string to be repeated. The expression must return a value that is a built-in single-byte character / varying string. The argument cannot be a string constant.

**integer**

*integer* must be a positive large integer value that specifies the number of times to repeat the string.

---

**Note**

If any argument can be null, the result can be null; if any argument’s value is null, the result is the null value.

---

### SECOND function

The SECOND function accepts a single argument and returns an integer data type.

```sql
SECOND(expression)
```

The expression must return one of the following built-in data types:

- **TIME**
- **TIME DURATION**
  
  A time duration is the result of subtracting two times.
- **TIMESTAMP**
- **TIMESTAMP WITH TIMEZONE**
- **TIMESTAMP DURATION**
  
  A timestamp duration occurs as the result of subtracting two timestamps of precision=6. The result is the second portion of the derived time or time/timestamp duration.

---

**Note**

A CHAR/VARCHAR argument is not accepted unless converted by a TIME, TIMESTAMP, or TIMESTAMP_TZ function.

If the argument can be null, the result can be null; if the argument’s value is null, the result is the null value.
**Example**

SECOND (TIME ('13.30.05'))

Will result in an integer of value = 05

It is invalid to specify SECOND ('13.30.05') since the argument is of type CHAR.

---

**Example**

SECOND (TIME ('13.30.05')-TIME ('12.02.01'))

Will result in an integer of value = 4

---

**SMALLINT function**

The SMALLINT function returns a small integer data type.

```
SMALLINT(numericExpression)  
SMALLINT(stringExpression)
```

The arguments are defined as follows:

**numericExpression**

`numericExpression` must be of data type INTEGER. Other data types are not allowed but can be converted to an integer using the INT() function.

If the result of the conversion exceeds the allowable values for a SMALLINT, an overflow condition occurs. The valid range is -32768 through 32767.

---

**Note**

If the argument is a constant, then it must adhere to integer value rules. Specifying SMALLINT(1.5) causes an error because 1.5 is not an integer.

---

**stringExpression**

A string expression can be in any external numerical format. It is first converted to its corresponding format, then converted to a SMALLINT. If the CCSID of the string is not in EBCDIC encoding scheme, it is cast to EBCDIC CCSID(37) before conversion takes place.

---

**Note**

If the argument can be null, the result can be null; if the argument’s value is null, the result is the null value.
**SOUNDEX function**

The SOUNDEX function returns a four-character code that represents the sound of the single argument. The utility can compare the result to the results of the SOUNDEX function of other strings.

(expression) must return a character, or a varying character. The string is cast to Unicode SBCS CCSID 367 before generating the SOUNDEX code.

**Note**

You cannot specify expression as a constant string, but you can specify it as CHAR(expression,n).

The data type of the result is CHAR (4) in Unicode SBCS CCSID.

The SOUNDEX function is useful for finding strings for which the sound is known but the precise spelling is not. The function’s assumptions about the way letters and combinations of letters sound can help you search for words with similar sounds.

The SOUNDEX function produces the same code as the DB2 SOUNDEX function. The Special Census method generates the code.

**Example**

SELECT SOUNDEX(CHAR('ashcraft', 8)) FROM SYSIBM.SYSDUMMY1

Results in 'A226'

**STRIP function**

The STRIP function removes blanks or other characters from the end, the beginning, or both ends of a string expression. The data type of the result is a varying length character string.

(stringExpression, direction, trimCharacter)

The length attribute of the result is the same as the length attribute of the string expression. The actual length of the result is the length of the string expression minus the number of characters removed. If all of the characters are removed, the result is an empty string.

The arguments are defined as follows:
**stringExpression**

stringExpression must be of type CHAR or VARCHAR.

**direction**

direction must be one of the following values:

- B or Both (the default value)
- T or Trailing
- L or Leading

**trimCharacter**

trimCharacter is a single SBCS character (by default a blank). The utility converts the character to the CCSID of the table before the trim operations start.

**Note**

If the argument can be null, the result can be null; if argument's value is null, the result is the null value.

---

### SUBSTR function

SUBSTR function returns a portion of a string of the same data type as stringExpression.

```
SUBSTR( stringExpression, startIntegerExpression, lengthIntegerExpression )
```

The arguments are defined as follows:

**stringExpression**

stringExpression must be one of the following data types:

- CHAR
- VARCHAR
- GRAPHIC
- VARGRAPHIC
**startIntegerExpression**

`startIntegerExpression` specifies which position in `stringExpression` to use as the first character of the substring. The value of the `startIntegerExpression` must be between 1 and the length attribute of `stringExpression`. (The length attribute of a varying-length string is its maximum length.) A value of 1 indicates that the first character of the substring is the first character of `stringExpression`.

**lengthIntegerExpression**

Specifying `lengthIntegerExpression` is optional. If you include it, the utility pads the substring on the right with the necessary number of characters to satisfy the length criteria. The utility uses the pad character for the current table.

The varying attribute of the result substring depends on `lengthIntegerExpression`. If you omit `lengthIntegerExpression`, the varying attribute of the substring is the same as `stringExpression`, and the length of the result substring is the same as the source string.

If `lengthIntegerExpression` is a constant with a value of 255 or less, the resulting substring is a fixed character or graphic string of the specified length. If `lengthIntegerExpression` is not a constant the resulting substring is a varying character or graphic string with the same maximum length as `stringExpression`.

---

**Note**

If any of the arguments can be null, the result can be null; if any argument's value is null, the result is the null value.

---

**TIME function**

The TIME function returns an internal representation of a time data type.

```
TIME(expression)
```

`expression` must return one of the following built-in data types:

- **TIME**
  The result is the specified time.
- **TIMESTAMP**
  The result is the time part of the timestamp.
- **TIMESTAMP WITH TIMEZONE**
  
  *expression* is cast to **TIMESTAMP WITHOUT TIME ZONE**, and then the time portion is extracted.

- **CHAR**
  
  The string must be a valid string representation of a date or timestamp with or without time zone. If the encoding of the character string is not EBCDIC, it will be cast to an EBCDIC CCSID(37) before conversion takes place.

  **Note**

  If the argument can be null, the result can be null; if the argument’s value is null, the result is the null value.

### TIMESTAMP function

The **TIMESTAMP** function returns a **TIMESTAMP WITHOUT TIME ZONE** value from its argument or arguments.

```
TIMESTAMP( expression1, expression2, n )
```

The rules for the arguments depend on whether you specify the second argument.

**If only one argument is specified without a precision of n**

The argument must be an expression that returns a value of one of the following built-in data types:

- **DATE**
- **TIMESTAMP**
- Character string representation of timestamp

  **Note**

  A time zone in a string representation of a timestamp results in an invalid format. A character string with an actual length of 8 is assumed to be a TOD clock value.

  A character string with an actual length of 14 represents a valid date and time in the form `yyyyxxddhhmmss`, where `yyyy` is the year, `xx` is the month, `dd` is the day, `hh` is the hour, `mm` is the minute, and `ss` is the seconds.

  The resulting timestamp has a precision of 6.
If only one argument is specified with a precision of $n$

The argument must be an expression that returns a value of one of the following built-in data types:

- Date
- Timestamp
- Character string representation of timestamp

The resulting timestamp has a precision of $n$, where $n$ is between 0 and 12 inclusive.

**Note**

A time zone in a string representation of a timestamp results in an invalid format.

If both arguments are specified

The first argument must be an expression that returns a value of one of the following built-in data types:

- Date
- Timestamp
- Character string representation of timestamp

The second argument must be an expression that returns a value of one of the following built-in data types:

- Time
- Character string representation of time

**Note**

If any of the arguments can be null, the result can be null; if any argument's value is null, the result is the null value.

**TIMESTAMP_TZ function**

The `TIMESTAMP_TZ` function returns a `TIMESTAMP WITH TIME ZONE` value from the input arguments.

```
TIMESTAMP_TZ( expression1, expression2, n )
```
The arguments are defined as follows:

**expression1**

_expression1_ is an expression that returns a value of one of the following built-in data types:

- TIMESTAMP WITHOUT TIMEZONE
- TIMESTAMP WITH TIMEZONE
- Character string representation of timestamp without time zone

**expression2**

If you specify _expression2_, _expression1_ must be a timestamp without a time zone, or a string representation of a timestamp without a time zone.

_expression2_ returns a character string that represents a time zone in the format of \( \pm th:tm \) with values ranging from -12:59 to +14:00. _th_ represents the time zone hour, and _tm_ represents the time zone minute.

You can specify _expression2_ as a constant integer in the range of 0 through 12. This value determines the precision of the result.

---

**Note**

If any argument can be null, the result can be null; if any argument's value is null, the result is the null value.

---

**TRANSLATE function**

The TRANSLATE function returns a fixed or varying character string that has a value in which one or more characters of the first argument might have been converted to other characters.

```
TRANSLATE( stringExpression, toString, fromString, pad )
```

The arguments are defined as follows:

**stringExpression**

_stringExpression_ is an expression that specifies the string to be converted. _stringExpression_ must return a value that is a built-in single-byte character string data type. _stringExpression cannot_ be a string constant.
**toString**

`toString` is a fixed-length character constant that is assumed to be in the same encoding scheme as the `stringExpression`. The translate function might not work correctly if the encoding schemes are not the same. The maximum length of `toString` is 256 characters.

If you specify `toString` you must also specify `fromString`. If the length of `toString` is less than the length of `fromString`, `toString` is padded to the length of `fromString` with the `pad` or a blank. If the length of `toString` is greater than `fromString`, the extra characters in `toString` are ignored without warning.

**fromString**

`fromString` is a fixed-length character constant that is assumed to be in the same encoding scheme as the `stringExpression`. The translate function may not work correctly if the encoding schemes are not the same. Alternatively, you can specify the character constant as a hexadecimal constant. The maximum length of `fromString` is 256 characters.

**pad**

`pad` character is optional and defaults to an EBCDIC blank. You can specify `pad` as a character string constant or hexadecimal constant with a length not exceeding one character.

`fromString` is considered the input translate table, `tablei`. `toString` is the output translate table, `tableo`.

**Note**

If only `string expression` is specified, the translate function is treated as an UPPER function.

If the argument can be null, the result can be null; if argument's value is null, the result is the null value.

TRANSLATE searches `tablei` for each character in `stringExpression`. If it finds the character, the corresponding character in `tableo` is used in the result string; if `tablei` contains duplicates, TRANSLATE uses the first (leftmost) occurrence. If it does not find the character, then TRANSLATE uses the original character in `string`. The result string is always the same length as `string`.

**UPPER or UCASE function**

The UPPER or UCASE function returns a string in which all the characters have been converted to uppercase characters.
Note
If the argument can be null, the result can be null; if argument's value is null, the result is the null value.

UPPER or UCASE without UNI locale

The arguments are defined as follows:

stringExpression

stringExpression can be one of the following:

- CHAR SBCS in EBCDIC encoding scheme
- VARCHAR SBCS in EBCDIC encoding scheme

localeNameString

localeNameString is a string constant in EBCDIC that specifies a valid locale name. You do not specify the localeNameString argument, the utility treats the locale as blank.

The only accepted localeNameString types are:

- Blank—SBCS lowercase characters A-Z are converted to SBCS uppercase characters.
- EN_US—SBCS lowercase characters A-Z are converted to SBCS uppercase characters.
- DE_DE—SBCSDeutsch lowercase characters are converted to Deutsch SBCS uppercase characters.

UPPER or UCASE with UNI locale

The arguments are defined as follows:

stringExpression

stringExpression can be defined in the Unicode encoding scheme, and in one of the following data types:

- CHAR SBCS or mixed
- VARCHAR SBCS or mixed
- GRAPHIC or VARGRAPHIC

**UNI**

Currently the NGT SQL language supports only the UNI locale.

**length**

The *length* argument defines the length of the output expressed in bytes. The result string is padded when there is a varying source string.

