IBM® DB2 Query Patroller™

DB2 Query Patroller Guide: Installation, Administration, and Usage

Version 8.2

GC09-7658-01

IBM® DB2 Query Patroller™

DB2 Query Patroller Guide: Installation, Administration, and Usage

Version 8.2

GC09-7658-01

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

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

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

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

Contents

About this book

This book provides information about how to install, configure and use DB2 Query Patroller Version 8 (Query Patroller) to manage query workloads against a database.

This manual is intended for database administrators, data warehouse support personnel, and other DB2 Universal Database users who are responsible for database administration tasks or who want to understand how to exploit the features of Query Patroller in their organization.

How this book is structured

This book contains information on the following Query Patroller topics:

Getting started with DB2 Query Patroller

- Chapter 1, "DB2 Query Patroller overview," on page 3 presents an overview of Query Patroller and its components.

Installing, setting up, and migrating Query Patroller

- Chapter 2, "Query Patroller installation environment overview," on page 11 explains the different installation environments for Query Patroller.
- Chapter 3, "Installing Query Patroller (UNIX)," on page 15 provides steps for installing Query Patroller on UNIX systems.
- Chapter 4, "Installing Query Patroller with the DB2 Setup wizard (UNIX)," on page 17 includes prerequisites, installation steps, and verification information when the DB2 Setup wizard is used.
- Chapter 5, "Installing Query Patroller manually (UNIX)," on page 43 provides steps for manually installing Query Patroller on UNIX systems. Including detailed installation prerequisites and steps for verifying the installation.
- Chapter 6, "Installing Query Patroller (Windows)," on page 51 provides steps for installing Query Patroller on Windows systems.
- Chapter 7, "Installing Query Patroller with the DB2 Setup wizard (Windows)," on page 53 includes prerequisites and installation steps when the DB2 Setup wizard is used.
- Chapter 8, "Setting up Query Patroller server," on page 77 provides steps for setting up the Query Patroller server after installation.
- Chapter 9, "Migrating Query Patroller," on page 79 provides steps for migrating Query Patroller Version 7 data and settings to Query Patroller Version 8.
- Chapter 10, "Next Steps," on page 83 provides pointers to the steps to take after you have completed the installation, setup, and migration of Query Patroller.

Planning query management to solve business problems

- Chapter 11, "Query Patroller solutions for business problems," on page 89 includes several scenarios that demonstrate how different Query Patroller features are used to address realistic business problems in a fictional organization.
- Chapter 12, "Query Patroller background," on page 95 provides the background concepts involved in configuring and using Query Patroller.

- Chapter 13, “Defining your query management strategy,” on page 107 describes the high-level considerations involved in deciding how to exploit the different features of Query Patroller in your environment.
- Chapter 14, “Configuring Query Patroller to implement your query management strategy,” on page 111 steps you through the decisions necessary to configuring different components of Query Patroller.

Setting up your Query Patroller system

- Chapter 15, “Administering Query Patroller system settings,” on page 123 describes the tasks involved in performing system-level configuration.
- Chapter 16, “Administering query classes,” on page 127 describes the tasks involved in creating and removing query classes.

Managing users

- Chapter 17, “Administering operators,” on page 133 defines the concepts of Query Patroller operators and operator profiles and describes the tasks for administering them.
- Chapter 18, “Administering submitters,” on page 137 defines the concepts of Query Patroller submitters and submitter profiles and describes the tasks for administering them.
- Chapter 19, “Administering query submission preferences,” on page 147 introduces the concept of Query Patroller submission preferences and describes the tasks for administering them.

Using Query Patroller

- Chapter 20, “Managing queries with Query Patroller,” on page 153 describes the tasks involved in administering queries that are managed by Query Patroller.
- Chapter 21, “Using historical analysis,” on page 161 describes the different uses for historical analysis and provides information about how to use the report and graph features to explore historical query activity in your system.
- Chapter 22, “Managing space,” on page 169 provides steps for eliminating old query data and result sets to conserve space on your system.

Query Patroller tuning and troubleshooting

- Chapter 23, “Tuning Query Patroller,” on page 179 provides guidance on how to adjust your configuration settings to address specific performance or resource issues.
- Chapter 25, “Query Patroller, Version 8 limitations and restrictions,” on page 185 describes the known limitations and restrictions of Query Patroller, Version 8.

Appendixes

- Appendix A, “Query Patroller commands,” on page 191 lists the Query Patroller commands.
- Appendix B, “Query Patroller control tables,” on page 263 describes the control tables that Query Patroller requires to process queries.
- Appendix C, “Query Patroller messages,” on page 273 lists the messages that can be returned by Query Patroller. Messages are explained and suggested actions are described.
- Appendix D, “Query Patroller graphical user interface,” on page 291 describes how to perform tasks using the Query Patroller graphical user interface.
- Appendix E, “Submitter tasks,” on page 299 describes how to set your own submission preferences.

Part 1. Getting started with DB2 Query Patroller

Chapter 1. DB2 Query Patroller overview

Query Patroller

DB2® Query Patroller is a powerful query management system that you can use to proactively and dynamically control the flow of queries against your DB2 database in the following key ways:

- Define separate query classes for queries of different sizes to better share system resources among queries and to prevent smaller queries from getting stuck behind larger ones
- Give queries submitted by certain users high priority so that these queries run sooner
- Automatically put large queries on hold so that they can be canceled or scheduled to run during off-peak hours
- Track and cancel runaway queries

The features of Query Patroller allow you to regulate your database's query workload so that small queries and high-priority queries can run promptly and your system resources are used efficiently. In addition, information about completed queries can be collected and analysed to determine trends across queries, heavy users, and frequently used tables and indexes.

Administrators can use Query Patroller to:

- Set resource usage policies at the system level and at the user level
- Actively monitor and manage system usage by canceling or rescheduling queries that could impact database performance
- Generate reports that assist in identifying trends in database usage such as which objects are being accessed, and which individuals or groups of users are the biggest contributors to the workload

Query submitters can use Query Patroller to:

- Monitor the queries they have submitted
- Store query results for future retrieval and reuse, effectively eliminating the need for repetitive query submission
- Set a variety of preferences to customize their query submissions, such as whether to receive e-mail notification when a query completes

Related concepts:

- "Query Patroller components" on page 4
- "Query processing by Query Patroller" on page 95
- "Installation tasks overview" on page 7

Related tasks:

- "Query Patroller administration tasks overview" on page 85

Query Patroller components

DB2® Query Patroller is a client and server solution consisting of the following components:

- Query Patroller server
- Query Patroller Center
- Query Patroller command line support

DB2 Query Patroller™ can be deployed on a system running DB2 Enterprise Server Edition.

Query Patroller server:

When you install Query Patroller server, the following software elements are deployed to the target computer:

Query Patroller stored procedures
Query Patroller stored procedures are called by other Query Patroller components to perform the necessary database tasks.

Control tables
When Query Patroller is set up to manage queries issued against a database, the DB2QP schema, control tables, triggers, functions, and procedures are created within that database. The control tables store all of the information that Query Patroller requires to manage queries. This information includes the following:

- Query Patroller system properties settings
- Query class information
- Submitter information, including query submission preferences
- Operator information
- Managed query properties information
- Historical query properties information
- Query result information
- Historical analysis data
- Scheduled purge job details

For example, the SUBMITTER_PROFILE table contains information such as the submitter's ID, authority level, and the maximum number of queries that the user can have running simultaneously. When the user submits a query, Query Patroller references the SUBMITTER_PROFILE table for these parameters.

Log files
Diagnostic information about errors is recorded in four different Query Patroller log files:

qpsetup.log
Query Patroller writes to qpsetup.log during installation and when the **qpsetup** command is issued. On UNIX® operating systems, qpsetup.log resides in the INSTANCE/db2dump directory, where INSTANCE is the directory where you installed DB2. On Windows, qpsetup.log resides in the directory specified in the *diagpath* database configuration parameter. The qpsetup.log file is intended for use by Query Patroller administrators.

qpmigrate.log

Query Patroller writes to qpmigrate.log during installation and when the **qpmigrate** command is issued. On UNIX operating systems, qpmigrate.log resides in the INSTANCE/db2dump directory, where INSTANCE is the directory where you installed DB2. On Windows, qpmigrate.log resides in the directory specified in the *diagpath* database configuration parameter. The qpmigrate.log file is intended for use by Query Patroller administrators.

qpuser.log

Query Patroller starts writing to the qpuser.log file when the system becomes active. The information written to the qpuser.log file is used for problem determination and is intended for use by Query Patroller administrators. On UNIX operating systems, qpuser.log resides in the INSTANCE/db2dump directory, where INSTANCE is the directory where you installed DB2. On Windows, qpuser.log resides in the directory specified in the *diagpath* database configuration parameter.

qpdiag.log

Query Patroller starts writing to the qpdiag.log file when the system becomes active. The information written to the qpdiag.log file is used for problem determination and is intended for use by DB2 technical support. On UNIX operating systems, qpdiag.log resides in the INSTANCE/db2dump directory, where INSTANCE is the directory where you installed DB2. On Windows, qpdiag.log resides in the directory specified in the *diagpath* database configuration parameter.

Use a text editor to view the log files on the server where Query Patroller is installed. The most recent events are recorded at the end of the file. Generally, each entry contains the following parts:

- A time stamp
- Instance, database, and partition name details
- Process ID (PID) and name, or Thread ID (TID) and name
- The component reporting the error
- A diagnostic message (usually beginning with ″DQP″) explaining the error

The log files grow continuously. Occasionally, they should be backed up then erased. A new log file is generated automatically the next time it is required by Query Patroller.

Query Patroller Center:

The Query Patroller Center is a graphical user interface that allows administrators to manage Query Patroller system properties, users, and queries, and to view historical analysis reports. The Query Patroller Center also allows query submitters to manage their queries, save query results, and customize their query submission preferences.

The look and functionality of the Query Patroller Center varies depending on different factors, such as the authority of the user and whether the DB2 administration tools are also installed.

An administrator has access to the Query Patroller Center’s full functionality. The following list shows some of the tasks that administrators can do with Query Patroller Center:

- Manage the Query Patroller system parameters
- Create, update, or delete profiles for Query Patroller submitters and operators
- Create, update, or delete submission preferences for Query Patroller submitters
- Create, update, or delete query classes
- Monitor and manage queries that have been intercepted by the Query Patroller system
- Generate and analyze reports that display database usage history

A submitter has access to a subset of the Query Patroller Center’s functionality. The following list shows some of the tasks that submitters can do with Query Patroller Center:

- Monitor and manage queries that they have submitted through the Query Patroller system
- Store results of the queries that they have submitted for future retrieval
- Show or file results of the queries that they have submitted
- Create, update, or delete their own query submission preferences

Query Patroller command line support:

Command line support enables Query Patroller administrators and submitters to perform most Query Patroller tasks from the DB2 CLP or from the operating system’s command line prompt. Query Patroller commands can also be combined with shell scripts or languages such as Perl, awk, and REXX.

Related concepts:

- “Query Patroller” on page 3
- “Query processing by Query Patroller” on page 95

Related reference:

- “DB2 Query Patroller control tables” on page 263
- “Query Patroller command line support” on page 191
- Chapter 23, “Tuning Query Patroller,” on page 179

Installation tasks overview

The following chapters describe the steps to take for installing Query Patroller, Version 8, setting up your Query Patroller server, and migrating data and settings from Query Patroller, Version 7.

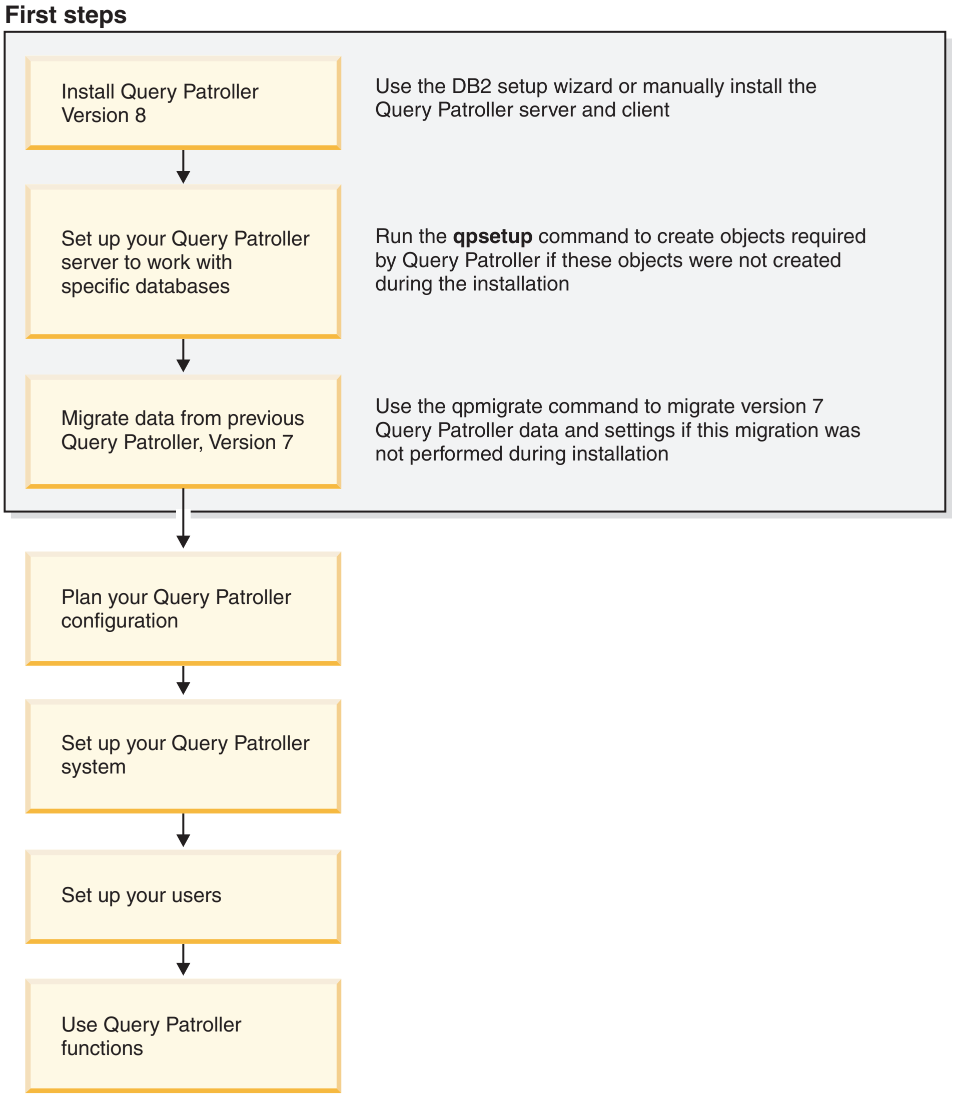

First steps
Install Query Patroller Version 8
Use the DB2 setup wizard or manually install the Query Patroller server and client
Set up your Query Patroller server to work with specific databases
Run the **qpsetup** command to create objects required by Query Patroller if these objects were not created during the installation
Migrate data from previous Query Patroller, Version 7
Use the qpmigrate command to migrate version 7 Query Patroller data and settings if this migration was not performed during installation
Plan your Query Patroller configuration
Set up your Query Patroller system
Set up your users
Use Query Patroller functions

Figure 1. Query Patroller installation tasks overview

Related concepts:

- "Query Patroller" on page 3
- "Query Patroller components" on page 4
- Chapter 2, "Query Patroller installation environment overview," on page 11
- "Automatic migration of Query Patroller server with the DB2 Setup wizard" on page 79

Related tasks:

- "Setting up Query Patroller server manually" on page 77
- "Migrating Query Patroller server manually" on page 80

- “Query Patroller administration tasks overview” on page 85
- “Enabling Query Patroller to intercept queries” on page 83
- Chapter 3, “Installing Query Patroller (UNIX),” on page 15
- Chapter 6, “Installing Query Patroller (Windows),” on page 51

Part 2. Installing, setting up, and migrating Query Patroller

Chapter 2. Query Patroller installation environment overview

This topic provides an overview of two typical Query Patroller installation environments. After reading this topic you should understand what components make up Query Patroller, and understand the nonpartitioned and partitioned environments in which they can operate.

A typical Query Patroller installation environment consists of the following computers:

- Query Patroller server installed on all partitions in your DB2® Enterprise Server Edition environment.
- One or more system administration workstations with Query Patroller client tools installed on them.
- One or more DB2 clients with Query Patroller client tools installed on them.
- One or more DB2 clients without the Query Patroller client tools installed on them.
- One or more Business Intelligence clients without the Query Patroller client tools installed on them.

Query Patroller server and client tools

Query Patroller server:

The Query Patroller server is installed on all of the computers that make up your DB2 Universal Database™ Enterprise Server Edition (Version 8.1.2 or later) environment. It is installed in either a nonpartitioned or a partitioned environment. The Query Patroller server accepts, analyzes, prioritizes, and schedules queries that run against your database. Query Patroller server can also notify users when their queries are completed.

Query Patroller client tools (system administration workstation):

You can install the Query Patroller client tools on the DB2 client that will function as the system administration workstation in your environment. One or more of your DB2 clients can function as a system administration workstation. You should install these tools on any machine that will perform remote administration of the data warehouse.

The Query Patroller client tools that get installed on your system administration client are the Query Patroller Center and the Query Patroller command line support. These tools enable you to configure and manage your Query Patroller server, to create and delete user profiles, to manage queries and result destinations, and to monitor the usage history of a database.

Query Patroller client tools (DB2 clients):

You can install the Query Patroller client tools on your DB2 clients. You can install either the Query Patroller Center or the Query Patroller command line support (or both). In order to install the Query Patroller Center on your clients they must all have either a DB2 client product installed on them (for example, a DB2 Run-Time client or a DB2 Application Development client), any DB2 Connect™ product (for example, DB2 Connect Personal Edition or DB2 Connect Enterprise Edition), or any DB2 server product (for example, DB2 Enterprise Server Edition or DB2 Workgroup Server Edition). All of these products must be Version 8.1.2 or later. You may also have your DB2 clients that do not have the Query Patroller Center or the Query Patroller command line support installed on them, submit queries to your Query Patroller server.

Note: You may also have a number of Business Intelligence clients with third-party GUI tools (such as Business Objects) that can submit queries. These clients first submit their queries to a dedicated Business Intelligence server which, in turn, directs the queries to the Query Patroller server for processing.

Query Patroller typical environments

DB2 Enterprise Server Edition (ESE) nonpartitioned environment:

You can install Query Patroller in a DB2 Enterprise Server Edition nonpartitioned environment. For example, a typical nonpartitioned environment consists of the following computers:

- One Query Patroller server with DB2 Enterprise Server Edition (Version 8.1.2 or later) installed on it.
- One system administration workstation with Query Patroller client tools installed on it.
- Two DB2 Run-Time clients (UNIX and Windows).

Figure 1 illustrates how these computers are typically set up in a DB2 Enterprise Server Edition nonpartitioned environment.

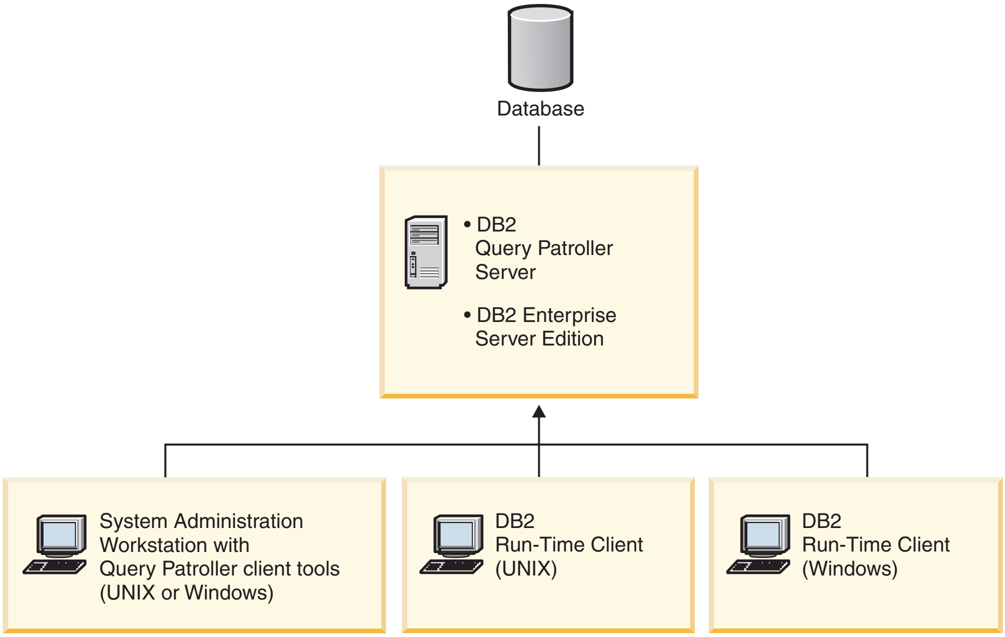

Database
• DB2 Query Patroller Server
• DB2 Enterprise Server Edition
System Administration Workstation with Query Patroller client tools (UNIX or Windows)
DB2 Run-Time Client (UNIX)
DB2 Run-Time Client (Windows)

Figure 2. Query Patroller nonpartitioned installation environment

In a typical nonpartitioned database environment, the Query Patroller server is installed on the computer that is running DB2 Enterprise Server Edition (Version 8.1.2 or later).

DB2 Enterprise Server Edition partitioned environment:

You can install Query Patroller in a DB2 Enterprise Server Edition partitioned environment. For example, a typical partitioned environment consists of the following computers:

- Query Patroller server installed on all partitions in your DB2 Enterprise Server Edition environment (Version 8.1.2 or later).
- One Business Intelligence server for receiving and directing queries from Business Intelligence clients.
- One system administration workstation with Query Patroller client tools installed on it.
- Two DB2 clients with Query Patroller client tools installed on them.
- Two DB2 Run-Time clients (UNIX and Windows).

Figure 2 illustrates how these computers are typically set up in a DB2 Enterprise Server Edition partitioned environment.

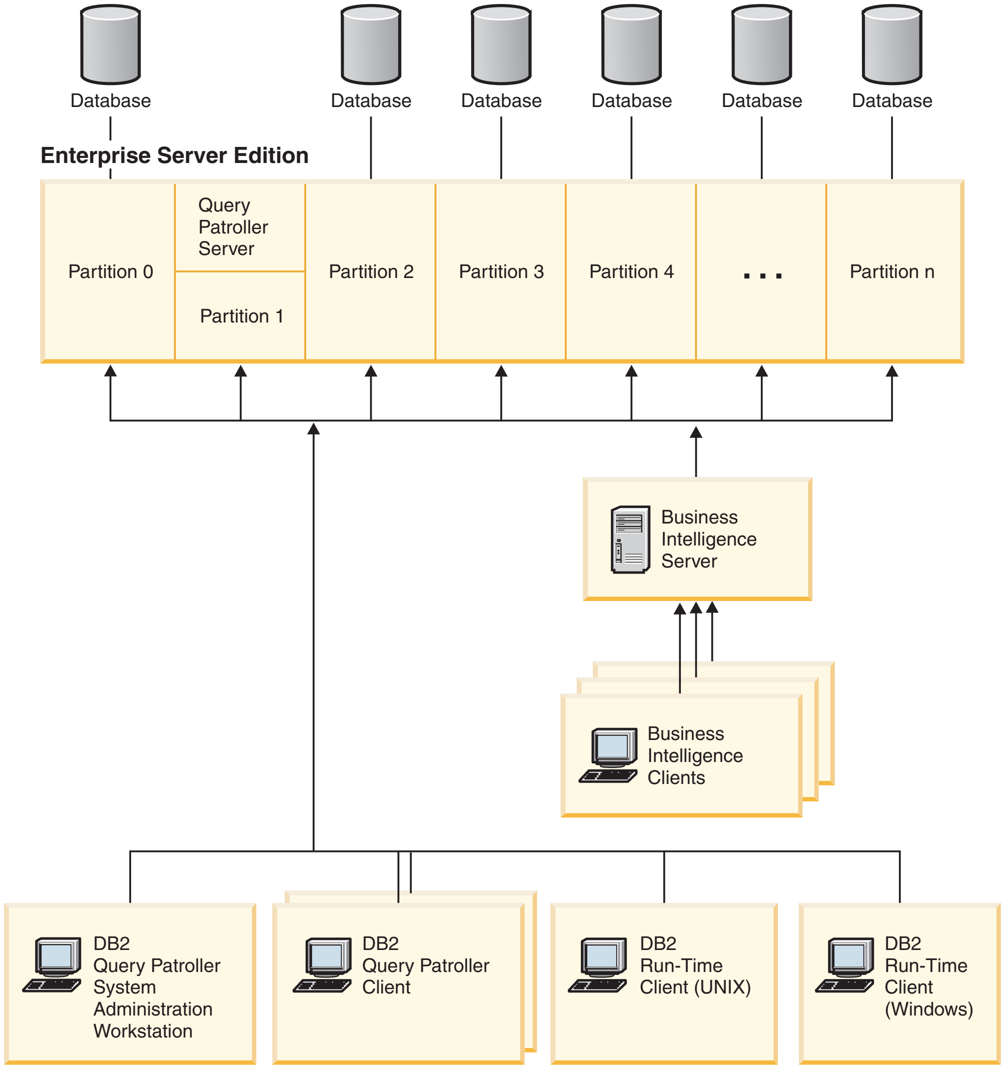

Database
Database
Database
Database
Database
Database
Enterprise Server Edition
Partition 0
Query Patroller Server
Partition 1
Partition 2
Partition 3
Partition 4
. . .
Partition n
Business Intelligence Server
Business Intelligence Clients
DB2 Query Patroller System Administration Workstation
DB2 Query Patroller Client
DB2 Run-Time Client (UNIX)
DB2 Run-Time Client (Windows)

Figure 3. Query Patroller partitioned installation environment

In a typical partitioned database environment, the Query Patroller server is installed on all of the computers running DB2 Enterprise Server Edition (Version 8.1.2 or later). This will enable all of your DB2 clients (with the Query Patroller client tools installed on them) to connect to any of the coordinating partitions in your environment.

You are now ready to install Query Patroller on either a UNIX-based operating system environment, or on a Windows® environment.

Related concepts:

- "Query Patroller configuration roadmap" on page 109
- "Scenario: Managing query submitter needs by configuring submitter profiles" on page 89
- "Scenario: Running large emergency queries" on page 91
- "Scenario: Handling very large queries" on page 90
- "Scenario: Managing queries of different sizes using query classes" on page 92
- "Query Patroller" on page 3
- "Query Patroller components" on page 4

Related tasks:

- "Installing Query Patroller server using the DB2 Setup wizard (Windows)" on page 55
- "Installing Query Patroller server using the DB2 Setup wizard (UNIX)" on page 21
- "Installing the Query Patroller client tools using the DB2 Setup wizard (UNIX)" on page 34
- "Installing the Query Patroller client tools using the DB2 Setup Wizard (Windows)" on page 69

Chapter 3. Installing Query Patroller (UNIX)

If you know which Query Patroller components and tools you want to install, and the kind of environment you will be installing on, you can follow the steps below to install Query Patroller on your UNIX-based server and your UNIX-based clients.

Prerequisites:

Before you install Query Patroller you must:

- Ensure that you know the kind of environment in which you are going to install Query Patroller.
- Verify that the computer on which you will install Query Patroller server meets the necessary installation, memory, and disk requirements.
- Verify that the computers on which you will install the Query Patroller client tools meet the necessary installation, memory, and disk requirements.
- Ensure that the DYN_QUERY_MGMT configuration parameter is set to DISABLE before you install Query Patroller. This parameter must be set to ENABLE for Query Patroller to intercept and manage queries, but it must be set to DISABLE prior to installation to prevent the interception of any internal queries run by the installer.

Procedure:

To install Query Patroller:

1. Install Query Patroller server with the DB2 Setup wizard (UNIX).
2. *Optional*: Manually install Query Patroller server (UNIX)
3. Verify the installation of Query Patroller server.
4. Install the Query Patroller client tools with the DB2 Setup wizard (UNIX).
5. Install the DB2 Information Center (UNIX)
6. *Optional*: Manually install the Query Patroller client tools (UNIX)

Related tasks:

- "Installing Query Patroller server using the DB2 Setup wizard (UNIX)" on page 21
- "Installing the Query Patroller client tools using the DB2 Setup wizard (UNIX)" on page 34
- "Verifying the installation of Query Patroller server" on page 29
- "Installing the DB2 Information Center using the DB2 Setup wizard (UNIX)" on page 307

Chapter 4. Installing Query Patroller with the DB2 Setup wizard (UNIX)

Installing the Query Patroller server with the DB2 Setup wizard (UNIX)

Installation requirements for Query Patroller server (UNIX)

The hardware, operating system, software, and communication requirements for Query Patroller server on a UNIX-based operating system are as follows:

Hardware requirements

One of the following, depending upon your UNIX-based operating system:

- IBM RISC/6000 (AIX)
- eServer pSeries (AIX)
- HP 9000 series 700 or 800 system (HP-UX)
- Intel 32–bit (Linux)
- Intel 64–bit (Linux)
- AMD 64–bit (Linux)
- S/390 9672 Generation 5 or later, Multiprise 3000 or eServer zSeries (Linux)
- IBM eServer iSeries that support Linux (Linux)
- IBM eServer pSeries that support Linux (Linux)
- Solaris UltraSPARC-based computer (Solaris Operating Environment)

Operating system requirements

AIX

Version 4.3.3 (32-bit)

- With maintenance level 9 and APARs IY22308, and IY32690 ; or
- With maintenance level 10 or later and APAR IY32690 (ML-10 includes IY22308); or
- With maintenance level 11 with no APARs required

Version 5.1.0 (32-bit)

- With maintenance level 2 or later and APARs IY31254, IY32217, IY32905, IY29345 and IY31080; or
- With maintenance level 3 or later

Version 5.1.0 (64-bit)

- With maintenance level 2 or later and APARs IY31254, IY32217, IY32905, IY33023, IY29345, IY32466 and IY31080; or
- With maintenance level 3 or later

Version 5.2.0 (32-bit and 64-bit)

Currently has no APARs requirements.

Note: You can query your system to see if a particular APAR is installed with the **instfix -v -i -k** <*APAR*> command. For example, `instfix -v -i -k IY31254`.

The following AIX file sets are required to install or run Query Patroller in languages other than English:

- X11.fnt.ucs.ttf (AIX Windows Unicode TrueType Fonts)
- xlC.rte 5.0.2.x
- For Asian languages, the following file sets are also required:
  - X11.fnt.ucs.ttf_CN (for zh_CN or Zh_CN)
  - X11.fnt.ucs.ttf_KR (for ko_KR)
  - X11.fnt.ucs.ttf_TW (for zh_TW or Zh_TW)
- On AIX Version 4.3.3 the following file sets are required:
  - xlC.aix43.rte 5.0.2.x
  - OpenGL.OpenGL_X.rte.base 4.3.3.76
  - OpenGL.OpenGL_X.rte.soft 4.3.3.75
  - X11.adt.lib 4.3.3.10
- On AIX Version 5.x the following file set is required:
  - xlC.aix50.rte 5.0.2.x

AIX file sets can be downloaded from:
http://techsupport.services.ibm.com/server/fixes

Linux

- For the latest information on supported Linux distribution and kernel levels, point your browser to http://www.ibm.com/db2/linux/validate

HP-UX

- Query Patroller can run on HP-UX 11i (11.11) for systems with PA-RISC 2.x (PA-8x00) processors with:
  - December 2001 GOLDBASE11i bundle (HP-UX)
  - December 2001 GOLDAPPS11i bundle (HP-UX)
  - patch PHSS_26560, PHKL_28489, and PHCO_27434 (HP-UX)
  - Patches for Java SDK 1.3.1. See http://www.hp.com/products1/unix/java/patches/index.html for required patches.

Solaris Operating Environment

- Solaris 7 (32-bit) ″Recommended & Security Patches″ + 107226-17 + 107153-01 + 106327-10
- Solaris 7 (64-bit) ″Recommended & Security Patches″ + 107226-17 + 107153-01 + 106300-11 + 106327-10
- Solaris 8 (32-bit) ″Recommended & Security Patches″ + 108921-12 + 108940-24 + 108434-03 and 108528-12
- Solaris 8 (64-bit) ″Recommended & Security Patches″ + 108921-12 + 108940-24 + 108435-03 + 108434-03 and 108528-12
- Solaris 9 (32-bit)
- Solaris 9 (64-bit)

″Recommended & Security Patches″ can be obtained from the http://sunsolve.sun.com Web site. From the SunSolve Online Web site, click on the ″Patches″ menu item in the left panel and select ″Recommended & Security Patches″ from the ″Browse & Download Patches″ section.

The J2SE Solaris Patch Clusters are also required. They can be obtained from the http://sunsolve.sun.com Web site. From the SunSolve Online

Web site, click on the ″Patches″ menu item in the left panel and select ″Recommended Patch Clusters″ from the ″Browse & Download Patches″ section.

The SUNWlibC software is required to install Query Patroller on Solaris.

For Query Patroller on 64-bit Fujitsu PRIMEPOWER systems, you require the following:

- Solaris 8 Kernel Update Patch 108528-16 or later to get the fix for patch 912040-01.
- Solaris 9 Kernel Update Patch 112233-01 or later to get the fix for patch 912041-01.

The Fujitsu PRIMEPOWER patches for the Solaris Operating Environment can be downloaded from FTSI at: http://download.ftsi.fujitsu.com/.

Software requirements

- To install Query Patroller server, you must have installed either one of the following:
  - DB2 Enterprise Server Edition Version 8.1.2 and FixPak 2+
  - DB2 Enterprise Server Edition that is later than Version 8.1.2
- For 32-bit environments you require the IBM Software Development Kit (SDK) Version 1.3.1 or Version 1.4.1 to run Query Patroller server, the Query Patroller Java-based tools (such as the Query Patroller Center), and to create and run Java applications, including stored procedures and user-defined functions. During the installation process, if the correct level of the IBM Software Developer’s Kit for Java (SDK) is not already installed, the SDK will be installed if it is required.
- For 64-bit environments you require the IBM Software Development Kit (SDK) Version 1.4.1 to run Query Patroller server, the Query Patroller Java-based tools (such as the Query Patroller Center), and to create and run Java applications, including stored procedures and user-defined functions. During the installation process, if the correct level of the IBM Software Developer’s Kit for Java (SDK) is not already installed, the SDK will be installed if it is required. For LinuxIA64 you require the IBM Software Developer’s Kit for Java (SDK) Version 1.3.1. For the latest DB2 Java application development updates, visit the Web page at: http://www.ibm.com/software/data/db2/udb/ad/v8/java.
- Netscape 6.2 or Microsoft Internet Explorer 5.5 (or later) is required to view the online installation help.

Communication requirements

- TCP/IP

Related concepts:

- Chapter 2, “Query Patroller installation environment overview,” on page 11

Related tasks:

- “Installing Query Patroller server using the DB2 Setup wizard (UNIX)” on page 21
- Chapter 3, “Installing Query Patroller (UNIX),” on page 15

Related reference:

- “Memory requirements for Query Patroller server (UNIX)” on page 20

- “Disk requirements for Query Patroller server (UNIX)” on page 20

Memory requirements for Query Patroller server (UNIX)

At a minimum, Query Patroller server requires 256 MB of RAM. Additional memory may be required.