---

**Note**

- *length* is supported only for the UNI locale.
- The value of *length* cannot exceed 255 for `stringExpression` when defined as a CHAR value, and cannot exceed 32704 for a varying string.
- The *length* argument must be a positive integer constant.
- The value of *length* must be bigger than the length of the source string. If less, the function returns an error.

---

**VALUE/COALESCE/IF NULL function**

The VALUE function argument list *must have a homogeneous data type*. That is, if the first argument is SMALLINT, the rest of the arguments must be of type SMALLINT.

A minimum of two expressions and max of 255 expressions can be specified. The first argument determines the data type of the result and therefore cannot be a constant.

```sql
VALUE( expression1, expression2, ...expression255 )
```

```sql
COALESCE( expression1, expression2, ...expression255 )
```

```sql
IFNULL( expression1, expression2 )
```
Note
The function returns a non-null value of the first non-null expression and of the same data type.
The result can be null only if all arguments can be null. The result is null only if all arguments are null.
The argument's data types must be the same. If one is SMALLINT, all other arguments must be SMALLINT. If one is a decimal then all must be decimal but can have different precisions and scale; the largest precision and scale will be selected as the final result.

VARCHAR function

The VARCHAR function converts a fixed or varying length character string into a variable-length character string.

```
VARCHAR( stringExpression, length )
```

The arguments are defined as follows:

stringExpression

stringExpression is a fixed or variable-length character string.

If stringExpression length exceeds its maximum length, the utility generates an error message and terminates the unit of work (UOW).

length

length is an integer constant of the maximum length of the result. If the value of length is shorter than the length of stringExpression, then the function truncates the result to the specified length, without warning.

Note
If the argument can be null, the result can be null; if argument's value is null, the result is the null value.

WEEK function

The WEEK function returns an integer in a range of 1 through 54 that represents the week of the year. The week starts with Sunday, and January 1 is always in the first week.
expression must return one of the following built-in data types:

- DATE
- TIMESTAMP
- TIMESTAMP WITH TIMEZONE
- Valid string representation of a date or timestamp with or without a time zone

**Note**
You cannot specify expression as a constant.
If the argument can be null, the result can be null; if the argument’s value is null, the result is the null value.

**Example**

```
UNLOAD FORMAT DSNTIAUL
SELECT LEAGUE_NUM, TEAM, WEEK(BOWL_DATES)
FROM BOWL2005.SUMMER_LEAGUES
REORG TABLESPACE BOWL2005.LEAGUES2
DISCARD FROM TABLE BOWL2005.SUMMER_LEAGUES
WHERE (WEEK(BOWL_DATES) BETWEEN 30 AND 45
```

**YEAR function**

The YEAR function returns an integer data type.

expression must return one of the following built-in data types:

- DATE
- DATE DURATION
  
  A date duration occurs as the result of subtracting two dates.
- TIMESTAMP
- TIMESTAMP WITH TIMEZONE
- **TIMESTAMP DURATION**

  A timestamp duration occurs as the result of subtracting two timestamps with precision=6. The result is the year portion of the derived date or date/timestamp duration.

  **Note**

  A CHAR/VARCHAR argument is not accepted unless converted by a DATE, TIMESTAMP, or TIMESTAMP_TZ function.

  If the argument can be null, the result can be null; if the argument’s value is null, the result is the null value.

  **Example**

  ```sql
  YEAR (DATE ('1991-12-31'))
  Will result in an integer of value = 1991
  It is invalid to specify YEAR ('1991-12-31') since the argument is of type CHAR.
  ```

  **Example**

  ```sql
  YEAR (DATE ('1991-12-31')-DATE ('1995-12-31'))
  Will result in an integer of value = -4
  ```

- **ZONED function**

  The ZONED function returns a fixed-length character string that is composed of converting the decimal number to a string of digits without a decimal point, and with a sign byte at the right of the string depicted in a zoned format.

  ```sql
  ZONED( decimalExpression )
  ```

  The length of the result is $p$, where $p$ is the precision of the `decimalExpression`.

  The ZONED function produces similar results as the DIGITS function but with special treatment at the rightmost byte, which is in the form $sn$, where:

  - $s$ is the sign of the result expressed in a hexadecimal constant of X'D' if the result is negative, or X'F' if positive
  - $n$ is the rightmost digit of the decimal expression

  **Example**

  Decimal number -123.4 results in hexadecimal 'F1F2F3D' or 123M.
  +123.4 results in hexadecimal 'F1F2F3F4' or 1234
The result is not translated to the current table's encoding scheme. If the argument can be null, the result can be null; if the argument's value is null, the result is the null value.

**@S370 function**

The @S370 function encodes the numeric data represented by the character expression argument into IBM DB2 internal format.

```
@S370(decimalExpression)
```

This function is mainly used in the LOAD utility product to convert numeric data in decoded format to encoded format in order to perform arithmetic and other advanced functions.

**@ROWID function**

The @ROWID function generates a ROWID value consistent with DB2 ROWID column specifications. The function returns the result as VARCHAR(14) data type SBCS for bit data.

Figure 9: @ROWID function syntax

```
@ROWID()
```

*Note*

The utility might use the FLOAT function to convert the @ROWID function to a real number between 0 and 1 for use as a random number generator.

**Date and time durations**

NGT SQL language supports the following types of durations: labeled, date, time, and timestamp.

**Labeled durations**

NGT SQL labeled durations are large integers. In contrast, IBM DB2 internally treats labeled durations as DECIMAL(15,0) numbers.
In NGT SQL, applying a labeled duration to a numerical expression whose type is not a large integer causes a syntax error. To correct the error, you can apply the INTEGER() function on the expression, before the labeled duration.

--- Example ---
Assume that column 'DEC15' is a DECIMAL(15,0). The following expression causes a syntax error because you can apply the labeled duration (DAYS) only to a large integer:

CURRENT DATE = DATE('1995-01-01') + DEC15 DAYS

Adding the INTEGER function corrects the error:

CURRENT DATE = DATE('1995-01-01') + INTEGER(DEC15) DAYS

--- Date durations ---
NGT utility products can treat any DECIMAL(8,0) or DECIMAL(9,0) field as a date duration. Date durations can be negative or positive, and you can use them to decrement or increment DATES, respectively.

--- Example ---
If \( x \) is the date duration, \( DATE + x \) is equivalent to the following expression:

\[
DATE + \text{YEAR}(x) \times \text{YEARS} + \text{MONTH}(x) \times \text{MONTHS} + \text{DAY}(x) \times \text{DAYS}
\]

--- Note ---
DB2 supports only DEC(8,0). Also, negative dates are not well defined in DB2.

--- Time durations ---
NGT utility products can treat any DECIMAL(6,0) or DECIMAL(7,0) field as a TIME duration.

--- Note ---
DB2 supports only DEC(6,0).

--- Timestamp durations ---
NGT utility products can treat any DECIMAL(20,6) or DECIMAL(21,6) field as a TIMESTAMP duration.
Comparisons

NGT SQL language supports the following comparisons: numeric, decimal, numeric string, and constant.

Numeric comparisons

When you compare two numerical expressions of different types, NGT SQL converts one type to the type of the other and then compares them.

The conversion rules are the same as outlined in SQL reference with the exceptions of FLOAT and DECIMAL comparisons. IBM DB2 converts the decimal to FLOAT and then carries out the comparison in FLOAT format. This technique produces correct results if the decimal precision is greater than 18.

Example

If the following two numbers are compared under DB2, the result will be EQUAL; which is incorrect

```
DEC310=1234567890123456789012345678901
FLOAT=.12345678901234568 E +31
```

To correct this example, NGT SQL converts the float to a decimal number, and then compares them in decimal format. This technique might produce undesirable results when a floating-point fraction cannot be represented exactly in decimal format. In this case, BMC recommends using either the FLOAT() or the DECIMAL() function to control this process.

Decimal comparisons

When comparing two decimal numbers that have different precision and scale, NGT SQL converts the two numbers to a uniform precision and scale, which are derived from the following formulas:

- \[ P = \min(31, \max(S_1, S_2) + \max(P_1 - S_1, P_2 - S_2)) \]
- \[ S = \max(S_1, S_2) \]
Assume that number 1 is DEC(31,31) and has a value of .31; number 2 is DEC(31,0) and has a value of 31. The uniform precision is:

\[ P = \min(31, \max(31,0) + \max(0,31)) \]
\[ = \min(31, 62) \]
\[ = 31 \]

The uniform scale is:

\[ S = \max(31,0) \]
\[ = 31 \]

Therefore, number 2 needs to be converted to DEC(31,31). However, this conversion will result in an overflow.

### Numeric string comparisons

DB2 requires numeric constants to be free of quotation marks and does not accept hexadecimal constants in numeric comparisons. NGT, however, is more flexible and does allow these specifications.

**Example**

If INT1 is a numeric column, NGT allows one of the following constants:

- `INT1=1`
- `INT1='1'`
- `INT1=X'F1'`

### Constant comparison

NGT SQL does not support constant comparison, and results in a syntax errors.

For example, the following comparisons are not permitted:

- `1=1`
- `'A'='A'`

### NGT SQL language keywords

NGT SQL language supports the following keywords: NOT, IN, and LIKE.

**NOT prefix operator**

NGT SQL does not support the NOT prefix operator but does support:

- `IS NOT NULL`
- NOT LIKE
- NOT IN
- The relational comparison not equal (\(^=\))

**IN predicate**

In the NGT utility products, the IN predicate supports the standard DB2 subselect.

To process the subselect, the NGT utility products complete this process:

1. Extract the subselect statement.
2. Pass the SQL to DB2 for processing.
3. Collect the output.
4. Filter the output from DB2 to exclude null and duplicate rows.
5. Place the output in an IBM MVS/ESA data space.
6. Search the output by using a high-speed search engine.

The data type of the IN predicate operands must match exactly:

- The left operand of the IN predicate data type must match the right operand, which is the result that DB2 returned.
- The precision and scale of one operand must match the precision and scale of the other.

If the precision and scale are not the same, you can change them using the INTEGER, FLOAT, or DECIMAL DB2 built-in function. To improve performance when conversion is required, BMC recommends that you convert the right hand operand to the type and length of the left hand operand.

For example, assume that the WHERE clause is specified as follows:

```
WHERE C1 IN (SELECT C2 FROM TB1)
```

The following table shows the possible data type combinations of C1 and C2, and the action required.
Table 11: IN predicate data type conversion actions

<table>
<thead>
<tr>
<th>Data type C1</th>
<th>Data type C2</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMALLINT</td>
<td>INTEGER</td>
<td>INTEGER(C1)</td>
</tr>
<tr>
<td>SMALLINT</td>
<td>DECIMAL(P,S)</td>
<td>DECIMAL(C1,P,S)</td>
</tr>
<tr>
<td>SMALLINT</td>
<td>REAL</td>
<td>(C1) IN (SELECT FLOAT(B) FROM TB1)</td>
</tr>
<tr>
<td>SMALLINT</td>
<td>FLOAT</td>
<td>FLOAT(C1)</td>
</tr>
<tr>
<td>INTEGER</td>
<td>SMALLINT</td>
<td>(SELECT INTEGER(C2) FROM TB1)</td>
</tr>
<tr>
<td>INTEGER</td>
<td>DECIMAL(P,S)</td>
<td>DECIMAL(C1,P,S)</td>
</tr>
<tr>
<td>INTEGER</td>
<td>REAL</td>
<td>FLOAT(C1) IN (SELECT FLOAT(B) FROM TB1)</td>
</tr>
<tr>
<td>INTEGER</td>
<td>FLOAT</td>
<td>FLOAT(C1)</td>
</tr>
<tr>
<td>REAL</td>
<td>SMALLINT</td>
<td>FLOAT(C1) IN (SELECT FLOAT(B) FROM TB1)</td>
</tr>
<tr>
<td>REAL</td>
<td>INTEGER</td>
<td>FLOAT(C1) IN (SELECT FLOAT(B) FROM TB1)</td>
</tr>
<tr>
<td>REAL</td>
<td>DECIMAL</td>
<td>DECIMAL(C1,P,S)</td>
</tr>
<tr>
<td>REAL</td>
<td>FLOAT</td>
<td>FLOAT(C1)</td>
</tr>
<tr>
<td>FLOAT</td>
<td>SMALLINT</td>
<td>(SELECT FLOAT(C2) FROM TB1)</td>
</tr>
<tr>
<td>FLOAT</td>
<td>INTEGER</td>
<td>(SELECT FLOAT(C2) FROM TB1)</td>
</tr>
<tr>
<td>FLOAT</td>
<td>DECIMAL</td>
<td>(SELECT FLOAT(C2) FROM TB1)</td>
</tr>
<tr>
<td>FLOAT</td>
<td>REAL</td>
<td>(SELECT FLOAT(C2) FROM TB1)</td>
</tr>
<tr>
<td>DECIMAL(P,S)</td>
<td>SMALLINT</td>
<td>(SELECT DECIMAL(C2,P,S) FROM TB1)</td>
</tr>
<tr>
<td>DECIMAL(P,S)</td>
<td>INTEGER</td>
<td>(SELECT DECIMAL(C2,P,S) FROM TB1)</td>
</tr>
<tr>
<td>DECIMAL(P,S)</td>
<td>REAL</td>
<td>(SELECT DECIMAL(C2,P,S) FROM TB1)</td>
</tr>
<tr>
<td>DECIMAL(P,S)</td>
<td>FLOAT</td>
<td>(SELECT DECIMAL(C2,P,S) FROM TB1)</td>
</tr>
<tr>
<td>DECIMAL(P1,S1)</td>
<td>DECIMAL</td>
<td>DECIMAL(C1,P,S) IN (SELECT DECIMAL(C2,P,S) FROM TB1) Where: p = min(31,max*s1,s2)+maxp1-s1,p2-s2, s=max(s1,s2)</td>
</tr>
</tbody>
</table>

To further accelerate the search process, or if the data does not reside in the DB2 table, use the external file technique. This technique utilizes non DB2 syntax and requires an input file that has been generated by either the NGT Unload utility or the DB2 REORG UNLOAD utility.

The full syntax of the feature is defined as follows:

```sql
WHERE columnName IN (FILE(ddname) FORMAT(format))
```
1. **columnName** is the left operand of the IN predicate. This operand must be a column name and cannot be a constant. The attributes of this column dictate the data type, length, and null attribute of the first field of the external file.

2. **ddname** is the name of the DD statement in the current job step that references the external data file.

3. **format** is the internal format of the data in the external file. Specify FORMAT(UNLOAD). For more information about unload formats, see the *BMC Next Generation Technology Unload for DB2 for z/OS Reference Manual*.

Processing an IN predicate by using an external file causes NGT utility products to complete these actions:

1. Read the specified DDNAME (which can be on DISK or TAPE).
2. Filter out duplicate and null rows.
3. Place the data in an MVS/ESA data space.
4. Searched the data via a high-speed search engine.

Processing of the IN predicate having an external file causes NGT utility products to read the specified DDNAME (which can be on DISK or TAPE), filter out duplicate and null rows, and place the data in an MVS/ESA data space. It is searched via a high speed search engine.

The IN predicate external file can contain several data fields or columns. However, only the first data field is extracted from the file. The remaining data fields are ignored. The first data field is recognized by using the same attributes of the left operand and can be the concatenation of columns.

---

**Example**

Assume that the following conditions exist:

- File(DD1) contains the following records in hexadecimal format:
  
  C1C2C3C4C5

- The data type for C1 is VARCHAR.

Executing the following expression returns incorrect output because the record in the file does not look like a VARCHAR:

```
WHERE C1 IN(FILE(DD1) FORMAT(ARCHIVE))
```

In one WHERE clause, you can have up to 200 IN predicates with different files, subselects, or both. You can use Boolean logic to form very complex pattern matching on large amounts of data at very high speeds. However, WHERE clause
syntax errors might occur if memory space is exceeded; in that case attempt to reduce the select data or add more paging data sets to your paging subsystem.

--- Note ---
If you are unloading to a file to be referenced by the NGT Reorg WHERE ... IN FILE(ddname), specify the same +DIGITS parameter on both the Unload job and the Reorg job.

---

**LIKE predicate**

The LIKE predicate supports only character data (no graphics).

It also supports patterns constructed dynamically, such as:

```
A1 LIKE SUBSTR(A2,1,1)||'%'||'CDE'
```

A1 and A2 represent character columns.

---

**Reserved words**

The NGT utilities do not use reserved words or keywords.

---

**Restrictions using NULL/NOT NULL predicate**

If you are using the NULL/NOT NULL predicate, you must ensure that the underlying column has the NULL attribute defined.

--- Example ---

The following expression causes a syntax error if column 'COL1' was defined as NOT NULL:

```
COL1 IS NOT NULL
```

---

SQL statements that are ordinarily coded using the IBM MOD function must instead use the operator `//` (double forward slash) as a substitute. For example instead of entering WHERE clause `MOD(DAYS(CURRENT DATE), 2) = 1`, enter `WHERE DAYS(CURRENT DATE) // 2 = 1`. 

---
Comparing DB2 schemas

If you have a license for the BMC Object Administration for DB2 solution, you can use BMC Workbench to compare two sets of data structures. The objects can reside in a DB2 catalog or a DDL file.

The comparison process enables you to perform the following tasks:

- Record changes made to a local or remote subsystem
- Understand structure changes between two subsystems to help you determine which changes to migrate

For more information, view the Quick Course "Workbench for DB2 - Comparing DB2 Schemas."

Comparing two schemas

Comparing schemas involves generating and viewing a set of reports. The reports list the objects that need to be created, altered, or dropped to make the two structures identical. BMC Workbench generates the Change Definition Language (CDL) that defines the changes that must be applied to make the schema's structures identical. You can save the CDL to a dataset and edit the CDL in the Scratchpad editor.

BMC Workbench enables you to make the following comparisons:

- DB2 catalog to DB2 catalog
- DDL to DDL
- DB2 catalog to DDL
- DDL to DB2 catalog
**Note**

BMC Workbench can generate DDL statements containing lines that are longer than 72 characters. These DDL statements cannot be used in a schema comparison.

The same set of reports is available for each type of comparison. For more information, see “Schema-comparison reports” on page 189.

Use the following procedure to create and display a Schema Compare View for each comparison that you want to see. You can also navigate to and run a saved schema comparison.

**To create a Schema Compare View**

The comparison requires two inputs:

- The schema or set of DB2 objects that has the structure definitions that you want to change (called the Primary Input in the procedure)
- The schema that has the preferred structure definitions, to serve as the basis for changing the other schema (called the Secondary Input)

**Note**

After you have specified the two data structures for comparison, you can click **Finish**, without changing the default dependent objects, attributes, and options values.

1. In the Schema Management perspective, click **Add Schema Compare View**.

   The Define a Comparison wizard opens.

2. On the wizard's **Step 1-General** page, complete the following fields and click **Next**:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a name for the report.</td>
</tr>
<tr>
<td>Select a report</td>
<td>Select to create a new report or use an existing report.</td>
</tr>
</tbody>
</table>

3. On the wizard’s **Step 2** page, take the appropriate action as follows:

   - **If you are using an existing report, complete these actions:**
     
     1. On the wizard's **Step 2 - Comparison Dataset** page, navigate to and select the data set that contains the report and click **Next**.
        
        For more information, see “Viewing and editing a data set” on page 229.
2 (optional) In the wizard’s **Step 3 - Options** page, select **Enable Trace** to provide informational messages in the View Messages dialog.

3 Click **Finish**.

4 Click to rerun the report.

- **If creating a new report**, on the wizard’s **Step 2-Type of Comparison** page, complete the following fields and click **Next**:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compare Type</td>
<td>Select the type of comparison that you require.</td>
</tr>
<tr>
<td>Output Compare Report</td>
<td>Navigate to and select a data set to save the output of the comparison.</td>
</tr>
</tbody>
</table>

4 On the wizard’s **Step 3 - Data Structures** page, complete the following fields according to the type of comparison that you are performing:

### Table 12: Catalog to catalog

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DB2 objects that do not have the desired structure (primary input)</strong></td>
<td></td>
</tr>
<tr>
<td>Object Type</td>
<td>Select the object type.</td>
</tr>
<tr>
<td>DB2 Connection</td>
<td>Select the DB2 connection from your list of favorites or add a DB2 connection to the list. (For more information, see “Connecting to DB2 subsystems” on page 29.)</td>
</tr>
<tr>
<td>Filter</td>
<td>Specify a filter for the name of the DB2 objects that you want to compare.</td>
</tr>
<tr>
<td><strong>DB2 objects that have the desired structure (secondary input)</strong></td>
<td></td>
</tr>
<tr>
<td>DB2 Connection</td>
<td>Select the DB2 connection from the list.</td>
</tr>
<tr>
<td>Filter</td>
<td>Specify a filter for the name of the DB2 objects that you want to compare.</td>
</tr>
</tbody>
</table>

### Table 13: DDL to DDL

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DB2 objects that do not have the desired structure (primary input)</strong></td>
<td></td>
</tr>
<tr>
<td>DB2 Connection</td>
<td>Select the DB2 connection from your list of favorites or add a DB2 connection to the list, as explained in “Connecting to DB2 subsystems” on page 29.)</td>
</tr>
<tr>
<td>DDL Data Set</td>
<td>Select the DDL data set. This data set defines the scope of the comparison. For more information, see “Viewing and editing a data set” on page 229.</td>
</tr>
<tr>
<td><strong>DB2 objects that have the desired structure (secondary input)</strong></td>
<td></td>
</tr>
</tbody>
</table>
### Table 14: Catalog to DDL

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 objects that do not have the desired structure (primary input)</td>
<td>Select the DB2 connection where the schema is stored.</td>
</tr>
<tr>
<td>DB2 Connection</td>
<td>Select the DB2 connection where the schema is stored.</td>
</tr>
<tr>
<td>DB2 objects that have the desired structure (secondary input)</td>
<td>Select the DDL data set. This data set defines the scope of the comparison.</td>
</tr>
<tr>
<td>DDL Data Set</td>
<td>Select the DDL data set. This data set defines the scope of the comparison.</td>
</tr>
<tr>
<td></td>
<td>For more information, see “Viewing and editing a data set” on page 229.</td>
</tr>
</tbody>
</table>

### Table 15: DDL to Catalog

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 objects that do not have the desired structure (primary input)</td>
<td>Select the DDL data set. This data set defines the scope of the comparison.</td>
</tr>
<tr>
<td>DDL Data set</td>
<td>Select the DDL data set. This data set defines the scope of the comparison.</td>
</tr>
<tr>
<td></td>
<td>For more information, see “Viewing and editing a data set” on page 229.</td>
</tr>
<tr>
<td>DB2 objects that have the desired structure (secondary input)</td>
<td>Select the DB2 connection where the schema is stored.</td>
</tr>
<tr>
<td>DB2 Connection</td>
<td>Select the DB2 connection where the schema is stored.</td>
</tr>
</tbody>
</table>

5 Click **Next**.

6 On the wizard’s **Step 4 - Dependent objects** page, select the dependent objects to include in the comparison and click **Next**.

**Note**

To compare a DB2 catalog to a DB2 catalog, the **Resolve object or creator name differences** checkbox is enabled by default.