When determining memory requirements, remember:
- Additional memory may be required for non-DB2 software that is running on your system.
- Additional memory is required to support DB2 clients that have the Query Patroller client tools installed on them.
- Specific performance requirements may determine the amount of memory needed.
- Memory requirements are affected by the size and complexity of your database system.

Related concepts:
- Chapter 2, “Query Patroller installation environment overview,” on page 11

Related tasks:
- “Installing Query Patroller server using the DB2 Setup wizard (UNIX)” on page 21
- Chapter 3, “Installing Query Patroller (UNIX),” on page 15

Related reference:
- “Disk requirements for Query Patroller server (UNIX)” on page 20
- “Installation requirements for Query Patroller server (UNIX)” on page 17

Disk requirements for Query Patroller server (UNIX)

The disk space required for Query Patroller server (UNIX) depends on the type of installation you choose. When you install Query Patroller server using the DB2 Setup wizard, size estimates are dynamically provided by the installation program based on installation type and component selection. Disk space is needed for the following:
- To store the product code.
- To store data that will be generated when using Query Patroller (for example, the contents of the control tables).

Remember to include disk space allowance for required software and communication products.

Insufficient disk space management:

If the space required to install the selected components exceeds the space found in the path you specify for installing the components, the DB2 Setup wizard issues an error about the insufficient space. If you choose, you can continue the installation. However, if the space for the files being installed is, in fact, insufficient, then the Query Patroller server installation stops when there is no more space. When this occurs, the installation is rolled back. You will then see a final dialog with the appropriate error messages. You can then exit the installation.

Note: This only applies to installations on HP-UX, Linux, and Solaris operating environments.

Related concepts:

- Chapter 2, “Query Patroller installation environment overview,” on page 11

Related tasks:

- “Installing Query Patroller server using the DB2 Setup wizard (UNIX)” on page 21
- Chapter 3, “Installing Query Patroller (UNIX),” on page 15

Related reference:

- “Memory requirements for Query Patroller server (UNIX)” on page 20
- “Installation requirements for Query Patroller server (UNIX)” on page 17

Installing Query Patroller server using the DB2 Setup wizard (UNIX)

This task is part of the larger task of *Installing Query Patroller (UNIX)*.

This task outlines the steps for installing Query Patroller server on a UNIX-based operating system using the DB2 Setup wizard.

Prerequisites:

Ensure that your computer meets the following requirements:

- Installation requirements for Query Patroller server (UNIX).
- Memory requirements for Query Patroller server (UNIX).
- Disk requirements for Query Patroller server (UNIX).
- The instance running on your DB2 server is stopped. To stop an instance, log on to the system as a user with SYSADM authority and issue the **db2stop** command at a UNIX shell prompt.
- A database whose queries you want to manage using the Query Patroller server.
- All of the languages that the clients will use when submitting queries to Query Patroller server must be installed on the server in order to eliminate any server to client dependencies.

Procedure:

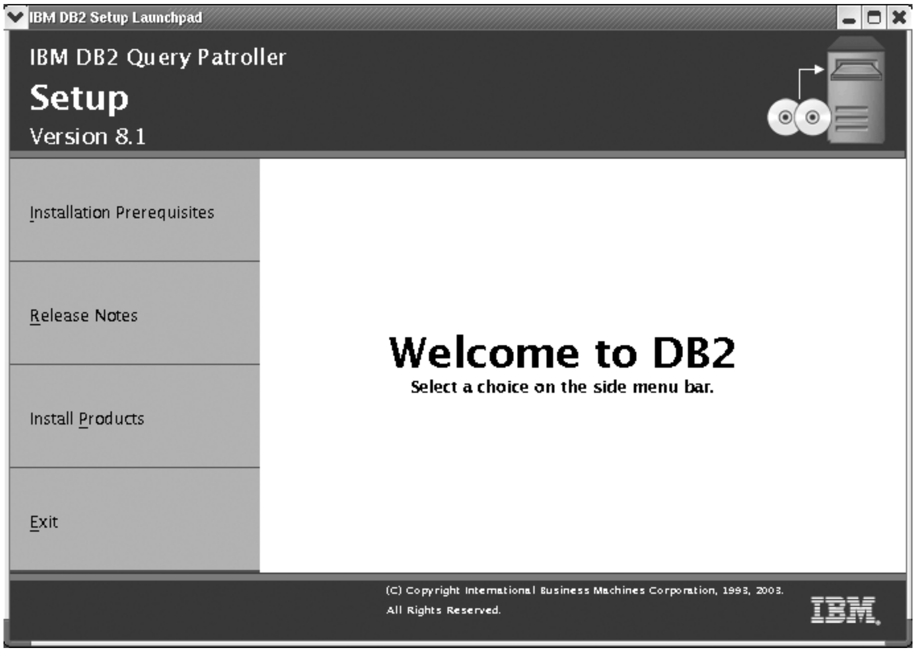
To install Query Patroller server (UNIX):

1. Log on to the system as a user with root authority.
2. Shut down any other programs so that the DB2 Setup wizard can update files as required.
3. Check the CD-ROM label to ensure that you are using the CD-ROM with the appropriate language.
4. Mount the Query Patroller product CD-ROM.
5. Change to the directory where the CD-ROM is mounted by entering the following command:

   ```
   cd /cdrom
   ```

   where */cdrom* represents the mount point of the CD-ROM.

6. Enter the **./db2setup** command to start the DB2 Setup wizard. The IBM DB2 Setup Launchpad window opens.
7. Choose **Install Products** from the DB2 Launchpad.



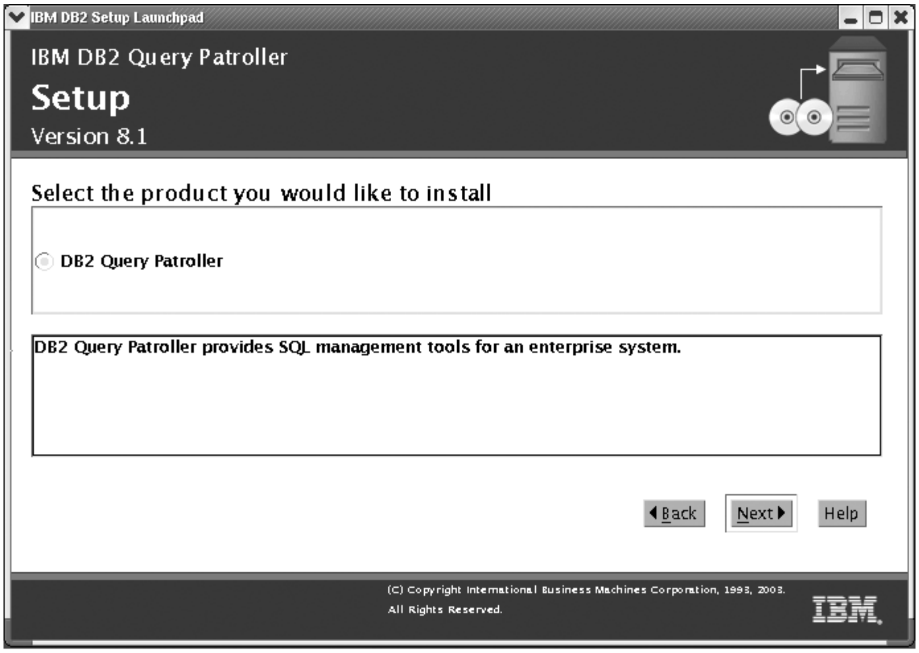
Welcome to DB2

Select a choice on the side menu bar.

(C) Copyright International Business Machines Corporation, 1993, 2003. All Rights Reserved.

Figure 4. Query Patroller welcome screen (UNIX)

8. On the **Select the product you would like to install** window, click **Next**.



Select the product you would like to install

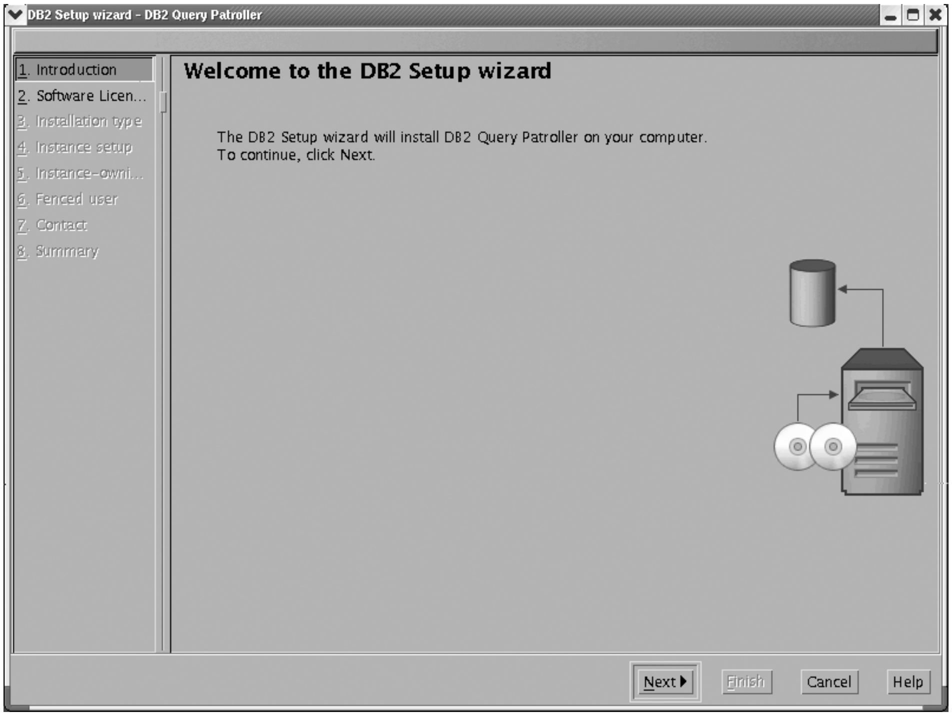
○ DB2 Query Patroller

DB2 Query Patroller provides SQL management tools for an enterprise system.

(C) Copyright International Business Machines Corporation, 1993, 2003. All Rights Reserved.

Figure 5. Query Patroller product screen (UNIX)

9. On the **Welcome to the DB2 Setup wizard** window, click **Next**.

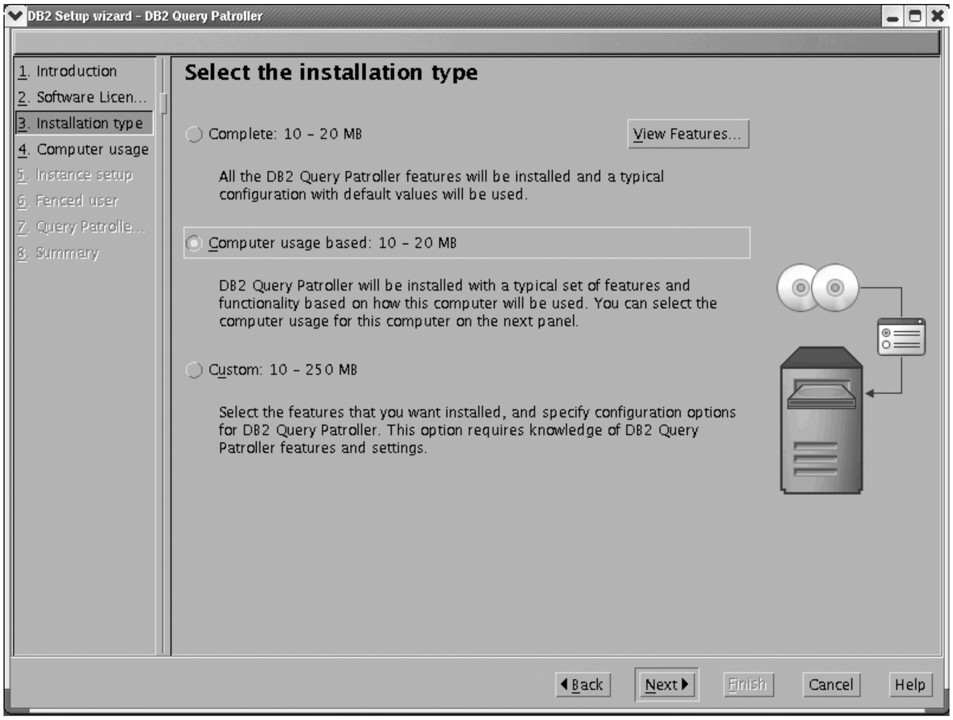


Welcome to the DB2 Setup wizard

The DB2 Setup wizard will install DB2 Query Patroller on your computer.
To continue, click Next.

Figure 6. Query Patroller DB2 Setup wizard welcome (UNIX)

10. On the **Software License Agreement** window, select **Accept** if you accept the terms. Then click **Next**. If you do not accept the terms, then select **Decline**. Then click **Cancel** to end the installation.
11. On the **Select the installation type** window, select **Computer usage based** and click **Next**.



Select the installation type

○ Complete: 10 - 20 MB

All the DB2 Query Patroller features will be installed and a typical configuration with default values will be used.

○ Computer usage based: 10 - 20 MB

DB2 Query Patroller will be installed with a typical set of features and functionality based on how this computer will be used. You can select the computer usage for this computer on the next panel.

○ Custom: 10 - 250 MB

Select the features that you want installed, and specify configuration options for DB2 Query Patroller. This option requires knowledge of DB2 Query Patroller features and settings.

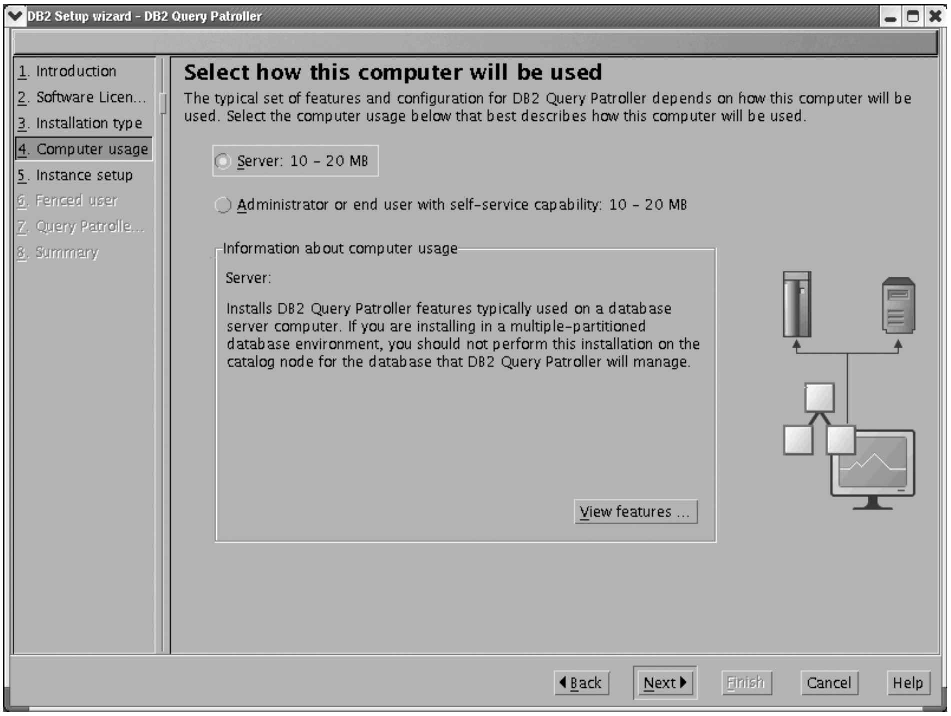
Figure 7. Query Patroller select installation type (UNIX)

12. On the **Select how this computer will be used** window, select **Server** and click **Next**

    Note:

    > If you select **Computer usage based** from the **Select the installation type** window using the DB2 Setup wizard, and you select **Server** from the **Select how this computer will be used** window, then the Query Patroller Center will not be installed. If you want to install the Query Patroller Center, then select either the **Complete** or the **Custom** installation types from the **Select the installation type** window.

.



Select how this computer will be used

The typical set of features and configuration for DB2 Query Patroller depends on how this computer will be used. Select the computer usage below that best describes how this computer will be used.

● Server: 10 – 20 MB

○ Administrator or end user with self-service capability: 10 – 20 MB

Information about computer usage

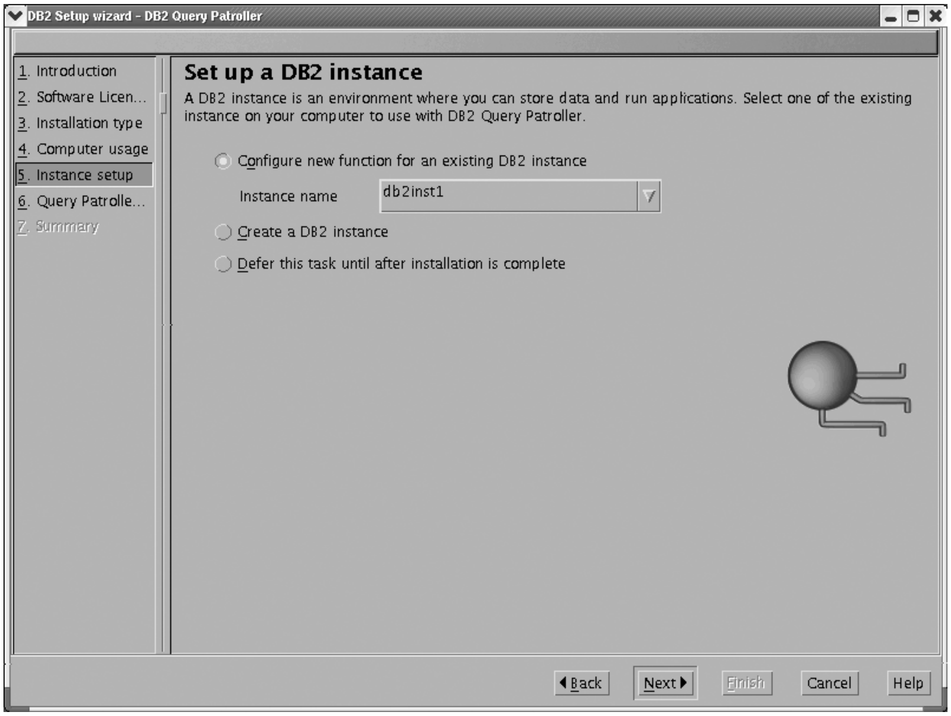
Server:

Installs DB2 Query Patroller features typically used on a database server computer. If you are installing in a multiple-partitioned database environment, you should not perform this installation on the catalog node for the database that DB2 Query Patroller will manage.

Figure 8. Query Patroller computer use (UNIX)

13. On the **Set up a DB2 instance** window select whether you want to configure an existing DB2 instance, or whether you want to defer this task until after the installation is complete. Select the instance you want to configure from

Instance name. Once you have made your selection, click **Next**.



Set up a DB2 instance

A DB2 instance is an environment where you can store data and run applications. Select one of the existing instance on your computer to use with DB2 Query Patroller.

● Configure new function for an existing DB2 instance

Instance name: db2inst1

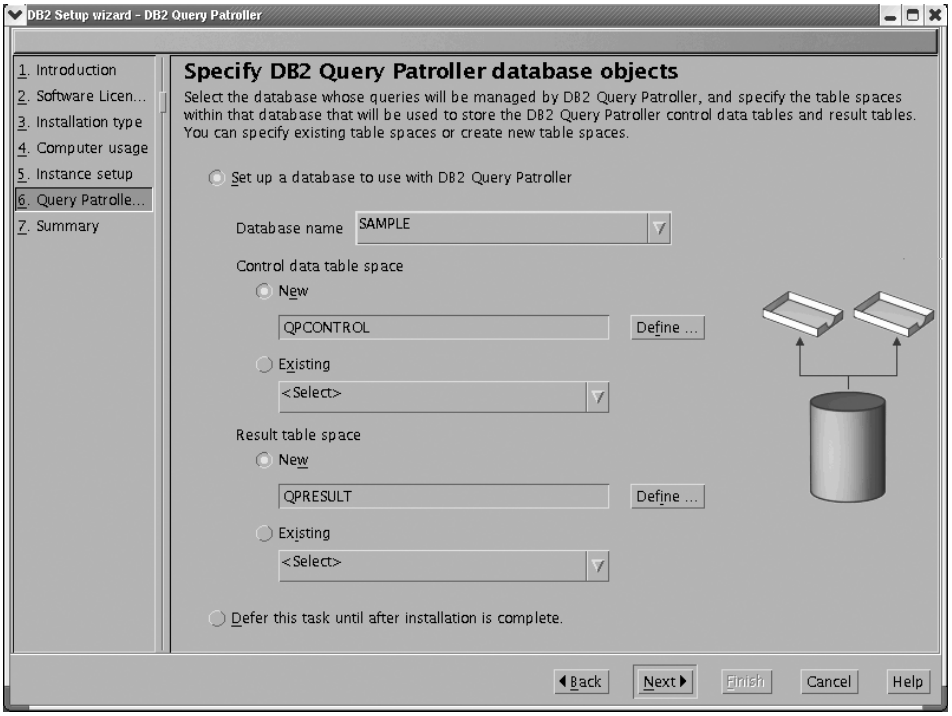
○ Create a DB2 instance

○ Defer this task until after installation is complete

Figure 9. Query Patroller set up DB2 instance (UNIX)

14. On the **Specify DB2 Query Patroller database objects** window, select the database whose queries will be managed by DB2 Query Patroller. Select **Set up a database to use with DB2 Query Patroller**, then select a database from **Database name**. Then select whether you want to create a new or existing control data table space, or whether you want to create a new or existing result table space. Click **New** and click **Define** if you want to create either a new control data table space or a new result table space. If you want to use existing table spaces, then select **Existing** and choose the table spaces from the menu. Once you have selected your table spaces click **Next**.

    Note: If the database you select is a Version 7 Query Patroller database, then you will be prompted to specify whether you want to migrate the Version 7 data and settings to the Version 8 Query Patroller system. If you choose not to migrate during the installation, you can migrate after the installation using the **qpmigrate** command.



Specify DB2 Query Patroller database objects

Select the database whose queries will be managed by DB2 Query Patroller, and specify the table spaces within that database that will be used to store the DB2 Query Patroller control data tables and result tables. You can specify existing table spaces or create new table spaces.

● Set up a database to use with DB2 Query Patroller

Database name: SAMPLE

Control data table space

● New: QPCONTROL

○ Existing

<Select>

Result table space

● New: QPRESULT

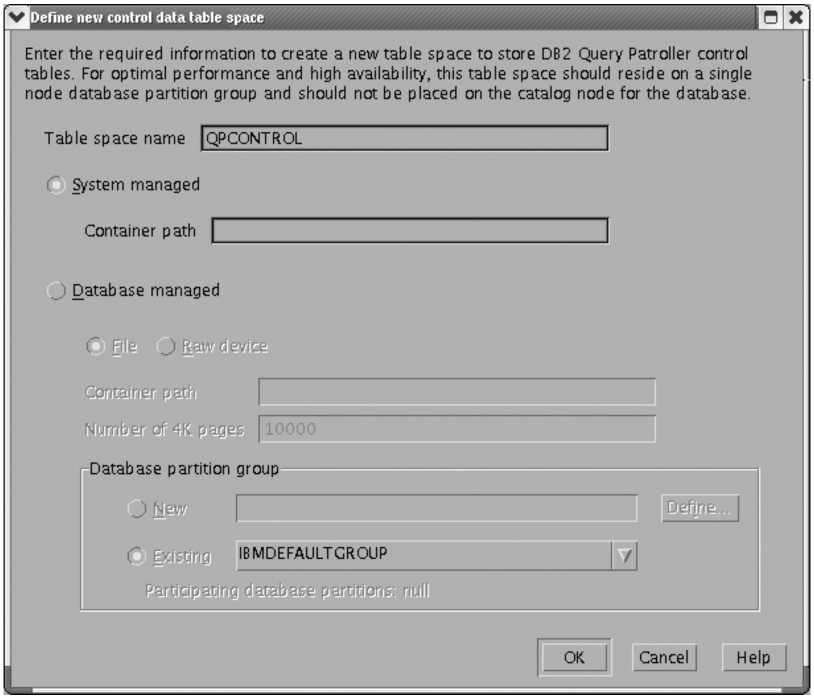
○ Existing: <Select>

○ Defer this task until after installation is complete.

Figure 10. Query Patroller specify database objects (UNIX)

15. If you select to create a new control data table space then the **Define new control data table space** window appears. When you are finished filling in the values in this window, click **OK**. The **Specify DB2 Query Patroller database objects** window will appear again with the name of the new control data table space in the text field. Once you have selected your table spaces click

Next.



Define new control data table space

Enter the required information to create a new table space to store DB2 Query Patroller control tables. For optimal performance and high availability, this table space should reside on a single node database partition group and should not be placed on the catalog node for the database.

Table space name: QPCONTROL

● System managed

Container path:

○ Database managed

● File ○ Raw device

Container path:

Number of 4K pages: 10000

Database partition group

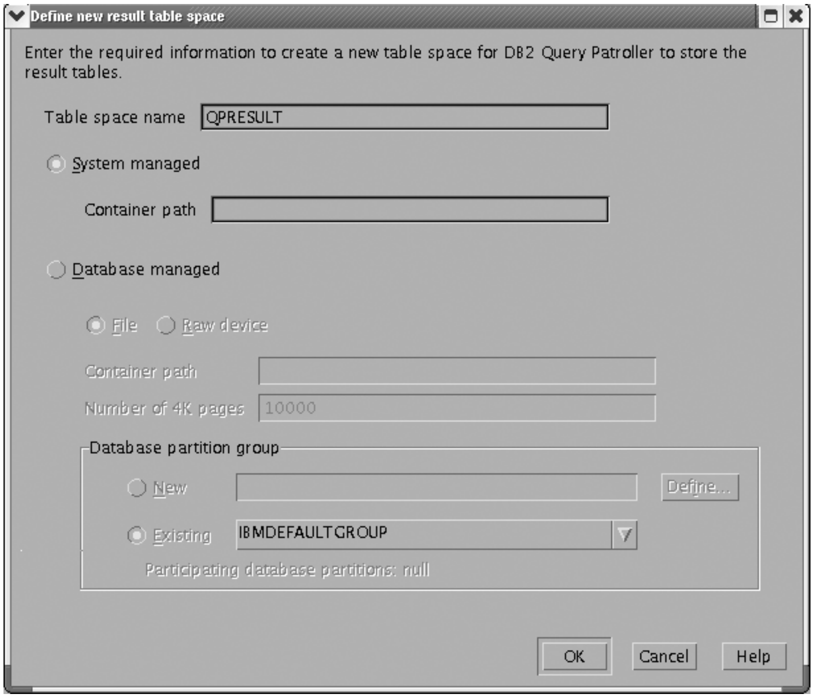
○ New

● Existing: IBMDEFAULTGROUP

Participating database partitions: null

Figure 11. Query Patroller define new control data table space (UNIX)

16. If you select to create a new result table space then the **Define new result table space** window appears. When you are finished filling in the values in this window, click **OK**. The **Specify DB2 Query Patroller database objects** window will appear again with the name of the new result table space in the text field. Once you have selected your table spaces click **Next**.



Define new result table space

Enter the required information to create a new table space for DB2 Query Patroller to store the result tables.

Table space name: QPRESULT

● System managed

Container path:

○ Database managed

● File ○ Raw device

Container path:

Number of 4K pages: 10000

Database partition group

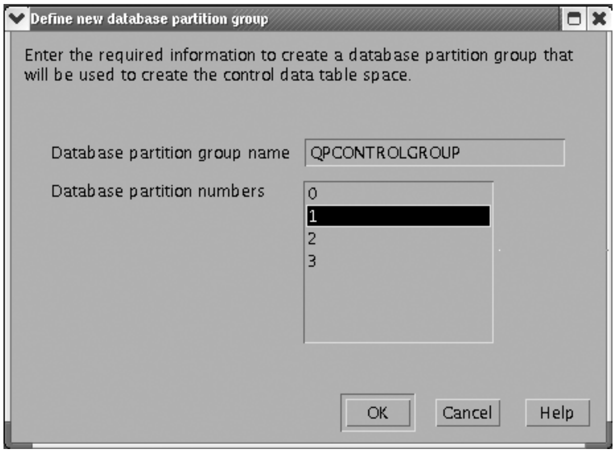
○ New

● Existing: IBMDEFAULTGROUP

Participating database partitions: null

Figure 12. Query Patroller define result table space (UNIX)

17. If you are installing Query Patroller server in a partitioned environment the **Define new database partition group** window appears. Fill in the required information in this window and click **OK**.



Define new database partition group

Enter the required information to create a database partition group that will be used to create the control data table space.

Database partition group name QPCONTROLGROUP

Database partition numbers 0 1 2 3

Figure 13. Query Patroller define new database partition group (UNIX)

18. On the **Start copying files** window, check the settings in **Current settings**. If you are satisfied with the settings, click **Finish** to begin copying files.

Online help is available to guide you through the installation. To invoke the online help, click the **Help** button on the bottom right hand corner of each installation window, or press **F1**. You can click the **Cancel** button at any time to end the installation.

When you have completed your installation, Query Patroller server will be installed in one of the following directories:

AIX /usr/opt/db2_08_01

HP-UX, Linux, Solaris Operating Environment
/opt/IBM/db2/V8.1

You have completed the task of installing Query Patroller server on a UNIX-based operating system using the DB2 Setup wizard. You are now ready to verify the installation of Query Patroller server, and to install the Query Patroller client tools on your UNIX-based DB2 clients using the DB2 Setup wizard.

Related concepts:

- Chapter 2, "Query Patroller installation environment overview," on page 11

Related tasks:

- "Installing Query Patroller server using the DB2 Setup wizard (Windows)" on page 55
- "Setting up Query Patroller server manually" on page 77
- "Migrating Query Patroller server manually" on page 80
- "Installing the Query Patroller client tools using the DB2 Setup wizard (UNIX)" on page 34
- "Verifying the installation of Query Patroller server" on page 29

- Chapter 3, “Installing Query Patroller (UNIX),” on page 15

Related reference:

- “Memory requirements for Query Patroller server (UNIX)” on page 20
- “Disk requirements for Query Patroller server (UNIX)” on page 20
- “Installation requirements for Query Patroller server (UNIX)” on page 17

Verifying the installation of Query Patroller server

After you have installed Query Patroller server, you can verify your installation by submitting a query and checking its status.

Prerequisites:

- You must set up Query Patroller server by issuing the **qpsetup** command. This must be issued prior to verifying the installation of Query Patroller server. You have the choice of either having the **qpsetup** command be issued automatically during the installation, or issuing it manually after the installation at a UNIX shell prompt or a Windows command prompt.
- Make sure the DB2 instance that the database belongs to is running on the server.

Procedure:

To verify your Query Patroller server installation:

1. Log on to the database that is managed by Query Patroller as a user with DBADM authority.
2. Start Query Patroller by issuing the **qpstart** command:

   ```
   qpstart dbname
   ```

   where *dbname* is the name of the database that is managed by Query Patroller. You should receive output that is similar to the following:

   ```
    IBM DB2 Query Patroller V8, (c) Copyright IBM Corp. 1998-2003.
    All rights reserved.
    Initializing.
    Query Patroller started.
   ```

   Note: On UNIX-based operating systems the **qpstart** command will release control back to the UNIX shell prompt. You do not need to open up a second UNIX shell prompt to continue with the rest of this procedure. On Windows the control is released back to the Windows command prompt immediately.
3. Enable the *dyn_query_mgmt* database configuration parameter by issuing the following command:

   ```
   db2 update db cfg for dbname using DYN_QUERY_MGMT enable
   ```

   where *dbname* is the name of the database that is managed by Query Patroller.
4. Submit a query that will be intercepted by Query Patroller server. This query must have an estimated cost of 15000 timerons or more. The estimated cost must be 15000 timerons or more because queries with costs lower than this are not managed by Query Patroller. This is determined by the default Query Patroller parameters. For example, submit the following query::

   ```
   db2 select count(*) from syscat.tables,syscat.tables,syscat.tables
   ```

Wait for the query to return. If you find that the cost of this query is not high enough (it is less than 15000 timerons), then append enough tables to the command until the cost is greater than the minimal cost of 15000 timerons. For example:

```
db2 select count(*) from syscat.tables,syscat.tables,syscat.tables,...
```

The above query is just an example. You can choose to submit a query that is more appropriate to your particular system. However, ensure that the query has a cost that is greater than 15000 timerons.

5. Check the status of the query that you submitted by entering the following command:

   ```
   qp -d dbname list queries
   ```

   where *dbname* is the name of the database that is managed by Query Patroller. If the query has completed successfully, you should receive output that is similar to the following:

   ```
     ID     Status     Created                    Completed
     ======================================================================
     1      Done       2003-10-30 18:36:37.615000 2003-10-30 18:36:37.615000
   ```

   If the query was aborted, you should receive output that is similar to the following:

   ```
     ID     Status     Created                    Completed
     ======================================================================
     1      Aborted    2003-10-30 18:36:37.615000 2003-10-30 18:36:37.615000
   ```

   You can find out why the query was aborted by issuing the following:

   ```
   qp -d dbname get query 1
   ```

   Under ″Message Description″ you will see the error that caused the query to be aborted. Investigate the error and resubmit the query.

6. Stop the Query Patroller service by entering the **qpstop** command. You should receive output that is similar to the following:

   ```
    IBM DB2 Query Patroller V8, (c) Copyright IBM Corp. 1998-2003.
    All rights reserved.
    Stopping Query Patroller.
    Query Patroller stopped.
   ```

Related concepts:

- Chapter 2, “Query Patroller installation environment overview,” on page 11

Related tasks:

- “Installing Query Patroller server using the DB2 Setup wizard (Windows)” on page 55
- “Installing Query Patroller server using the DB2 Setup wizard (UNIX)” on page 21
- “Setting up Query Patroller server manually” on page 77
- “Migrating Query Patroller server manually” on page 80
- Chapter 3, “Installing Query Patroller (UNIX),” on page 15

Installing the Query Patroller client tools with the DB2 Setup wizard (UNIX)

Installation requirements for the Query Patroller client tools (UNIX)

This topic lists the hardware, operating system, software, and communication requirements for the Query Patroller client tools on a UNIX-based operating system.

Hardware requirements

One of the following, depending upon your system:

- IBM RISC/6000 (AIX)
- eServer pSeries (AIX)
- HP 9000 series 700 or 800 system (HP-UX)
- Intel 32–bit or 64–bit(Linux)
- Solaris SPARC-based computer (Solaris Operating Environment)

Operating system requirements

AIX:

One of the following:

Version 4.3.3 (32–bit)

- With maintenance level 9 and APARs IY22308 and IY32690; or
- With maintenance level 10 or later and APAR IY32690

Version 5.1.0 (32–bit or 64–bit)

- With maintenance level 2 or later and APARs IY32905, and IY29345; or
- With maintenance level 3 or later

Version 5.2.0 (32–bit or 64–bit)

Currently there are no APARs requirements.