When this option is enabled, the product excludes databases, table spaces, synonyms, and aliases from the comparison because duplicates might occur. The product performs this name conversion to resolve the differences in the object owners, schemas, and database names.

7 On the wizard’s **Step 5 - Attributes** page, select the attributes for each object type to include in the comparison and click **Next**.

8 On the wizard’s **Step 6 - Options** page, specify the option values for the current definition.
These values do not affect the global compare options that you specify by using the Options button.

9  Click **Finish**.

The Reports tab displays the reports for this comparison. For more information, see “Schema-comparison reports” on page 189.

10 *(optional)* If you want to view the generated CDL for the comparison, click **Show CDL** to view the CDL in the Scratchpad.

---

**Note**

You cannot run the CDL from the Scratchpad.

11 *(optional)* If you want to rerun the comparison, click **Run**.

**To copy a Schema Compare View**

You can also copy all the definitions (name, type and output, data structures, dependent objects, attributes, and options) of an existing Schema Compare View.

1  In the Navigation pane, right-click a Schema Compare View.

2  Click **Copy Definition**.

   A copy of the selected Schema Compare View is created.

3  Edit the Schema Compare View.

4  Save the Workspace.

**Schema-comparison reports**

When you run a schema comparison, you can generate and view reports that display the required changes to match the schemas by statement type or by object type, or show the detailed CDL for each action.
Overview by Statement Type report

The report displays the number of each CDL statement (ALTER, DROP, and CREATE) required to match the two schemas. The report also displays a breakdown of the objects that each statement affects.

Figure 10: Sample Overview by Statement Type report

Table 16: Legend—Overview by Statement Type report

<table>
<thead>
<tr>
<th>#</th>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Show Compare Report button</td>
<td>Updates the compare reports</td>
</tr>
</tbody>
</table>
### Description | Details
--- | ---
2 | **Show CDL button** Displays the CDL statements in the Scratchpad. 
**Note:** You cannot run the CDL from the Scratchpad.
3 | Reports tab Displays the reports
4 | Report description Describes what the pie chart is showing.
For example, the description in the figure indicates that the pie chart shows results for the CREATE action.
5 | Pie chart section Shows the objects breakdown for the selected statement (ALTER, CREATE, DROP) in the column chart
6 | Report selector Enables you to select the following reports:
- Overview by Statement Type
- Overview by Object Type
- Details
7 | **Schema Management options button** Displays the Schema Management options.
See “Setting Schema Management options” on page 196.
8 | Help button Displays the online Help
9 | **Show object color legend button** Shows or hides the object-color legend
10 | Object color legend Indicates the color that corresponds to each object type in the pie chart
11 | CDL statement detail pane Shows the detailed CDL statement required to perform the action for the selected object (in this example, a CREATE statement for the table space)
12 | List of CDL statements Lists the CDL statements (in this example, a list of the CREATE TABLESPACE statements). The detailed CDL for the selected statement is displayed.
13 | Column chart showing the count of CREATE, ALTER, and DROP statements required to make the first structure identical to the second structure When you click on a column, displays the object type breakdown
When you double-click on a column, displays a list of all the CDL statements for the selected column

This report is interactive. You can change the information displayed by clicking or hovering over report elements.

You can navigate through the report as follows:

<table>
<thead>
<tr>
<th>To</th>
<th>Do this</th>
</tr>
</thead>
<tbody>
<tr>
<td>List statements for a statement type</td>
<td>Double-click the relevant statement column in the column chart.</td>
</tr>
<tr>
<td>List statements for the selected object type and statement type</td>
<td>Double-click the relevant section of the pie chart.</td>
</tr>
</tbody>
</table>
To | Do this
---|---
Update the object breakdown in the pie chart | Click the relevant statement column in the column chart.
Update a list of statements for the selected object type and statement type | Click the relevant section of the pie chart.
View the count information | Hover over the relevant column of the column chart or relevant segment of the pie chart.

**Example**

This example lets you view the number of dropped table spaces:

1. In the Summary by Statement Type column chart, double-click the column representing the DROP action.
2. In the Object Type Breakdown pie chart, hover over the segment representing table spaces, then double-click the segment representing table spaces.
   A list of DROP TABLESPACE statements is displayed.

**Example**

This example lets you View a CREATE INDEX statement for a specific index:

1. In the Summary by Statement Type column chart, click the column representing the CREATE action.
2. In the Object Type Breakdown pie chart, double-click the segment representing Indexes.
   A list of CREATE INDEX statements is displayed.
3. In the CREATE INDEX Statements panel (left pane), select the CREATE INDEX statement for the specific index.
   The detailed CDL statement is displayed in the right pane.

**Related Information**

- “Setting Schema Management options” on page 196
- “Overview by Object Type report” on page 193
- “Details report” on page 195
Overview by Object Type report

This report displays the number of DROP, CREATE, and ALTER statements required for each object type. You can select and view a specific CDL statement for an object.

Figure 11: Sample Overview by Object Type report

Table 17: Legend —Overview by Object Type report

<table>
<thead>
<tr>
<th>#</th>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Show Compare Report button</td>
<td>Updates the compare reports</td>
</tr>
<tr>
<td>2</td>
<td>Show CDL button</td>
<td>Displays the CDL statements in the Scratchpad</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> You cannot run the CDL from the Scratchpad.</td>
</tr>
<tr>
<td>3</td>
<td>Details for each object type</td>
<td>For each object type, displays the number of DROP, CREATE, and ALTER CDL statements and the number of fetches</td>
</tr>
<tr>
<td>#</td>
<td>Description</td>
<td>Details</td>
</tr>
<tr>
<td>----</td>
<td>----------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>4</td>
<td>CDL statements pane</td>
<td>Lets you select a statement in order to view its details in the pane on the right</td>
</tr>
<tr>
<td>5</td>
<td>CDL details pane</td>
<td>Displays the details of the selected CDL statement</td>
</tr>
</tbody>
</table>
| 6  | **Schema Management options** button | Displays the Schema Management options  
See “Setting Schema Management options” on page 196. |
| 7  | **Help** button     | Displays the online Help                                                |

**Related Information**

- “Setting Schema Management options” on page 196
- “Overview by Statement Type report” on page 190
- “Details report” on page 195
Details report

This report displays a filterable list of all actions needed in order to make the two schemas match. You can filter the list by objects or by actions. You can select an action or object, and view the CDL statements.

Figure 12: Details report

Table 18: Legend

<table>
<thead>
<tr>
<th>#</th>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CDL statement pane</td>
<td>Lets you select a statement in order to view its details in the pane on the right</td>
</tr>
<tr>
<td>2</td>
<td>Object filters</td>
<td>By default, displays all objects (with object filters shown in green) Click a filter to hide the associated object.</td>
</tr>
<tr>
<td>3</td>
<td>All objects filter button</td>
<td>Displays all objects Click to select or deselect all filters.</td>
</tr>
<tr>
<td>4</td>
<td>CDL details pane</td>
<td>Displays the details of the selected CDL statement</td>
</tr>
<tr>
<td>5</td>
<td>Action filters</td>
<td>Hides the associated action</td>
</tr>
</tbody>
</table>
### Setting Schema Management options

You can use the **Options** button to set optional behavior for the Schema Management feature. These settings also serve as the default option settings for all new Schema Compare views.

**Note**

You can override the options settings for a specific Schema Compare View in the Define a Comparison wizard. For more information, see “Comparing two schemas” on page 185.

The following options can be specified:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use European format for decimal numbers (EURO)</td>
<td>Select whether to use the European decimal format (1.234.567,89). This value corresponds to the CHANGE MANAGER EURO installation option.</td>
<td>No</td>
</tr>
<tr>
<td>Use padded values for columns of type VARCHAR (PADDED)</td>
<td>Select whether to pad VARCHAR values. This option corresponds to the CHANGE MANAGER OVERRIDE(IXPADDLEDY) and OVERRIDE(IXPADDLEDN) ALUIN keywords, and to the PADIX DSNZPARM.</td>
<td>No</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
<td>Default value</td>
</tr>
<tr>
<td>------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
</tbody>
</table>
| Include the primary key constraint in the scope      | Select whether to include the primary key constraint in the scope.  
This option corresponds to the CHANGE MANAGER OVERRIDE( KEEP-ALL-PRIMARY ) ALUIN keyword.                                                    | No            |
| Include white spaces in native SQL stored procedure comparisons | Select whether to include white spaces in native SQL stored procedures.  
This option corresponds to the CHANGE MANAGER CMPSPTXT installation option, and the SPTEXT( EXACT ) or SPTEXT( NOEXACT ) ALUIN keyword. | Yes           |
| CCSID                                                | Select the default encoding scheme for databases that are created.  
This option corresponds to the CHANGE MANAGER CCSID installation option.                                                                           | EBCDIC        |
| CISIZE                                               | Select N to create VSAM data sets with a control interval size (CISIZE) of 4 KB, regardless of the page size (buffer pool size) of the table space. Select Y to support larger CISIZE values.  
This option corresponds to the DSVCI DSNZPARM.                                                      | Blank         |
| Length of inline LOB columns                         | Enter a value for the inline LOB column.  
This option corresponds to the LOB_INLINE_LENGTH DSNZPARM.                                                                                         | 0             |
| OVERRIDE                                             | Enter one or more space-separated CHANGE MANAGER OVERRIDE keywords (maximum of 57 characters).  
**Note:** Do not enclose the OVERRIDE keywords between parentheses.  
For more information, see “OVERRIDE keywords” on page 197.                                          | NA            |
| Enable Trace                                         | Select whether to include informational messages. Consult with BMC Customer Support before using this option.                                                                                               | No            |

**OVERRIDE keywords**

You can use OVERRIDE keywords in the **OVERRIDE** field to override the comparison default values.

You can enter a list of space-separated keywords, containing a maximum of 57 characters, into the field.
Table 19: OVERRIDE keywords

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CKSEGSIZE</td>
<td>Shows the differences between a simple table space and a segmented table space defined with a SEGSIZE of 4</td>
</tr>
<tr>
<td>ENQFAILRC4</td>
<td>Produces a return code of 4 when the deletion of a baseline fails because the baseline is in use. This keyword overrides the default return code of 0. You must manually insert this keyword in the ALUIN input stream.</td>
</tr>
<tr>
<td>FKNN</td>
<td>Shows the differences between two foreign keys when one has a three-part explicit name and the other has no name (and part three is not specified)</td>
</tr>
<tr>
<td>GENOBID</td>
<td>Specifies the value of the OBID as a comment for a CREATE TABLE statement in the CDL. This comment is used when you compare a file or DB2 catalog to either a catalog baseline or a DB2 catalog. The comment also appears in the baseline report.</td>
</tr>
<tr>
<td>INCLUDEPARENTS</td>
<td>Includes the parent objects of tables or table spaces in the scope of a catalog-to-catalog comparison. This keyword applies when you are specifying the object types and object names for primary input and secondary input to explicitly define the scope of the comparison.</td>
</tr>
<tr>
<td>IXPADDDEDN</td>
<td>Uses the default of PADDED NO for an index, regardless of the value in the DB2 subsystem parameter DSNZPARM</td>
</tr>
<tr>
<td>IXPADDREDY</td>
<td>Uses the default of PADDED YES for an index, regardless of the value in the DB2 subsystem parameter DSNZPARM</td>
</tr>
<tr>
<td>KEEP-ALL-PRIMARY</td>
<td>Includes the primary key constraint in a baseline or comparison when you exclude unique constraints</td>
</tr>
<tr>
<td>LONGVARCOLS</td>
<td>Shows the differences between two columns when one column has a LONG VARCHAR data type and the other column has a VARCHAR data type</td>
</tr>
<tr>
<td>NO-PRIMARY-CONSTRAINTS</td>
<td>Ignores the primary key constraint in a baseline or comparison when you exclude unique constraints</td>
</tr>
<tr>
<td>NOAPPEND</td>
<td>Ignores the APPEND attribute of a table in a comparison</td>
</tr>
<tr>
<td>NOCLONES</td>
<td>Ignores clone tables in a comparison</td>
</tr>
<tr>
<td>NODSSIZE</td>
<td>Ignores the DSSIZE attribute of a table space in a comparison</td>
</tr>
<tr>
<td>NOENFORCED</td>
<td>Ignores the ENFORCED attribute of a foreign key in a table in a comparison</td>
</tr>
<tr>
<td>Keyword</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NOGBPCACHE</td>
<td>Ignores the GBPCACHE attribute of table spaces in a comparison</td>
</tr>
<tr>
<td>NOINLINELOB</td>
<td>Ignores the INLINE attribute of table spaces in a comparison</td>
</tr>
<tr>
<td>NOIXCLOSE</td>
<td>Ignores the CLOSE attribute of an index in comparison</td>
</tr>
<tr>
<td>NOIXCOMPRESS</td>
<td>Ignores the COMPRESS attribute of an index in a comparison</td>
</tr>
<tr>
<td>NOIXCOPY</td>
<td>Ignores the COPY attribute of an index in a comparison</td>
</tr>
<tr>
<td>NOIXPADDDED</td>
<td>Ignores the PADDDED attribute of an index in a comparison</td>
</tr>
<tr>
<td>NOLIMITKEYS</td>
<td>Ignores the LIMITKEY attribute of a partitioned table space in a comparison</td>
</tr>
<tr>
<td>NOLOG</td>
<td>Ignores the LOG attribute of a table space in a comparison</td>
</tr>
<tr>
<td>NOPiecesize</td>
<td>Ignores the PIECESIZE attribute of an index in a comparison</td>
</tr>
<tr>
<td>NOSPACTIVE</td>
<td>Ignores the active version attribute for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPASUTIME</td>
<td>Ignores the ASUTIME option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPCCSID</td>
<td>Ignores the CCSID option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPCCOLLID</td>
<td>Ignores the package collection ID option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPCommitReturn</td>
<td>Ignores the commit on return option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPConAccreso</td>
<td>Ignores the concurrent access resolution option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPCurrentData</td>
<td>Ignores the CURRENTDATA option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPDataAcc</td>
<td>Ignores the SQL DATA ACCESS option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPDateFormat</td>
<td>Ignores the date format option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPDBINFO</td>
<td>Ignores the DBINFO option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPDebugMode</td>
<td>Ignores the DEBUG MODE option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPDecimal</td>
<td>Ignores the DECIMAL option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPDeferPrepare</td>
<td>Ignores the DEFER PREPARE option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPDeterministic</td>
<td>Ignores the deterministic option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPDynrules</td>
<td>Ignores the dynamic rules option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPExplain</td>
<td>Ignores the EXPLAIN option for stored procedures in a comparison</td>
</tr>
<tr>
<td>Keyword</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NOSPEXTNAME</td>
<td>Ignores the external name option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPFORUPD</td>
<td>Ignores the FOR UPDATE clause option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPIMMEDWRITE</td>
<td>Ignores the IMMEDIATE WRITE option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPISODELEVEL</td>
<td>Ignores the isolation level option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPJAVA</td>
<td>Ignores the JAR SCHEMA, JAR ID, and JAVA SIGNATURE options for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPKEEPDYN</td>
<td>Ignores the KEEP DYNAMIC option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPLANGUAGE</td>
<td>Ignores the implementation language option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPMAXFAIL</td>
<td>Ignores the maximum number of failures option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPOPTHINT</td>
<td>Ignores the OPPTHINT option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPPACKOWNER</td>
<td>Ignores the package owner option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPPARAMETERS</td>
<td>Ignores the parameter attributes for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPARMCCSID</td>
<td>Ignores the PARAMETER CCSID option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPAPMDATATYPE</td>
<td>Ignores the data type of the parameter for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPARMFORBIT</td>
<td>Ignores the subtype of the parameter for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPARMINOUT</td>
<td>Ignores the row type of the parameter for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPARMLOC</td>
<td>Ignores the location of the parameter for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPARMNAME</td>
<td>Ignores the name of the parameter for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPARMORDER</td>
<td>Ignores the ordinal number of the parameter for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPARMSTYLE</td>
<td>Ignores the PARAMETER STYLE option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPARMVCHAR</td>
<td>Ignores the varying length string parameter for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPPROGRAMTYPE</td>
<td>Ignores the PROGRAM TYPE option for stored procedures in a comparison</td>
</tr>
<tr>
<td>Keyword</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NOSPQUALIFIER</td>
<td>Ignores the qualifier option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPREGISTERS</td>
<td>Ignores the SPECIAL REGISTER option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPRELEASEAT</td>
<td>Ignores the release option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPRESULTSET</td>
<td>Ignores the result sets option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPRUNOPTS</td>
<td>Ignores the RUNOPTS option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPSECURITY</td>
<td>Ignores the EXTERNAL SECURITY option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPSQLPATH</td>
<td>Ignores the SQL path (PATHSCHEMAS) option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPSTAYRESIDENT</td>
<td>Ignores the STAY RESIDENT option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPTEXT</td>
<td>Ignores the text option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPSTFORMAT</td>
<td>Ignores the time format option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPVALIDATE</td>
<td>Ignores the VALIDATE option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPWLMENV</td>
<td>Ignores the WLM environment option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSPWLMMNEST</td>
<td>Ignores the WLM environment for nested calls option for stored procedures in a comparison</td>
</tr>
<tr>
<td>NOSTOREDPROCS</td>
<td>Ignores stored procedures in the scope of a comparison</td>
</tr>
<tr>
<td>NOSUBTYPE</td>
<td>Ignores the subtype of a column of a table in a comparison</td>
</tr>
<tr>
<td>NOTBAUDIT</td>
<td>Ignores the AUDIT attribute of a table in a comparison</td>
</tr>
</tbody>
</table>
| NOTBCOLORER         | Ignores differences in the order of the columns in a table during a comparison  
<p>|                     | If you specify NOTBCOLORER and you add a new column to the table, the product adds the column to the end of the table. |
| NOTBDEFVALUE        | Ignores the default value of a column in a table in a comparison           |
| NOTBTSAUTO          | Does not apply automatic change rules to match all of the table's table space names when comparing at the table level |
| NOTSCCLOSE          | Ignores the CLOSE attribute of a table space in a comparison               |
| NOVWQQUAL           | Ignores the QUALIFIED attribute (explicit qualifier) of a view in a comparison |</p>
<table>
<thead>
<tr>
<th><strong>Keyword</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
</table>
| **SPOWNER**     | Shows the differences between the owners of stored procedures  
By default, the product does not compare the owners.                                    |
| **SPTEXT-TRACE**| Provides a hexadecimal dump of the text for a stored procedure when all of the following conditions are met:  
- The product retrieves the information from the SYSIBM catalog.  
- Compare detects a difference between the text for the primary input and secondary input. |
| **UCNN**        | Shows the differences between two unique constraints when one has a three-part explicit name and the other has no name (and part three is not specified) |
Managing Backup and Recovery

The Recovery Management perspective lets you view the status of the recovery process, verify the recoverability of individual objects to a specified recovery point, and estimate the elapsed time for a backup or a recovery.

Note
This feature is available if you have a license for the BMC Recovery for DB2 solution.

For more information, view the Quick Course "Workbench for DB2 - Assessing Backup and Recovery Reports."

Viewing recovery progress

Use the following procedure to view the progress of a recovery, and identify which objects in the selected object set were recovered, rebuilt, or not recovered.

To view a Recovery Progress report

1. In the Recovery Management perspective, click Add View and select Progress Report from the menu.

2. Complete the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a name for the report</td>
</tr>
<tr>
<td>RTO Target</td>
<td>Accept the default value or enter the maximum acceptable Recovery Time Objective (hh:mm) for this report. To define the default value, see “Setting Recovery Management options” on page 215.</td>
</tr>
<tr>
<td>DB2 Connection</td>
<td>Select a DB2 connection from your list of favorites. If necessary, you can add a connection. See “Connecting to DB2 subsystems” on page 29.</td>
</tr>
<tr>
<td>Object Set</td>
<td>Navigate to and select an object set.</td>
</tr>
</tbody>
</table>
### Viewing recovery progress

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Time</td>
<td>Enter a time to begin the Recovery process.</td>
</tr>
<tr>
<td>Include Indexes</td>
<td>Select to include index information.</td>
</tr>
</tbody>
</table>

3. Click **Run**.

The Recovery Progress Report tab is displayed.

In this interactive report, you can refresh and update the results whenever you want. You can click each bar of the bar chart to see a list of the recovered, rebuilt, and not recovered objects in the Details tab.

**Figure 13: Recovery Progress Report**

![Recovery Progress Report](image)

**Table 20: Legend**

<table>
<thead>
<tr>
<th>#</th>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Run button</td>
<td>Executes the report</td>
</tr>
<tr>
<td>2</td>
<td>Progress By selector</td>
<td>Lets you choose to display progress by size or by count</td>
</tr>
<tr>
<td>3</td>
<td>Size unit selector</td>
<td><em>(Progress by Size only)</em> Lets you choose to display the size in MB, GB, or TB By default, the size is displayed in MB.</td>
</tr>
<tr>
<td>#</td>
<td>Description</td>
<td>Details</td>
</tr>
<tr>
<td>----</td>
<td>-----------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>4</td>
<td>Progress bar for a table-space partition recovery</td>
<td>Shows how much of the recovery has been completed. You can click the progress bar to view the list of recovered and not recovered table spaces.</td>
</tr>
<tr>
<td>5</td>
<td>Progress bar for an index partition recovery</td>
<td>Shows how much of the recovery has been completed. You can click the progress bar to view the list of recovered, rebuilt, and not recovered index partitions.</td>
</tr>
<tr>
<td>6</td>
<td>Progress bar for all the objects being recovered</td>
<td>Lets you view the list of recovered and not recovered objects.</td>
</tr>
<tr>
<td>7</td>
<td><strong>Time to Completion</strong> field</td>
<td>Shows the estimated time when the recovery will finish</td>
</tr>
<tr>
<td></td>
<td></td>
<td>![green checkmark] indicates that the recovery will finish within your defined Recovery Time Objective (RTO).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>![red exclamation mark] indicates that the time will exceed your RTO.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For more information, see <a href="#">Step 2 on page 26</a>.</td>
</tr>
<tr>
<td>8</td>
<td>Recovery Manager options</td>
<td>Displays the Recovery Manager options</td>
</tr>
<tr>
<td></td>
<td></td>
<td>See “Setting Recovery Management options” on page 215.</td>
</tr>
<tr>
<td>9</td>
<td>Refresh button</td>
<td>Refreshes the report</td>
</tr>
<tr>
<td>10</td>
<td>Color legend</td>
<td>Indicates the colors that the bar charts use to convey recovery-completion status</td>
</tr>
</tbody>
</table>

---

**Related Information**

- “Setting Recovery Management options” on page 215
- “Verifying recoverability” on page 205
- “Estimating backup time” on page 209
- “Estimating recovery time” on page 212

---

**Verifying recoverability**

You can run the Recoverability report to validate the recoverability of each object in an object set to a selected recovery point. The report identifies any objects that are not recoverable. For example, the report identifies specific data sets that do not have available copies. You should run the Recoverability report as part of your routine disaster recovery preparations.
Note
The recoverability information is obtained from the most recent execution of the ARMBGPV program for the selected recovery point. Clicking refreshes the recoverability information. For more information, see “Refreshing the recoverability information” on page 208.