The following AIX file sets are required to install or run DB2 in languages other than English:

- X11.fnt.ucs.ttf (AIX Windows Unicode TrueType Fonts)
- X11.fnt.ucs.com (AIX Windows Common Fonts)
- xlC.rte 5.0.2.x
- For Asian languages, the following file sets are also required:
  - X11.fnt.ucs.ttf_CN (for zh_CN or Zh_CN)
  - X11.fnt.ucs.ttf_KR (for ko_KR)
  - X11.fnt.ucs.ttf_TW (for zh_TW or Zh_TW)
- On AIX Version 4.3.3 the following file set is required:
  - xlC.aix43.rte 5.0.2.x
- On AIX Version 5.x the following file set is required:
  - xlC.aix50.rte 5.0.2.x

AIX file sets can be downloaded from: http://techsupport.services.ibm.com/server/fixes

Linux:

- To check for the latest technical information on the supported levels of Linux, point your browser to http://ibm.com/db2/linux/validate

HP-UX:

- HP-UX 11i (11.11) for systems with PA-RISC 2.x (PA-8x00) processors with:
  - December 2002 GOLDBASE11i bundle
  - December 2002 GOLDAPPS11i bundle
  - Patches PHSS_26560, PHCO_27434, and PHKL_28489
  - Patches for Java SDK 1.3.1. See http://www.hp.com/products1/unix/java/patches/index.html for required patches.

Solaris Operating Environment:

- Solaris Version 7 or later
- The following patches are required for the Solaris Operating Environment Version 7 and later:
  - Solaris 7 (32-bit) ″Recommended & Security Patches″ + 107226-17 + 107153-01 + 106327-10
  - Solaris 7 (64-bit) ″Recommended & Security Patches″ + 107226-17 + 107153-01 + 106300-11 + 106327-10
  - Solaris 8 (32-bit) ″Recommended & Security Patches″ + 108921-12 + 108940-24 + 108434-03 and 108528-12
  - Solaris 8 (64-bit) ″Recommended & Security Patches″ + 108921-12 + 108940-24 + 108435-03 + 108434-03 and 108528-12
  - Solaris 9 (32-bit)
  - Solaris 9 (64-bit)

The following software is required to install Query Patroller on Solaris Operating Environment:

- SUNWlibC

Software requirements

- A DB2 product must be installed on the computer that you will install the Query Patroller client tools on. The DB2 product must be either Version 8.1.2 with FixPak 2+, or a product that is later than Version 8.1.2. The following products are appropriate prerequisites:
  - Any DB2 client product (for example, DB2 Run-Time client or DB2 Application Development client).
  - Any DB2 Connect product (for example, DB2 Connect Personal Edition or DB2 Connect Enterprise Server Edition).
  - Any DB2 server product (for example, DB2 Enterprise Server Edition or DB2 Workgroup Server Edition).
- For 32-bit environments you require the IBM Software Development Kit (SDK) Version 1.3.1 or Version 1.4.1 to run Query Patroller server, the Query Patroller Java-based tools (such as the Query Patroller Center), and to create and run Java applications, including stored procedures and user-defined functions. During the installation process, if the correct level of the IBM Software Developer’s Kit for Java (SDK) is not already installed, the SDK will be installed if it is required.

- For 64-bit environments you require the IBM Software Development Kit (SDK) Version 1.4.1 to run Query Patroller server, the Query Patroller Java-based tools (such as the Query Patroller Center), and to create and run Java applications, including stored procedures and user-defined functions. During the installation process, if the correct level of the IBM Software Developer's Kit for Java (SDK) is not already installed, the SDK will be installed if it is required. For LinuxIA64 you require the IBM Software Developer's Kit for Java (SDK) Version 1.3.1. For the latest DB2 Java application development updates, visit the Web page at: http://www.ibm.com/software/data/db2/udb/ad/v8/java.
- Netscape 6.2 or Microsoft Internet Explorer 5.5 (or later) is required to view the online installation help.

Communication requirements

- TCP/IP.

Related concepts:

- Chapter 2, "Query Patroller installation environment overview," on page 11

Related tasks:

- "Installing the Query Patroller client tools using the DB2 Setup wizard (UNIX)" on page 34
- Chapter 3, "Installing Query Patroller (UNIX)," on page 15

Related reference:

- "Memory requirements for the Query Patroller client tools (UNIX)" on page 33
- "Disk requirements for the Query Patroller client tools (UNIX)" on page 34

Memory requirements for the Query Patroller client tools (UNIX)

The following list outlines the recommended minimum memory requirements for running the Query Patroller client tools on a DB2 client (UNIX):

- To run the Query Patroller client tools on a system administration workstation requires an additional amount of 64 MB of RAM beyond the amount of RAM required to run your UNIX-based operating system.
- The amount of memory required to run the Query Patroller client tools on a DB2 client that submits queries to the Query Patroller server depends on:
  - the UNIX-based operating system you are using and
  - the database applications you are running.

  It should be sufficient to use the minimum memory requirements of the operating system as the minimum requirements for running these tools on a DB2 client.

Note: Performance may be affected if less than the recommended minimum memory requirements are used.

Related concepts:

- Chapter 2, "Query Patroller installation environment overview," on page 11

Related tasks:

- "Installing the Query Patroller client tools using the DB2 Setup wizard (UNIX)" on page 34
- Chapter 3, "Installing Query Patroller (UNIX)," on page 15

Related reference:

- "Installation requirements for the Query Patroller client tools (UNIX)" on page 31
- "Disk requirements for the Query Patroller client tools (UNIX)" on page 34

Disk requirements for the Query Patroller client tools (UNIX)

The actual fixed disk requirements of your installation may vary depending on your file system and the Query Patroller client tools you install. Ensure that you have included a disk space allowance for your application development tools (if necessary), and communication products.

When you install the Query Patroller clients on a DB2 client using the DB2 Setup wizard, size estimates are dynamically provided by the installation program based on installation type and component selection.

Related concepts:

- Chapter 2, "Query Patroller installation environment overview," on page 11

Related tasks:

- "Installing the Query Patroller client tools using the DB2 Setup wizard (UNIX)" on page 34
- Chapter 3, "Installing Query Patroller (UNIX)," on page 15

Related reference:

- "Installation requirements for the Query Patroller client tools (UNIX)" on page 31
- "Memory requirements for the Query Patroller client tools (UNIX)" on page 33

Installing the Query Patroller client tools using the DB2 Setup wizard (UNIX)

This task is part of the larger task of *Installing Query Patroller (UNIX)*.

This task outlines the steps for installing the Query Patroller client tools on a UNIX-based DB2 client using the DB2 Setup wizard.

Prerequisites:

The following requirements must be met before you begin installing the Query Patroller client tools on a UNIX-based DB2 client:

- Installation requirements for the Query Patroller client tools (UNIX)
- Memory requirements for the Query Patroller client tools (UNIX)
- Disk requirements for the Query Patroller client tools (UNIX)
- Installing the Query Patroller client tools on a DB2 client running on the Solaris Operating Environment or on HP-UX requires that you update your kernel configuration parameters and restart your system.

- A DB2 product (Version 8.1.2 or later) must be installed on the computer that you will install the DB2 Query Patroller client tools on. The following products are appropriate prerequisites:
  - Any DB2 client product (for example, DB2 Run-Time client or DB2 Application Development client)
  - Any DB2 Connect product (for example, DB2 Connect Personal Edition or DB2 Connect Enterprise Server Edition.
  - Any DB2 server product (for example, DB2 Enterprise Server Edition or DB2 Workgroup Server Edition)

Procedure:

After updating your kernel configuration parameters and rebooting your system (required for Solaris and HP-UX), you can install the Query Patroller client tools.

To install the Query Patroller client tools on a UNIX-based DB2 client:

1. Log in as a user with root authority.
2. Shut down any other programs so that the DB2 Setup wizard can update files as required.
3. Check the Query Patroller product CD-ROM label to ensure that you are using the CD-ROM with the appropriate language.
4. Mount the Query Patroller product CD-ROM.
5. Change to the directory where the CD-ROM is mounted by entering the following command:

   ```
   cd /cdrom
   ```

   where */cdrom* represents the mount point of the CD-ROM.
6. Enter the **./db2setup** command to start the DB2 Setup wizard. The IBM DB2 Setup Launchpad window opens.
7. Choose **Install Products** from the DB2 Launchpad.
8. Click **Next** on the **Select the product you would like to install** window.
9. Click **Next** on the **Welcome to the DB2 Setup wizard** window.
10. On the **Software License Agreement** window, select **Accept** if you accept the terms, then click **Next**. If you do not accept the terms, then select **Decline**. Click **Cancel** to end the installation.
11. On the **Select the installation type** window, select the **Computer usage based** button and click **Next**.
12. On the **Select how this computer will be used** window, select **Administrator or end user with self-service capability** and click **Next**.
13. Proceed by following the setup program's prompts. Online help is available to guide you through the remaining steps. To invoke the online help, click the **Help** button on the bottom right hand corner of each installation window, or press **F1**. You can click the **Cancel** button at any time to end the installation.

When you have completed your installation, the Query Patroller client tools will be installed in one of the following directories:

AIX /usr/opt/db2_08_01

HP-UX, Linux, Solaris Operating Environment
/opt/IBM/db2/V8.1

You are now ready to configure your Query Patroller tool-enabled DB2 client (UNIX) to access a remote Query Patroller server.

Related concepts:

- Chapter 2, "Query Patroller installation environment overview," on page 11

Related tasks:

- Chapter 3, "Installing Query Patroller (UNIX)," on page 15

Related reference:

- "Installation requirements for the Query Patroller client tools (UNIX)" on page 31
- "Memory requirements for the Query Patroller client tools (UNIX)" on page 33
- "Disk requirements for the Query Patroller client tools (UNIX)" on page 34

Configuring a client to Query Patroller server connection using the command line processor

This task describes how to configure a connection from a client to a Query Patroller server using the command line processor (CLP).

Prerequisites:

Before you configure a client to Query Patroller server connection:

- Communications must be configured on the client computer. Communications must be TCP/IP.
- Communications must be configured on the Query Patroller server. Communications must be TCP/IP.

Procedure:

To configure a client to Query Patroller server connection using the command line processor:

1. Catalog the TCP/IP node on the DB2 client.
2. Catalog the database on the DB2 client.
3. Test the client to server connection.

Related tasks:

- "Cataloging the TCP/IP node on the DB2 client" on page 36
- "Cataloging a database using the CLP" on page 38
- "Testing the client to server connection using the CLP" on page 40

Cataloging the TCP/IP node on the DB2 client

Cataloging the TCP/IP node adds an entry to the DB2 client's node directory that describes the remote node. This entry specifies the chosen alias (*node_name*), the *hostname* (or *ip_address*), and the *svcename* (or *port_number*) that the client uses to access the remote host.

Prerequisites:

- You must have System Administrative (SYSADM) or System Controller (SYSCTRL) authority, or have the `catalog_noauth` option set to `ON`. You cannot catalog a node using root authority.

Procedure:

To catalog a TCP/IP node:

1. Log on to the system as a user with System Administrative (SYSADM) or System Controller (SYSCTRL) authority.
2. If you are using a UNIX client, set up the instance environment. Run the start-up script:

   For bash, Bourne or Korn shell

   ```
   . INSTHOME/sqllib/db2profile
   ```

   For C shell

   ```
   source INSTHOME/sqllib/db2cshrc
   ```

   where *INSTHOME* represents the home directory of the instance.
3. Start the DB2 command line processor. On Windows, issue the **db2cmd** command from a command prompt. On UNIX, issue the **db2** command from a command prompt.
4. Catalog the node by entering the following commands in the command line processor:

   ```
   db2 => catalog tcpip node node_name remote hostname|ip_address
     server service_name|port_number [remote_instance instance_name]
     [system system_name] [ostype  os_type]

   db2 => terminate
   ```

   where:

   - `node_name` represents a nickname you can set for the machine that has the database you want to catalog.
   - `remote_instance` represents the name of the server instance to which an attachment is being made.
   - `system` represents the DB2 system name that is used to identify the server.
   - `ostype` represents the operating system type of the server.

   Notes:

   a. The **terminate** command is needed to refresh the directory cache.
   b. Although `remote_instance`, `system`, and `ostype` are optional, they are required for users who want to use the DB2 tools.
   c. The *service_name* used on the client does not have to be the same as the one on the server. However, the port numbers that they map to *must* match.

   Note:

   For example, to catalog a node you want to call *earth* on a remote server *earth.torolab.ibm.com* that is using port number *50000*, you would enter the following from a **db2** prompt:

   ```
   db2 => catalog tcpip node earth remote earth server 50000
   DB20000I  The CATALOG TCPIP NODE command completed successfully.
   DB21056W  Directory changes may not be effective until the directory cache is
   refreshed.
   ```

```
db2 => terminate
DB20000I  The TERMINATE command completed successfully.
```

Related tasks:

- "Configuring TCP/IP communications on the client using the CLP" in the *Installation and Configuration Supplement*

Related reference:

- "CATALOG TCPIP NODE Command" in the *Command Reference*

Cataloging the database and testing the connection

Cataloging a database using the CLP

This task describes how to catalog a database using the DB2 command line processor (CLP).

Before a client application can access a remote database, the database must be cataloged on the client. When you create a database, the database is automatically cataloged on the server with a database alias that is the same as the database name, unless a different database alias was specified.

The information in the database directory, along with the information in the node directory (unless cataloging a local database where a node is not needed), is used on the DB2 client to establish a connection to the remote database.

Prerequisites:

- You require a valid DB2 user ID.
- DB2 does not support using root authority to catalog a database.
- If you are cataloging a database on a system that has a DB2 server or DB2 Connect product installed, the user ID must have System Administrative (SYSADM) or System Controller (SYSCTRL) authority on the instance.
- The following parameter values are applicable when cataloging a *remote* database:
  - Database name
  - Database alias
  - Node name
  - Authentication type (optional)
  - Comment (optional)

  You can refer to the parameter values worksheet for cataloging a database for more information about these parameters and to record the values that you use.
- The following parameter values are applicable when cataloging a *local* database:
  - Database name
  - Database alias
  - Authentication type (optional)
  - Comment (optional)

  Local databases can be uncataloged and re-cataloged at any time.

Procedure:

To catalog a database on the client, perform the following steps.

1. Log on to the system with a valid DB2 user ID.
2. Optional. Update the Your Value column in the Parameter values worksheet for cataloging a database.
3. If you are using DB2 on a UNIX platform, set up the instance environment. Run the start-up script:

   For bash, Bourne or Korn shell

   ```
   . INSTHOME/sqllib/db2profile
   ```

   For C shell

   ```
   source INSTHOME/sqllib/db2cshrc
   ```

   where: *INSTHOME* represents the home directory of the instance.
4. Start the DB2 command line processor. On Windows, you can do this by issuing the **db2cmd** command from a command prompt. On UNIX, you can do this by issuing the **db2** command from a command prompt.
5. Catalog the database by entering the following commands in the command line processor:

   ```
   db2 => catalog database database_name as database_alias at
      node node_name [ authentication auth_value ]
   ```

   where:

   - database_name represents the name of the database you want to catalog.
   - database_alias represents an alternate name or nickname for the database you want to catalog.
   - node_name represents a nickname you can set for the machine that has the database you want to catalog.
   - auth_value represents the authentication value for remote databases. Specifying an authentication type can result in a performance benefit. SERVER, CLIENT, SERVER_ENCRYPT, and KERBEROS are the authentication value options.

Example:

For example, to catalog a remote database called *sample* so that it has the local database alias *mysample*, on the node *db2node* using authentication *server*, enter the following commands:

```
db2 => catalog database sample as mysample at node db2node
       authentication server

db2 => terminate
```

Related tasks:

- "Testing the client to server connection using the CLP" on page 40

Related reference:

- "Parameter values worksheet for cataloging a database" in the *Installation and Configuration Supplement*
- "CATALOG DATABASE Command" in the *Command Reference*

Testing the client to server connection using the CLP

After cataloging the node and the database, you should connect to the database to test the connection.

Prerequisites:

- The database node and database must be cataloged before you can test the connection.
- The values for *userid* and *password* must be valid for the system on which they are authenticated. By default, authentication takes place on the server.
- The database manager must be started. If it isn't started, then you can start the database manager by entering the **db2start** command on the database server.

Procedure:

To test the client to server connection:

1. If you are using DB2 on a UNIX platform, set up the instance environment. Run the start-up script:

   For bash, Bourne or Korn shell

   ```
   . INSTHOME/sqllib/db2profile
   ```

   For C shell

   ```
   source INSTHOME/sqllib/db2cshrc
   ```

   where: *INSTHOME* represents the home directory of the instance.
2. Start the DB2 command line processor. On Windows, issue the **db2cmd** command from a command prompt. On UNIX, issue the **db2** command from a command prompt.
3. Type the following command on the client to connect to the remote database:

   ```
   db2 => connect to database_alias user userid
   ```

   For example, enter the following command:

   ```
   connect to mysample user jtris
   ```

   You are then be prompted to enter your password.

If the connection is successful, you receive a message showing the name of the database to which you have connected. A message similar to the following is given:

```
Database Connection Information
Database server = DB2/NT 8.1.0
SQL authorization ID = JSTRIS
Local database alias = mysample
```

You can now work with the database. For example, to retrieve a list of all the table names listed in the system catalog table, enter the following SQL command:

```
select tabname from syscat.tables
```

An implicit connection occurs when you issue an SQL statement followed by the **db2 terminate** command. To define a default database, run the **db2set db2dbdft = <dbname>** command. After running this command, you can, for example, run the **db2 select * from <table>** command without first connecting to a database. This command uses the value defined in **db2dbdft**. To connect to a database other than the default, you must use the CONNECT command to explicitly connect to the database of your choice.

When you are finished using the database connection, enter the **connect reset** command to end the database connection.

Related reference:

- "db2start - Start DB2 Command" in the *Command Reference*
- "db2set - DB2 Profile Registry Command" in the *Command Reference*

Chapter 5. Installing Query Patroller manually (UNIX)

If you choose not to install Query Patroller using the DB2 Setup wizard, you can use the information in the following sections to install Query Patroller manually.

Note: You must run the **qpsetup** command after you finish the manual installation and before you verify if the manual installation was successful.

Installing a DB2 product manually

It is recommended that you install DB2 products using the DB2 Setup wizard or by using a response file. The DB2 Setup wizard provides an easy-to-use graphical interface with installation help, user and group creation, protocol configuration, and instance creation.

A response file installation provides the same advantages, but without the means of a graphical interface. In addition, by using a response file you can take advantage of advanced configuration capabilities such as setting individual DBM configuration parameters or setting profile registry variables.

However, if you do not prefer these installation methods, then you can manually install DB2 products on UNIX-based operating systems using the **db2_install** script, or by using your UNIX-based operating system's native installation utility.

Prerequisites:

Refer to the installation documentation for the particular DB2 product you want to install. For example, if you want to install DB2 Enterprise Server Edition, then refer to the *Quick Beginnings for DB2 Servers* documentation to review installation prerequisites and other important setup information.

Restrictions:

You *cannot* manually install a DB2 product on Windows operating systems. DB2 products can be only installed on Windows operating systems using the DB2 Setup wizard or a response file.

Procedure:

Select a manual installation method:

- Install DB2 using the db2_install script (can be used to install on any of the supported UNIX-based operating systems)
- Install DB2 on AIX using SMIT
- Install DB2 for Linux using rpm
- Install DB2 for HP-UX using swinstall
- Install DB2 for Solaris Operating Environment using pkgadd

Related concepts:

- "Installation methods for DB2 UDB (Windows and UNIX)" in the *Quick Beginnings for DB2 Servers*

Related tasks:

- "Installing a DB2 product using the db2_install script" on page 44
- "Installing a DB2 product on AIX using SMIT" on page 46
- "Installing a DB2 product on Linux using rpm" on page 47
- "Installing a DB2 product on HP-UX using swinstall" on page 48
- "Installing a DB2 product on Solaris Operating Environments using pkgadd" on page 47
- "Setting up DB2 servers after manual installation" in the *Installation and Configuration Supplement*

Installing a DB2 product using the db2_install script

The db2_install script installs all of the components for a particular DB2 product on your UNIX-based operating system using its native installation utility. The db2_install script only installs support for English. This means that help, messages, and tool interfaces are in English only. Tasks such as user creation and configuration that might otherwise be performed for you during an interactive installation (DB2 Setup wizard) or a response file installation will have to be performed after the product is installed.

Prerequisites:

Before you install a DB2 product using the db2_install script:

- You must have root authority.
- You should refer to the installation documentation for the particular DB2 product you want to install. For example, if you want to install DB2 Enterprise Server Edition, then refer to the *Quick Beginnings for DB2 Servers* documentation to review installation prerequisites and other important setup information.

Restrictions:

The db2_install script cannot be used to install DB2 on Windows operating systems. DB2 for Windows must be installed using the DB2 Setup wizard or a response file.

Procedure:

To install a DB2 product using the db2_install script:

1. Log in as a user with root authority.
2. Insert and mount the appropriate CD-ROM.
3. Enter the **./db2_install** command at the shell prompt to start the db2_install script. The db2_install script can be found in the root directory on your DB2 Version 8 product CD-ROM. The db2_install script prompts you for one of the following keywords (depending upon what DB2 product you are installing):

   DB2.EXP
   DB2 Express Edition

   DB2.ESE
   DB2 Enterprise Server Edition

   DB2.WSE
   DB2 Workgroup Server Edition and DB2 Workgroup Server Unlimited Edition

DB2.PE
DB2 Personal Edition

DB2.CONEE
DB2 Connect Enterprise Edition, DB2 Connect Unlimited Edition, and DB2 Connect Application Server Edition.

DB2.CONPE
DB2 Connect Personal Edition

DB2.ADMCL
DB2 Administration Client

DB2.ADCL
DB2 Application Development Client

DB2.RTCL
DB2 Run-Time Client

DB2.DLM
DB2 Data Links Manager

DB2.GSE
DB2 Spatial Extender

DB2.WM
DB2 Warehouse Manager

DB2.WMC
DB2 Warehouse Manager Connectors

DB2.QP
DB2 Query Patroller

DB2.CUBE
DB2 Cube Views

DB2.LSDC
DB2 Information Integrator Non-Relational Wrappers

DB2.RCON
DB2 Information Integrator Relational Wrappers

4. Enter the keyword for the product you want to install. If you specify more than one product keyword, separate the keywords by spaces.

The installation directory for DB2 software is on:

- /usr/opt/db2_08_01 for AIX
- /opt/IBM/db2/V8.1 for HP-UX, Linux, or Solaris Operating Environment

For the Solaris Operating Environment you have the option of specifying a different base directory for DB2. The default base directory for Solaris Operating Environment is /opt. If you choose to install DB2 to a different base directory, then links will be set for the default DB2 installation directory, /opt/IBM/db2/V8.1. Parts of the product are dependant upon the default installation directory. Creating links allows DB2 to physically exist on a base directory other than /opt.

Related tasks:

- "Mounting the CD-ROM (AIX)" in the *Quick Beginnings for DB2 Servers*
- "Mounting the CD-ROM (HP-UX)" in the *Quick Beginnings for DB2 Servers*
- "Mounting the CD-ROM (Linux)" in the *Quick Beginnings for DB2 Servers*

- "Setting up DB2 servers after manual installation" in the *Installation and Configuration Supplement*

Installing a DB2 product on AIX using SMIT

This task describes how to install a DB2 product using the System Management Interface Tool (SMIT) on AIX. Tasks such as user creation and configuration that might otherwise be performed for you during an interactive installation (DB2 Setup wizard), or during a response file installation, must be performed after the product is installed.

Prerequisites:

Before you install DB2 on AIX using SMIT:

- You must have root authority.
- You should refer to the installation documentation for the particular DB2 product you want to install. For example, if you want to install DB2 Enterprise Server Edition, then refer to the *Quick Beginnings for DB2 Servers* documentation to review installation prerequisites and other important setup information.

Procedure:

To install a DB2 product using the SMIT on AIX:

1. Log in as a user with root authority.
2. Insert and mount the appropriate CD-ROM.
3. Enter the **smit** command at the shell prompt. The smit GUI opens.
4. Under the **System Management** list, select **Software Installation and Maintenance** --> **Install and Update Software** --> **Install Software**.
5. In the ***INPUT device / directory for software** text field specify the input device or directory for the installation media, or click the **List** button to display all input devices or directories.
6. Select the mounted CD-ROM drive from the list. The input device or directory for the installation media will then appear in the ***INPUT device / directory for software** text field. Click **OK**.
7. In the ***SOFTWARE to install** text field specify the components that you want to install, or click the **List** button to display the installable software products. Click **OK**.

   Note: Each DB2 product CD-ROM provides a file that lists the components available for installation. Use this list to identify required, typical, and optional DB2 components for the product that you want to install. The component list is in a file called ComponentList.htm and is located in the /db2/*plat* directory on your CD-ROM where *plat* is the platform that you are installing on.

When the installation is complete your DB2 software will be installed in the /usr/opt/db2_08_01 directory.

Related tasks:

- "Mounting the CD-ROM (AIX)" in the *Quick Beginnings for DB2 Servers*
- "Setting up DB2 servers after manual installation" in the *Installation and Configuration Supplement*

Installing a DB2 product on Solaris Operating Environments using pkgadd

This task describes how to install a DB2 product using the **pkgadd** utility on Solaris Operating Environments. Tasks such as user creation and configuration that might otherwise be performed for you during an interactive installation (DB2 Setup wizard), or during a response file installation, must be performed after the product is installed.

Prerequisites:

Before you install a DB2 product on Solaris Operating Environments using **pkgadd**:

- You must have root authority.
- You should refer to the installation documentation for the particular DB2 product you want to install. For example, if you want to install DB2 Enterprise Server Edition, then refer to the *Quick Beginnings for DB2 Servers* documentation to review installation prerequisites and other important setup information.

Procedure:

To install a DB2 product on Solaris Operating Environments using **pkgadd**:

1. Log in as a user with root authority.
2. Insert and mount the appropriate CD-ROM.
3. Identify components you want to install. Each DB2 product CD-ROM provides a file that lists the components available for installation. Use this list to identify required, typical, and optional DB2 components. The component list is in a file called ComponentList.htm and is located in the /db2/*plat* directory on your CD-ROM where *plat* is the platform that you are installing on.
4. Run the **pkgadd** command for each component you want to install:

   ```
   pkgadd component_name
   ```

   For example, if you want to install the Control Center, install the db2wcc81 component by entering the following command:

   ```
   pkgadd db2wcc81
   ```

When installation is complete your DB2 software will be installed in the /opt/IBM/db2/V8.1 directory.

Related tasks:

- “Installing a DB2 product using the db2_install script” on page 44
- “Setting up DB2 servers after manual installation” in the *Installation and Configuration Supplement*

Installing a DB2 product on Linux using rpm

This task describes how to install a DB2 product using the **rpm** command on Linux. Tasks such as user creation and configuration that might otherwise be performed for you during an interactive installation (DB2 Setup wizard) or during a response file installation, must be performed after the product is installed.

Prerequisites:

Before you install a DB2 product for Linux using the **rpm** command:

- You must have root authority.
- You should refer to the installation documentation for the particular DB2 product you want to install. For example, if you want to install DB2 Enterprise Server Edition, then refer to the *Quick Beginnings for DB2 Servers* documentation to review installation prerequisites and other important setup information.

Procedure:

To install a DB2 product on Linux using the **rpm** command:

1. Log in as a user with root authority.
2. Insert and mount the appropriate CD-ROM.
3. Identify the components you want to install. Each DB2 product CD-ROM provides a file that lists the components available for installation. Use this list to identify required, typical, and optional DB2 components. The component list is in a file called ComponentList.htm and is located in the /db2/*plat* directory on your CD-ROM where *plat* is the platform that you are installing on.
4. Run the **rpm** command for each component you want to install:

   ```
   rpm -ivh component_name
   ```

   For example, if you want to install the Control Center, install the db2cc81-8.1.0-0.i386.rpm component by entering the following command:

   ```
   rpm -ivh IBM_db2cc81-8.1.0-0.i386.rpm
   ```

When the installation is complete your DB2 software will be installed in the /opt/IBM/db2/V8.1 directory.

Related tasks:

- "Mounting the CD-ROM (Linux)" in the *Quick Beginnings for DB2 Servers*
- "Setting up DB2 servers after manual installation" in the *Installation and Configuration Supplement*

Installing a DB2 product on HP-UX using swinstall

This task describes how to install a DB2 product using the **swinstall** utility on HP-UX. Tasks such as user creation and configuration that might otherwise be performed for you during an interactive installation (DB2 Setup wizard), or during a response file installation, must be performed after the product is installed.

Prerequisites:

Before you install a DB2 product on HP-UX using the **swinstall** utility:

- You must have root authority.
- You should refer to the installation documentation for the particular DB2 product you want to install. For example, if you want to install DB2 Enterprise Server Edition, then refer to the *Quick Beginnings for DB2 Servers* documentation to review installation prerequisites and other important setup information.

Procedure:

To install a DB2 product on HP-UX using the **swinstall** utility:

1. Log in as a user with root authority.
2. Insert and mount the appropriate CD-ROM.

3. Run the **swinstall** utility using the following command at the shell prompt:

   ```
   swinstall -x autoselect_dependencies=true
   ```

   This command opens the **Software Selection** window and the **Specify Source** window. If necessary, change the host name in the **Source Host Name...** text field in the **Specify Source** window.
4. In the **Source Depot Path...** field enter /cdrom/db2/hpux where */cdrom* represents the CD-ROM mount directory.
5. Click **OK** to return to the **Software Selection** window.
6. The **Software Selection** window contains a list of available software to install. Identify the components you want to install. Each DB2 product CD-ROM provides a file that lists the components available for installation. Use this list to identify required, typical, and optional DB2 components. The component list is in a file called ComponentList.htm and is located in the /db2 directory on your CD-ROM.
7. Select **Mark for Install** from the **Actions** menu to choose the components to be installed.
8. Select **OK** if the following message appears:

   ```
   In addition to the software you just marked, other software was
   automatically marked to resolve dependencies.  This message will
   not appear again.
   ```
9. Select **Install (analysis)** from the **Actions** menu to begin the product installation and to open the **Install Analysis** window.
10. Select **OK** in the **Install Analysis** window when the **Status** field displays a Ready message.
11. Select **Yes** in the Confirmation window to confirm that you want to install the software.

    View the Install window to read processing data while the software is being installed, until the **Status** field indicates Ready and the Note window opens. The **swinstall** program loads the file set, and runs the control scripts for the file set.
12. Select **Exit** from the **File** menu to exit from **swinstall**.

When the installation is complete your DB2 software will be installed in the /opt/IBM/db2/V8.1 directory.

Related tasks:

- "Mounting the CD-ROM (HP-UX)" in the *Quick Beginnings for DB2 Servers*
- "Setting up DB2 servers after manual installation" in the *Installation and Configuration Supplement*

Chapter 6. Installing Query Patroller (Windows)

If you know which Query Patroller components and tools you want to install, and the kind of environment you will be installing on, you can follow the steps below to install Query Patroller on your Windows server and your Windows clients.

Prerequisites:

Before you install Query Patroller you must:

- Ensure that you know the kind of environment in which you are going to install Query Patroller.
- Verify that the computer on which you will install Query Patroller server meets the necessary installation, memory, and disk requirements.
- Verify that the computers on which you will install the Query Patroller client tools meet the necessary installation, memory, and disk requirements.
- Ensure that the DYN_QUERY_MGMT configuration parameter is set to DISABLE before you install Query Patroller. This parameter must be set to ENABLE for Query Patroller to intercept and manage queries, but it must be set to DISABLE prior to installation to prevent the interception of any internal queries run by the installer.

Procedure:

To install Query Patroller (Windows):

1. Install Query Patroller server with the DB2 Setup wizard (Windows).
2. Verify the installation of Query Patroller server.
3. Install the Query Patroller client tools with the DB2 Setup wizard (Windows).
4. Install the DB2 Information Center (Windows)

Related tasks:

- “Installing Query Patroller server using the DB2 Setup wizard (Windows)” on page 55
- “Installing the Query Patroller client tools using the DB2 Setup Wizard (Windows)” on page 69
- “Verifying the installation of Query Patroller server” on page 29
- “Installing the DB2 Information Center using the DB2 Setup wizard (Windows)” on page 309

Chapter 7. Installing Query Patroller with the DB2 Setup wizard (Windows)

Installing the Query Patroller server with the DB2 Setup wizard (Windows)

Installation requirements for Query Patroller server (Windows)

This topic lists the hardware, operating system, software, and communication requirements for Query Patroller server on Windows.

Hardware requirements

- For 32-bit Query Patroller servers, a Pentium or Pentium-compatible CPU is required. For 64-bit Query Patroller servers, an Itanium or Itanium-compatible CPU is required.

Operating system requirements

- Windows NT Version 4 with Service Pack 6a or later
- Windows 2000 Professional, Standard Server, Advanced Server, and Datacenter Servers. Windows 2000 Service Pack 2 or later is required for Windows Terminal Server.
- Windows Server 2003 (32-bit and 64-bit)

Software requirements

- To install Query Patroller server, you must have installed either one of the following:
  - DB2 Enterprise Server Edition Version 8.1.2 and FixPak 2+
  - DB2 Enterprise Server Edition that is later than Version 8.1.2
- For 32-bit environments you require the IBM Software Development Kit (SDK) Version 1.3.1 or Version 1.4.1 to run Query Patroller server, the Query Patroller Java-based tools (such as the Query Patroller Center), and to create and run Java applications, including stored procedures and user-defined functions. During the installation process, the IBM Software Development Kit (SDK) will be installed if it is required.
- For 64-bit environments you require the IBM Software Development Kit (SDK) Version 1.4.1 to run Query Patroller server, the Query Patroller Java-based tools (such as the Query Patroller Center), and to create and run Java applications, including stored procedures and user-defined functions. During the installation process, the IBM Software Development Kit (SDK) will be installed if it is required. For the latest DB2 Java application development updates, visit the Web page at: http://www.ibm.com/software/data/db2/udb/ad/v8/java.
- Netscape 6.2 or Microsoft Internet Explorer 5.5 is required to view the online installation help.

Communication requirements

- TCP/IP.

Related concepts:

- Chapter 2, "Query Patroller installation environment overview," on page 11

Related tasks:

- "Installing Query Patroller server using the DB2 Setup wizard (Windows)" on page 55
- Chapter 6, "Installing Query Patroller (Windows)," on page 51

Related reference:

- "Memory requirements for Query Patroller server (Windows)" on page 54
- "Disk requirements for Query Patroller server (Windows)" on page 54

Memory requirements for Query Patroller server (Windows)

At a minimum, Query Patroller server requires 256 MB of RAM. Additional memory may be required.

When determining memory requirements, remember:

- Additional memory may be required for non-DB2 software that is running on your system.
- Additional memory is required to support DB2 clients that have the Query Patroller client tools installed on them.
- Specific performance requirements may determine the amount of memory needed.
- Memory requirements are affected by the size and complexity of your database system.