To run the Recoverability report

1. In the Recovery Management perspective, click Add View and select Recoverability Report.

2. In the Definition tab, complete the following fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a name for the report.</td>
</tr>
<tr>
<td>DB2 Connection</td>
<td>Select a DB2 connection from your list of favorites. If necessary, you can add a connection. See “Connecting to DB2 subsystems” on page 29.</td>
</tr>
<tr>
<td>Object Set</td>
<td>Navigate to and select an object set.</td>
</tr>
<tr>
<td>Recovery Point</td>
<td>Select any of the following recovery points for the object set:</td>
</tr>
<tr>
<td></td>
<td>■ To Current: recover to the current time stamp</td>
</tr>
<tr>
<td></td>
<td>■ To Copy: recover to the last incremental copy</td>
</tr>
<tr>
<td></td>
<td>■ To Full copy: recover to the last full copy</td>
</tr>
<tr>
<td></td>
<td>■ To Quiesce: recover to the last quiesce point</td>
</tr>
<tr>
<td></td>
<td>■ To Log Mark: recover to a log mark that you defined in the Log Master for DB2 product.</td>
</tr>
<tr>
<td></td>
<td>If you select this option, you must also enter the log mark name.</td>
</tr>
</tbody>
</table>

3. Click Run.
The Recoverability report tab displays the count of unrecoverable objects and why they cannot be recovered. Clicking View Details lists unrecoverable objects in the Details tab.

**Figure 14: Recoverability report**

**Table 21: Legend**

<table>
<thead>
<tr>
<th>#</th>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Run button</td>
<td>Executes the report</td>
</tr>
<tr>
<td>2 &amp; 3</td>
<td>Recovery exceptions (two examples shown)</td>
<td>Provides the count of excepted objects and describes the recovery exceptions</td>
</tr>
<tr>
<td>4</td>
<td>View Details button</td>
<td>Lists objects for the recovery exception in the Details tab</td>
</tr>
<tr>
<td>5</td>
<td>Recovery Manager options</td>
<td>Displays the Recovery Manager options</td>
</tr>
</tbody>
</table>

**Related Information**

- “Setting Recovery Management options” on page 215
- “Viewing recovery progress” on page 203
- “Estimating backup time” on page 209
- “Estimating recovery time” on page 212
Refreshing the recoverability information

You should refresh the recoverability information to ensure that the Recoverability report reflects the most recent execution of the ARMBGPV program for the selected recovery point. The Refresh Recoverability option lets you navigate to and submit one or more ARMBGPV jobs to refresh the recoverability data.

*Note*

For more information about the ARMBGPV program, see the *RECOVERY MANAGER for DB2 User Guide*.

To refresh the recoverability information

1. In the Recovery Management perspective, open a Recoverability report.
2. Click **Refresh Recoverability Data**.
3. In the Refresh Recoverability Data panel, navigate to and select a data set containing an ARMBGPV job for an object set.
4. *(optional)* Enter a description of the ARMBGPV job.
5. *(optional)* Click **Add Data Set** and repeat Step 3 on page 208. Continue until you have added all of the recovery JCL to be run.
6. *(optional)* Click **Show JCL** to view the ARMBGPV job JCL in the Scratchpad.
7. *(optional)* In the Scratchpad, edit and save the JCL.
8. Click **Submit** to run the ARMBGPV job.

If you run the Recoverability report, it will reflect the updated recoverability data.

Estimating backup or recovery time

You can create What-If reports to estimate the required time to backup or recover an object set. By using recovery and backup estimation, you can deal with problem objects and fine-tune your recovery strategy prior to running a backup or recovery scenario.
Related Information

- “Estimating backup time” on page 209
- “Estimating recovery time” on page 212

Estimating backup time

BMC Workbench lets you run a Backup Time Estimation report for an object set. This report shows which backup strategies are estimated to fall within the Backup Time Objective (BTO) that you defined. You can create What-If backup time estimations by hypothetically increasing the size or number of objects in the data set and running the report.

To view a Backup Time Estimation report

1. In the Recovery Management perspective, click Add View and select Time Estimation report from the list.

   The Generate Time Estimation wizard opens.

2. On the wizard’s Step 1 - General page, complete the following fields and click Next:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a name for the report.</td>
</tr>
<tr>
<td>Estimate Backup Time</td>
<td>Select Estimate Backup Time.</td>
</tr>
<tr>
<td>BTO Target</td>
<td>Accept the default value or enter the maximum acceptable Backup Time Objective (hh:mm) for this report. To define the default value, see “Setting Recovery Management options” on page 215.</td>
</tr>
<tr>
<td>DB2 Connection</td>
<td>Select a DB2 connection from your list of favorites. If necessary, you can add a connection. See “Connecting to DB2 subsystems” on page 29.</td>
</tr>
<tr>
<td>Object Set</td>
<td>Navigate to and select an object set.</td>
</tr>
</tbody>
</table>

3. On the wizard’s Step 2 - Backup page, accept the default values or enter new values and click Next:
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| Outsize | Specify the size threshold for making copies to an alternate DD or output descriptor. Valid values are:  
- 0 through 4294967295 KB  
- 0 through 4194303 MB  
- 0 through 4095 GB  
**Note:** If you enter an Outsize value greater than 0, the hybrid elapsed time is the combined time of two estimates. |
| Index Size Threshold | Specify the threshold size at which indexes are backed up rather than rebuilt. (valid values are KB 0 - 4294967295, MB 0 - 4194303, GB 0 - 4095) Valid values are:  
- 0 through 4294967295 KB  
- 0 through 4194303 MB  
- 0 through 4095 GB |
| MAXTASKS | Define the default number of multitasking subtasks that can be performed in parallel when you are making copies. Valid values are 1 through 32. |
| Include Indexes | Select to include indexes in the backup estimation, if they are not already in the object set. |
| IO Factor | Specify the I/O factor. The valid values are 1 through 10000, and the default is 100. A factor of 0 will calculate a new factor estimate. |

4 On the wizard’s **Step 3 - Growth** page, accept the default values or enter new values:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of data sets increased by</td>
<td>Enter a percentage or number increase to generate a What-If backup estimation based on an increase in the number of data sets. Valid values are 0 through 32767.</td>
</tr>
<tr>
<td>Size of data sets increased by</td>
<td>Enter a percentage or number increase to generate a What-If backup estimation based on an increase in the size of the data sets. Valid values are 0 through 32767.</td>
</tr>
</tbody>
</table>

5 Click **Finish**.
The General, Backup, and Growth tabs are populated and the Time Estimation Report tab is displayed.

Figure 15: Time Estimation Report

Table 22: Legend

<table>
<thead>
<tr>
<th>#</th>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Run button</td>
<td>Executes the report.</td>
</tr>
</tbody>
</table>
| 2  | Tabs                         | - General (see Step 2 on page 209)  
- Backup (see Step 3 on page 209)  
- Growth (see Step 4 on page 210)  
- Time Estimation Report Tab (see Step 5 on page 204) |
| 3  | Details about the actual objects | Indicates the actual number of objects in your selected group, their actual size, and the applicable I/O factor                           |
| 4  | Details about the projected objects | Projects what the number and size of objects would be, based on the What-If percentage or number changes that you entered in the Growth tab (see Step 4 on page 210) |
| 5  | Recovery Manager options     | Displays the Recovery Manager options  
See “Setting Recovery Management options” on page 215.                                                                                 |
Estimating backup or recovery time

<table>
<thead>
<tr>
<th>#</th>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Elapsed time</td>
<td>Estimates the elapsed time needed for each displayed backup strategy&lt;br&gt;&lt;br&gt;Note: If you enter an Outsize value greater than 0, the hybrid elapsed time is the combined time of two estimates.</td>
</tr>
<tr>
<td>7</td>
<td>Backup strategy</td>
<td>Estimates values for multiple backup strategies</td>
</tr>
<tr>
<td>8</td>
<td>BTO indicator</td>
<td>Indicates whether the backup time estimation falls within your Backup Time Objective (BTO):&lt;br&gt;&lt;br&gt;☑️ indicates that the time is within your defined BTO&lt;br&gt;&lt;br&gt;⚠️ indicates that the time exceeds your BTO</td>
</tr>
</tbody>
</table>

---

**Related Information**

- “Setting Recovery Management options” on page 215
- “Verifying recoverability” on page 205
- “Viewing recovery progress” on page 203

---

**Estimating recovery time**

BMC Workbench lets you run a Recovery Time Estimation report for an object set. The report shows whether either recovery strategy is applicable and shows recovery time estimations for both (if applicable) backout and forward recovery strategies. The report also shows whether the recovery time estimations fall within the Recovery Time Objective (RTO) that you defined.

**To view a Recovery Time Estimation report**

1. In the Recovery Management perspective, click **Add View**.

   The Generate Time Estimation wizard opens.

2. On the wizard’s **Step1 - General** page, complete the following fields, and click **Next**.
### Field | Description
---|---
Name | Enter a name for the report.
Estimate Recovery Time | Select *Estimate Recovery Time*
RTO Target | Accept the default value or enter the maximum acceptable Recovery Time Objective *(hh:mm)* for this report. To define the default value, see “Setting Recovery Management options” on page 215.
DB2 Connection | Select a DB2 connection from your list of favorites. If necessary, you can add a connection. See “Connecting to DB2 subsystems” on page 29.
Object Set | Navigate to and select an object set.

3 On the wizard’s **Step 2 - Recovery** page, accept the default values or enter new values and click **Next**:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recovery point</td>
<td>Accept the current date and time, or enter values.</td>
</tr>
<tr>
<td>Number of Jobs</td>
<td>Specify the number of jobs in the recovery run. Valid values are 1 (the default) through 99.</td>
</tr>
<tr>
<td>MAXLSORT</td>
<td>Specify the maximum number of log sorts that can run concurrently. Valid values are 1 (the default) through 32.</td>
</tr>
<tr>
<td>Rebuild Indexes</td>
<td>Select to rebuild indexes in the recovery estimation.</td>
</tr>
<tr>
<td>Include Indexes</td>
<td>Select to include indexes in the backup estimation if they are not already in the object set.</td>
</tr>
<tr>
<td>IO Factor</td>
<td>Specify the I/O factor. The valid values are 1 through 10000, and the default is 100. A factor of 0 will calculate a new factor estimate.</td>
</tr>
</tbody>
</table>

4 On the wizard’s **Step 3 - Growth** page, accept the default values or enter new values:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of data sets increased by</td>
<td>Enter a percentage or number increase to generate a What-If recovery estimation based on an increase in the number of data sets. Valid values are 0 through 32767.</td>
</tr>
<tr>
<td>Size of data sets increased by</td>
<td>Enter a percentage or number increase to generate a What-If recovery estimation based on an increase in the size of the data sets. Valid values are 0 through 32767.</td>
</tr>
</tbody>
</table>

5 Click **Finish**.
The General, Recovery, and Growth tabs are populated with your input, and the Time Estimation report tab is displayed.

**Figure 16: Recovery Time Estimation report**

![Recovery Time Estimation report]

**Table 23: Legend**

<table>
<thead>
<tr>
<th>#</th>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Run button</td>
<td>Executes the report.</td>
</tr>
<tr>
<td>2</td>
<td>Tabs</td>
<td>Categorizes the information as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- General (see Step 2 on page 209)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Recovery (see Step 3 on page 209)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Growth (see Step 4 on page 210)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Time Estimation Report (see Step 5 on page 204)</td>
</tr>
<tr>
<td>3</td>
<td>Actual data</td>
<td>Indicates the actual number of objects in your selected group, their</td>
</tr>
<tr>
<td></td>
<td></td>
<td>actual size, and the applicable I/O factor</td>
</tr>
<tr>
<td>4</td>
<td>Projected data</td>
<td>Projects what the number and size of objects would be, based on the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>What-If percentage or number changes that you entered in the Growth</td>
</tr>
<tr>
<td></td>
<td></td>
<td>tab (see Step 4 on page 210)</td>
</tr>
</tbody>
</table>
Setting Recovery Management options

You can optionally define global backup and recovery target times. BMC Workbench can use these values to calculate whether performance times will exceed these values.