Related concepts:

- Chapter 2, "Query Patroller installation environment overview," on page 11

Related tasks:

- "Installing Query Patroller server using the DB2 Setup wizard (Windows)" on page 55
- Chapter 6, "Installing Query Patroller (Windows)," on page 51

Related reference:

- "Installation requirements for Query Patroller server (Windows)" on page 53
- "Disk requirements for Query Patroller server (Windows)" on page 54

Disk requirements for Query Patroller server (Windows)

The disk space required for Query Patroller server depends on the type of installation you choose and the type of disk drive upon which Query Patroller is installed. When you install Query Patroller server using the DB2 Setup wizard, size estimates are dynamically provided by the installation program based on installation type and component selection. Disk space is needed for the following:

- To store the product code.
- To store data that will be generated when using Query Patroller (for example, the contents of the control tables).

Remember to include disk space allowance for required software and communication products.

Insufficient disk space management:

If the space required to install the selected components exceeds the space found in the path you specify for installing the components, the DB2 Setup wizard issues a warning about the insufficient space. If you choose, you can continue the installation. However, if the space for the files being installed is, in fact, insufficient, then the Query Patroller server installation stops when there is no more space. When this occurs, the installation is rolled back. You will then see a final dialog with the appropriate error messages. You can then exit the installation.

Related concepts:

- Chapter 2, “Query Patroller installation environment overview,” on page 11

Related tasks:

- “Installing Query Patroller server using the DB2 Setup wizard (Windows)” on page 55
- Chapter 6, “Installing Query Patroller (Windows),” on page 51

Related reference:

- “Installation requirements for Query Patroller server (Windows)” on page 53
- “Memory requirements for Query Patroller server (Windows)” on page 54

Installing Query Patroller server using the DB2 Setup wizard (Windows)

This task is part of the larger task of *Installing Query Patroller (Windows)*.

This task outlines the steps for installing Query Patroller server on Windows using the DB2 Setup wizard.

Prerequisites:

Ensure that your computer meets the following requirements:

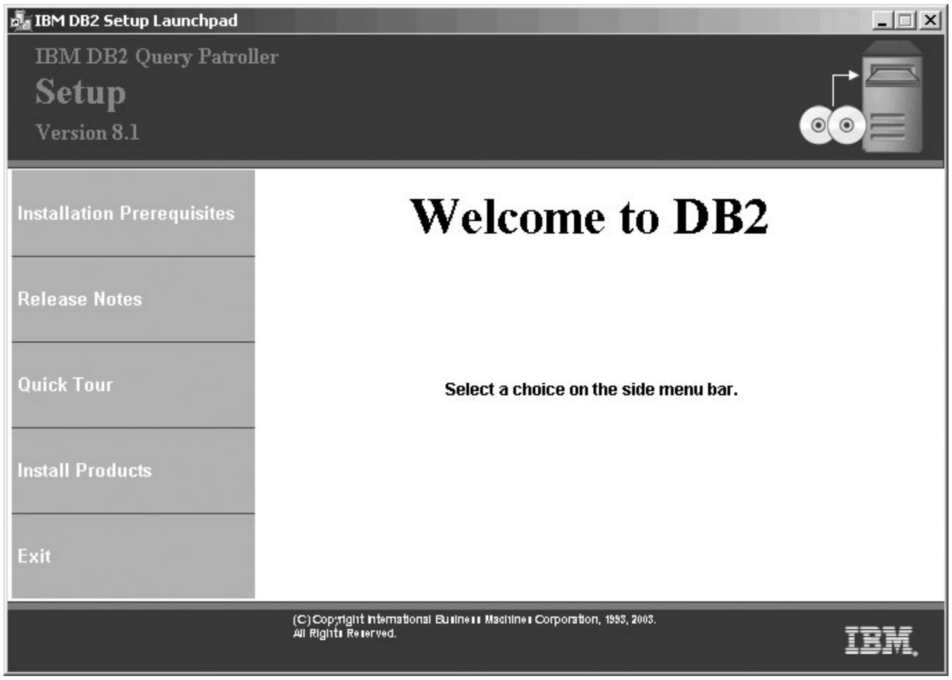
- Installation requirements for Query Patroller server (Windows)
- Memory requirements for Query Patroller server (Windows)
- Disk requirements for Query Patroller server (Windows)
- The instance running on your DB2 server is stopped. To stop an instance, log on to the system as a user with SYSADM authority and issue the **db2stop** command at a Windows command prompt.
- A database whose queries you want to manage using the Query Patroller server.
- All of the languages that the clients will use when submitting queries to Query Patroller server must be installed on the server in order to eliminate any server to client dependencies.

Procedure:

To install Query Patroller server on Windows:

1. Log on to the system with the Administrator account you will use to install Query Patroller server.
2. Close all programs so the installation program can update files as required.
3. Insert the CD-ROM into the drive. If enabled, the auto-run feature automatically starts the DB2 Setup Launchpad. From this window you can proceed directly to the installation.

4. Choose **Install Products** from the DB2 Launchpad.



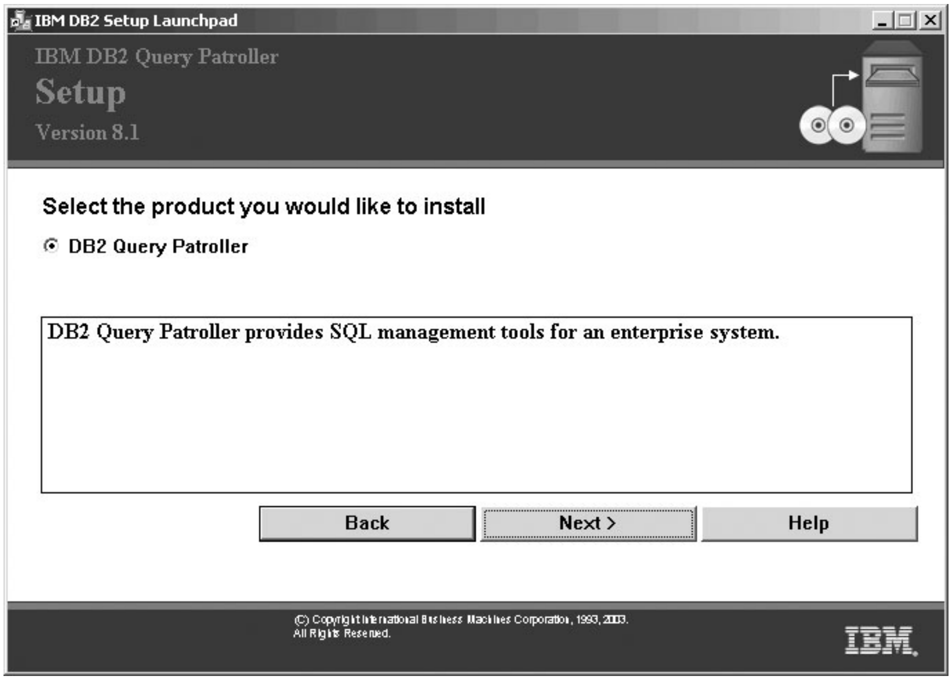
Welcome to DB2

Select a choice on the side menu bar.

(C) Copyright International Business Machines Corporation, 1993, 2003.
All Rights Reserved.

Figure 14. Query Patroller welcome screen (Windows)

5. On the **Select the product you would like to install** window, click **Next**.



Select the product you would like to install

● DB2 Query Patroller

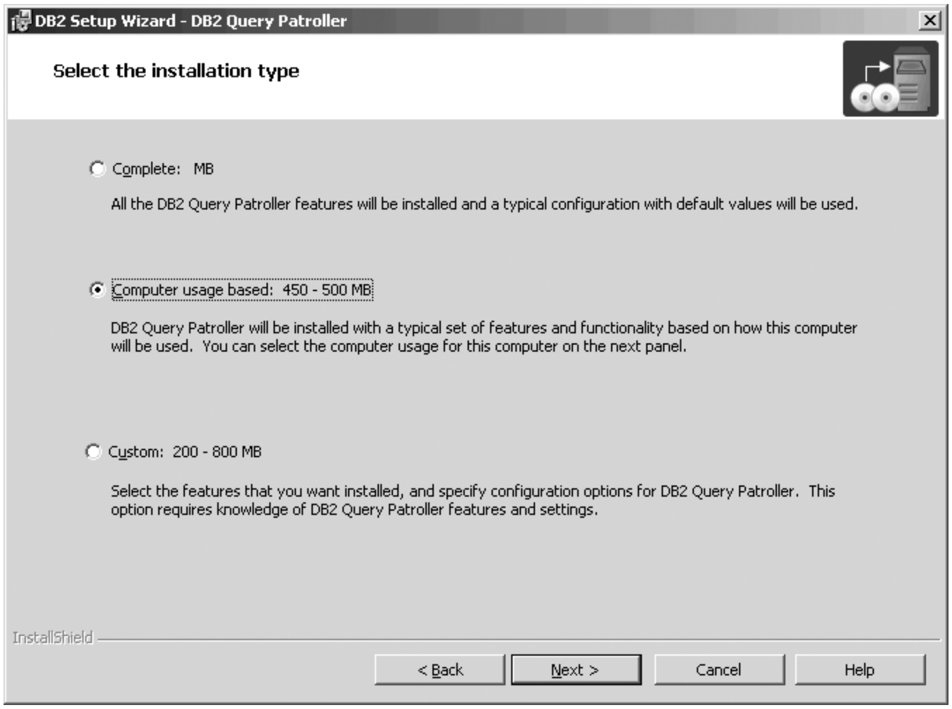
DB2 Query Patroller provides SQL management tools for an enterprise system.

(C) Copyright International Business Machines Corporation, 1993, 2003.
All Rights Reserved.

Figure 15. Query Patroller product description (Windows)

6. On the **Welcome to the DB2 Setup wizard** window, click **Next**.

7. On the **License Agreement** window, select **I accept the terms in the license agreement** if you accept the terms. Then click **Next**. If you do not accept the terms, then select **I do not accept the terms in the license agreement**. Then click **Cancel** to end the installation.
8. On the **Select the installation type** window, select **Computer usage based** and click **Next**.



DB2 Setup Wizard - DB2 Query Patroller

Select the installation type

○ Complete: MB

All the DB2 Query Patroller features will be installed and a typical configuration with default values will be used.

● Computer usage based: 450 - 500 MB

DB2 Query Patroller will be installed with a typical set of features and functionality based on how this computer will be used. You can select the computer usage for this computer on the next panel.

○ Custom: 200 - 800 MB

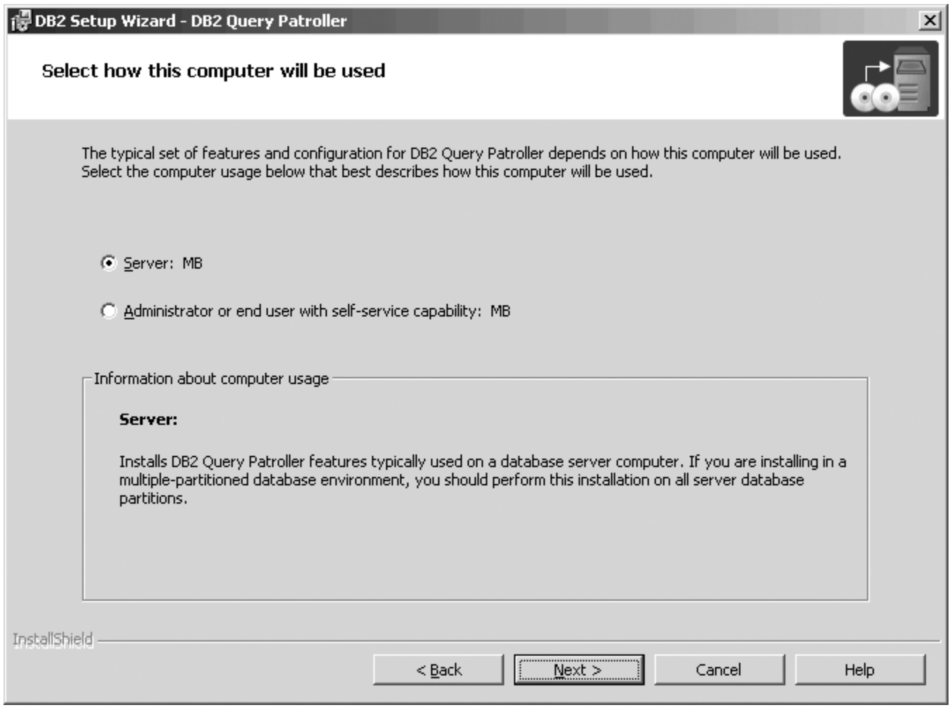
Select the features that you want installed, and specify configuration options for DB2 Query Patroller. This option requires knowledge of DB2 Query Patroller features and settings.

Figure 16. Query Patroller select installation type (Windows)

9. On the **Select how this computer will be used** window, select **Server** and click **Next**.

Note:

If you select **Computer usage based** from the **Select the installation type** window using the DB2 Setup wizard, and you select **Server** from the **Select how this computer will be used** window, then the Query Patroller Center will not be installed. If you want to install the Query Patroller Center, then select either the **Complete** or the **Custom** installation types from the **Select the installation type** window.



DB2 Setup Wizard - DB2 Query Patroller

Select how this computer will be used

The typical set of features and configuration for DB2 Query Patroller depends on how this computer will be used. Select the computer usage below that best describes how this computer will be used.

● Server: MB

○ Administrator or end user with self-service capability: MB

Information about computer usage

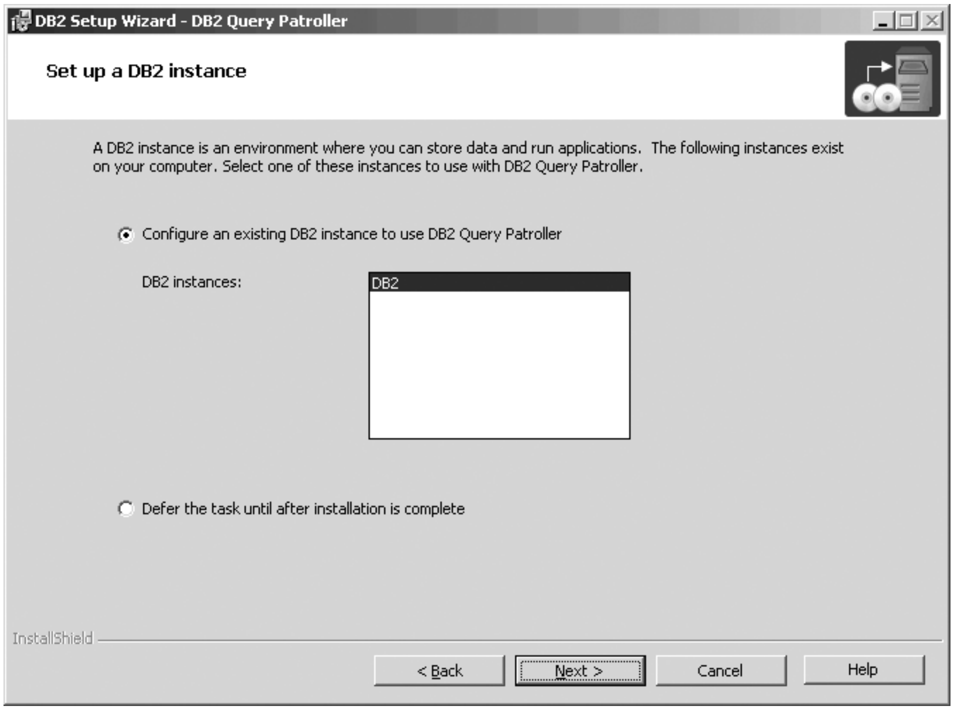
Server:

Installs DB2 Query Patroller features typically used on a database server computer. If you are installing in a multiple-partitioned database environment, you should perform this installation on all server database partitions.

Figure 17. Query Patroller computer use (Windows)

10. On the **Select installation folder** check the **Confirm installation folder** section. Click **Next** if you want to install Query Patroller server on the drive and in the directory that is displayed.
11. On the **Set up a DB2 instance** window select whether you want to configure an existing DB2 instance, or whether you want to defer this task until after the installation is complete. All of the existing DB2 instances appear in **DB2 instances**. If your selected instance does not have any existing database associated with it, then a warning dialog will appear informing you to either select another instance, or to defer this task until after the installation is

complete. Once you have made your selection, click **Next**.



DB2 Setup Wizard - DB2 Query Patroller

Set up a DB2 instance

A DB2 instance is an environment where you can store data and run applications. The following instances exist on your computer. Select one of these instances to use with DB2 Query Patroller.

● Configure an existing DB2 instance to use DB2 Query Patroller

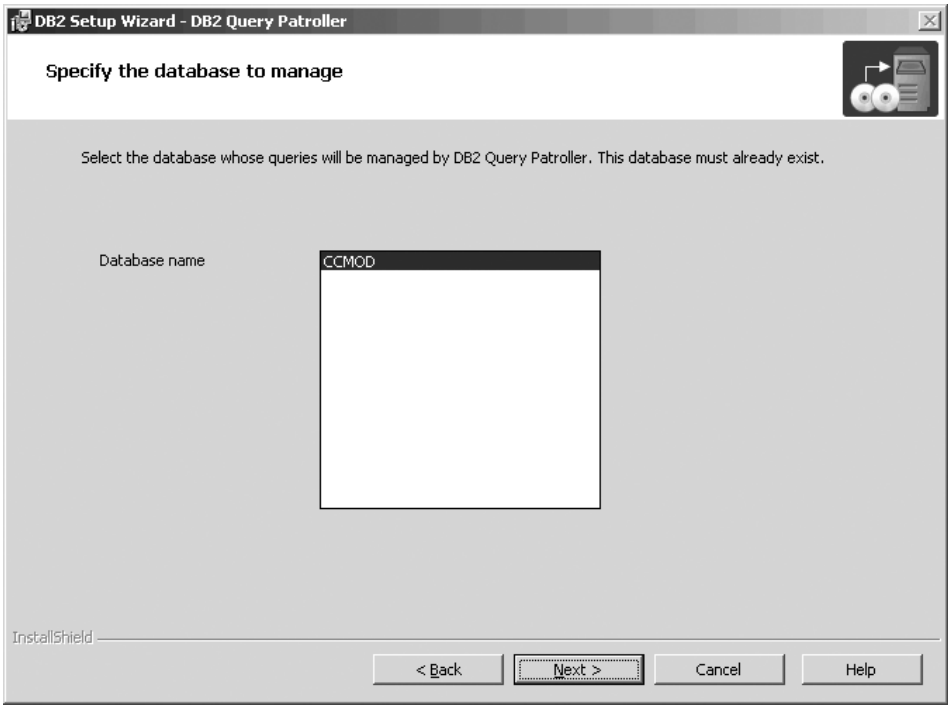
DB2 instances: DB2

○ Defer the task until after installation is complete

Figure 18. Query Patroller set up a DB2 instance (Windows)

12. On the **Specify the database to manage** window, select an already existing database that will be managed by Query Patroller. The database you select will appear in **Database name**. Once you have made your selection, click **Next**.

    Note: If the database you select has Version 7 Query Patroller data and settings, then you will be prompted to specify whether you want to migrate the Version 7 data and settings to the Version 8 Query Patroller system. If you choose not to migrate during the installation, you can migrate after the installation using the **qpmigrate** command.



DB2 Setup Wizard - DB2 Query Patroller

Specify the database to manage

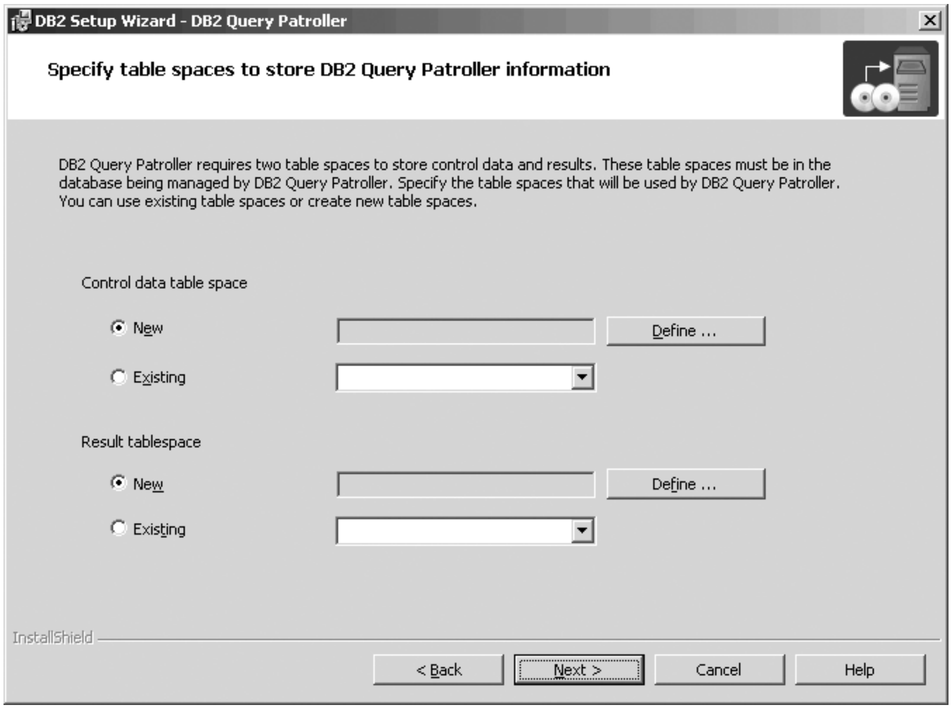
Select the database whose queries will be managed by DB2 Query Patroller. This database must already exist.

Database name: CCMOD

Figure 19. Query Patroller specify the database to manage (Windows)

13. On the **Specify table spaces to store DB2 Query Patroller information** window, select whether you want to create a new or existing control data table space, or whether you want to create a new or existing result table space. Click **New** and then click **Define** if you want to create either a new control data table space or a new result table space. If you want to use existing table spaces, then select **Existing** and choose the table spaces from the

menu. Once you have selected your table spaces click **Next**.



DB2 Setup Wizard - DB2 Query Patroller

Specify table spaces to store DB2 Query Patroller information

DB2 Query Patroller requires two table spaces to store control data and results. These table spaces must be in the database being managed by DB2 Query Patroller. Specify the table spaces that will be used by DB2 Query Patroller. You can use existing table spaces or create new table spaces.

Control data table space

● New

○ Existing

Result tablespace

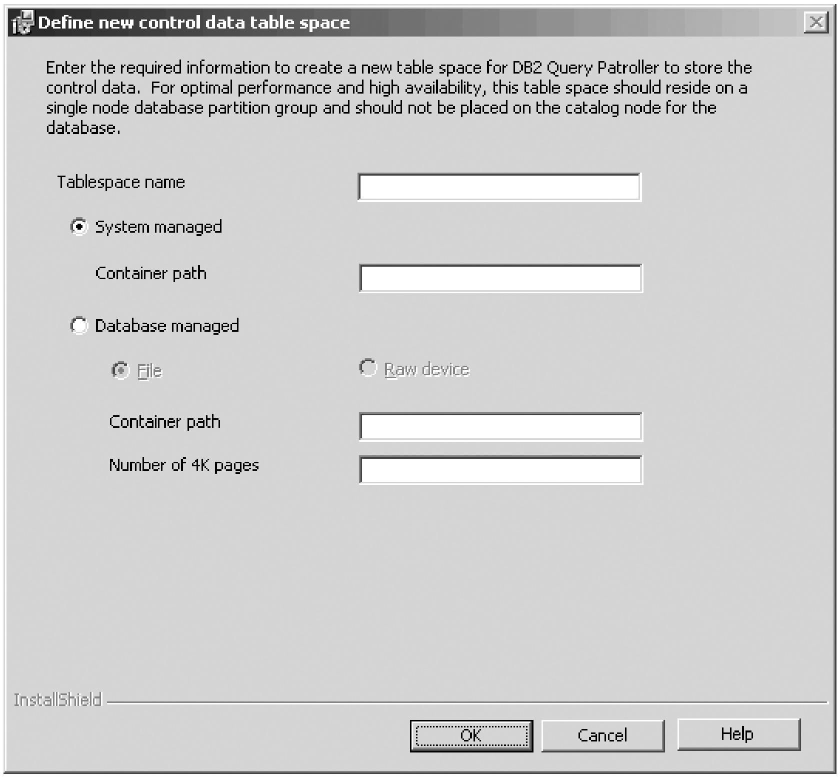
● New

○ Existing

Figure 20. Query Patroller specify table spaces (Windows)

14. If you select to create a new control data table space then the **Define new control data table space** window appears. When you are finished filling in the values in this window, click **OK**. The **Specify table spaces to store DB2 Query Patroller information** window will appear again with the name of the new control data table space in the text field. If you are installing Query Patroller in a partitioned environment, then the **Database partition group** box appears on the bottom of the window. Click **Existing** if you want to use an existing database partition group. If you want to create a new database partition group, click **New** and then click **Define**. The **Define new database partition group** window appears . Fill in the required information in this window and click **OK**. The **Define new control data table space** window

appears again.Once you have selected your table spaces click **Next**.



Define new control data table space

Enter the required information to create a new table space for DB2 Query Patroller to store the control data. For optimal performance and high availability, this table space should reside on a single node database partition group and should not be placed on the catalog node for the database.

Tablespace name

● System managed

Container path

○ Database managed

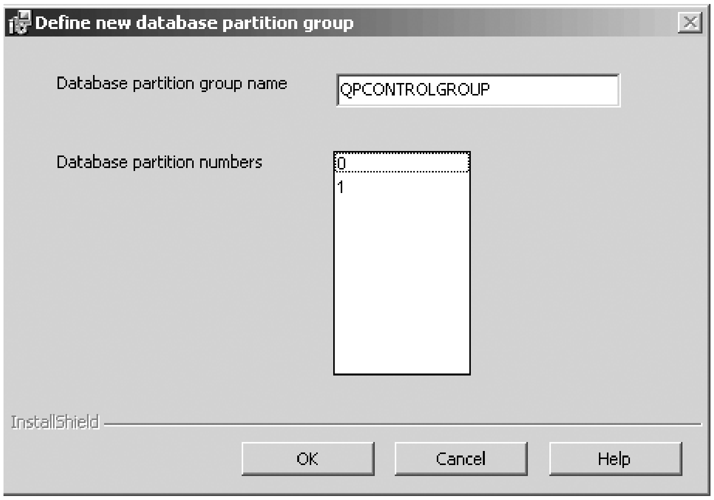
● File ○ Raw device

Container path

Number of 4K pages

Figure 21. Query Patroller define new control data table space (Windows)

15. If you are installing Query Patroller server in a partitioned environment the **Define new database partition group** window appears. Fill in the required information in this window and click **OK**.



Define new database partition group

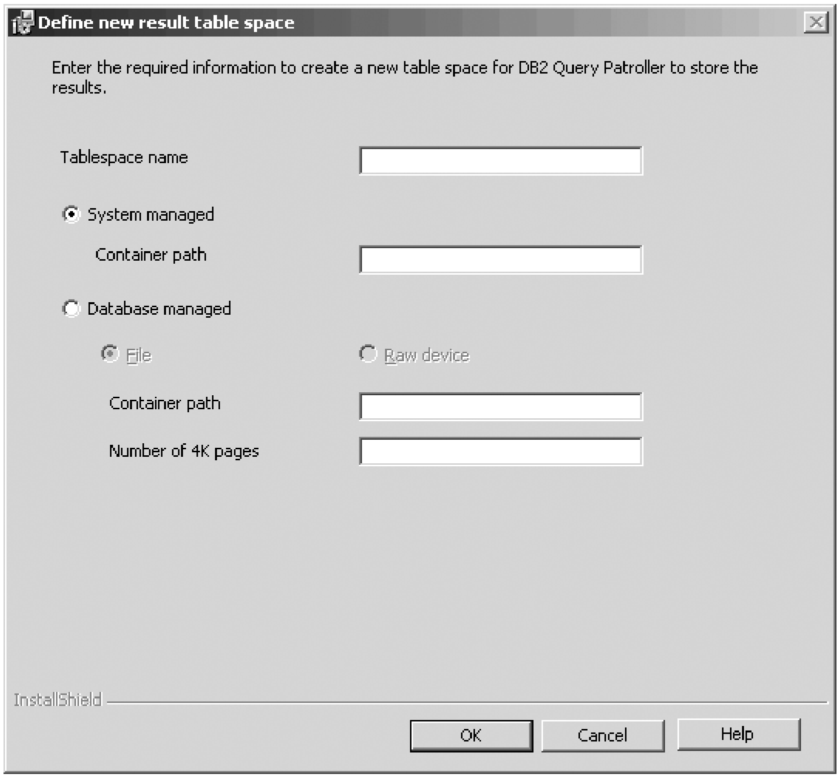
Database partition group name: QPCONTROLGROUP

Database partition numbers: 0, 1

Figure 22. Query Patroller define new database partition group (Windows)

16. If you select to create a new result table space then the **Define new result table space** window appears. When you are finished filling in the values in this window, click **OK**. The **Specify table spaces to store DB2 Query Patroller**

information window will appear again with the name of the new result table space in the text field. If you are installing Query Patroller in a partitioned environment, then the **Database partition group** box appears on the bottom of the window. Click **Existing** if you want to use an existing database partition group. If you want to create a new database partition group, click **New** and then click **Define**. The **Define new database partition group** window appears (see above). Fill in the required information in this window and click **OK**. The **Define new result table space** window appears again.Once you have selected your table spaces click **Next**.



Define new result table space

Enter the required information to create a new table space for DB2 Query Patroller to store the results.

Tablespace name

● System managed

Container path

○ Database managed

● File ○ Raw device

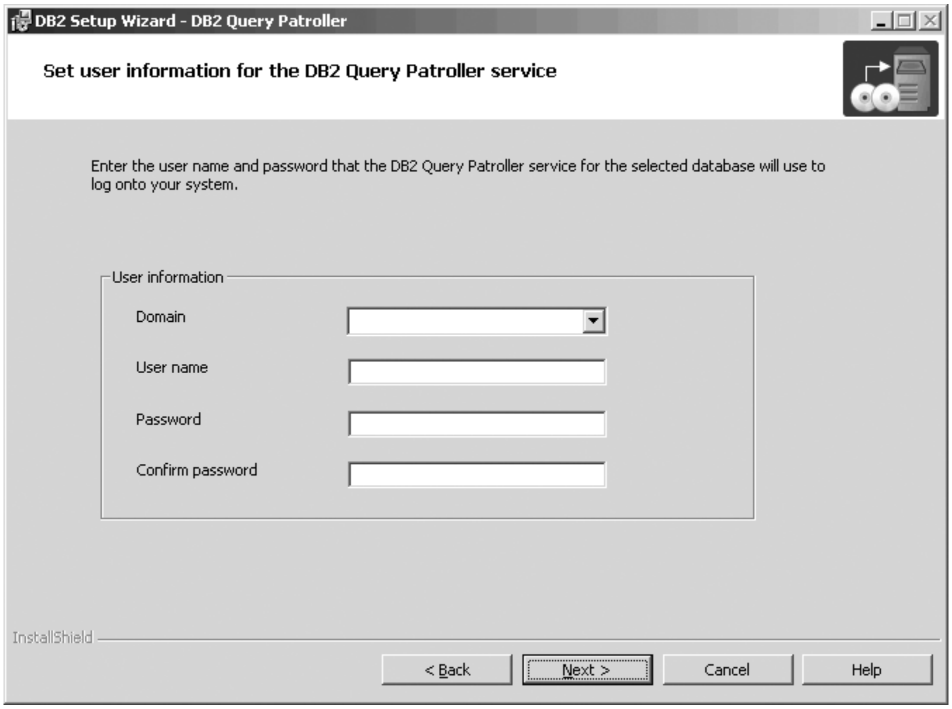
Container path

Number of 4K pages

Figure 23. Query Patroller define new result table space (Windows)

17. On the **Set user information for the DB2 Query Patroller service** window, fill in your user information in the **User information** section. Once you have

filled in your information, click **Next**.



DB2 Setup Wizard - DB2 Query Patroller

Set user information for the DB2 Query Patroller service

Enter the user name and password that the DB2 Query Patroller service for the selected database will use to log onto your system.

Figure 24. Query Patroller set user information for service (Windows)

18. On the **Start copying files** window, check the settings in **Current settings**. If you are satisfied with the settings, click **Install** to begin copying files.

Online help is available to guide you through the installation. To invoke the online help, click the **Help** button on the bottom right hand corner of each installation window, or press **F1**. You can click the **Cancel** button at any time to end the installation.

For information on errors encountered during installation, see the db2.log file. The db2.log file stores general information and error messages resulting from install and uninstall activities. By default, the db2.log file is located in the My Documents\DB2LOG\ directory. The location of the My Documents directory depends on your computer settings. For information on the most recent Query Patroller installations see the db2wi.log. The contents of this log are put at the end of the db2.log file.

You have completed the task of installing Query Patroller server on Windows using the DB2 Setup wizard. You are now ready to verify the installation of Query Patroller server, and to install the Query Patroller client tools on your DB2 clients (Windows) using the DB2 Setup wizard.

Related concepts:

- Chapter 2, “Query Patroller installation environment overview,” on page 11

Related tasks:

- “Installing Query Patroller server using the DB2 Setup wizard (UNIX)” on page 21
- “Setting up Query Patroller server manually” on page 77

- “Migrating Query Patroller server manually” on page 80
- “Installing the Query Patroller client tools using the DB2 Setup Wizard (Windows)” on page 69
- “Verifying the installation of Query Patroller server” on page 29
- Chapter 6, “Installing Query Patroller (Windows),” on page 51

Related reference:

- “Installation requirements for Query Patroller server (Windows)” on page 53
- “Memory requirements for Query Patroller server (Windows)” on page 54
- “Disk requirements for Query Patroller server (Windows)” on page 54

Verifying the installation of Query Patroller server

After you have installed Query Patroller server, you can verify your installation by submitting a query and checking its status.

Prerequisites:

- You must set up Query Patroller server by issuing the **qpsetup** command. This must be issued prior to verifying the installation of Query Patroller server. You have the choice of either having the **qpsetup** command be issued automatically during the installation, or issuing it manually after the installation at a UNIX shell prompt or a Windows command prompt.
- Make sure the DB2 instance that the database belongs to is running on the server.

Procedure:

To verify your Query Patroller server installation:

1. Log on to the database that is managed by Query Patroller as a user with DBADM authority.
2. Start Query Patroller by issuing the **qpstart** command:

   ```
   qpstart dbname
   ```

   where *dbname* is the name of the database that is managed by Query Patroller. You should receive output that is similar to the following:

   ```
    IBM DB2 Query Patroller V8, (c) Copyright IBM Corp. 1998-2003.
    All rights reserved.
    Initializing.
    Query Patroller started.
   ```

   Note: On UNIX-based operating systems the **qpstart** command will release control back to the UNIX shell prompt. You do not need to open up a second UNIX shell prompt to continue with the rest of this procedure. On Windows the control is released back to the Windows command prompt immediately.
3. Enable the *dyn_query_mgmt* database configuration parameter by issuing the following command:

   ```
   db2 update db cfg for dbname using DYN_QUERY_MGMT enable
   ```

   where *dbname* is the name of the database that is managed by Query Patroller.
4. Submit a query that will be intercepted by Query Patroller server. This query must have an estimated cost of 15000 timerons or more. The estimated cost

must be 15000 timerons or more because queries with costs lower than this are not managed by Query Patroller. This is determined by the default Query Patroller parameters. For example, submit the following query::

```
db2 select count(*) from syscat.tables,syscat.tables,syscat.tables
```

Wait for the query to return. If you find that the cost of this query is not high enough (it is less than 15000 timerons), then append enough tables to the command until the cost is greater than the minimal cost of 15000 timerons. For example:

```
db2 select count(*) from syscat.tables,syscat.tables,syscat.tables,...
```

The above query is just an example. You can choose to submit a query that is more appropriate to your particular system. However, ensure that the query has a cost that is greater than 15000 timerons.

5. Check the status of the query that you submitted by entering the following command:

```
qp -d dbname list queries
```

where *dbname* is the name of the database that is managed by Query Patroller. If the query has completed successfully, you should receive output that is similar to the following:

```
  ID     Status     Created                    Completed
  ======================================================================
  1      Done       2003-10-30 18:36:37.615000 2003-10-30 18:36:37.615000
```

If the query was aborted, you should receive output that is similar to the following:

```
  ID     Status     Created                    Completed
  ======================================================================
  1      Aborted    2003-10-30 18:36:37.615000 2003-10-30 18:36:37.615000
```

You can find out why the query was aborted by issuing the following:

```
qp -d dbname get query 1
```

Under ″Message Description″ you will see the error that caused the query to be aborted. Investigate the error and resubmit the query.

6. Stop the Query Patroller service by entering the **qpstop** command. You should receive output that is similar to the following:

```
 IBM DB2 Query Patroller V8, (c) Copyright IBM Corp. 1998-2003.
 All rights reserved.
 Stopping Query Patroller.
 Query Patroller stopped.
```

Related concepts:

- Chapter 2, "Query Patroller installation environment overview," on page 11

Related tasks:

- "Installing Query Patroller server using the DB2 Setup wizard (Windows)" on page 55
- "Installing Query Patroller server using the DB2 Setup wizard (UNIX)" on page 21
- "Setting up Query Patroller server manually" on page 77
- "Migrating Query Patroller server manually" on page 80
- Chapter 3, "Installing Query Patroller (UNIX)," on page 15

Installing the Query Patroller client tools with the DB2 Setup wizard (Windows)

Installation requirements for the Query Patroller client tools (Windows)

This topic lists the operating system, software, and communication requirements for the Query Patroller client tools on Windows.

Operating system requirements

One of the following:

- Windows 98
- Windows ME
- Windows NT Version 4.0 with Service Pack 6a or later
- Windows NT Server 4.0, Terminal Server Edition (only supports the DB2 Run-Time Client) with Service Pack 6 or later for Terminal Server
- Windows 2000
- Windows XP (32-bit and 64-bit editions)
- Windows Server 2003 (32-bit and 64-bit editions)

Software requirements

- A DB2 product must be installed on the computer that you will install the Query Patroller client tools on. The DB2 product must be either Version 8.1.2 with FixPak 2+, or a product that is later than Version 8.1.2. The following products are appropriate prerequisites:
  - Any DB2 client product (for example, DB2 Run-Time client or DB2 Application Development client).
  - Any DB2 Connect product (for example, DB2 Connect Personal Edition or DB2 Connect Enterprise Server Edition).
  - Any DB2 server product (for example, DB2 Enterprise Server Edition or DB2 Workgroup Server Edition).
- For 32-bit environments you require the IBM Software Development Kit (SDK) Version 1.3.1 or Version 1.4.1 to run Query Patroller server, the Query Patroller Java-based tools (such as the Query Patroller Center), and to create and run Java applications, including stored procedures and user-defined functions. During the installation process, the IBM Software Development Kit (SDK) will be installed if it is required.
- For 64-bit environments you require the IBM Software Development Kit (SDK) Version 1.4.1 to run Query Patroller server, the Query Patroller Java-based tools (such as the Query Patroller Center), and to create and run Java applications, including stored procedures and user-defined functions. During the installation process, the IBM Software Development Kit (SDK) will be installed if it is required. For the latest DB2 Java application development updates, visit the Web page at: http://www.ibm.com/software/data/db2/udb/ad/v8/java.
- Netscape 6.2 or Microsoft Internet Explorer 5.5 is required to view the online installation help.

Communication requirements

- TCP/IP.

Related concepts:

- Chapter 2, “Query Patroller installation environment overview,” on page 11

Related tasks:

- “Installing the Query Patroller client tools using the DB2 Setup Wizard (Windows)” on page 69
- Chapter 6, “Installing Query Patroller (Windows),” on page 51

Related reference:

- “Memory requirements for the Query Patroller client tools (Windows)” on page 68
- “Disk requirements for the Query Patroller client tools (Windows)” on page 69

Memory requirements for the Query Patroller client tools (Windows)

The following list outlines the recommended minimum memory requirements for running the Query Patroller client tools on a DB2 client (Windows):

- To run the Query Patroller client tools on a system administration workstation requires an additional amount of 64 MB of RAM beyond the amount of RAM required to run your Windows operating system. For example, to run Query Patroller Center on a system administration workstation running Windows 2000 Professional, you need a minimum of 64 MB of RAM for the operating system plus an additional amount of 64 MB of RAM for the tools.
- The amount of memory required to run the Query Patroller client tools on a DB2 client that submits queries to the Query Patroller server depends on:
  - the Windows operating system you are using and
  - the database applications you are running.

  It should be sufficient to use the minimum memory requirements of the Windows operating system as the minimum requirements for running these tools on a DB2 client.

Note: Performance may be affected if less than the recommended minimum memory requirements are used.

Related concepts:

- Chapter 2, “Query Patroller installation environment overview,” on page 11

Related tasks:

- “Installing the Query Patroller client tools using the DB2 Setup Wizard (Windows)” on page 69
- Chapter 6, “Installing Query Patroller (Windows),” on page 51

Related reference:

- “Installation requirements for the Query Patroller client tools (Windows)” on page 67
- “Disk requirements for the Query Patroller client tools (Windows)” on page 69

Disk requirements for the Query Patroller client tools (Windows)

The actual fixed disk requirements of your installation may vary depending on your file system and the Query Patroller client tools you install. Ensure that you have included a disk space allowance for your application development tools (if necessary), and communication products.

When you install the Query Patroller client tools using the DB2 Setup wizard, size estimates are dynamically provided by the installation program based on installation type and component selection.

Related concepts:

- Chapter 2, “Query Patroller installation environment overview,” on page 11

Related tasks:

- “Installing the Query Patroller client tools using the DB2 Setup Wizard (Windows)” on page 69
- Chapter 6, “Installing Query Patroller (Windows),” on page 51

Related reference:

- “Installation requirements for the Query Patroller client tools (Windows)” on page 67
- “Memory requirements for the Query Patroller client tools (Windows)” on page 68

Installing the Query Patroller client tools using the DB2 Setup Wizard (Windows)

This task is part of the larger task of *Installing Query Patroller (Windows)*.

This task outlines the steps for installing the Query Patroller client tools on a DB2 client for Windows using the DB2 Setup wizard.

Prerequisites:

The following requirements must be met before you begin installing the Query Patroller client tools on a DB2 client for Windows:

- Installation requirements for the Query Patroller client tools (Windows)
- Memory requirements for the Query Patroller client tools (Windows)
- Disk requirements for the Query Patroller client tools (Windows)
- A DB2 product (Version 8.1.2 or later) must be installed on the computer that you will install the Query Patroller client tools on. The following products are appropriate prerequisites:
  - Any DB2 client product (for example, DB2 Run-Time client or DB2 Application Development client)
  - Any DB2 Connect product (for example, DB2 Connect Personal Edition or DB2 Connect Enterprise Server Edition.
  - Any DB2 server product (for example, DB2 Enterprise Server Edition or DB2 Workgroup Server Edition)

Procedure:

To install the Query Patroller client tools on Windows:

1. Log on to the system with the Administrator account you will use to install the Query Patroller client tools.
2. Shut down any other programs so that the DB2 Setup wizard can update files as required.
3. Check the Query Patroller product CD-ROM label to ensure that you are using the CD-ROM with the appropriate language.
4. Insert the CD-ROM into the drive. If enabled, the auto-run feature automatically starts the DB2 Setup Launchpad. From this window you can proceed directly to the installation.
5. The DB2 Setup wizard determines the system language, and launches the setup program for that language. If you want to run the setup program in a different language, or if the setup program fails to auto-start, you can start the DB2 Setup wizard manually.

   To start the DB2 Setup wizard manually:

   a. Click **Start** –> **Run**.

   b. In the **Open** field, enter the following command:

      ```
      x:\setup /i language
      ```

      where:

      - *x:* represents your CD-ROM drive
      - *language* represents the territory identifier for your language (for example, EN for English).

      If the /i flag is not specified, the installation program runs in the default language of the operating system.

   c. Click **OK**.
6. Choose **Install Products** once the DB2 Launchpad opens.
7. Click **Next** on the **Select the product you would like to install** window.
8. On the **Welcome to the DB2 Setup wizard** window, click **Next**.
9. On the **License Agreement** window, select **I accept the terms in the license agreement** if you accept the terms. Click **Next**. If you do not accept the terms, then select **I do not accept the terms in the license agreement**. Click **Cancel** to end the installation.
10. On the **Select the installation type** window select **Computer usage based** and click **Next**.
11. On the **Select how this computer will be used** window select **Administrator or end user with self-service capability** and click **Next**.
12. Proceed by following the setup program's prompts. Online help is available to guide you through the remaining steps. To invoke the online help, click the **Help** button on the bottom right hand corner of each installation window, or press **F1**. You can click the **Cancel** button at any time to end the installation.

You are now ready to configure your Query Patroller tool-enabled DB2 client (Windows) to access a remote Query Patroller server.

Related concepts:

- Chapter 2, "Query Patroller installation environment overview," on page 11

Related tasks:

- Chapter 6, "Installing Query Patroller (Windows)," on page 51

Related reference:

- "Installation requirements for the Query Patroller client tools (Windows)" on page 67
- "Memory requirements for the Query Patroller client tools (Windows)" on page 68
- "Disk requirements for the Query Patroller client tools (Windows)" on page 69

Configuring a client to Query Patroller server connection using the command line processor

This task describes how to configure a connection from a client to a Query Patroller server using the command line processor (CLP).

Prerequisites:

Before you configure a client to Query Patroller server connection:

- Communications must be configured on the client computer. Communications must be TCP/IP.
- Communications must be configured on the Query Patroller server. Communications must be TCP/IP.

Procedure:

To configure a client to Query Patroller server connection using the command line processor:

1. Catalog the TCP/IP node on the DB2 client.
2. Catalog the database on the DB2 client.
3. Test the client to server connection.

Related tasks:

- "Cataloging the TCP/IP node on the DB2 client" on page 36
- "Cataloging a database using the CLP" on page 38
- "Testing the client to server connection using the CLP" on page 40

Cataloging the TCP/IP node on the client

This is part of the main task of *Configuring a client to server connection using the CLP*.

Cataloging the TCP/IP node adds an entry to the DB2 client's node directory to describe the remote node, the chosen node_name, and the hostname. This entry specifies the chosen alias (*node_name*), the *hostname* (or *ip_address*), and the *svcename* (or *port_number*) that the client will use to access the remote host.

Procedure:

To catalog a TCP/IP node, perform the following steps:

1. Log on to the system as a user with System Administrative (SYSADM) or System Controller (SYSCTRL) authority. You can also log on to the system without these authority levels if you have the `catalog_noauth` option set to `ON`.

2. If you are using a UNIX client, set up the instance environment and invoke the DB2 command line processor. Run the start-up script as follows:

   ```
   . INSTHOME/sqllib/db2profile    (for bash, Bourne or Korn shell)
   source INSTHOME/sqllib/db2cshrc (for C shell)
   ```

   where *INSTHOME* is the home directory of the instance.
3. Catalog the node by entering the following commands from a **db2** prompt:

   ```
   catalog tcpip node  node_name remote hostname|ip_address\
      server service_name|port_number\
      [remote_instance instance_name] [system system_name] [ostype  os_type]
   terminate
   ```

   Where:
   - system is the system name of the remote server;
   - ostype is the operating system of the remote server system.

   Specifying the remote_instance, system, and ostype is optional, but recommended for users who want to use the DB2 tools. The *service_name* used on the client does not have to be the same as the one on the server. However, the port numbers that they map to *must* match.

   For example, to catalog the remote host *myserver* on the node called *db2node*, using the service name *server1*, enter the following from a **db2** prompt:

   ```
   catalog tcpip node db2node remote myserver server server1
   terminate
   ```

   To catalog a remote server with the IP address *9.21.15.235* on the node called *db2node*, using the port number *3700*, enter the following from a **db2** prompt:

   ```
   catalog tcpip node db2node remote 9.21.15.235 server 3700
   terminate
   ```

   Note: The **terminate** command is needed to refresh the directory cache.

The next step is to catalog the database on the client.

Related tasks:

- "Configuring TCP/IP communications on the client using the CLP" in the *Installation and Configuration Supplement*

Related reference:

- "CATALOG TCPIP NODE Command" in the *Command Reference*

Cataloging the database and testing the connection

Cataloging a database using the CLP

This task describes how to catalog a database using the CLP.

Before a client application can access a remote database, the database must be cataloged on the client. When you create a database, the database is automatically cataloged on the server with a database alias that is the same as the database name, unless a different database alias was specified. The information in the database directory, along with the information in the node directory (unless cataloging a local database where a node is not needed), is used on the DB2 client to establish a connection to the remote database.

Prerequisites:

Before you catalog the database:

- You require a valid DB2 user ID
- If you are cataloging a database on a system that has a DB2 server or DB2 Connect product installed, the user ID must have System Administrative (SYSADM) or System Controller (SYSCTRL) authority on the instance.
- The following parameter values are applicable when cataloging a *remote* database:
  - Database name
  - Database alias
  - Node name
  - Authentication type (optional)
  - Comment (optional)

  Refer to the Parameter values worksheet for cataloging a database for more information about these parameters.
- The following parameter values are applicable when cataloging a *local* database:
  - Database name
  - Database alias
  - Authentication type (optional)
  - Comment (optional)

  Local databases can be uncataloged and re-cataloged at any time.

Procedure:

To catalog a database on the client, perform the following steps.

1. Log on to the system with a valid DB2 user ID. If you are cataloging a database on a system that has a DB2 server or DB2 Connect server installed, log onto this system as a user with System Administrative (SYSADM) or System Controller (SYSCTRL) authority on the instance.
2. Update the Your Value column in the Parameter values worksheet for cataloging a database.
3. If you are using DB2 on a UNIX platform, set up the instance environment. Run the start-up script as follows:

   ```
   . INSTHOME/sqllib/db2profile    (for bash, Bourne or Korn shell)
   source INSTHOME/sqllib/db2cshrc (for C shell)
   ```

   where *INSTHOME* is the home directory of the instance.
4. Start the DB2 command line processor. You can do this by issuing the **db2** command from a DB2 command window.
5. Catalog the database by entering the following commands in the command line processor:

   ```
   catalog database database_name as database_alias at\
   node node_name authentication auth_value
   ```

   For example, to catalog a remote database called *sample* so that it has the local database alias *mysample*, on the node *db2node*, enter the following commands:

   ```
   catalog database sample as mysample at node db2node\
   authentication server
   terminate
   ```

The next step is to test the client to server connection.

Related tasks:

- "Testing the client to server connection using the CLP" on page 40

Related reference:

- "Parameter values worksheet for cataloging a database" in the *Installation and Configuration Supplement*
- "CATALOG DATABASE Command" in the *Command Reference*

Testing the client to server connection using the CLP

This task is part of the main task of *Configuring a client to server connection using the command line processor (CLP)*.

After cataloging the node and the database, you should connect to the database to test the connection.

Prerequisites:

The following prerequisites apply:

1. The database node and database must be cataloged before you can test the connection.
2. The values for *userid* and *password* must be valid for the system on which they are authenticated. By default, authentication takes place on the server.
3. Start the database manager by entering the **db2start** command on the database server (if it was not already started).

Procedure:

To test the client to server connection:

1. If you are using a UNIX client, run the start-up script as follows:

   ```
   . INSTHOME/sqllib/db2profile    (for Bash, Bourne, or Korn shell)
   source INSTHOME/sqllib/db2cshrc (for C shell)
   ```

   where *INSTHOME* represents the home directory of the instance.
2. Using the CLP, enter the following command on the client to connect to the remote database:

   ```
   connect to database_alias user userid
   ```

   For example, enter the following command:

   ```
   connect to mysample user jsmith
   ```

   You will then be prompted to enter your password.

If the connection is successful, you will receive a message showing the name of the database to which you have connected. A message similar to the following will be given:

```
Database Connection Information
Database server = DB2/NT 8.1.0
SQL authorization ID = JSMITH
Local database alias = mysample
```

You are now able to work with the database. For example, to retrieve a list of all the table names listed in the system catalog table, enter the following SQL command:

```
select tabname from syscat.tables
```

An implicit connection occurs when you issue an SQL statement without being connected to any database. To define a default database, run the **db2set db2dbdft = <dbname>** command. After running this command, you can, for example, run the **db2 get * from <table>** command without first connecting to a database. This command will use the value defined in **db2dbdft**. To connect to a database other than the default, you must use the CONNECT command to explicitly connect to the database of your choice.

When you are finished using the database connection, enter the **connect reset** command to end the database connection.

Related reference:

- "db2start - Start DB2 Command" in the *Command Reference*
- "db2set - DB2 Profile Registry Command" in the *Command Reference*

Chapter 8. Setting up Query Patroller server

Setting up Query Patroller server manually

This task outlines how to manually set up Query Patroller server on a specified DB2 database. The DB2 Setup wizard will automatically run the **qpsetup** command during the Query Patroller server installation (UNIX or Windows). However, if you decide not to automatically run this command during the installation you can manually run it after the installation by issuing the **qpsetup** command from a UNIX shell prompt or from a Windows command prompt. You can use **qpsetup** in the following two situations:

1. You can use **qpsetup** if you install the Query Patroller server files without setting up Query Patroller server.
2. You can use **qpsetup** if you decide to manage a different database after you install Query Patroller server.

The **qpsetup** command performs the following:

- Creates either an SMS or a DMS table space for the Query Patroller control tables, if the specified table space does not exist.
- Creates either an SMS or a DMS table space for the Query Patroller result tables, if the specified table space does not exist.
- Creates Query Patroller control tables on the specified table space, if they do not exist.
- Binds Query Patroller packages to the database.

You need to run the **qpsetup** command for each database that you want to use Query Patroller with.

Restrictions:

- Only one table space container can be specified. This is to avoid an overly-complicated installation process and command line input. If you require more than one container for the table space, then the table space must be created before running the **qpsetup** command.
- You cannot specify the buffer pool. The table space will be created using the default buffer pool, IBMDEFAULTBP. If you require the table space to use a different buffer pool, then the buffer pool and the table space must be created before running the **qpsetup** command.
- Since the table space will be created using the default buffer pool, IBMDEFAULTBP, which has a 4K page size, the page size for the table space will be 4K as well. If you require table space with page sizes other than 4K, the table space and its buffer pool must be created before running the **qpsetup** command.
- If you specify a partition group that spans more than one logical database partition in the CONTROL_DBPARTITIONGROUP and RESULT_DBPARTITIONGROUP parameter of the **qpsetup** command, then the following restriction applies:
  - You can specify a database partition expression for container string syntax when creating either SMS or DMS containers. You would typically specify the database partition expression if you were using multiple logical database partitions in the partitioned database system. This ensures that container names are unique across database partition servers. When you specify the

expression, the database partition number is part of the container name or, if you specify additional arguments, the result of the argument is part of the container name.

.

Prerequisites:

- Query Patroller server must be installed on your computer.
- A DB2 instance must be running on your computer. This is the instance that contains the database that Query Patroller server will monitor.
- You must have SYSADM authority to create new table spaces using the **qpsetup** command.
- You require DBADM authority to use existing table spaces using the **qpsetup** command.

Procedure:

To set up Query Patroller server on a specified DB2 database:

1. Check the appropriate command syntax and command parameters before issuing the **qpsetup** command. For more information see the **qpsetup** command topic in the related links.
2. Issue the **qpsetup** command at a UNIX shell prompt or a Windows command prompt, depending on your operating system.

Related tasks:

- "Installing Query Patroller server using the DB2 Setup wizard (Windows)" on page 55
- "Installing Query Patroller server using the DB2 Setup wizard (UNIX)" on page 21
- "Migrating Query Patroller server manually" on page 80
- "Verifying the installation of Query Patroller server" on page 29
- Chapter 3, "Installing Query Patroller (UNIX)," on page 15
- Chapter 6, "Installing Query Patroller (Windows)," on page 51

Related reference:

- "qpmigrate - Migrate Query Patroller data and settings" on page 222
- "qpsetup - Set up Query Patroller server" on page 224

Chapter 9. Migrating Query Patroller

Automatic migration of Query Patroller server with the DB2 Setup wizard

The DB2® Setup wizard will automatically run the **qpmigrate** command during the Query Patroller server installation (UNIX or Windows). If the DB2 Setup wizard detects Query Patroller Version 7 data and settings on the server, a message will appear during the installation informing you of this fact. You can then decide whether you want the DB2 Setup wizard to automatically migrate the Version 7 data and settings, or whether you want to defer migrating this until after the installation is finished. Ease of use is the primary advantage of having the DB2 Setup wizard automatically run this command during the installation.

The following items get automatically migrated during the Query Patroller server installation:

- Query Patroller system settings
- Query Patroller query classes
- Query Patroller submitter profiles
- Query Patroller operator profiles
- Query Patroller submission preferences files

If you choose *not* to have the DB2 Setup wizard automatically run the **qpmigrate** command during the installation, make sure that these items get migrated after the installation is finished. You can do this by manually migrating these items by issuing the **qpmigrate** command from a UNIX® shell prompt or a Windows® command prompt using the appropriate command syntax and command parameters.

When migrating a Version 7 administrator profile to Query Patroller, Version 8, an operator profile is created. This profile is automatically granted the maximum authority level for all Query Patroller privileges. However, the migrated profile is not automatically granted DBADM authority on the Query Patroller database. This means that the migrated administrator profile does not have privileges to generate historical data or create, update, or remove operator profiles. If the user or users with this migrated profile need to perform these tasks, you must manually grant DBADM authority to the user after migration.

Note: The following items do *not* get migrated by the **qpmigrate** command:

- The Query Patroller Version 7 result table space value. (This value is from the DB2 registry DQP_RES_TBLSPC). This value is not migrated because the Query Patroller Version 8 result table space value should have been correctly set up when the **qpsetup** command was run during the installation process.
- Query Patroller Version 7 settings that are defined on your DB2 clients. These settings are not migrated because the **qpmigrate** command is run on the server computer, not on the client computers. For example, client settings such as DQP_ABORTRESULT and DQP_SHARE are not migrated to the query submission preferences.
- The Query Patroller Version 7 e-mail notification feature. This feature is not migrated because the Query Patroller Version 8 e-mail notification

feature requires different parameters than the corresponding Query Patroller Version 7 feature. You need to manually set up and enable your e-mail notification settings after migration, if desired. For Query Patroller Version 8, the e-mail notification feature is disabled by default.

- The Query Patroller Version 7 group profile named ″PUBLIC″. This group profile is not migrated because in Query Patroller Version 8 a group submitter profile named ″PUBLIC″ is automatically created during installation. It therefore pre-exists and has a meaning that is different from the Version 7 group profile.

Therefore, you may need to further customize and fine tune your DB2 Query Patroller server after automatically running the **qpmigrate** command.

Related tasks:

- "Installing Query Patroller server using the DB2 Setup wizard (Windows)" on page 55
- "Installing Query Patroller server using the DB2 Setup wizard (UNIX)" on page 21
- "Migrating Query Patroller server manually" on page 80

Related reference:

- "qpmigrate - Migrate Query Patroller data and settings" on page 222

Migrating Query Patroller server manually

This task outlines how to manually migrate Query Patroller Version 7 data and settings to the Query Patroller Version 8 system.

The DB2 Setup wizard will automatically run the **qpmigrate** command during the Query Patroller server installation (UNIX or Windows). However, if you decide *not* to automatically run this command during the installation you can manually run it after the installation by issuing the **qpmigrate** command from a UNIX shell prompt or from a Windows command prompt. The primary advantage of issuing this command after the installation is that you can read the command syntax and command parameters of the **qpmigrate** command before you migrate the Query Patroller Version 7 data and settings. You will then have a better understanding of the migration process.

Restrictions:

- The **qpmigrate** command must be run on the server computer, not on the client computers. Therefore, no Version 7 Query Patroller client settings can be migrated (for example, DQP_ABORTRESULT and DQP_SHARE are not migrated to the query submission preferences).
- The Version 8 Query Patroller e-mail notification feature needs different types of parameters than the corresponding Version 7 feature. Therefore the Version 7 e-mail notification feature does not get migrated. You need to manually set up your e-mail notification for Version 8 after migration (if desired).
- Query Patroller server must not be running when you run the **qpmigrate** command.
- You should only run the **qpmigrate** command *once* to migrate your Query Patroller server settings, and (optionally at a later time) *once* to clean up Query Patroller Version 7 control tables and result tables (using the *cleanup* parameter)

Prerequisites:

- You need DBADM authority to run the **qpmigrate** command.
- The DB2 instance and corresponding DB2 database must be migrated from Version 7 to Version 8 before running the **qpmigrate** command.
- Query Patroller Version 8 must be installed on the server.
- The **qpsetup** command must be run prior to running the **qpmigrate** command.
- The DB2 instance that the database belongs to must be running on the server.
- The **qpmigrate** command must be run prior to customizing your Query Patroller system. If you customize your Query Patroller Version 8 system prior to running the **qpmigrate** command, then **qpmigrate** may overwrite previously customized profiles and system parameters. Consider the following:
  - System settings in the QP_SYSTEM table will be overwritten by the migrated values (for those parameters that exist in Version 7).
  - Existing query classes in the QUERY_CLASS table will not be overwritten. Migrated query classes will be added to QUERY_CLASS. If the maximum limit of 99 query classes is reached, then no more query classes will be migrated or added. If a migrated query class has the same max_cost as an existing query class, then it will not be migrated or added.
  - If user profile related entries (such as SUBMITTER_PROFILE, OPERATOR_PROFILE, and SUBMISSION_PREFERENCES) already exist for a DB2 user ID or DB2 group ID, then **qpmigrate** will not overwrite the values from the corresponding Version 7 profile with the same user name or group name.

  Note: There is one exception to this prerequisite: you should inspect and update (if necessary) the settings of the default PUBLIC submission preferences to the desired values before running the **qpmigrate** command. As a result of this update all new query submission preferences that are created by **qpmigrate** will inherit all of the desired values from the default PUBLIC submission preferences.

Procedure:

To migrate a Query Patroller server:

1. Check the appropriate command syntax and command parameters before issuing the **qpmigrate** command. For more information see the **qpmigrate** command topic in the related links.
2. Issue the **qpmigrate** command from a UNIX shell prompt or a Windows command prompt, depending upon your operating system.

After you run the **qpmigrate** command always check the message log file qpmigrate.log for important messages. This log is located in the \sqllib\<instance>\ directory on the Windows operating system, and in the /sqllib/db2dump/ directory on UNIX-based operating systems. These paths are the default locations for Windows and UNIX. You can change these paths by using the DB2 database manager configuration parameter *diagpath* on a Windows command prompt or a UNIX shell prompt.

When migrating a Version 7 administrator profile to Query Patroller, Version 8, an operator profile is created. This profile is automatically granted the maximum authority level for all Query Patroller privileges. However, the migrated profile is not automatically granted DBADM authority on the Query Patroller database. This means that the migrated administrator profile does not have privileges to generate historical data or create, update, or remove operator profiles. If the user or users with this migrated profile need to perform these tasks, you must manually grant DBADM authority to the user after migration.

Related tasks:

- "Installing Query Patroller server using the DB2 Setup wizard (Windows)" on page 55
- "Installing Query Patroller server using the DB2 Setup wizard (UNIX)" on page 21
- "Setting up Query Patroller server manually" on page 77
- "Verifying the installation of Query Patroller server" on page 29
- Chapter 3, "Installing Query Patroller (UNIX)," on page 15
- Chapter 6, "Installing Query Patroller (Windows)," on page 51

Related reference:

- "qpmigrate - Migrate Query Patroller data and settings" on page 222
- "qpsetup - Set up Query Patroller server" on page 224

Chapter 10. Next Steps

Now that you have successfully installed Query Patroller, you need to start the Query Patroller server and enable Query Patroller to intercept your queries. Then, you can begin performing administration tasks with Query Patroller.

Starting Query Patroller

You might need to start or stop Query Patroller during normal business operations; for example, you must stop and restart Query Patroller after creating, modifying, or removing query classes for the changes you have made to take effect.

Prerequisites:

- You must have DBADM authority.
- DB2 must be started.

Procedure:

To start Query Patroller, issue the **qpstart** command.

Related tasks:

- "Starting DB2 UDB on UNIX" in the *Administration Guide: Implementation*
- "Starting DB2 UDB on Windows" in the *Administration Guide: Implementation*
- "Stopping Query Patroller" on page 84

Related reference:

- "qpstart - Start Query Patroller" on page 228

Enabling Query Patroller to intercept queries

Before you can use the interception, management, and historical analysis functions of Query Patroller, you must enable Query Patroller to intercept queries by setting the *dyn_query_mgmt* configuration parameter.

Procedure:

To enable Query Patroller to intercept queries, use any of the following methods.

Note: In a partitioned environment, Query Patroller must be enabled for all partitions.

DB2 Control Center method:

1. Open the Database Configuration window.
2. Click on the **Value** for DYN_QUERY_MGMT. A … button appears.
3. Click …. The Change Database Configuration Parameter window opens.
4. Click the **Enable** radio button.
5. Click **OK** to close the Change Database Configuration Parameter window.

6. Click **OK** to close the Database Configuration window and save your changes.

Command line method:

Issue the **UPDATE DATABASE CONFIGURATION** command and set *dyn_query_mgmt* to ENABLE for the database you want Query Patroller to intercept queries from.

Related concepts:

- "Query interception and management in Query Patroller" on page 105

Related reference:

- "dyn_query_mgmt - Dynamic SQL query management configuration parameter" in the *Administration Guide: Performance*

Stopping Query Patroller

You might need to start or stop Query Patroller during normal business operations; for example, you must stop and restart Query Patroller after creating, modifying, or removing query classes for the changes you have made to take effect. After you have stopped Query Patroller, forced queries will be in an inconsistent state until you restart Query Patroller and query recovery is completed.

Prerequisites:

- You must have DBADM authority.
- Query Patroller must be started.

Procedure:

To stop Query Patroller, issue the **qpstop** command.

Related tasks:

- "Starting Query Patroller" on page 83

Related reference:

- "qpstop - Stop Query Patroller" on page 229

Query Patroller administration tasks overview

Once you have completed your installation, set up, and migration, and you have enabled Query Patroller to intercept queries, you can begin to administer and use your Query Patroller system. The following diagram presents an overview of the different Query Patroller post-installation tasks.

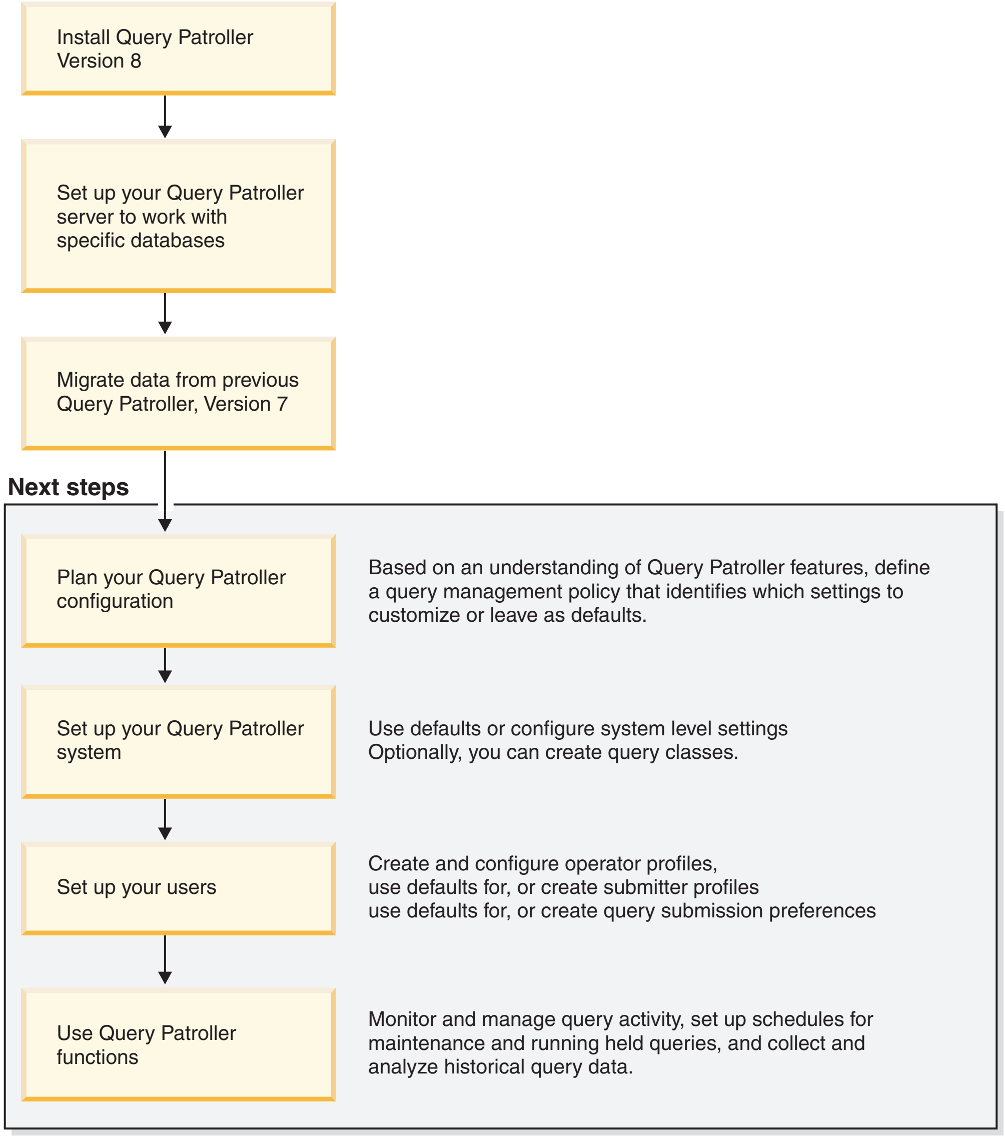

Install Query Patroller Version 8
Set up your Query Patroller server to work with specific databases
Migrate data from previous Query Patroller, Version 7
Next steps
Plan your Query Patroller configuration
Based on an understanding of Query Patroller features, define a query management policy that identifies which settings to customize or leave as defaults.
Set up your Query Patroller system
Use defaults or configure system level settings Optionally, you can create query classes.
Set up your users
Create and configure operator profiles, use defaults for, or create submitter profiles use defaults for, or create query submission preferences
Use Query Patroller functions
Monitor and manage query activity, set up schedules for maintenance and running held queries, and collect and analyze historical query data.