BMC Workbench uses these values as default values for any new report.

### Table 24: Recovery Management options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backup Time Objectives (BTO) Target</td>
<td>Enter a value (hh:mm).</td>
</tr>
<tr>
<td>Recovery Time Objectives (RTO) Target</td>
<td>Enter a value (hh:mm).</td>
</tr>
</tbody>
</table>
Managing object sets

Object sets are collections of objects that resolve to indexes and table spaces. You can use BMC Workbench to manage object sets.

Object sets are stored in the BMC common repository on each IBM DB2 subsystem, which enables sharing object sets across the following products:

- DASD MANAGER PLUS
- RECOVERY MANAGER for DB2
- NGT Copy
- NGT Recover
- BMC Workbench

**Note**

You can edit the object sets that you create in BMC Workbench, but not the object sets created in other products.

You can use BMC Workbench to:

- Create an object set
- Edit an object set
- Delete an object set
- View the list of DB2 objects (table spaces and indexes) to which an object set resolves
- View a filtered list of object sets

For more information, view the Quick Course "Workbench for DB2 - Managing object sets."
Browsing to an object set

You can browse to an object set either by adding a new object set view or by opening an existing object set view.

For each object set view, you create an object set filter that defines the object sets that are accessible from the object set view. Object set views are saved in the workspace.

Tip

In the DB2 Navigator perspective, you can view the definition of an object set by clicking on the toolbar.

To open an existing object set view

1 Open a workspace.

2 Open the DB2 Navigator perspective and, in the navigation pane, select the DB2 connection that contains the object set.

3 Expand the list of object set views, and select the view whose filter will list the object set you want to see.

   All object sets matching that filter are listed in the Results pane.

4 In the Results pane, navigate to and double-click the object set.

   The object set definition is displayed in the Product Tools perspective.

To add an object set view

1 Open or create a workspace.

2 In the DB2 Navigator perspective, click the Add View menu arrow, and then click Add Object Set View.

   Tip

   You can also right-click in the navigation pane to select this action.
   The default value of the Add View menu is always the last option that you created.

3 In the Add Object Set View dialog box, complete the following fields:
<table>
<thead>
<tr>
<th>Field</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 Connection</td>
<td>Use either of the following methods:</td>
</tr>
<tr>
<td></td>
<td>■ Type the full or partial name of a DB2 connection to select the first subsystem that matches this value in the list of favorite connections.</td>
</tr>
<tr>
<td></td>
<td>■ Use the menu arrow to select from your list of favorite connections.</td>
</tr>
<tr>
<td></td>
<td><strong>Tip:</strong> You can add DB2 connections to your list of favorite connections by clicking and specifying which connections to include.</td>
</tr>
<tr>
<td>Object Set Filter</td>
<td>Enter a filter for an object set name. You can use the * and the % wildcards.</td>
</tr>
</tbody>
</table>

4 Click **OK**.

The object set view is listed under the DB2 connection in the DB2 navigation pane.

All object sets matching that filter are listed in the Results pane.

5 Click **to save the workspace.**

Removing an object set view does not affect any saved object set.

**Related Information**

- “Generating a list of resolved objects” on page 224

---

**Creating and editing object sets**

You can create an entirely new object set or create one based on an existing object set.

In BMC Workbench, you must create an object set within an object set view. The object set is saved to the DB2 subsystem's object set database. The object set is then available to other BMC products.

**Note**

If you cannot see the Product Tools perspective, click **Layout** and add the perspective to the layout. For more information, see “Managing a workspace layout” on page 32.
Tip

In the DB2 Navigator perspective, you can view the definition of an object set by clicking } on the toolbar.

To create a new object set

1. In the Product Tools perspective, click the Add View + menu arrow, and then click Add Object Set View.

   Tip

   You can also right-click in the navigation pane to select this action.
   The default value of the Add View menu is always the last option that you created.

2. In the Add Object Set View dialog box, select Create New Object Set.

3. Select a DB2 Connection, and click OK.

   Use either of the following methods:

   ■ Type the full or partial name of a DB2 connection to select the first subsystem that matches this value in the list of favorite connections.

   ■ Use the menu arrow to select from your list of favorite connections.

   Tip

   You can add DB2 connections to your list of favorite connections by clicking and specifying which connections to include.

4. In the Results pane, complete the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Set</td>
<td>Enter a name for the object set.</td>
</tr>
<tr>
<td>Description</td>
<td>(optional) Enter a description of the object set.</td>
</tr>
</tbody>
</table>

5. To add specifications to the object set, click Add Specification:

   a. Click + and - to toggle between included objects or excluded objects.

   b. Click the Select Object Type menu arrow, and select an object type.
c Enter a filter for the object type, you enter a name or use the * and % wildcards.

d Select how you want partitions included in the object set:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>By space</td>
<td>Include partitioned objects at index space or table space level. All partitions are included.</td>
</tr>
<tr>
<td>By partition</td>
<td>Include partitioned objects by partition. All partitions are included.</td>
</tr>
<tr>
<td>By range</td>
<td>Include a specific range of partitions.</td>
</tr>
</tbody>
</table>

e If you selected **By range**, select a beginning and end of the partition range.

**Note**
If you select **By space** or **By partition**, these fields are dimmed.

f Click **Show Details** to access the Object Set Specification Details dialog box, which displays two tabs:

- The Object List tab which lets you specify objects from the object list for exclusion
  
  1 In the Included Objects pane, select objects for exclusion.

  2 Click **Add**.

  3 Click **OK**.

  An excluded group of objects is created.

- The Related Objects tab which lets you specify related objects for inclusion.

**Tip**
At any time, you can click ✭ to revert to the last saved specification.

6 *(optional)* To reorder the display of the object specifications click the up and down arrows 🕰️ 🕰️.

7 *(optional)* To copy an object specification, click **Copy Specification** 🖉.

8 Click **Save Object Set** 📦.

The object is saved to the DB2 subsystem’s object set database.
To create a new object set by using an existing object set

Tip
You can use this procedure to copy and then make changes to an object set that was defined in another product, for example NGT Recover.

1 In the Product Tools perspective, click the Add View menu arrow, and click Add Object Set View.

Tip
You can also right-click in the DB2 navigation pane to select this action. The default value of the Add View menu is always the last option that you created.

2 In the Add Object Set View dialog box, select Create Like Object Set.

3 Select a DB2 Connection, and click OK.

Use either of the following methods:

- Type the full or partial name of a DB2 connection to select the first subsystem that matches this value in the list of favorite connections.

- Use the menu arrow to select from your list of favorite connections.

Tip
You can add DB2 connections to your list of favorite connections by clicking and specifying which connections to include.

4 Navigate to and select an existing object set.

5 Click OK.

6 In the Object Set field, change the object Set name.

7 Follow the instructions from Step 5 on page 220.

To edit an object set

You can edit only object sets that were created using BMC Workbench.
In the Product Tools perspective, click the Add View menu arrow, and click **Add Object Set View.**

**Tip**
You can also right-click in the DB2 navigation pane to select this action. The default value of the Add View menu is always the last option that you created.

In the **Add Object Set View** dialog box, select **Edit Object Set.**

Select a DB2 Connection, and click **OK.**

Use either of the following methods:

- Type the full or partial name of a DB2 connection to select the first subsystem that matches this value in the list of favorite connections.
- Use the menu arrow to select from your list of favorite connections.

**Tip**
You can add DB2 connections to your list of favorite connections by clicking and specifying which connections to include.

Navigate to and select an existing object set.

Click **OK.**

Follow the instructions from **Step 5 on page 220.**

**Related Information**

- “Deleting an object set” on page 223
- “Generating a list of resolved objects” on page 224

### Deleting an object set

Use the following procedure to delete an object set. You can delete only object sets that were created using BMC Workbench.

1. Open the workspace containing the object set view.
2 Click the Product Tools tab.

3 From the navigation pane, select the object set that you want to delete.

4 From the Results pane toolbar, click **Delete Object Set**.

**Tip**

This button is available only if the object set was created in BMC Workbench. It is dimmed if the object set was created in any other product.

5 When asked to confirm the deletion, click **Yes**.

**Related Information**

- “Creating and editing object sets” on page 219

---

### Generating a list of resolved objects

When you resolve an object set, you can include table spaces and indexes, or just one of them. Use the following procedure to resolve an object set.

The navigation pane lists, under the relevant DB2 connection, all object sets that are created or viewed in BMC Workbench.

**To generate a list of resolved objects**

1 Use one of the following procedures to display an object set definition in the Product Tools perspective:

   - In the DB2 Navigator perspective, select an object set.
     - For more information, see “Browsing to an object set” on page 218.

   - In the Product Tools perspective, create a new object set.
     - For more information, see “Creating and editing object sets” on page 219.

   - In the Product Tools perspective, select an object set from the Navigation pane.

   In the Results pane, the object set definition is displayed in the Specification tab. The object set properties are displayed in the Properties tab.
**Note**

If the object set was not created in Workbench, the details on the Specification tab are dimmed.

2 Click the Resolve button.

3 Select whether you want to include table spaces, indexes or both.

4 Click **OK** to use the default storage group (SYSDEFLT), or optionally use the following procedure to select another storage group to use when resolving the object set:

   a Click **Advanced**.

   b In the Advanced Options dialog box, select the storage group from the list of available storage groups.

   c Click **OK**.

5 On the Resolve dialog box, click **OK**.

The Resolved Objects tab appears, listing the table spaces and indexes resolved from the object set.

---

**Related Information**

- “Creating and editing object sets” on page 219
- “Browsing to an object set” on page 218
Generating a list of resolved objects
Managing JES jobs

You can browse Job Entry Subsystem (JES) jobs running on the LPAR where the UIM server is active.

Viewing JES jobs

Use the following procedure to view JES jobs, job data sets, and job output.

To view JES jobs, job data sets, and job output

1. In the Job Browser perspective, click Add Job Filter.

   **Tip**
   
   You can remove job filters, by clicking Remove a Job Filter.

2. In the Add Job Filter dialog box, enter a job name.
   
   The wildcard * is supported.

3. Click OK.
   
   The application searches for jobs that satisfy the filter value. Matching jobs are listed in the results pane.

4. To view the job output, perform one of the following tasks:
   
   ■ Double-click the required job.
   
   ■ Select the required job and click .

   Two panes are displayed in the output view:
- The upper pane displays the list of data sets.

- The lower pane displays the contents of the data set selected in the upper pane. By default, the job output of the first data set is selected and displayed.

5 (optional) Search for specific values and text strings in the job output.

Guidelines are as follows:

- All standard RegEx values are supported.

- You can use the scroll bars to navigate between the job data sets.

- You can click to reload an active job from the server; the jobs list and job output will be updated accordingly. This button is dimmed when you are viewing details of an inactive job.

6 When finished, click Back to Job List to return to the job list.

---

**Related Information**

- “Editing text files in Scratchpad” on page 70

- “Running a command from Scratchpad” on page 73
Managing data sets

You can navigate to and edit data set members.

The File Locator perspective provides a visual hierarchical display of the data sets and members, similar to that commonly found on personal computers.

Viewing and editing a data set

Use the following procedure to browse to, select, and edit any file on the IBM z/OS system that you are authorized to access.