Figure 25. Query Patroller administration tasks overview

Related concepts:

- "Query Patroller query classes" on page 101
- "Query Patroller thresholds" on page 100
- "Query interception and management in Query Patroller" on page 105
- "Query Patroller configuration roadmap" on page 109
- Chapter 2, "Query Patroller installation environment overview," on page 11
- "Definition of your query management policy" on page 107

- Chapter 25, “Query Patroller, Version 8 limitations and restrictions,” on page 185

Related tasks:

- “Updating the list of databases in Query Patroller” on page 124
- “Starting Query Patroller” on page 83
- “Enabling Query Patroller to intercept queries” on page 83
- “Enabling collection of historical data” on page 125

Part 3. Planning query management to solve business problems

Chapter 11. Query Patroller solutions for business problems

This chapter presents a set of scenarios that illustrate the use of Query Patroller features in a fictional business setting. These scenarios depict the query management strategies of Shopmart, a company that operates 34 retail grocery and department stores. Each scenario outlines a specific data warehousing challenge, and describes how Shopmart has addressed the problem by exploiting Query Patroller functions.

Scenario: Managing query submitter needs by configuring submitter profiles

The Shopmart data warehouse has three main types of users: sales representatives, sales managers, and business analysts. Each of these three groups has different requirements with respect to the types of queries that they submit and the response times that they require.

When the sales representatives submit queries, they are often seeking an immediate answer to a customer inquiry. At times, receiving query results quickly can make the difference between winning or losing an important account. The sales representatives use a query application to submit their queries.

The sales managers use the data warehouse primarily to prepare weekly reports. These queries are written with a customized query-building application and are therefore reasonably consistent in terms of the size of the queries and the efficiency of the SQL. Sales managers submit their queries under a time constraint: they must have up-to-date figures compiled in time to present at a weekly meeting.

The business analysts write complex ad hoc queries. When written by an analyst who has limited experience with SQL, these queries sometimes become very large and unwieldy, which can severely impact the performance of the data warehouse. The response time for these queries is not particularly critical, although the results should return within a matter of hours.

To reflect the different characteristics of these groups and their queries, the administrator, Mel, has created three Query Patroller submitter profiles based on the following existing DB2® user groups: Managers, Salesreps, and Analysts.

Because of the rapid response requirements of the sales representatives, the administrator, Mel, has configured the Salesreps submitter profile so that queries from this group of submitters will not be intercepted by Query Patroller. This avoids the performance cost associated with having Query Patroller intercept and manage these queries.

To ensure that the report queries for sales managers execute relatively quickly, Mel has assigned a queue priority of 999 to the Managers submitter profile. Having a priority of 999 means that when a manager’s query is placed in a queue, it is given priority over queries from other submitters. Because the size of the sales manager’s queries is consistent, Mel has left the query cost threshold for this submitter profile as the default.

The business analysts may need to submit queries that are larger than the default cost allowed. However, to prevent the business analysts from submitting queries that are too costly, Mel has set the maximum query cost threshold for the Analysts submitter profile to 100 million. If an analyst submits a query whose cost exceeds this threshold, the query will be held by Query Patroller.

These Shopmart submitter profile settings are summarized in the following table.

Table 1. Shopmart submitter profile settings

Submitter profile	Intercepted by Query Patroller?	Maximum query cost (in timerons)	Queue priority
Salesreps	N	n/a	n/a
Managers	Y	Default (10 000 000)	999
Analysts	Y	100 000 000	Default (500)

Related concepts:

- "Query Patroller thresholds" on page 100
- "Query Patroller submitter profiles" on page 137

Related tasks:

- "Creating submitter profiles for users and groups" on page 141
- "Setting submitter resource limits" on page 143

Scenario: Handling very large queries

Jane is a business analyst examining customer buying patterns in Shopmart stores. She needs to make recommendations regarding the suite of services to be offered at a new Shopmart that is currently under plan.

She suspects that there is an important relationship between the type of fresh produce a customer purchases and the type of in-store services such a customer uses, such as dry cleaning or photofinishing. To explore such a relationship, Jane submits a very large query, involving dozens of tables from different departments.

In order to ensure that an unusually large query from a business analyst does not tie up system resources and prevent smaller queries from running, the system administrator, Mel, has set the Query Patroller thresholds so that extremely large queries from the business analysts will be held. When Query Patroller holds a query that exceeds the size allowed for the business analysts, Query Patroller alerts the submitter that the query is being held. A decision can then be made about whether the query should be run immediately, scheduled for later execution, or canceled. Mel has set up the Shopmart system so that any queries that are still held at the end of the day will be run overnight.

The size of Jane's query exceeds the maximum size specified in the Query Patroller group profile for business analysts. Query Patroller holds the query, and Jane gets an error message from Query Patroller saying that her query is being held. So she calls Quentin from the data warehouse support team.

Quentin examines the query to make sure that the query is not unusually large due to an error in the SQL statement. He decides that the query is legitimately large and therefore does not warrant being canceled. Quentin now has two options for dealing with Jane's held query: he can release the query to run as soon as resources are available, or he can let the query run at the scheduled time for held queries.

Quentin tells Jane that he will allow her query to run even though it exceeds her normal cost threshold. However, to minimize the impact on other users, he explains that he has decided to set the query to run overnight.

Related concepts:

- "Scenario: Managing query submitter needs by configuring submitter profiles" on page 89
- "Submitter profile configuration" on page 113
- "Definition of your query management policy" on page 107

Scenario: Running large emergency queries

There has been an emergency recall of a line of Shopmart healthcare products sold between May and October of last year. The legal department has asked Arun, the pharmacy sales manager, for a detailed report on all sales of these products during this period and on any remaining stock.

Although this means executing a very large query at a time of day when the system is very busy, it is urgent that the query run immediately. Arun notifies Mel, the system administrator, that this query cannot be scheduled for a later time but must run right away.

To safeguard against a single large query crippling the entire system, Mel has set the Query Patroller thresholds in query submitter profiles so that extremely large queries from any group of users, including the sales managers, are held.

Knowing how important Arun's query is, Mel decides to temporarily raise the queue priority value in Arun's submitter profile and asks Arun to wait until she has done this before he submits his query. Raising Arun's queue priority means that if the query is queued, it will become the first query to run as soon as some other query completes.

When submitted, the query is intercepted and held by the system because it exceeds the maximum query cost for Arun's submitter profile. Mel releases the query from a held state to allow it to run as soon as sufficient resources are available.

After the query executes, Mel resets the queue priority value in Arun's submitter profile to its original value.

Related concepts:

- "Submitter profile configuration" on page 113
- "Scenario: Handling very large queries" on page 90
- "Definition of your query management policy" on page 107

Related tasks:

- "Changing submitter profiles for users and groups"--*Help: Query Patroller*

Related reference:

- “UPDATE SUBMITTER_PROFILE” on page 251

Scenario: Managing queries of different sizes using query classes

Head office requires each product line manager to review the corporation-wide sales figures from the previous week to prepare for the Monday afternoon sales meeting. Since the sales figures are not available until after close of business on Saturday, the managers usually run their weekly sales queries between 8:30 and 11:00 a.m. Monday morning.

Even though the system could potentially run all of the weekly sales queries simultaneously, these queries would monopolize system resources and prevent other users from running small queries, such as customer account queries, at the same time. To address this problem, the system administrator, Mel, has set up query classes to allocate system resources to both smaller queries and larger queries.

Query classes are Query Patroller mechanisms for grouping and running queries according to size. For each query class, you can specify the size of the queries in the class and the number of queries that can run concurrently. The queries in each query class are queued separately, so you can control the flow of queries against a database.

Small queries take only seconds to complete, so in setting up the query class for small queries, Mel has not set a limit on the number of queries from this class that can run simultaneously. To limit the amount of resources that the large weekly sales queries can tie up, she has set the maximum number of queries for the query class for large queries to 10. This means that if 15 weekly sales queries are submitted at once, 5 of these will be placed in a queue until some of the running queries in this query class complete.

When Bill, the dairy sales manager, submits his weekly sales query on Monday at 9:55 a.m., there are already 12 other managers trying to run similar queries. Meanwhile, Alphonso, a credit representative, is submitting a query to verify some customer account information.

Because of the query classes that Mel has set up, Alphonso’s small query runs immediately. Bill’s query will be placed in the queue and will run in the large query class once some of the other sales queries complete.

Related concepts:

- “Query Patroller configuration roadmap” on page 109
- “Query class configuration” on page 116
- “Submitter profile configuration” on page 113
- “Definition of your query management policy” on page 107

Related tasks:

- “Creating query classes for Query Patroller” on page 128

Scenario: Using historical analysis to improve performance

The Shopmart data warehouse has grown very large in the past few years, and the database administrator, Mel, is looking for ways to restructure the database and improve performance.

Query Patroller has been collecting query activity data on the data warehouse for the past six months. The historical analysis function uses this data to generate reports on which database objects have been used, by which submitters, and when.

Mel uses the historical analysis Tables Not Hit report to identify tables that have not been accessed by any queries in the past six months. Several of these tables are very large and are located on the most powerful processors in the data warehouse. Because the historical data suggests that these tables are infrequently or never accessed, Mel decides to move them to a slower access device, freeing up space for more heavily-used tables.

Mel also uses another report, Tables Hit, to look at the most frequently accessed tables in the database. Once he has identified the tables that appear to be the most critical to the data warehouse users, he drills down to look more closely at the columns that are accessed to find suitable candidates for indexes. He also looks at the Indexes not hit report to see which indexes are not being used and should be removed.

After Mel has implemented these changes to the Shopmart data warehouse, he can monitor the effect of these changes on query performance. To do this, he examines the historical analysis graphs and reports on query execution time and compares the current average execution time of queries with the average execution time prior to the database changes.

Related concepts:

- “Uses for historical analysis reports” on page 161

Related tasks:

- “Collecting historical data with Query Patroller” on page 163

Chapter 12. Query Patroller background

This chapter provides the conceptual information that you will need to define your query management strategy.

Query processing by Query Patroller

Query Patroller interacts with DB2® to execute several processes when a query is submitted. This topic covers the processing that takes place from the time a query is submitted until the time results are generated.

Query submission:

Query Patroller can intercept queries submitted through a variety of methods, including:

- A query application
- A middle tier business intelligence tool
- The DB2 graphical user interface (dynamically)
- A command line interface

When a query is submitted, DB2 checks the *dyn_query_mgmt* database configuration parameter. If the parameter is set to ENABLE, then Query Patroller captures information about the query such as the SQL statement and the submitter's ID. Query Patroller also evaluates other information at this point, such as the query submitter's minimum cost to manage value for a query (MIN_COST_TO_MANAGE), and the submitter's maximum cost of a query value (MAX_COST_ALLOWED).

Cost analysis:

After a query is submitted, the DB2 query optimizer performs a cost estimation on the query. The query optimizer provides Query Patroller with an estimated cost to run the query, measured in timerons.

Query interception:

Query Patroller determines whether to intercept the query or allow it to run directly against the database.

You can set properties at the system level or at the submitter level to specify which queries Query Patroller intercepts. At the system level, query interception is determined based on the application name. At the submitter level, query interception is determined based on a submitter profile property.

If the query is not intercepted by Query Patroller, the query runs directly against the database without further Query Patroller involvement.

Query management:

If the query's estimated cost is less than the submitter's minimum cost to manage value for a query (MIN_COST_TO_MANAGE), Query Patroller allows the query to run directly against the database.

If the query’s estimated cost is greater than the submitter’s minimum cost to manage value for a query, then Query Patroller assesses, prioritizes, and holds or queues the query.

Query Patroller holds the query under either of the following conditions:

- The estimated cost of the query exceeds the submitter’s maximum cost of a query value (MAX_COST_ALLOWED).
- The estimated cost of the query exceeds the maximum workload cost value for the system (MAX_TOTAL_COST).

Queries that are in a held state will not run unless an administrator or operator manually releases the query, or a scheduled release job runs.

Query Patroller queues the query under any of the following conditions:

- The maximum number of queries value for the submitter (MAX_QUERIES_ALLOWED) has already been reached.
- The maximum number of queries value for the system (MAX_TOTAL_QUERIES) has already been reached.
- The estimated cost of the query plus the current workload cost exceeds the maximum workload cost value for the system (MAX_TOTAL_COST).
- The maximum number of queries value for the query class in which the query runs (MAX_QUERIES) value has already been reached.

Query Patroller routinely checks the query queue to identify queries that are eligible to execute. For example, if a submitter is allowed to submit five concurrent queries, then the sixth query is put in the queued state. However, as soon as one of the other five queries completes, Query Patroller executes the sixth query, provided that the query falls within other system and submitter thresholds.

If query classes have not been defined, the query executes within the default query class. If query classes have been defined, the query executes within the appropriate query class.

Query execution:

DB2 executes the query, and Query Patroller directs the result set to one of two destinations, based on settings in the submitter’s submission preferences, illustrated in Table 2.

Table 2. Result set destinations

Destination	Query Patroller Center method	Command line method
The original DB2 agent process executes the query and returns the result set back to the application that submitted the query.	**Wait until the results are returned** is selected in the Query Submission Preferences window.	UPDATE SUBMISSION_PREFERENCES or ADD SUBMISSION_PREFERENCES command is issued with a value of ’A’ for the RESULT_DESTINATION parameter.

Table 2. Result set destinations (continued)

Destination	Query Patroller Center method	Command line method
A new DB2 agent process is created to execute the query and store the result set in a result table.	**Release the application and retrieve the results from a result table** is selected in the Query Submission Preferences window	UPDATE SUBMISSION_PREFERENCES or ADD SUBMISSION_PREFERENCES command is issued with a value of ’T’ for the RESULT_DESTINATION parameter.

Query executions can be canceled or put in background any time from submission until they have been completed.

Notification:

If the submitter indicated that the query’s result set is to be directed to a result table and e-mail notification has been set up for the submitter, Query Patroller sends e-mail notification to the submitter when the result table is created.

The following diagram depicts the query processing described in this topic.

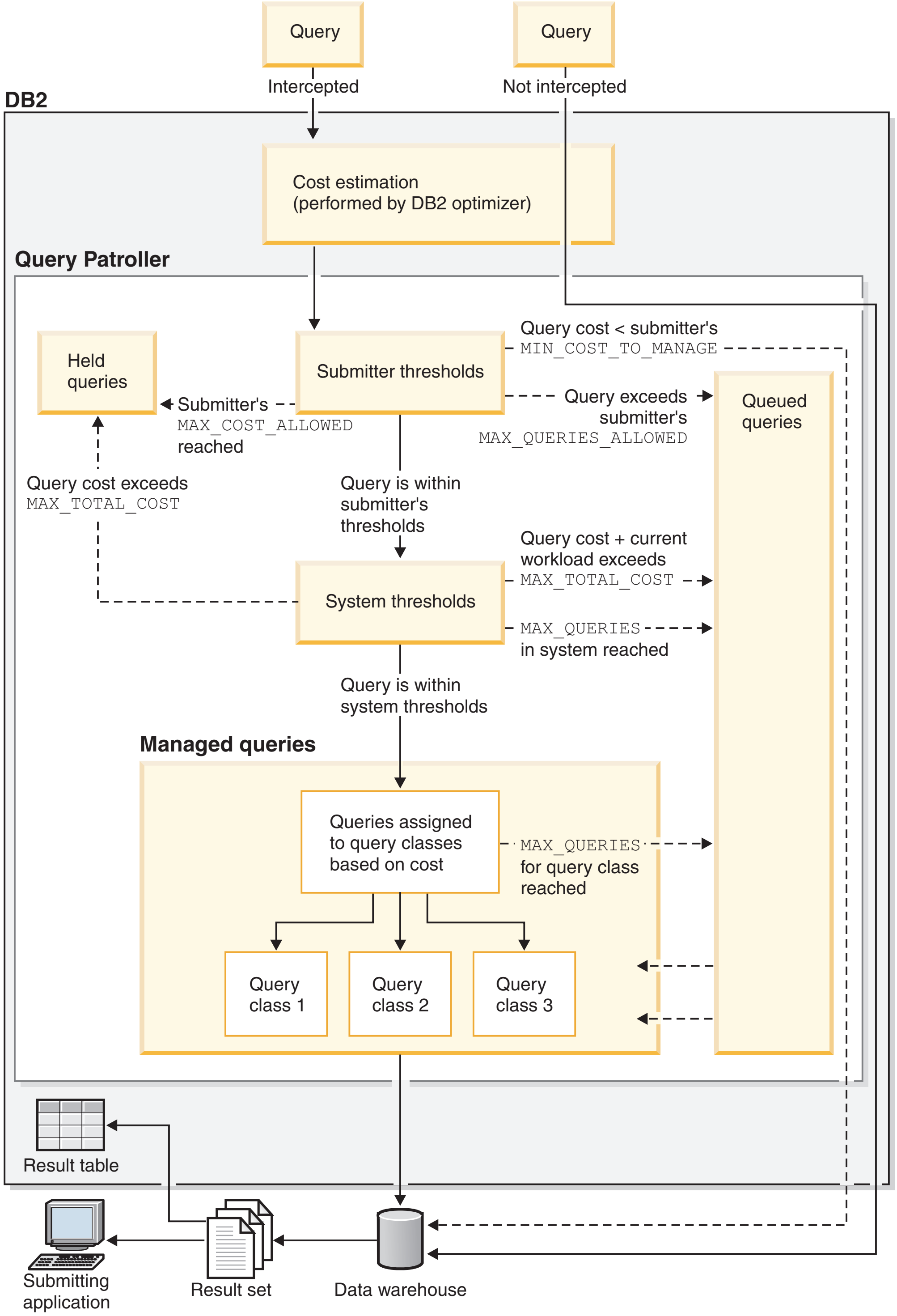


Figure 26. Query processing by Query Patroller

Related concepts:

- "Query Patroller configuration roadmap" on page 109
- "Query class configuration" on page 116
- "Submitter profile configuration" on page 113
- "Cost estimation in Query Patroller" on page 99

- “Query Patroller” on page 3
- “Query Patroller components” on page 4

Related reference:

- “UPDATE SUBMISSION_PREFERENCES” on page 248
- “Query Patroller system threshold settings” on page 255
- “Query interception settings” on page 257
- “E-mail notification settings” on page 261
- “ADD SUBMISSION_PREFERENCES” on page 198

Cost estimation in Query Patroller

Query Patroller evaluates each query that it intercepts based on the estimated size, or cost, of the query. The estimated cost of a query is the estimated total system resources that will be used in the execution of the query. A number of key Query Patroller thresholds are set based on the estimated cost of queries. On Query Patroller graphical user interfaces, this estimated cost is simply referred to as *cost*, for example, ″Maximum cost of a query.″

Query Patroller relies on DB2® for information on the estimated cost of queries. For any query submitted to DB2, there might be many different methods, or access plans, for retrieving the data from all of the required tables. An access plan is the data access strategy, which includes index usage, sort methods, locking semantics, and join methods.

The DB2 query optimizer analyzes the different access plans for a query to determine which of these plans is the most efficient in terms of estimated resource cost. In calculating the estimated cost of an access plan, the query optimizer considers both CPU cost (the number of instructions) and I/O (the number of seeks and page transfers). You can capture the details of the access plan that the query optimizer has selected to execute a query. These details are stored in explain tables, which can be queried or displayed using the DB2 explain facility.

The DB2 unit of measure for cost is the timeron. A timeron does not directly equate to any actual CPU execution time, but gives a relative measure of the estimated resources required to execute a particular query according to the selected access plan. Although the estimated cost of a query in timerons does not predict the execution time of that query, it can provide a basis for comparing two queries in the same environment under the same workload. For example, given the same external factors such as hardware capacity and concurrent workload, a very costly query involving complex joins and large table scans can be expected to have a longer execution time than a cheaper, simpler query.

DB2 passes on to Query Patroller the estimated cost of executing a query. Query Patroller uses this information to determine whether or not the query exceeds certain cost thresholds that have been defined for the system.

Note: To calculate the estimated cost of a query, the query optimizer relies on current table statistics. For this reason, it is recommended that you run the RUNSTATS command periodically to give the query optimizer the most accurate information on which to base its calculations.

Related concepts:

- “Optimizer use of distribution statistics” in the *Administration Guide: Performance*

- “SQL explain facility” in the *Administration Guide: Performance*
- “Explain tools” in the *Administration Guide: Performance*
- “The explain tables and organization of explain information” in the *Administration Guide: Performance*

Query Patroller thresholds

Query Patroller relies on resource thresholds to determine the flow of the workload that can run against a database. Thresholds can be set to control the number and size of queries run by a particular submitter or group, to control the size of the overall system workload, or both. Depending on the characteristics of your particular system and workload, you might want to set some or all of these thresholds. The main types of thresholds are submitter thresholds and system thresholds.

Submitter thresholds

Submitter thresholds are set in submitter profiles, so thresholds can be set for an individual submitter or group of submitters. Each individual user can belong to multiple groups, therefore a user may have multiple group profiles as well as an individual submitter profile. When a user has multiple profiles that they can submit queries under, the appropriate threshold setting is chosen based upon a set of rules that select the effective profile for the submitter. For information on how Query Patroller chooses which submitter profile to use for a submitter, see the Query Patroller submitter profiles topic.

The maximum cost threshold (the MAX_COST_ALLOWED parameter) for a submitter determines the maximum cost of query that a submitter can run. If a submitter tries to run a query whose estimated cost exceeds the maximum cost for his or her profile, then the query is held.

You might want to set this threshold if you have problems with runaway queries tying up system resources. By setting a size limit on individual queries, you can separate unreasonably large queries before they execute, and hold these queries so that you can decide to run them or cancel them on a case-by-case basis.

The maximum number of queries value for a submitter (MAX_QUERIES_ALLOWED) determines the number of queries that can be run simultaneously by a particular submitter. If a submitter tries to run a query when the maximum number of queries specified in his or her submitter profile is already running, the query will be queued until one of the queries completes.

You might want to set this threshold if you are allowing a submitter group to submit large queries, but you want to limit each individual submitter to a few queries. Setting this threshold can also address the problem of users resubmitting the same costly queries in rapid succession because the response time is slower than they expect.

You may want to set one or both submitter thresholds if you need to control the use of resources by particular users or groups. For example, if the data warehouse is primarily funded by one department, but another department is permitted limited use of the resources, you might want to set submitter thresholds so that the users or groups in the second department are restricted in the amount of resources they can tie up with their queries. The thresholds for the users and groups in the funding department can be more liberal in the amount of resources allowed for each submitter.

System thresholds

You can control the overall workload that executes against your database by setting system thresholds for cost (the MAX_TOTAL_COST parameter) and number of queries (the MAX_TOTAL_QUERIES parameter).

The maximum workload cost value for the system (MAX_TOTAL_COST parameter) determines the maximum size of the overall workload running against the database. The workload cost is calculated by adding the cost estimates of all the Query Patroller managed queries currently running in the system. If the execution of a new query will cause the aggregate cost of all queries currently running to exceed the maximum workload cost, the new query is placed in a queued state until the system can run the new query without exceeding the maximum workload cost.

You might want to set this threshold if you find that your CPU utilization is uncomfortably high when running your current workload, and you want to lower the risk of overloading the system by placing a restriction on workload size.

The maximum number of queries value for the system (MAX_TOTAL_QUERIES parameter) places a limit on the number of queries that can be running simultaneously in the system. When this threshold is reached, additional queries are placed in a queued state where they wait until the system can run the queries without exceeding the maximum number of queries value.

You might want to set this threshold if you have a need to limit the number of concurrent queries in order to avoid overloading the system.

Related concepts:

- “Query interception and management in Query Patroller” on page 105
- “Query Patroller submitter profiles” on page 137

Related tasks:

- “Setting query thresholds for the Query Patroller system” on page 123
- “Setting submitter resource limits” on page 143

Query Patroller query classes

Query classes are Query Patroller mechanisms for grouping and running queries according to size. Using query classes, you can control the flow of queries on a database so that system resources are shared among queries in the different size groupings. For example:

- You can ensure that smaller queries do not get stuck behind larger ones by specifying one query class for small queries and one for large queries. The small queries then run in their own query class and are queued separately from the large queries.
- You can prevent large queries from monopolizing system resources by specifying the number of queries in a single query class that can run at once, thus limiting the number of large queries that are processed concurrently.

Query classes categorize managed queries according to the estimated cost of each query. Each query class that is defined for a particular Query Patroller system accepts managed queries whose estimated cost falls within a specified range. For example, if query class 1 accepts queries costing up to 10 000 timerons, a query whose estimated cost is 7500 timerons will run in that class.

Each query class can be configured to limit the number of queries running concurrently in that query class. For example, query class 2 can be configured to allow 10 queries to run simultaneously by setting the maximum number of queries (MAX_QUERIES parameter) value for this query class to 10. Once this limit is reached, any additional queries that fit into that class are queued until one of the running queries in that class completes.

The upper limit for a query class is set by the maximum query cost (MAX_COST parameter) value for the query class. While you do not explicitly define a lower limit for a particular query class, a lower limit is enforced through the interaction of the following two factors:

1. A query always runs in the query class with the lowest estimated cost limit possible. For example, although a query with an estimated cost of 1200 timerons could run in either query class 1 with a MAX_COST value of 10 000 or query class 2 with a MAX_COST of 100 000, it will always run in query class 1. This means that query class 2 will never receive queries smaller than 10 001 timerons.
2. Each submitter profile contains a value for the minimum cost to manage a query (MIN_COST_TO_MANAGE parameter) that specifies the lowest cost a query can have in order for it to be managed by Query Patroller. For example, if the minimum query cost for the submitter profile defined for the managers group is 1000 timerons, any queries with an estimated cost smaller than 1000 timerons submitted by a member of this group will not be put into a query class. Instead, these queries will run without being queued.

Each query class defined within your system must have a unique value for the maximum cost of each query (MAX_COST) parameter.

Each query class is stored along with its parameter values as a row in the QUERY_CLASS control table.

Default query class

Every system will have a default query class even if no query classes have been defined by the administrator. The maximum query cost for this default query class is unlimited.

A query will run in the default query class under two conditions:

- The estimated cost of the query is less than or equal to the system maximum query cost (MAX_TOTAL_COST parameter) but larger than the maximum query cost for any user-defined query class. In cases such as this where query classes have been defined, only a single query can run at a time in the default query class.
- When no query classes have been defined by the administrator, all queries will run within the default query class. In cases such as this where no query classes have been defined, an unlimited number of queries can run at once in the default query class.

Note: The number of queries running simultaneously in this class can be indirectly constrained by the system-wide limit on the total number of queries in the system (MAX_TOTAL_QUERIES parameter).

Related concepts:

- "Query class configuration" on page 116
- "Scenario: Managing queries of different sizes using query classes" on page 92

Related tasks:

- "Creating query classes for Query Patroller" on page 128
- "Configuring query classes" on page 127

Related reference:

- "ADD QUERY_CLASS" on page 196

Query Patroller historical analysis

You can use the historical analysis functions provided by Query Patroller to analyze various aspects of your data warehouse use over time. This allows you to gather information for business reports, performance tuning (through the optimization of Query Patroller system thresholds), and the identification of redundant database objects.

You can find out whether certain tables, indexes, and columns are being used, and by which query submitters. From information such as the number of queries run and query execution time, you can see how resource usage varies by month, week, day, hour, or minute.

By using the Query Patroller historical analysis functions, you might be able to identify trends in data warehouse usage:

- The table, index, and column reports can help you identify which database objects are not being used and can therefore be removed to conserve storage space.
- Query reports by submitter, showing the number and size of queries, can help you identify which submitter groups are the most significant consumers of system resources.
- Reports on the number of queries run against the data warehouse over time can help you identify peak traffic times and off-peak times so that you can schedule maintenance tasks during off-peak times.
- Reports on query execution over time can help you identify how various system, hardware, and configuration changes have impacted performance.

By default, all successfully completed queries managed by Query Patroller are included in historical analysis. Additionally, you can include in your historical analysis successfully completed queries that were intercepted but not managed by Query Patroller.

In order to have the most recent information available for historical analysis, you must generate historical data. Generating historical data runs the SQL Explain facility against the queries that Query Patroller has saved for historical analysis and makes the information available in the Query Patroller historical analysis reports and graphs.

You can remove queries from historical analysis as you see fit by scheduling a regular purge of historical queries or by removing queries individually.

Related concepts:

- "Query interception and management in Query Patroller" on page 105
- "Getting started with the Query Patroller historical analysis interface" on page 295
- "Uses for historical analysis reports" on page 161

Related tasks:

- "Viewing historical query details using Query Patroller" on page 166
- "Managing historical queries" on page 170
- "Generating historical data using Query Patroller" on page 164
- "Collecting historical data with Query Patroller" on page 163
- "Enabling collection of historical data" on page 125

Result tables and result sets in Query Patroller

Each time Query Patroller executes a query, the rows returned form a result set. Query Patroller returns result sets to a result destination, which can be either the application through which the query was submitted or a result table.

A result table is a table that is created by Query Patroller to store the result set. A result table is created when:

- A query is held for any reason, then released and run to completion
- The submitter specifies in his submission preferences that a result table should be created
- A query is run in the background

When Query Patroller is installed, you must specify the table space where result tables will be stored. The name of the table space is stored in the Query Patroller system properties. You can later change or remove the table space name that is specified. If you remove the table space name specified in the Query Patroller system properties, Query Patroller will use a default table space.

The size of result tables can be limited for each submitter in the submitter's properties by restricting the number of rows that can be returned to the result table. Result tables consume storage space, so they should be removed periodically. You can remove result tables manually or you can schedule a purge job.

Query submitters can configure their submission preferences to handle result tables in certain ways:

- They can allow other Query Patroller users to view their result tables
- They can specify whether they want Query Patroller to return no results or truncated results if a query's result set is longer than the maximum allowed.

  Note: If no results are returned, the query is considered to have failed (the query status is Aborted). If the query results are truncated, the query is considered to have completed successfully (the query status is Done; the result status is Truncated).

Related tasks:

- "Setting Query Patroller maintenance schedules for queries and result tables" on page 169
- "Dropping result tables manually using Query Patroller" on page 173

Query interception and management in Query Patroller

After queries are submitted, Query Patroller performs two stages of evaluation on them to determine the level of interaction Query Patroller will have with each query.

1. Query Patroller first evaluates the queries to determine if they meet specific criteria that allow them to bypass Query Patroller. Unintercepted queries are not managed, nor are they used for the collection of data for historical analysis.
2. Query Patroller then evaluates the intercepted queries to determine how each query will be handled. Depending on Query Patroller's evaluation of a query, one of the following occurs:
   - The query is both managed and historical data is collected for it.
   - The query is not managed, but historical data is collected for it.
   - The query is neither managed, nor is historical data collected for it.

Note: In order for Query Patroller to perform these evaluations and subsequent query interception and management, the *dyn_query_mgmt* database configuration parameter must be set to ENABLE.

Query interception

Query Patroller intercepts a query if the query meets the following criteria:

- The query is from an application whose queries you have specified you want to intercept in the Query Patroller system properties.
- The query is from a submitter whose submitter profile indicates that Query Patroller should intercept queries from the submitter.

If Query Patroller intercepts the query, it then evaluates the query to see if the query should be managed. Query Patroller can be configured to collect data for historical analysis on intercepted queries even if they do not meet the criteria for queries that should be managed.

Query management

Query Patroller manages an intercepted query based on the properties that have been set for the submitter of the query and on the Query Patroller system settings. Query Patroller also uses the query's estimated cost, which is assessed by the DB2® query optimizer. The management functions that Query Patroller performs on a query include prioritizing it (if applicable), assigning it to a query class (if applicable), and then either running, queuing, holding, or rejecting the query. If the cost of the query is smaller than the value specified in the submitter's profile for the minimum cost to manage a query (MIN_COST_TO_MANAGE), Query Patroller will not manage the query. Table 3 shows the differences in how Query Patroller treats managed and unmanaged queries.

Table 3. Query Patroller treatment of managed versus unmanaged queries

Query Patroller action	Managed queries	Unmanaged queries
Saves the query in the Query Patroller Center's Managed Queries folder	Yes	No

Table 3. Query Patroller treatment of managed versus unmanaged queries (continued)

Query Patroller action	Managed queries	Unmanaged queries
Saves the query in the Query Patroller Center's Historical Analysis folder	Yes	Optional
Prioritizes the query	Yes	No
Assigns the query to the appropriate query class	Yes	No
Runs, queues, holds, or rejects the query based on various thresholds	Yes	No
Returns the result set to the submitter's application, or creates a result table	Yes	No
Sends e–mail notification to submitter if a result table is created	Optional	No

Unintercepted queries

Query Patroller can be configured not to intercept queries from specific applications, or from certain submitters. Unintercepted queries are not managed by Query Patroller, so you cannot collect historical analysis data on them.

Related concepts:

- "Query Patroller query classes" on page 101
- "Query Patroller historical analysis" on page 103
- "Query Patroller thresholds" on page 100
- "Query processing by Query Patroller" on page 95

Chapter 13. Defining your query management strategy

Before you begin to use Query Patroller query management features, you need to define a strategy that is based on the characteristics of your workload, your user requirements, and the amount of system resources that you have available.

This chapter outlines the different decision points that you will come to in defining your query management strategy and describes the different business goals that can be set in your plan.

Definition of your query management policy

You can use Query Patroller to implement a query handling policy, specifying exactly how queries are handled from the time they are submitted. As part of such a policy, you need to define exactly which queries are intercepted or managed and which queries are left to run without interference. You can also specify the circumstances under which queries are held or queued based on user or system resource thresholds.

Setting up a query handling policy involves several decision points:

1. Deciding which types of queries will be intercepted and managed by Query Patroller
2. Deciding whether to limit overall database workload
3. Deciding which queries will be held and when
4. Devising a policy for handling held queries
5. Deciding whether to use query classes to manage queries of different sizes

Decide which types of queries will be intercepted and managed by Query Patroller:

Whenever Query Patroller intercepts a query to perform a cost estimate, this uses system resources. It also impacts the response time of the query. Similarly, once Query Patroller has intercepted a query, managing that query (assigning it to a query class, queuing it, checking the submitter priority, and so on) requires system resources and impacts the response time for the query. You need to weigh the benefits of intercepting and managing a query or group of queries against the impact that the overhead will have on performance and on the availability of system resources.

Given these considerations, you may not want Query Patroller to intercept certain kinds of queries:

- Queries that require almost instantaneous response, such as queries submitted by OLTP (online transaction processing) applications
- Queries that are submitted by users whose requirement for rapid response outweighs the benefits of interception and tracking by Query Patroller

You might want to roll out Query Patroller in stages and begin by intercepting queries from only a subset of submitters or applications. That way, you can test out your initial configuration without impacting all of your users.

Also, you may not want Query Patroller to manage queries that are too small to be a major impact on system performance. By setting a minimum query cost threshold for Query Patroller to manage queries, you can specify immediate processing of queries that are intercepted by Query Patroller but are estimated to be relatively small. This threshold is set for individual submitters or groups of submitters in their submitter profiles. Query Patroller provides a default value for this setting, but if you decide to customize this value for different submitters, you will need to determine an appropriate minimum query cost for management in your system.

Decide whether to limit overall managed query workload:

You have the option of setting a limit on the total number of managed queries that run concurrently as well as the total aggregate cost of all managed queries running concurrently. By default, these Query Patroller settings are set to unlimited. If you decide to control the overall managed query workload, you need to determine what the optimum workload is for your system.

Decide which queries will be held and when:

You can specify a cost limit on the size of queries submitted by individuals or groups of submitters. Queries exceeding this limit will be held by Query Patroller.

Query Patroller provides defaults for these cost limits on queries that you can use as a starting point, but if you decide to customize these settings, you will need to determine the appropriate query cost limits to set for the different submitters in your system.

Devise a policy for handling held queries:

You need to decide how queries will be handled once they have been held. For example, you will need to decide whether held queries will be individually screened by an administrator or operator to see if it reasonable to run the queries despite their size or whether they will be canceled automatically for exceeding size thresholds. You also have the option to run held queries on a schedule.

If you decide to run held queries on a schedule, you need to decide what that schedule should be, taking into consideration factors such as periods of peak database usage and maintenance activities.

Decide whether to use query classes to manage queries of different sizes:

Query classes can help you control the flow of queries against your database. If you need to limit the number of large queries running concurrently in your system, or you want to prevent smaller queries from being queued behind larger queries, you can create query classes to run different size queries.

Related concepts:

- "Query Patroller configuration roadmap" on page 109
- "Query class configuration" on page 116
- "Submitter profile configuration" on page 113

Related tasks:

- "Configuring query classes" on page 127
- "Configuring submitter profiles" on page 140

Query Patroller configuration roadmap

You can use Query Patroller to help you address specific business goals, such as preventing runaway queries from monopolizing your system. Table 4 indicates the specific configuration settings that apply to different business goals.

Table 4. Road map to configuration information

Goal	Parameter to set	Topics to read
To prevent runaway queries or limit the size of queries from particular submitters	Maximum cost of a query allowed (MAX_COST_ALLOWED) in submitter profiles	Submitter profile configuration
To limit the number of queries from a particular submitter or group	Maximum number of queries allowed (MAX_QUERIES_ALLOWED) in submitter profiles	Submitter profile configuration
To set the optimum number of concurrent queries in the system	Maximum total queries in system (MAX_TOTAL_QUERIES) in system settings	Query Patroller thresholds
To limit the amount of system resources used for all queries or for intercepted queries	Maximum system workload cost (MAX_TOTAL_COST) in system settings	Query Patroller thresholds
To allow the queries from some submitters or groups to bypass Query Patroller	Select do not intercept option (INTERCEPT set to 'N') in submitter profiles	Submitter profile configuration
To allow the queries from some applications to bypass interception by Query Patroller	Specify which applications to not intercept (INTERCEPT_APPLICATION and EXCLUDE_APPLICATIONS) in system settings	Setting query thresholds for the Query Patroller system
To specify particular applications for Query Patroller to intercept queries from	Specify which applications to intercept (INTERCEPT_APPLICATION and INCLUDE_APPLICATIONS) in system settings	Setting query thresholds for the Query Patroller system
To run small queries without Query Patroller managing them	Minimum cost to manage (MIN_COST_TO_MANAGE) in submitter profiles	Submitter profile configuration
To prioritize queries of particular users when these queries are in the queue	Queue priority (PRIORITY) in submitter profiles	Submitter profile configuration
To prevent small queries from getting queued behind large queries	Set up query classes based on grouping queries by size	Query class configuration
To limit the number of concurrent large queries	Set up a query class for large queries	Query class configuration
To manage disk space used by result tables and historical query information	Set up purge jobs for removing result tables, managed query information and historical query information	Setting Query Patroller maintenance schedules for queries and result tables

Related concepts:

- "Query Patroller thresholds" on page 100

- “Query interception and management in Query Patroller” on page 105
- “Query class configuration” on page 116
- “Submitter profile configuration” on page 113
- “Definition of your query management policy” on page 107

Related tasks:

- “Setting query thresholds for the Query Patroller system” on page 123
- “Setting Query Patroller maintenance schedules for queries and result tables” on page 169

Chapter 14. Configuring Query Patroller to implement your query management strategy

Once you have defined your strategy for managing queries with Query Patroller, you need to decide which specific parameters you need to configure to achieve your goals, and which parameters you can leave as defaults.

You might decide to run Query Patroller with the default settings in place for a trial period so you can collect and analyze information about your database activity using the historical analysis feature. After you have collected this data, you should have a better understanding of whether the default settings need to be adjusted, whether you need to configure or create additional submitter profiles, or whether you need to create query classes.

This chapter describes the steps to take in planning your configuration of the Query Patroller system-level parameters, submitter profiles, and query classes. After you have determined the appropriate configuration settings for your environment, you can configure these settings using the steps outlined in Chapter 15, “Administering Query Patroller system settings,” on page 123, Chapter 18, “Administering submitters,” on page 137, and Chapter 16, “Administering query classes,” on page 127.

Query Patroller system configuration

Query Patroller allows you to control the size of the query workload that runs concurrently against your database, either by setting a limit on the number of queries that run concurrently in the system, by setting a limit on the total cost of all queries that run concurrently, or both.

By default, both of these settings are set to unlimited. In most situations, these defaults will be sufficient because submitter-level resource limits and resource limits within query classes will effectively constrain the size of query workload that can run concurrently. It is strongly recommended that you use query classes and submitter-level resource limits to constrain database activity. However, if you wish to configure one or both of these system-level settings, you can use the following heuristics to guide you in determining the best settings for your system.

Setting total queries (MAX_TOTAL_QUERIES):

Your ability to effectively set a limit on the total number of managed queries depends on the nature of your database workload. Specifically, the degree of variation in the size of queries in your workload will determine whether or not it is beneficial to set this threshold.

If your workload is homogenous and includes queries that are similar in terms of estimated cost, you can determine the appropriate setting for this threshold by running test workloads with varying numbers of queries, and monitoring your system performance for how well it responds to the different size workloads.

If your workload is not homogenous, but is consistent in terms of composition, for example 75 percent small and 25 percent large queries, you can determine the appropriate setting for this threshold by running a set of test workload that mimic this composition. Monitor the performance of your system as you gradually increase the number of queries in the test workload to establish the optimal number of queries that should be allowed to run concurrently in your system.

Notes:

1. It is strongly recommended that you run these test workloads prior to defining any query classes in your system.
2. Queries that are unintercepted or unmanaged by Query Patroller are not counted for the purposes of the MAX_TOTAL_QUERIES threshold. This means that, if you expect a significant number of queries to run unintercepted or unmanaged, you will need to take these queries into account when determining how many queries you can run concurrently in your system. For example, if you determine that you can comfortably run 1000 queries concurrently on your system, and you typically run 200 queries that are unintercepted or unmanaged, then you would set your maximum total number of queries to 800.

If your database workload is inconsistent in terms of the size of queries it includes, it may be extremely difficult to determine an appropriate setting for the maximum number of concurrent queries (MAX_TOTAL_QUERIES) threshold since system performance will vary based on both the size and the number of queries in the workload.

After setting the maximum number of concurrent queries (MAX_TOTAL_QUERIES) threshold, continue to monitor performance to ensure you have the correct settings. You might have to reevaluate your settings after significant database restructuring or changes to your typical database workload.

Setting total workload cost (MAX_TOTAL_COST):

Query Patroller calculates the total cost of your current database workload by adding up the estimated cost of all managed queries currently running in the system. The estimated cost of each query is calculated on the entire execution of the query, which might be spread out across a few minutes or several hours.

It is often difficult to set a useful limit on the total cost of all queries (MAX_TOTAL_COST) in the system because ten queries that cost 100,000 timerons have the same total cost as a single query costing 1,000,000 timerons, even though it would likely be more taxing on your system resources to run ten queries simultaneously in one hour than to run a single query over several hours.

To effectively ascertain how many timerons your system can handle at one point in time, you need to figure out how many timerons of work your system can handle in a particular period of time, or time slice. The following steps will guide you in doing this.

1. Choose a query or set of queries that are representative of your workload. You should be familiar with these queries and have a good sense of how long these queries typically take to execute.
2. Perform some test executions of the query or queries.
3. Use the historical analysis Queries report to see the execution time for the queries and the estimated cost of the queries in timerons.
4. To get a more refined measurement of cost, divide the query execution time into time slices. For example, divide a 50 minute query into ten five-minute time slices. Assuming that cost is relatively constant throughout the execution of the query, you can estimate that the cost of execution during each time slice

is one tenth of the total cost of the query. For example, a query costing 100,000 timerons that takes 50 minutes might be divided into ten time slices costing 10,000 timerons each.

5. Determine how many of these representative queries you can run simultaneously before performance begins to deteriorate. For example, if you can run 20 such queries simultaneously before performance degradation, you might determine that your system can run 20,000,000 timerons concurrently.

   Note: Note: Queries that are unintercepted or unmanaged by Query Patroller are not counted for the purposes of the MAX_TOTAL_COST parameter. This means that, if you expect a significant number of queries to run unintercepted or unmanaged, you will need to take these queries into account when determining the maximum cost of the workload you can run concurrently in your system

6. Set the maximum number of concurrent queries (MAX_TOTAL_COST) threshold, either through the Query Patroller Center interface or the command line.
7. Continue to monitor performance to ensure you have the correct settings. You might have to reevaluate your settings after significant database restructuring or changes to your typical database workload.

Submitter profile configuration

Submitter profiles contain settings for submitter query interception, submitter resource limits, and submitter queue priority. Before creating your submitter profiles, you need to plan how you will configure these settings to best meet the needs of your users and your organization.

Step 1: Determine what type of submitter distinctions can be made in your system

Before you can plan for submitter profiles for your users and groups, you need to find out what submitter distinctions can be made based on the type of user information that is available to Query Patroller. A submitter may be a user, a group of users, an application, or a server that submits queries on behalf of a user.

In a two-tier setup where users are connecting directly to the database and submitting queries directly to Query Patroller, you can distinguish between every submitter who submits queries.

In a three-tier setup where users are submitting queries through a third-party submission application, you can only distinguish between submitters if the tool passes the appropriate user information on to Query Patroller. For example, if the submission application connects to the database with the DB2® ID of the user who submits the query, Query Patroller can use the submitter profile for the user to determine how to handle the query. If the submission application connects to the database using a different ID than the user ID submitting the query, Query Patroller uses the profile for the connecting ID to determine how to handle the query. In such cases, all queries submitted through the submission application will be identified as originating from a single submitter.

In situations where different user groups use different applications to submit their queries, and you can differentiate resource allocation, priority, or other user characteristics based on these groups.

Step 2: Determine the nature of the resource requirements that characterize each submitter or group of submitters

You may already be familiar with the warehouse usage patterns associated with the different departments or groups running queries against your database. However, if you do not know what kinds of queries are typically submitted by each group, you can use the Historical Analysis feature to collect historical data to get a representative sample of database activity and analyze the submitter activity data.

You can use the Submitters reports to view the following submitter activity information:

- the size of queries that are submitted by different submitters
- the number of queries submitted by a submitter or group of submitters within a given time frame
- which submitters generate unusually large queries

Step 3: Decide which individual or group submitters should be intercepted by Query Patroller

From the information about the query requirements of the different submitters in your system, you can get a sense of whether there are submitters in your organization whose queries are consistently so small that they are not worth intercepting. If such submitters can be identified, you may want to set up their submitter profiles such that queries from these submitters are not intercepted.

There may also be users in your organization whose response time requirements are so great that they cannot spare even the slightest performance cost that is incurred when a query is intercepted. These users should also be associated with submitter profiles whose queries will not be intercepted.

If you are deploying Query Patroller as a pilot or test project, you may want to intercept only the submitters who are participating in the project. The simplest way to do this is to set your test submitter profiles to be intercepted, and set the PUBLIC submitter profile to not be intercepted by Query Patroller.

Note: In a production environment, it is recommended that you intercept queries from any submitter who has the ability to submit ad hoc queries against the database.

Keep in mind that when a query is submitted under a submitter profile that is set to not be intercepted by Query Patroller (INTERCEPT=’N’) the query cannot be tracked for the purposes of Historical Analysis. This means that once a submitter profile is set to not be intercepted by Query Patroller, any query activity submitted under this profile will not appear in any resource usage reports that you generate.

Step 4: Decide which thresholds to set for submitters whose queries will be intercepted

There are several different resource thresholds that can be set within a submitter profile. Any or all of these thresholds can remain set to the default value, but in most cases where distinctions between submitter resource allocations need to be made, you will want to use one or more of these thresholds to control resource usage at the individual or group level.

Note: When setting resource thresholds for group profiles, keep in mind that the limit that you set applies to each individual in the group each time a query is submitted. This is not a resource limit for the entire group's submissions.

The thresholds are presented in the following table, along with their default values and their potential uses. For a more detailed description of each submitter profile parameter, see the description of the ADD_SUBMITTER_PROFILE command.

Table 5. Submitter threshold defaults and usage

Threshold	Parameter	Default value	Usage
Maximum query cost	MAX_COST_ALLOWED	10,000,000 timerons	Controls size of individual queries. Use to constrain any ″problem″ submitters or to prevent runaway queries.
Maximum number of queries	MAX_QUERIES_ALLOWED	100 queries	Controls maximum number of concurrent queries. Use to prevent submitters from monopolizing system resources by submitting too many queries at once.
Maximum result rows	MAX_RESULT_ROWS	1,000,000 rows	Controls the number of result rows that can be stored in a result table for a single query. Use to limit the amount of disk space taken up by large query results.
Minimum cost to manage	MIN_COST_TO_MANAGE	15,000 timerons	Determines whether a particular query will be managed based on size. Use to exclude small queries whose management would incur a noticeable performance hit relative to execution time.

From the information about the query activity by the different submitters in your system, you should be able to see the size of the queries that are typically submitted by each submitter or group of submitters. This will give you a sense of the value (in timerons) that would be appropriate for the maximum query cost (MAX_COST_ALLOWED) that you can assign to each submitter group. For some submitters, you may not want to change this value from the default. However, for submitters who occasionally submit queries that are excessively large, you can set a value in their submitter profile for the maximum query cost based on what you observe to be an acceptable size query, and allow Query Patroller to hold any queries that exceed this value. If an excessively large query is justified, a submitter will have to either notify the Query Patroller administrator to manually run the query despite its size, or the query can be run along with other held queries at a time when resource usage is lower.

Similarly, if you observe or know of a problem with certain individuals or groups submitting too many queries at once, you can set the maximum number of queries (MAX_QUERIES_ALLOWED) in their submitter profile to a value that seems reasonable given the resource requirements of the submitter.

Note: If your environment is one in which different submitters are not distinguished, you should set the maximum queries allowed (MAX_QUERIES_ALLOWED) threshold to be unlimited (-1) because all intercepted queries will be identified as belonging to a single submitter.

If disk space limitations are a concern, you can set a limit on the number of result rows (MAX_RESULT_ROWS) that are stored in a result table for a query. You can also see from the submitter activity reports what the typical sizes of result sets are for different submitters. This will give you an indication of what values are appropriate for this submitter threshold for the different profiles.

The minimum cost to manage (MIN_COST_TO_MANAGE) threshold can be set to allow small queries to bypass Query Patroller management. You may have to lower this value if you feel that not enough of your workload is being managed by Query Patroller, or you may have to raise this value if you feel that too many queries are being managed by Query Patroller.

You can still specify that these queries are tracked for the purposes of Historical Analysis, but be aware that this tracking will have a small performance cost since it involves table updates for every query.

Step 5: Determine the queue priority level for each submitter profile.

The queue priority level for a submitter profile determines the order in which queued queries are run. The higher the queue priority level for a submitter, the sooner his or her queries are selected from the queue to run. You may want to adjust the queue priority level for a submitter or group of submitters if you have some submitters whose queries need to be managed by Query Patroller, but which have a higher degree of urgency than other submitter's queries.

Since only managed queries are queued, queue priority only affects queries that are intercepted and managed.

Related concepts:

- "Query Patroller thresholds" on page 100
- "Query Patroller submitter profiles" on page 137
- "Scenario: Managing query submitter needs by configuring submitter profiles" on page 89
- "Scenario: Running large emergency queries" on page 91
- "Scenario: Managing queries of different sizes using query classes" on page 92
- "Cost estimation in Query Patroller" on page 99
- "Definition of your query management policy" on page 107

Related tasks:

- "Creating submitter profiles for users and groups" on page 141

Related reference:

- "UPDATE SUBMITTER_PROFILE" on page 251

Query class configuration

Query classes allow you to control the number of queries of a particular size that can run concurrently in your system. Before you create query classes for your system, you need to decide how many query classes to create and how to configure each query class.

Step 1: Characterize your workload

To determine the type and number of query classes to create, you first need to understand the characteristics of the workload that is typical for your system. In particular, you need to have a good sense of the size ranges of the different queries that run against your data warehouse, and the distribution or frequency of each size grouping.

If you do not have this information already, you can collect historical data to get a representative sample of database activity and use the Query Patroller Historical Analysis reports to determine what size groupings exist for your current workload.

Once you have collected the workload data, examine the ″Query activity over time″ report in the Historical Analysis folder of the Query Patroller Center. Sort the data by the Estimated Cost column by clicking on the column header.

This report will show you any natural size groupings that you can use as a guide in creating query classes.

For example, Table 6 shows the compiled figures for query activity in a given data warehouse over a two-week period.

Table 6. Distribution of queries in example workload

Query size (timerons)	Number of queries	Percent of total queries in workload
<8099	1588	88
10190–96444	206	11
>1030000	17	1

The queries in this workload fall naturally into three different size ranges: the estimated cost of the smallest (and most frequent) type of query fall below 8099 timerons. The next size range for queries in this workload is between 10190 and 96444 timerons. The largest and least frequent queries are larger than 1030000.

Based on these groupings, you can plan for your query classes. You may want to plan for a query class for each major size grouping that you identify.

Note: The creation of too many query classes can negatively impact performance so you will have to monitor your system performance after any configuration modifications.

Step 2: Decide on the maximum query cost for each query class

The maximum cost for queries in each query class should correspond to the upper end of the size range for the queries that the query class is intended to run.

Based on the data in Table 6, you might create three query classes to correspond to the three size groupings identified. These query classes are shown in Table 7.

Table 7. Logical size groupings for query classes

Query class	Maximum size of query (timerons)
1	10000
2	100000
3	10000000

If your data reveals a continuous distribution of queries across the entire range of query sizes and there are no natural size-based divisions in your query set, you might need to decide on arbitrary divisions for your query classes that correspond to small, medium, and large queries.

Note: Each query class in your system must have a distinct value for the maximum cost of a query.

Step 3: Decide on the maximum number of queries for each query class

To make your query classes allocate the use of system resources more effectively, you can limit the number of queries that can run concurrently in each query class. The key to setting the maximum number of queries for a query class is to weigh the query execution time against the distribution of a particular size of query in the workload.

When deciding on the maximum number of queries for each query class, it might help to think of your system like a parking lot for queries that provides both short-term parking (for example, 15 minutes) and long-term parking (for example, up to 10 hours). Even if the short-term parking customers represent 50% of the total customers for the parking lot, you would not want to allocate 50% of the parking spaces for short-term parking because short-term customers only use their parking spot briefly. This means that there would likely be many unused short-term spots at any given time. Meanwhile, you might have long-term parking customers waiting for long-term spots to become available.

For example, imagine that the maximum number of queries (MAX_TOTAL_QUERIES) for the system shown in Table 6 on page 117 and Table 7 on page 117 is set to 100. The queries that will run in query class 1 represent 88% of the overall workload. However, these queries take only moments to run. You might set the maximum number of queries (MAX_QUERIES) for this query class to 60. This number is large enough to ensure that small queries do not have to wait to run, but small enough that there will not be empty spots in this query class while larger queries in other query classes are queued. You might then set the maximum number of queries in query classes 2 and 3 to 30 and 10 queries respectively.

Another strategy in setting the maximum number of queries for query classes is to place limits on the number of larger queries only, leaving the number of small queries unlimited. By setting the maximum number of queries for the query class for larger queries you can limit the number of large queries running in the system. By setting the maximum number of queries for the smaller query class to unlimited, you ensure that smaller queries are not put in the queue even though sufficient resources are available to run them.

Step 4: Evaluate your query class definitions

You can evaluate the efficacy of your query class settings using the Managed Queries folder in the Query Patroller Center. In this folder, you can see how many queries are currently running in each query class, and how many queries are queued in each query class. If you check this view periodically as you run your workload, you can see if there is excessive queuing of queries in some classes and adjust the maximum number of queries for your query classes accordingly.

You can also use the Query Activity graph and report in the Historical Analysis folder to see whether queries have spent time in the queue during execution.

Related concepts:

- "Query Patroller query classes" on page 101
- "Scenario: Managing queries of different sizes using query classes" on page 92

- “Cost estimation in Query Patroller” on page 99
- “Definition of your query management policy” on page 107

Related tasks:

- “Creating query classes for Query Patroller” on page 128

Part 4. Setting up your Query Patroller system

Chapter 15. Administering Query Patroller system settings

This chapter provides information on how to configure system-level settings. Most of these tasks can be performed both through the Query Patroller Center graphical user interface or from the command line using the **UPDATE QP_SYSTEM** command.

Setting query thresholds for the Query Patroller system

You can control the overall workload that executes against your database by setting system thresholds for cost and number of queries. By default, these query thresholds are set as unlimited. You may want to set more restrictive thresholds initially or you may want to wait until you've collected historical data that can guide you.

First specify the applications you want to intercept, then set system-wide limits for managed queries.

Prerequisites:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile has the CONFIGURATION privilege with edit authority

Procedure:

To set query thresholds for the Query Patroller system, use one of the following methods.

Query Patroller Center method:

1. Open the Query Patroller System Properties notebook.
2. Click the **Thresholds** tab.
3. To specify the applications that you want to intercept, select one of the following options:
   - **All applications**
   - **No applications or only the applications listed**. Type the executable file name for applications that are to be intercepted by Query Patroller. Separate multiple applications with commas.
   - **All applications except those listed**. Type the executable file name for applications that are not to be intercepted by Query Patroller. Separate multiple applications with commas.

   Note: Application names are case-sensitive.
4. In the **Maximum number of queries** field, type the number of queries that can be running simultaneously on the system. The value you enter will depend on the following factors:
   - The performance of your database
   - The number of users who are submitting queries

- The average cost of the queries being submitted at any given time

To set the value to unlimited, leave the field blank.

5. In the **Maximum workload cost** field, type a number to represent the maximum workload cost threshold, in timerons. To set the value to unlimited, leave the field blank.
6. Click **OK** to accept your entries.

Command line method:

Issue the **UPDATE QP_SYSTEM** command using the following parameters:

- INTERCEPT_APPLICATION
- INCLUDE_APPLICATIONS
- EXCLUDE_APPLICATIONS
- MAX_TOTAL_QUERIES
- MAX_TOTAL_COST

Related concepts:

- "Query Patroller thresholds" on page 100

Related reference:

- "UPDATE QP_SYSTEM" on page 254

Updating the list of databases in Query Patroller

You can update the list of databases that you can select to work with in the Query Patroller Center.

Procedure:

To update the list of databases, use the following method.

Query Patroller Center method:

1. Open the Update Database List window.
2. Use the arrow buttons to move the databases that you want to add from the **Available databases** list to the **Selected databases** list.
3. Click **OK** to update the list of databases that you can work with in the Query Patroller Center.

Enabling e-mail notification of Query Patroller submitters

Enable e–mail notification if you want submitters to be notified when a query completes and a result table is created, or if an error occurs during the processing of a query that would have had its results sent to a result table.

Note: A result table is created for submitters when:

- The query submitter's submission preferences specify that the application should be released after a query is submitted
- A query was held and then run
- A query was run in the background

Prerequisites:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile has the CONFIGURATION privilege with edit authority

You must also have access to an SMTP mail server.

Procedure:

To enable e–mail notification, use one of the following methods.

Query Patroller Center method:

1. Open the Query Patroller System Properties notebook.
2. Click the **E–mail** tab.
3. Select the **Enable e–mail notification** check box.
4. In the **E–mail server** field, enter the host name or IP address of the SMTP e–mail server you want to use.
5. For submitters who do not have an e–mail address specified in their query submission preferences, select either **Do not send e–mail** or **Send e–mail to a designated address**. If you are sending to a designated address, such as the administrator's address or a support desk, enter the address in the field.

   Note: Submitters can specify the e–mail addresses they want to use in the Query Submission Preferences window. For more information, see Setting query submission preferences for another submitter.
6. Click **OK** to enable e–mail notification and close the notebook.

Command line method:

Issue the **UPDATE QP_SYSTEM** command using the following parameters:

- EMAIL_ENABLE
- EMAIL_SERVER
- SEND_DESIGNATED
- DESIGNATED_EMAIL_ADDRESS

Related concepts:

- "Query Patroller submitter profiles" on page 137

Related tasks:

- "Setting query submission preferences for another submitter" on page 147

Related reference:

- "UPDATE QP_SYSTEM" on page 254

Enabling collection of historical data

Query Patroller historical analysis can be a powerful tool for analyzing data warehouse usage. You can choose to collect historical data only for queries that were managed by Query Patroller, or for all queries intercepted by Query Patroller. Queries that are not intercepted cannot have historical data collected about them.

The default behavior is that only managed queries will have historical data collected.

Note: If you are planning to collect data regarding query execution time, you need to ensure that the DB2 timestamp and statement monitor switches are set to ’ON’. If you are planning to collect data about how many rows are returned by queries, you need to ensure that the DB2 statement monitor switch is set to ’ON’.

Prerequisites:

You must meet one of the following requirements:
- Have DBADM authority
- Be an operator whose profile has the CONFIGURATION privilege with edit authority

Procedure:

To enable collection of historical data, use one of the following methods.

Query Patroller Center method:
1. Open the Query Patroller System Properties notebook.
2. Click the **Options** tab.
3. Under **Historical analysis**, specify the queries that you want to save by selecting the **Only managed queries** or the **All intercepted queries** radio button.
4. Click **OK** to save your changes.

Command line method:

Issue the **UPDATE QP_SYSTEM** command with the QUERIES_TO_SAVE parameter.

Related concepts:
- “Query Patroller historical analysis” on page 103
- “Query interception and management in Query Patroller” on page 105

Related reference:
- “UPDATE QP_SYSTEM” on page 254

Chapter 16. Administering query classes

Query classes are Query Patroller mechanisms for grouping and running queries according to size. Using query classes, you can control the flow of queries on a database so that system resources are shared among queries in the different size groupings.

This chapter describes how to configure, create, and remove query classes in your Query Patroller system.

Configuring query classes

You can use query classes to control the number of managed queries of a particular size that can run concurrently in your system.

Prerequisites:

You must have read the topic ″Query class configuration″.

Procedure:

To configure query classes:

1. Using Query Patroller Historical Analysis reports, collect system and workload statistics.
2. To identify any natural size groupings for queries (based on estimated query cost) to use as a guide in creating query classes, examine the ″Query activity over time″ report which appears in the **Historical Analysis >> Queries** folder of the Query Patroller Center.
3. Sort the queries data by the **Estimated Cost** column by clicking on the column header.

   Note: If the **Estimated Cost** column does not appear, click the **Customize columns** icon in the bottom toolbar.
4. Decide on the maximum query cost (in timerons) for each query class.

   Note: Each query class in your system must have a distinct value for the maximum query cost. The maximum cost for queries in each query class should correspond to the upper end of the size range for queries that the query class is intended to run.
5. Decide on the maximum number of queries that can run concurrently in each query class.
6. Create query classes for Query Patroller.
7. Evaluate and refine your query class definitions. In the ″Managed Query″ report in the Query Patroller Center, you can see how many queries are currently running in each class and how many queries are queued in each query class.

   Note: If the **Query class**column does not appear, click the **Customize columns** icon in the bottom toolbar.

If you have not already done so, you can further control the flow of queries against your database by setting or modifying the system thresholds for managed queries or by setting or modifying submitter resource limits.

Related concepts:

- "Query class configuration" on page 116
- "Scenario: Managing queries of different sizes using query classes" on page 92
- "Definition of your query management policy" on page 107

Related tasks:

- "Creating query classes for Query Patroller" on page 128
- "Collecting historical data with Query Patroller" on page 163

Creating query classes for Query Patroller

By creating query classes, you can control the number of queries of various sizes that are running against the system at any one time. When you create a new query class, you specify its properties, and you can provide a useful description of that query class.

When you create query classes, you should consider the following factors:

- The performance of your database
- The number of users submitting queries
- The average cost of the queries being submitted at any given time

Query classes can be created, removed, or modified while Query Patroller is started. However, these changes will not take effect until Query Patroller has been stopped and restarted.

Prerequisites:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile has the CONFIGURATION privilege with edit authority

Procedure:

To create a new query class, use any of the following methods.

Query Patroller Center method:

1. Open the Create Query Class window.
2. In the **Maximum number of queries** field, specify the maximum number of queries of this query class that can be run simultaneously. Queries belonging to this query class and submitted after this value is reached are queued by Query Patroller. This number must be less than or equal to the **Maximum number of queries** set in the Query Patroller System Properties notebook. If **Maximum number of queries** for the system is unlimited, you can create a query class with an unlimited maximum number of queries by leaving this field blank.
3. In the **Maximum cost of a query** field, specify the maximum size for queries that belong to this query class. The query class that you are defining will contain queries that are smaller than this **Maximum cost of a query** amount,

and larger than the **Maximum cost of a query** amount of the next smallest query class. You cannot define two query classes that have the same **Maximum cost of a query** amounts.

4. Optional: In the **Query class description** field, enter a comment or a description for the query class that you are creating.
5. Click **OK** to create your query class and close the Create Query Class window. When you click **OK**, a Query class ID will be assigned by Query Patroller.

Command line method:

Issue the **ADD QUERY_CLASS** command.

Related concepts:

- "Query Patroller query classes" on page 101

Related tasks:

- "Changing query classes for Query Patroller"--*Help: Query Patroller*
- "Removing query classes for Query Patroller" on page 129

Related reference:

- "ADD QUERY_CLASS" on page 196

Removing query classes for Query Patroller

Remove a query class when you want to change the query classification that you are using. You might remove a query class if you discover that it is not being used, or if you want to resize all of your query classes.

Query classes can be created, removed, or modified while Query Patroller is started. However, these changes will not take effect until Query Patroller has been stopped and restarted.

Prerequisites:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile has the CONFIGURATION privilege with edit authority

Procedure:

To remove a query class, use any of the following methods.

Query Patroller Center method:

1. From the Query Patroller Center, expand the **Configuration** folder to find the **Query Classes** folder.
2. Click the **Query Classes** folder. Any existing query classes are displayed in the pane on the right side of the window (the contents pane).
3. In the contents pane, right-click the query class that you want to remove, and click **Remove** in the pop-up menu. A window opens where you can confirm that you want to remove the query class.

Command line method:

Issue the **REMOVE QUERY_CLASS** command.

Related concepts:

- "Query Patroller query classes" on page 101

Related tasks:

- "Creating query classes for Query Patroller" on page 128
- "Changing query classes for Query Patroller"--*Help: Query Patroller*

Related reference:

- "REMOVE QUERY_CLASS" on page 231

Part 5. Managing users

Chapter 17. Administering operators

Query Patroller operators

A Query Patroller operator is an ID used by Query Patroller to represent a user or a group of users who has a subset of administrator authorities and tasks defined in the operator's profile. Operator IDs are not defined by Query Patroller; they are created by selecting from existing DB2® user or group IDs.

Note: The ID that has DBADM authorities is automatically the Query Patroller administrator.

A Query Patroller operator performs some or all of the following administrative tasks.

- **Configuration**

  Involves creating or deleting query classes and setting system-wide thresholds, as well as other configuration tasks such as setting up e–mail notification.
- **Monitoring**

  Involves changing the status of queries, viewing the SQL of managed queries, and deleting result tables that are not needed.
- **User administration**

  Involves creating, modifying, and deleting submitter profiles. Also involves removing managed queries that have completed and are no longer needed, and manually removing held queries.
- **Historical analysis**

  Involves removing historical queries that are no longer needed.

Related concepts:

- "Query Patroller submitters" on page 137
- "Query Patroller operator profiles" on page 133

Query Patroller operator profiles

The role of a Query Patroller operator is usually filled by users in a front-line support role, such as help-desk representatives.

The kinds of tasks that an operator can perform depend on the level of authority that the operator has: view authority or edit authority.

For instance, an operator whose profile has the MONITORING privilege with edit authority can change the status of queries, view the SQL of managed queries, and delete result tables that are no longer needed. However, an operator whose profile has the MONITORING privilege with view authority can only view the status of queries.

Active and suspended operator profiles

When an operator profile is suspended, users associated with the profile can still access the Query Patroller Center as submitters, provided that they are associated with an active submitter profile. However, they cannot access any other parts of the Query Patroller Center.

Which operator profile Query Patroller uses

When an operator's user ID has multiple group operator profiles associated with it, Query Patroller determines the operator's privileges by merging the privileges of all the operator profiles that the user is associated with. The privileges are merged so that the operator is granted the highest set of privileges from the operator profiles.

Related concepts:

- "Query Patroller operators" on page 133
- "Query Patroller submitter profiles" on page 137

Related tasks:

- "Creating operator profiles for users and groups" on page 134
- "Changing operator profiles for users and groups"--*Help: Query Patroller*

Creating operator profiles for users and groups

You can create an Query Patroller operator profile based on an existing operator profile or you can create an operator profile with new settings.

Prerequisites:

You must have DBADM authority.

Procedure:

To create an operator profile, use one of the following methods.

Query Patroller Center method:

1. Open the Create Operator window:
   a. If you are creating an operator profile based on an existing operator profile.
   b. If you are creating an operator profile with new settings.
2. In the **Operator** field, type a name for the operator you are creating. The name must be typed in uppercase and it must exist as a DB2 authorization ID.
3. Use the **Profile type** field to select whether the operator profile applies to a user or a group.
4. Optional: If you want to temporarily suspend the new operator's ability to access all parts of the Query Patroller Center, select the **Access suspended** check box.

   Note: If this is a group operator profile, suspending access does not suspend all users belonging to that group, unless they have no other operator profiles. Suspending this profile will, however, prevent this profile from giving these users any additional privileges beyond those assigned to the operator profiles of the other groups to which they belong.