1. In the File Locator perspective, click Add Data Set Filter.
2. Enter a data set name.
   The wildcard * is supported.
3. Click OK to search for all data sets and data set members (files) that satisfy your filter.
   The matching data sets and data set members are displayed in a hierarchical tree on the navigator pane. Details of the data sets and data set members are displayed in the results pane.
4. In the hierarchical tree, navigate to the data set that contains the files that you require.
5. If you want to open and view the data set, perform one of the following actions in the results pane:
   - Double-click the file.
   - Select the file and click Open.
You can also right-click in the results pane and select 

---

6 *If you want to edit the file*, perform one of the following actions in the results pane:

- Double-click the file.
- Select the file and click **Edit**

---

7 In Scratchpad, view or edit the file, and save any changes.

8 If your data set is archived, respond to the displayed Restore Archive Data dialog box in either of the following ways:

- Click **Yes** to restore the data set.
- Click **No** to bypass restoring.

---

**Related Information**

- “Editing text files in Scratchpad” on page 70
- “Setting Scratchpad options” on page 75
Index

.csv 64

A

access to Workbench functions 27
ACTWBREP 34
ad hoc 92
advanced search 53
ALTER 69
Alter command 67
ARCHIVE 128
ARMBGPV 208

B

Backup Time Objectives 215
Bind command 67
BLOB 147
BMC Workbench options 35
BMC Workbench repository 32
browse 61
browsers
  supported 23
BTO 215

C

case sensitive search 70
CATALOG MANAGER 25
CATALOG MANAGER Options 65
catalog objects tab 101
CCSID option 196
CDL 185, 190
Change Definition Language 185
change the filter options 35
changes to the product 10

CISIZE option 196
CKSEGSIZE parameter, OVERRIDE ALUIN
  keyword 198
CLOB 147, 174
COALESCE function 173
column names 128
command
  ad-hoc 68
  edit 73
  run 73
  statement cache 68
compare
  tab 101
comparing statements 88
comparison 88
connection manager
  refresh button 31
Continue on error 113
conventions, documentation 10
copy 64
  view 60
copy a layout 36
copy a Schema Compare View 185
copy file 70
copy workspace 43
cost 100
create a Schema Compare View 185
Creating a DB2 statement cache view 58
csv format 63

D

data set
  search 229
database
    ALTER 69
date durations 178
DAYOFMONTH function 147
DAYOFWEEK function 147
DAYOFYEAR function 148
DB2 connection
  add 29
  delete 29
  favorite 29
  menu 29
  remove 29
  unsupported version 29
DB2 navigation pane 57
DB2 object view
  create 53
DB2 privileges 23
DB2 statement
  grayed out 78
  selecting 90
  unavailable 78
DBC 23
DBCLOB 147
DCL command 67
DDL command 67
DDL limitations 185
debug messages 33
DECIMAL 179, 181
definition
  perspective 18
  view 18
  workspace 18
degree 104
  set value for 92
delete
  view 59
Delete command 67
delete index. See drop index
Delete previous instances before migration 113
DIGITS function 151
Display command 67
documentation information 9
Drop command 67
drop index 85
drop recovery 75
  unavailable 75
durations
  date 178
  labeled 177
  time 178
  timestamp 178

E
edit

ad-hoc DDL view 70
command output 70
edit commands
  case sensitive search 70
  copy file 70
  find 70
  redo 70
  RegEx 70
  replace 70
  save 70
  save all 70
  search 70
  undo 70
element
  search 229
ENQFAILRC4) parameter, OVERRIDE ALUIN keyword 198
ESCAPE clause 128
European format option 196
Explain diagram
  SQL tab 98
Explain tree
  SQL tab 98
  status information 98
export 63

F
filter options 35
find 70
FKNN parameter, OVERRIDE ALUIN keyword 198
FLOAT 179, 181
folder
  add 47, 48
  create 47
  delete 47
Free command 67
functions
  COALESCE 173
  DAYOFMONTH 147
  DAYOFWEEK 147
  DAYOFYEAR 148
  DIGITS 151
  IFNULL 173
  SOUNDEX 165
  SUBSTR 166
  VALUE 173
  WEEK 174
G
Generated statements 88
GENOBID parameter, OVERRIDE ALUIN keyword 198

H
HDDL command 67
dimmed 67
HDDL command unavailable 67

I
IFNULL function 173
IN predicate 181
INCLUDEPARENTS parameter, OVERRIDE ALUIN keyword 198
index
adding 83
ALTER 69
copy 86
drop 85
edit 87
index name 81
index on expression 83
index statistics 81
Index Term 119–121, 123, 125, 217, 219, 223, 224
Insert command 67
INTEGER 181
IO Factor 212
IS 128
IS NOT 128
IXPADDEDDN parameter, OVERRIDE ALUIN keyword 198
IXPADDEDDY parameter, OVERRIDE ALUIN keyword 198

J
job data sets 227
job output 227

K
KEEP-ALL-PRIMARY parameter, OVERRIDE ALUIN keyword 198
key columns 81

L
labeled durations 177
layout 32
layout options 35
Length of inline LOB columns option 196
LIKE clause 128
LIKE predicate 184
LONGVARCOLS parameter, OVERRIDE ALUIN keyword 198

M
MAXLSORT 212
message rules 100
messages
debug 33
error 33
information 33
warning 33
migrating statistics 107
migration stats 107, 110

N
native procedure
ALTER 69
NO-PRIMARY-CONSTRAINTS parameter, OVERRIDE ALUIN keyword 198
NOAPPEND parameter, OVERRIDE ALUIN keyword 198
NOCLONES parameter, OVERRIDE ALUIN keyword 198
NODSSIZE parameter, OVERRIDE ALUIN keyword 198
NOENFORCED parameter, OVERRIDE ALUIN keyword 198
NOGBPCACHE parameter, OVERRIDE ALUIN keyword 199
<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOINLINELOB parameter, OVERRIDE ALUIN</td>
<td>199</td>
</tr>
<tr>
<td>NOIXCLOSE parameter OVERRIDE ALUIN</td>
<td>199</td>
</tr>
<tr>
<td>NOIXCOMPRESS parameter, OVERRIDE ALUIN</td>
<td>199</td>
</tr>
<tr>
<td>NOIXCOPY parameter, OVERRIDE ALUIN</td>
<td>199</td>
</tr>
<tr>
<td>NOIXPADDLED parameter, OVERRIDE ALUIN</td>
<td>199</td>
</tr>
<tr>
<td>NOLIMITKEYS parameter, OVERRIDE ALUIN</td>
<td>199</td>
</tr>
<tr>
<td>NOLOG parameter, OVERRIDE ALUIN</td>
<td>199</td>
</tr>
<tr>
<td>NOPIECESIZE parameter, OVERRIDE ALUIN</td>
<td>199</td>
</tr>
<tr>
<td>NOSPACTIVE parameter, OVERRIDE ALUIN</td>
<td>199</td>
</tr>
<tr>
<td>NOSPASUTIME parameter, OVERRIDE ALUIN</td>
<td>199</td>
</tr>
<tr>
<td>NOSPCCSID parameter, OVERRIDE ALUIN</td>
<td>199</td>
</tr>
<tr>
<td>NOSPCLASS ID parameter, OVERRIDE ALUIN</td>
<td>199</td>
</tr>
<tr>
<td>NOSPDEEDBMODE parameter, OVERRIDE ALUIN</td>
<td>199</td>
</tr>
<tr>
<td>NOSPDEGREE parameter, OVERRIDE ALUIN</td>
<td>199</td>
</tr>
<tr>
<td>NOSPDETERMINISTIC parameter, OVERRIDE ALUIN</td>
<td>199</td>
</tr>
</tbody>
</table>
NOSPRESULTSET parameter, OVERRIDE ALUIN keyword 201
NOSPROUNDING parameter, OVERRIDE ALUIN keyword 201
NOSPUNOPTS parameter, OVERRIDE ALUIN keyword 201
NOSPSECURITY parameter, OVERRIDE ALUIN keyword 201
NOSPSQLPATH parameter, OVERRIDE ALUIN keyword 201
NOSPSTAYRESIDENT parameter, OVERRIDE ALUIN keyword 201
NOSPTEXT parameter, OVERRIDE ALUIN keyword 201
NOSPTFORMAT parameter, OVERRIDE ALUIN keyword 201
NOSPVALIDATE parameter, OVERRIDE ALUIN keyword 201
NOSPWLNMNEST parameter, OVERRIDE ALUIN keyword 201
NOSTOREDPROCS parameter, OVERRIDE ALUIN keyword 201
NOTBAUDIT parameter, OVERRIDE ALUIN keyword 201
NOTBCOLOORDER parameter, OVERRIDE ALUIN keyword 201
NOTBDEFVALUE parameter, OVERRIDE ALUIN keyword 201
NOTBTSAUTO parameter, OVERRIDE ALUIN keyword 201
NOTSCLOSE parameter
   OVERRIDE ALUIN keyword 201
NOWVQUAL parameter, OVERRIDE ALUIN keyword 201
NULL/NOT NULL predicate 184

supported 23
OVERRIDE 196
OVERRIDE option 196

P
package
   selecting a statement 91
padded values option 196
paste
   view 60
Performance perspective 93
perspective
   definition 18
plan table owner 104
   set value for 92
predicates 100
   IN 181
   LIKE 184
   NULL/NOT NULL 184
primary key constraint 196
product changes 10
publications, related 9

R
Rebind command 67
Recovery Point 205
Recovery Time Objectives 215
redo 70
RegEx 70
related objects
   selecting a table 101
   selecting an object type 101
   tab 101
toolbar 62
   viewing 101
related publications 9
remove file view 70
remove index. See drop index
reordered row format option 196
replace 70
reserved words 184
RTO 215
rule set 104

O
object icons 101
object set view 218
Object Types for HDDL 65
online application 104
options
   BMC Workbench 35
   DB2 Navigator 65
   Scratchpad 75
OS
S
SAF authorization 23
SAF resource 27
save 70
save all workspaces 43
save workspace 43
Schema Compare View
   copy 185
   create 185
schema management
   CCSID option 196
   CISIZE option 196
   European format option 196
   Length of inline LOB columns option 196
   padded values option 196
   primary key constraint option 196
   reordered row format option 196
   white spaces in native SQL stored procedure option 196
scratchpad 70
search 70
   advanced 53
Select command 67
select options 75, 104
select statement 90
server information 33
service unit
   cost per 104
set the default layout 37
SOUNDEX function 165
SPOWNER parameter, OVERRIDE ALUIN
   keyword 202
SPTEXT-TRACE parameter, OVERRIDE ALUIN
   keyword 202
SQL statement
   edit 96
   validation 96
Start command 67
statement selection 90
statistic migration options 113
   Continue on error 113
   Delete previous instances before migration 113
statistics migration 107
stats migration 107, 110
status
   severe 98
   warning 98
status of workspace 39
step type
   index 98
   operator 98
   query block 98
   table 98
stogroup
   ALTER 69
Stop command 67
string delimiter 53
SUBSTR function 166
subsystem
   not supported 53
summary of changes 10
superuser authorization 27
supported browsers 23
supported OS 23
SYNC 115
syntax statement conventions 13
T
tab
   indexes 97
table
   ALTER 69
table qualifier 104
      set value for 92
table space
   ALTER 69
template 48
      create 48
      delete 49
      displaying 46
      managing 46
      remove 49
templates
      move 50
time durations 178
timeron
      cost per 104
timestamp durations 178
Timestamp to use for STATSTIME 113
total cost 98
TSO 25
U

UCNN parameter, OVERRIDE ALUIN keyword 202
UCPKPALL installation option 198
UIM 23
UIM properties 33
undo 70
UNLOAD 128
Update command 67
User Interface Middleware properties 33

V

VALUE function 173
view
copy 60
definition 18
delete 59
paste 60
viewing
job data sets 227
job output 227

W

WEEK function 174

What-If 79
What-If index
dimmed 81
unavailable 81
white spaces in native SQL stored procedure option 196
workflow
SQL tuning 77
workspace
close 45
creating from template 41
creating new 40
definition 18
delete 45
edit 46
open 44
remove 45
saving 43
workspace icon 39

Z

zoom
SQL tab 98