5. Use the **Configuration** field to specify the level of access the operator will have to the Configuration tree element in the Query Patroller Center. The level of access you specify determines whether or not the operator can create or delete query classes or set system-wide thresholds.
6. Use the **Monitoring** field to specify the level of access the operator will have to the Monitoring tree element in the Query Patroller Center. The level of access you specify determines whether or not the operator can modify the status of a query or delete a result table.
7. Use the **User administration** field to specify the level of access the operator will have to the User Administration tree element in the Query Patroller Center. The level of access you specify determines whether or not the operator can create, modify, or delete submitter profiles.
8. Use the **Historical analysis** field to specify the level of access the operator will have to the Historical Analysis tree element in the Query Patroller Center. The level of access you specify determines whether or not the operator can remove and view historical query data.
9. Click **OK** to create the new operator profile.

Command line method:

Issue the **ADD OPERATOR_PROFILE** command.

Related concepts:

- "Query Patroller operators" on page 133

Related tasks:

- "Changing operator profiles for users and groups"--*Help: Query Patroller*
- "Suspending or restoring operator privileges for users and groups" on page 135
- "Removing operator profiles for users and groups"--*Help: Query Patroller*

Related reference:

- "ADD OPERATOR_PROFILE" on page 193

Suspending or restoring operator privileges for users and groups

You can temporarily suspend an operator's ability to access all parts of the Query Patroller Center, then restore these privileges at a later time.

Notes:

1. A user with a suspended operator profile can still access the Query Patroller Center as a submitter if he has an active submitter profile.
2. If you want to suspend or restore an operator's access to only some parts of the Query Patroller Center, see Changing operator profiles for users and groups.

Prerequisites:

You must have DBADM authority.

Procedure:

To suspend privileges for an operator, use one of the following methods.

Query Patroller Center method:

1. Open the Operator Properties window.
2. Select the **Access suspended** check box to suspend the operator's ability to access all parts of the Query Patroller Center, or clear it to restore the submitter's ability to access the indicated areas of the Query Patroller Center.

   Note: If this is a group operator profile, suspending access does not suspend all users belonging to that group, unless they have no other operator profiles. Suspending this profile will, however, prevent this profile from giving these users any additional privileges beyond those assigned to the operator profiles of the other groups to which they belong.
3. Click **OK** to close the Operator Properties window.

Command line method:

Issue the **UPDATE OPERATOR_PROFILE** command using the SUSPENDED parameter.

Related concepts:

- "Query Patroller operators" on page 133

Related tasks:

- "Changing operator profiles for users and groups"--*Help: Query Patroller*
- "Creating operator profiles for users and groups" on page 134
- "Removing operator profiles for users and groups"--*Help: Query Patroller*

Related reference:

- "UPDATE OPERATOR_PROFILE" on page 243

Chapter 18. Administering submitters

Query Patroller submitters

A submitter is an ID used by Query Patroller to represent any person, group, or application that submits queries. A submitter ID is not defined by Query Patroller; the submitter ID is the SQL authorization ID that DB2® assigns when the submitter connects to the instance (a case-sensitive ID, usually in uppercase). Query Patroller uses submitter IDs to differentiate the users or groups of users who submit queries through Query Patroller.

Query Patroller's ability to identify individual submitters varies depending on the environment in which you are using Query Patroller:

- In a two-tier environment in which users submit queries directly to Query Patroller, the DB2 ID associated with each submitter is passed along to Query Patroller.
- In a three-tier environment, Query Patroller treats all queries that are submitted through the application as if they come from the same submitter. In such an environment, the application server in the second tier uses the same ID to connect to the database regardless of which user is submitting a query.

Related concepts:

- "Query Patroller submitter profiles" on page 137
- "Query Patroller query submission preferences" on page 147

Related tasks:

- "Creating submitter profiles for users and groups" on page 141
- "Changing submitter profiles for users and groups"--*Help: Query Patroller*
- "Setting submitter resource limits" on page 143

Query Patroller submitter profiles

A Query Patroller submitter profile is a set of characteristics that define:

- Whether Query Patroller should intercept queries from a submitter
- If the submitter's queries are intercepted, what resource limits are applied to those queries
- What priority level the submitter's queries have in a queue
- The submitter's charge-back account code (to be used for cost tracking purposes)

You can create submitter profiles for individual users and for groups, but this is not required. During the Query Patroller installation process, a submitter profile called PUBLIC is created. By default, all submitters use this profile unless they also belong to a more restrictive profile. For information, see "Which submitter profile Query Patroller uses" on page 139.

If you want to assign different characteristics to some submitters than you assign to others, the submitters must use different submitter profiles. For instance, if you want to track queries submitted by the Marketing department to determine the resources that they use, you will need to create a group submitter profile for submitters in that department. This means that you must use a DB2® group ID for the Marketing submitters.

Submitter resource limits

You can set submitter resource limits to ensure that no individual submitter or group of submitters uses too many system resources. You can set limits on the number of queries that a submitter can run simultaneously and on the maximum cost (in timerons) that any single query from a submitter can incur. See the Query Patroller thresholds topic.

To limit the amount of disk space taken up by large query results, you can limit the number of result rows that can be stored in a result table for a single query from a submitter.

Interception and management of queries from a particular submitter profile

You can specify in the submitter profile if queries from the associated submitter should be allowed to bypass Query Patroller. Unintercepted queries are not managed, nor are they used for the collection of data for historical analysis.

You can also specify in the submitter profile that if a query from the submitter is below a certain size (in timerons) Query Patroller will not manage that query. Specify this using the minimum cost to manage value for the submitter (MIN_COST_TO_MANAGE).

Submitter queue priority

If you want to ensure that queries from certain users have priority over other queries when there is a queue, you can assign a higher queue priority to those users. The submitter queue priority is a numeric value between 0 and 999 that defines the priority of queries submitted by the associated submitter in a queue of queries. By default, all submitters have a queue priority of 500. You may, for example, want to assign a queue priority of 700 to submitters that perform queries needed for monetary transactions to ensure that these transactions are handled quickly.

For queue priority settings to work effectively, you should plan in advance the queue priority values you want to assign to different users.

Submitter charge-back accounts

A charge-back account is an alphanumeric account code to be used for cost tracking purposes. You can use the charge-back parameter to sort submitters into logical groupings to track usage costs. To obtain historical analysis data for charge-back accounts, view the TRACK_QUERY_INFO table.

Active and suspended submitter profiles

Users must have an active submitter profile in order to submit queries. With DB2 Version 8, every DB2 user ID belongs to the PUBLIC group, so if the PUBLIC submitter profile is active then by default Query Patroller will allow every DB2 user to submit queries. To prevent an individual user from submitting queries without suspending the PUBLIC submitter profile, you can create an individual profile for the submitter and suspend the individual submitter profile. To create or suspend a submitter profile, you must be an administrator or an operator whose profile has the USER ADMINISTRATION privilege with edit authority.

Which submitter profile Query Patroller uses

The query submission preferences that can be set for a submitter include specifying the group submitter profile Query Patroller uses for that submitter. The default query submission preference for a submitter profile is PUBLIC. You can override the default query submission preference for a submitter profile by changing the query submission preferences of each submitter.

Query Patroller identifies which submitter profile to use when processing a query based on the following criteria, in order of importance:

1. Query Patroller chooses the submitter's own profile, if one exists. This submitter profile type is USER.
2. If the submitter does not have a USER submitter profile, but does have query submission preferences, Query Patroller chooses the active group profile that matches the group profile specified in the query submission preferences.
3. If there is no group submitter profile specified in the query submission preferences, or if the specified group submitter profile is not active, Query Patroller chooses the active group profile that is most restrictive by looking at the following characteristics in the following order of importance:
    a. Whether Query Patroller does or does not intercept queries (BYPASS Y/N) from the group
    b. The minimum cost to manage value for the group (MIN_COST_TO_MANAGE)
    c. The maximum cost of a query value for the group (MAX_COST_ALLOWED)
    d. The maximum number of queries value for the group (MAX_QUERIES_ALLOWED)
    e. The queue priority of the group
    f. The maximum number of return rows value for the group (MAX_RESULT_ROWS)
4. If more than one of the active group submitter profiles have the same degree of restrictiveness, Query Patroller chooses a profile arbitrarily.
5. If there is no active group submitter profile found, Query Patroller chooses the PUBLIC profile,
6. If the PUBLIC submitter profile is not active, Query Patroller returns an SQL error that states that a submitter profile was not found.

For example, consider a sales manager who has three different IDs that she uses when submitting queries:

- She uses the *sales_dept* group ID when accessing sales data.
- She uses the *sales_transactions* group ID when performing sales transactions.
- She uses the *managers* group ID when accessing the records of employees in her department.

Each of these group IDs has a group submitter profile. The settings for each of these group profiles are defined as shown in Table 8 on page 140.

Table 8. Profile settings

Settings	sales_dept	sales_transactions	managers
Intercept	Y	Y	Y
Minimum cost to manage value (in timerons)	10 000	10 000	15 000
Maximum cost of a query allowed value (in timerons)	700 000	250 000	1 000 000
Maximum number of queries value	20	30	20
Queue priority	500	700	500
Maximum number of return rows value	1 000,000	400 000	1 200 000

The sales manager does not have her own submission profile (of type USER), and her submission preferences do not specify which group profile she wants to use to submit queries. Query Patroller identifies which submitter profile to use when processing the sales manager's queries by determining which submitter profile is most restrictive:

- Queries submitted by all three groups can be intercepted, so they are all equally restrictive.
- The minimum cost to manage a query specified for the *managers* group profile is 15 000 timerons; whereas the minimum cost to manage specified for both the *sales_dept* group profile and the *sales_transactions* group profile is 10 000 timerons. Since the *managers* group profile is less restrictive, it is not used.
- The value specified for maximum cost of a query allowed for the *sales_dept* group profile is 700 000 timerons, whereas the value is 250 000 timerons for the *sales_transactions* group profile. The value specified for the *sales_transactions* group profile is more restrictive.
- Query Patroller identifies the *sales_transactions* group profile as the submitter profile it should use when processing queries from the sales manager.

If the sales manager needs to submit larger queries, she must either request that the administrator create her a submission profile (of type USER) or she must change her submission preferences so that they specify which group profile she wants to use to submit queries.

Related concepts:

- "Query Patroller submitters" on page 137
- "Query interception and management in Query Patroller" on page 105
- "Query Patroller query submission preferences" on page 147

Related tasks:

- "Creating submitter profiles for users and groups" on page 141
- "Changing submitter profiles for users and groups"--*Help: Query Patroller*

Configuring submitter profiles

Before you create submitter profiles for your database, you need to make decisions about the type of submitters that will be using the database and what type of constraints you want to place on the resources used by each submitter or group of submitters.

Prerequisites:

You must read the Submitter profile configuration topic before completing this task.

Procedure:

To configure a submitter profile, use the following method.

1. Decide what type of submitter distinctions can be made in your system.
2. Optional: Collect historical data to get a representative sample of database activity.
3. Optional: Use the Historical Analysis submitters reports to view the following information:
   - the size of queries submitted by different submitters
   - the number of queries submitted by a submitter or group of submitters within a given time frame
   - which submitters generate unusually large queries
4. Decide which submitters or groups of submitters should be intercepted by Query Patroller.
5. Decide which thresholds to set for submitters whose queries will be intercepted.
6. Determine the priority level for each submitter profile.
7. Create submitter profiles for your system.
8. Evaluate your system performance and modify your submitter profiles if necessary.

Related concepts:

- "Query Patroller thresholds" on page 100
- "Query interception and management in Query Patroller" on page 105
- "Query Patroller submitter profiles" on page 137
- "Scenario: Managing query submitter needs by configuring submitter profiles" on page 89
- "Submitter profile configuration" on page 113
- "Scenario: Handling very large queries" on page 90
- "Definition of your query management policy" on page 107

Related tasks:

- "Collecting historical data with Query Patroller" on page 163
- "Creating submitter profiles for users and groups" on page 141
- "Setting submitter resource limits" on page 143

Creating submitter profiles for users and groups

You can create a Query Patroller submitter profile based on an existing submitter profile or you can create a submitter profile with new settings. In the submitter profile, you can:

- Define the submitter's charge-back account code (to be used for cost tracking purposes)
- Specify whether queries from the submitter should be intercepted

- If the submitter's queries can be intercepted, set resource limits for the submitter

Prerequisites:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile has the USER ADMINISTRATION privilege with edit authority

Procedure:

To create one or more submitter profiles, use one of the following methods.

Query Patroller Center method:

1. Open the Create Submitter notebook:
   - To create a submitter profile based on an existing submitter profile
   - To create a submitter profile with new settings
2. On the General page:
   a. In the **Submitter user ID** field, type the name that will be associated with this profile. The name is case sensitive and must match the SQL authorization ID in the Database Connection Information. The SQL authorization ID is usually same as the login ID, but in uppercase. You can type multiple values separated by commas, to create multiple submitter profiles.

   b. Use the **Profile type** field to select whether the submitter profile applies to a user or a group.

   c. Optional: In the **Charge-back account** field, type an alphanumeric account code to be used for cost tracking purposes. You can use the charge-back parameter to sort submitters into logical groupings to track usage costs.

      Examples:
      - If three submitters belong to the legal department, then you could type LEGAL in the **Charge-back account** field.
      - If an expense code of MK001 exists for the Marketing department, then you could type MK001 in the **Charge-back account** field whenever you add a submitter from the Marketing department.

   d. If you want to temporarily suspend the new submitter's ability to submit queries, select the **Access suspended** check box.
3. On the Resources page:
   a. If you do not want Query Patroller to intercept queries from the submitter you are creating, select the **Do not intercept queries from this submitter** check box. Query Patroller will neither perform cost evaluation nor create a query in the historical analysis view. Query Patroller will not manage queries submitted by submitters with this profile.

   b. In the **Minimum cost to manage** field, type a number greater than or equal to 0 and less than the **Maximum cost of a query** value. Queries costing less than this value will not be managed by Query Patroller.

   c. In the **Maximum number of queries** field, type the maximum number of queries that a submitter is permitted to run simultaneously. Additional queries will be queued. To allow an unlimited number of queries to run simultaneously, leave the field blank.

d. In the **Maximum cost of a query** field, type a number for the maximum query cost. If the submitter submits a query with an estimated cost that is higher than this value, the query is held. To set the value to unlimited, leave the field blank.

e. In the **Maximum size of a result table** field, type a value for the maximum number of result rows to be stored in a result table. If you leave the field blank, DB2 will always make this submitter's result table large enough to accommodate the entire result set.

f. In the **Queue priority** field, type a value from 0 to 999 to represent the priority assigned to a query when a query is submitted. Higher numbers indicate higher priority.

4. Click **OK** to create the new submitter.

Command line method:

Issue the **ADD SUBMITTER_PROFILE** command.

Related concepts:

- "Query Patroller submitter profiles" on page 137
- "Query Patroller submitters" on page 137

Related tasks:

- "Changing submitter profiles for users and groups"--*Help: Query Patroller*
- "Suspending or restoring submitter privileges for users and groups" on page 144
- "Removing submitter profiles for users and groups"--*Help: Query Patroller*
- "Setting query submission preferences for another submitter" on page 147

Related reference:

- "ADD SUBMITTER_PROFILE" on page 201

Setting submitter resource limits

You can specify whether or not you want a submitter's queries to be intercepted by Query Patroller. If you specify that they will be intercepted, you can also limit the resources that are used by a submitter, so that the submitter cannot use too many of the system's resources.

Prerequisites:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile has the USER ADMINISTRATION privilege with edit authority

Procedure:

To limit submitter resources, use one of the following methods.

Query Patroller Center method:

1. Open the Submitter Properties notebook.
2. Click the **Resources** tab.

3. If you do not want Query Patroller to intercept queries from the submitter you are creating, select the **Do not intercept queries from this submitter** check box. Query Patroller will neither perform cost evaluation nor create a query in the historical analysis view. Query Patroller will not manage queries submitted by submitters with this profile.
4. In the **Minimum cost to manage** field, type a number greater than or equal to 0 and less than the **Maximum cost of a query** value. Queries costing less than this value will not be managed by Query Patroller.
5. In the **Maximum number of queries** field, type the maximum number of queries that this submitter is permitted to run simultaneously. Additional queries will be queued. To allow an unlimited number of queries, to run simultaneously, leave the field blank.
6. In the **Maximum cost of a query** field, type a number for the maximum query cost. If the submitter submits a query with an estimated cost that is higher than this value, the query is held. To set the value to unlimited, leave the field blank.
7. In the **Maximum size of a result table** field, type a value for the maximum number of result rows to be stored in a result table. If you leave the field blank, DB2 will always make this submitter's result table large enough to accommodate the entire result set.
8. In the **Queue priority** field, type a value from 0 to 999 to represent the priority assigned to a query when a query is submitted. Higher numbers indicate higher priority.
9. Click **OK** to update the submitter profile.

Command line method:

Issue the **UPDATE SUBMITTER_PROFILE** command.

Related concepts:

- "Query Patroller submitters" on page 137
- "Query Patroller submitter profiles" on page 137

Related tasks:

- "Changing submitter profiles for users and groups"--*Help: Query Patroller*
- "Suspending or restoring submitter privileges for users and groups" on page 144

Related reference:

- "UPDATE SUBMITTER_PROFILE" on page 251

Suspending or restoring submitter privileges for users and groups

You can temporarily suspend a submitter's ability to submit queries, then restore these privileges at a later time.

Prerequisites:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile has the USER ADMINISTRATION privilege with edit authority

Procedure:

To suspend or restore a submitter's privileges, use one of the following methods.

Query Patroller Center method:

1. Open the Submitter Properties notebook.
2. On the General page, select the **Access suspended** check box to suspend the submitter's ability to submit queries, or clear it to restore the submitter's ability to submit queries.
3. Click **OK** to close the Submitter Properties notebook.

Command line method:

Issue the **UPDATE SUBMITTER_PROFILE** command using the SUSPENDED parameter.

Related concepts:

- "Query Patroller submitters" on page 137

Related tasks:

- "Creating submitter profiles for users and groups" on page 141
- "Changing submitter profiles for users and groups"--*Help: Query Patroller*
- "Setting submitter resource limits" on page 143

Related reference:

- "UPDATE SUBMITTER_PROFILE" on page 251

Chapter 19. Administering query submission preferences

Query Patroller query submission preferences

Query submission preferences are used to specify

- The submitter profile the submitter uses when submitting queries if the submitter does not have a user submitter profile, and has two or more group submitter profiles
- The location where Query Patroller should send the results of the submitter's queries
- Who can view the submitter's result tables
- What Query Patroller should do if the submitter's result tables are too large
- The e–mail address to be used for sending notifications to the submitter

The default query submission preferences are assigned to a submitter profile called PUBLIC. If an individual submitter requires submission preferences that are different from the default submission preferences, you need to create new submission preferences for that submitter.

Related concepts:

- "Query Patroller submitters" on page 137
- "Query Patroller submitter profiles" on page 137

Related tasks:

- "Setting query submission preferences for another submitter" on page 147

Setting query submission preferences for another submitter

The default query submission preferences are called PUBLIC. Submitters who do not have their own submission preferences, use the settings specified in the PUBLIC submission preferences. If you do not want a submitter to use the values specified in the PUBLIC submission preferences, you can set different submission preferences for the submitter.

Prerequisites:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile has the USER ADMINISTRATION privilege with edit authority

Procedure:

To update preferences for query submission, use one of the following methods.

Query Patroller Center method:

1. Open the Query Submission Preferences window.
   - To create new query submission preferences based on existing query submission preferences

- To create new query submission preferences with new settings
- To modify existing query submission preferences

2. If you are creating new query submission preferences, in the **Submitter** field enter the name of the submitter or the user ID that was used to start the application that submits queries.
3. If the submitter or end user belongs to multiple group profiles, use the **Submitter profile to use** field to select which group profile the submitter will use when submitting queries. If the submitter or end user has a submitter profile whose type is user, this field shows the submitter's user ID, and you cannot change this value.

   If you lack a submitter profile whose type is user, yet belong to multiple group submitter profiles, this field lists the group submitter profiles. Select the group submitter profile that you want to submit your queries with. The Select automatically entry applies the profile with the lowest resource limits.
4. Specify what will happen after a query is submitted:
   - To specify that the application that submitted the query will wait for the result set to return while Query Patroller manages the query, select the **Wait until the results are returned** radio button. This option is the default.

     When this option is selected, the application that submitted the query can become unresponsive until the result set is returned.
   - To specify that the result set will be stored in a DB2 table and the application that submitted the query will become free for further processing, select the **Release the application and retrieve the results from a result table** radio button.
5. Specify the access levels you want to use for your result tables:
   - To specify that only the submitter can access the result tables, select the **Restrict access to the submitter** radio button.
   - To make the result table accessible to specific DB2 users select the **Grant access to other users or groups** radio button. Type the names of specific users or groups in the associated field, separated by commas. The DB2 users listed must have access to the database where the query was submitted.
6. Specify how to handle result sets that are longer than the maximum allowed:
   - To specify that a truncated set of results should be stored in a result table, select the **Return a truncated set of results** radio button.
   - To specify that the incomplete results should not be stored in a result table, select the **Do not return any results** radio button.
7. In the **E–mail address** field, type the e–mail address to be used for sending notifications to the submitter. You can type multiple values separated by commas, to send e–mail to more than one address. Notifications are sent to the address or addresses specified when a query completes or if a query encounters an error, but only in cases where a result table is created.

   Note: A result table is created for submitters when:
   - The **Release the application** option is specified in the Query Submission Preferences window.
   - The **Wait until results are returned** option is specified in the Query Submission Preferences window, yet the query was placed in a held state before completing.
   - The status of a query is changed to **Run query in the background**.
8. Click **OK** to set preferences for query submission.

Command line method:

To create new query submission preferences, issue the **ADD SUBMISSION_PREFERENCES** command.

To change existing query submission preferences, issue the **UPDATE SUBMISSION_PREFERENCES** command.

Related concepts:

- "Query Patroller submitters" on page 137

Related tasks:

- "Creating submitter profiles for users and groups" on page 141
- "Removing query submission preferences for submitters"--*Help: Query Patroller*

Related reference:

- "UPDATE SUBMISSION_PREFERENCES" on page 248

Part 6. Using Query Patroller

Chapter 20. Managing queries with Query Patroller

Managed query status

While a query is being managed by Query Patroller, a query can be in one of several different states. Query status information can be obtained in several ways: from the Query Patroller Center Managed Queries folder, from the **Query status** field of the Managed Query Properties notebook, and by issuing the GET QUERY command.

If a query is managed by Query Patroller, it will pass through the following states during normal processing:

Initial The query has been intercepted by the Query Patroller server. (This status is never displayed, because the query moves quickly to a subsequent status.)

Running
The query is in progress, and has been passed to DB2® for execution.

Done The query completed successfully.

In some cases, queries pass through other states during processing:

Held The cost of the query exceeds the submitter's threshold. A held query can be released manually or it can be released automatically by a scheduled job. Releasing a held query will place it in *released* state.

Released
The query was held, but has been released by an administrator or an operator whose profile has the MONITORING privilege with edit authority, or it was released automatically by a scheduled job. A released query is processed by the Query Controller and placed in *running* state or *queued* state, depending on the current system workload.

Queued
The query is waiting to run. A query can be queued if one or more of the following thresholds is exceeded:

- Maximum number of queries value for the system (MAX_TOTAL_QUERIES)
- Maximum number of queries value for the submitter (MAX_QUERIES_ALLOWED)
- Maximum workload cost value for the system (MAX_TOTAL_COST)
- Maximum number of queries value for the query class in which the query runs (MAX_QUERIES)

The query will run when the situation that caused it to be queued has changed. For example, a query that was queued because the maximum number of queries for that query class has been exceeded will run when the number of queries belonging to that query class falls below the maximum for the query class. However, if a query caused more than one threshold to be exceeded, then even after the first threshold is no longer exceeded, the second threshold may still be exceeded. The query will run when it no longer causes any of these thresholds to be exceeded.

Canceled
The query was canceled by the administrator, submitter, or an operator whose profile has the MONITORING privilege with edit authority. Only *running*, *held*, *released*, and *queued* queries can be *canceled*.

Aborted
The query was terminated by DB2 because of an error.

Rejected
The query was prevented from running.

Unknown
The status of the query cannot be determined. A query with status *unknown* is no longer *running*, but Query Patroller cannot determine whether the query completed or failed.

Related concepts:

- Chapter 25, "Query Patroller, Version 8 limitations and restrictions," on page 185

Related reference:

- "GET QUERY" on page 210
- "UPDATE QP_SYSTEM" on page 254

Changing the status of queries using Query Patroller

Depending on your authority level, you may change the status of queries managed by Query Patroller in several ways: cancel a query, release a held query, or run a query in the background.

Canceling queries
Cancel a query if you realize after it has been submitted that it contains an error, or that its cost is too high. For example, you might receive notification that your query is being held because its cost exceeds the maximum amount of system resource that is allowed for any one of your queries. Canceling a query places it in the canceled state.

Releasing queries from the held state
Release a held query if you decide that a particular query should run, even though the query exceeds the submitter's maximum query cost. Releasing a held query places the query in the running state or the queued state, depending on the current system workload.

Running queries in the background
Run a query in the background if your query submission preferences specify that you will wait until the results of a query are returned, but you want to use your client application while that particular query runs. Running a query in the background places the query in the running state or the queued state, depending on the current system workload.

Prerequisites:

- To cancel a query, you must meet one of the following requirements:
  - Have DBADM authority
  - Be an operator whose profile has the MONITORING privilege with edit authority
  - Be the submitter of the query
- To release a held query, you must meet one of the following requirements:

  - Have DBADM authority
  - Be an operator whose profile has the MONITORING privilege with edit authority
- To run a query in the background, you must:
  - Be the submitter of the query

Procedure:

To change the status of a query:

Query Patroller Center method:

1. Open the Change Query Status window.
2. To cancel the query, click **Cancel query**.
3. To run the query, click **Release query from held state**.
4. To regain control of the submitting application, click **Run query in the background**. Query Patroller stops the execution of the query and resubmits it. The results of the query will be returned to a result table.
5. Click **OK** to change the status of the query as you have indicated and close the Change Query Status window.

Command line method:

To cancel the query, issue the **CANCEL QUERY** command.

To run the query, issue the **RUN HELD_QUERY** command.

To rerun the query in the background, issue the **RUN IN BACKGROUND QUERY** command.

Related concepts:

- "Managed query status" on page 153

Related tasks:

- "Dropping result tables manually using Query Patroller" on page 173

Related reference:

- "CANCEL QUERY" on page 204
- "RUN HELD_QUERY" on page 240
- "RUN IN BACKGROUND QUERY" on page 241

Viewing managed query details using Query Patroller

By viewing the properties of a query that has been managed by Query Patroller, you can see details such as information about the submitter of the query, the processing time, and the result table.

Prerequisites:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile has the MONITORING privilege with edit or view authority

- Be the submitter of the query

Procedure:

To view the details of a query:

Query Patroller Center method:

1. Open the Managed Query Properties notebook.
2. To view general information about the query, click the **General** tab.
   a. To view the SQL of the query in a new window, click **View SQL in Separate Window**.
   b. To view the profile of the submitter who submitted the query, click **Submitter Properties**. You must have DBADM authority or be an operator whose profile has the USER ADMINISTRATION privilege with edit or view authority to open the Submitter Properties window.
   c. If you want to change the status of the query, for example to cancel it, click **Change Status**.
3. To view query execution information and result table details, click the **Results** tab.
   a. To display the result table for the query, click **Show Results**.
   b. To save the result table for the query, click **Save Results**.

      Note: When saving the results of a query, a BLOB column value will not be saved. Instead it will be replaced with the keyword, ″BLOB″. A CLOB value will be truncated if it is greater than 32K in size.
   c. To delete the result table for the query, click **Drop Result Table**.
4. To view query event timestamps and processing durations, click the **Time** tab.
5. To view authorization IDs, application information, and user information, click the **Other** tab.
6. Click **Close** to close the Managed Query Properties notebook.

Command line method:

To view the details of a query, issue the **GET QUERY** command.

To save the results of a query, issue the **FILE RESULT** command.

To view the results of a query, issue the **SHOW RESULT** command.

Related tasks:

- “Viewing the SQL of managed queries using Query Patroller” on page 157
- “Viewing query submitter profiles”--*Help: Query Patroller*
- “Changing the status of queries using Query Patroller” on page 154

Related reference:

- “GET QUERY” on page 210
- “SHOW RESULT” on page 242
- “FILE RESULT” on page 207

Viewing the SQL of managed queries using Query Patroller

Use the SQL Statement window to view the SQL of a managed query. From this window, you can search for strings and SQL keywords in the statement, save the SQL statement to a file, print the SQL statement, and copy the SQL statement. You can paste the query into SQL Explain in order to find the access plan that the DB2 optimizer used for the SQL statement.

Prerequisites:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile has the MONITORING privilege with edit or view authority
- Be the submitter of the query

Procedure:

To view the SQL of a query, use any of the following methods.

Query Patroller Center method:

1. Open the SQL Statement window.
2. Click **Copy Text** to copy the SQL statement to the clipboard.
3. Click **Find** to find a specific text string in the SQL statement. This action opens a Find window.
4. Click **Save As** to open a standard Save As window. From there you can specify a file and location to save the SQL statement.
5. Click **Print** to open a standard Print window. From there you can select a printer and choose to print the SQL statement.
6. Click **Close** to close the SQL Statement window.

Related concepts:

- "Query interception and management in Query Patroller" on page 105
- "The SQL Compiler" in the *Federated Systems Guide*

Related tasks:

- "Finding specific text in the SQL statement of a managed query"--*Help: Query Patroller*

Viewing result tables using Query Patroller

A result table is created when:

- The query submitter's submission preferences specify that the application should be released after a query is submitted
- A query was held and then run
- A query was run in the background

You may want to view result tables in several situations:

- If you suspect the data you are interested in has not changed significantly since you last ran a query

- If you want to view the results of a query that was run in the past (Resubmitting the query may return different results.)
- If a query has been run by another Query Patroller submitter, and you have been granted access to that submitter's result tables

Prerequisites:

You must meet one of the following requirements:
- Have DBADM authority
- Be the submitter of the query
- Be granted access to the result table by the submitter of the query

Procedure:

To view the result table for a query, use any of the following methods.

Query Patroller Center method:

Note: You cannot use the Query Patroller Center to view another submitter's results, even if you have been granted access to them. You must use the command line.

1. Open the Show Results window. The Show Results window allows you to view a result table 50 rows at a time.
2. Click **Previous** or **Next** to move through the rows of the result table.
3. Click **Close** to close the Show Results window.

Command line method:

Issue the **SHOW RESULT** command.

Related concepts:
- "Result tables and result sets in Query Patroller" on page 104

Related tasks:
- "Viewing managed query details using Query Patroller" on page 155

Related reference:
- "SHOW RESULT" on page 242

Running held queries at a scheduled time

You can set held queries to run at a scheduled time and specify the maximum time that held queries can run. This allows you to schedule held queries to run during off-peak hours when there are fewer demands on your system.

Prerequisites:

You must meet one of the following requirements:
- Have DBADM authority
- Be an operator whose profile has the CONFIGURATION privilege with edit authority

Procedure:

To run held queries at a scheduled time, use one of the following methods.

Query Patroller Center method:

1. Open the Query Patroller System Properties notebook.
2. Click the **Held Queries** tab.
3. To set a time when held queries will be run:
   a. Select the **Run held queries at a scheduled time** check box.
   b. Click **Schedule Start Time** to open a window where you can manually schedule the time at which the queries will run. See Scheduling the start time for running held queries.

   Note: There is no default schedule for Run held queries. If you do not create a schedule, no held queries will be run. If you create a schedule that has an end date, after the end date passes, no more held queries will be run.
4. In the **How long to run held queries** field, specify the length of time over which held queries will be run. After the specified amount of time passes, any held queries that have not been run will be held until the next start time. Any queries that are currently running are allowed to run to completion.
5. Click **OK** to run held queries at a scheduled time.

Command line method:

Issue the **UPDATE QP_SYSTEM** command using the following parameters:

- RUN_HELD_QUERIES
- RUN_HELD_DURATION

Note: The **UPDATE QP_SYSTEM** command enables you to run held queries only if you have already manually set the schedule for running held queries through the Query Patroller Center. You cannot manually set schedules using the command line.

Related concepts:

- “Managed query status” on page 153

Related tasks:

- “Scheduling the start time for running held queries” on page 159

Related reference:

- “UPDATE QP_SYSTEM” on page 254

Scheduling the start time for running held queries

You can schedule held queries to run at a specific time, on a recurring basis. For instance, you can schedule held queries to run every Sunday at 1:00 in the morning.

Note: You can manually set schedules for running held queries only through the Query Patroller Center. You cannot manually set schedules using the command line.

Prerequisites:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile has the CONFIGURATION privilege with edit authority

Note: If you are an operator, but you do not have edit authority, you can still view the schedule, but you cannot change the schedule.

Procedure:

To schedule running held queries, use the following method.

Query Patroller Center method:

1. Open the Schedule Start Time for Running Held Queries window
2. To specify how often you want to run held queries, use the **Interval** and **Frequency** fields. Depending on the interval you select, you may also need to use the **Details** field.
3. To specify when to begin running held queries, use the **Date** and **Time** fields.
4. Specify an end date for running held queries. If there is no end date, select the **No end date** check box. Otherwise, to specify an end date, use the **Date** and **Time** fields.

   Note: The end date specified does not set the duration of a single task. It sets the end date for the recurring tasks. For instance, if you want to set held queries to run every Saturday for a duration of four hours, for the next seven weeks, use the **Date** and **Time** fields to specify an end date that is seven weeks from now.
5. Optional: To prevent the task from running, select the **Suspend schedule** check box.
6. Click the **Add** button to add the task to the **Schedule list**.
7. Click **OK** to return to the Query Patroller System Properties notebook, where you can set the duration of time that held queries will run.

Related concepts:

- "Scheduling — Overview"--*Help: Query Patroller*

Chapter 21. Using historical analysis

You can use Query Patroller historical analysis functions to analyze various aspects of your data warehouse activity over time. This chapter describes the different uses for historical analysis, and outlines how to generate and view historical data reports and graphs.

Uses for historical analysis reports

Historical analysis reports in the Query Patroller Center provide much useful information. These reports can be accessed by clicking the **Historical Analysis** folder, which expands to show the following folders:

- **Queries**
- **Tables Hit**
- **Tables Not Hit**
- **Indexes Hit**
- **Indexes Not Hit**
- **Submitters**

When data is available, the **Tables Hit** and **Submitters** folders can be further expanded to show more reports.

Most historical analysis reports can be viewed in table form and in graphical form. Use the **Show Table** button and the **Show Graph** or **Show Histogram** button to switch between the two views.

To sort a report (when you are looking at the table view), click the header of a column to sort the report by that column.

Table 9 provides a list of information you might want to know about your data warehouse, what you could use the information for, and how to find that information in the historical analysis reports. The reports can help you answer such questions as ″How long do queries spend in the queue before they are executed?″ and ″How many queries were processed last month?″

Table 9. Uses for historical analysis reports

Historical data	Use to	Reports to reference
Performance		
Average query execution time over time	Identify trends or variations in query performance; assess impact of hardware changes or warehouse redesign	Queries (sorted by **Execution Time**) Queries histogram (displaying **Queries Run -- Average Time** in the Y-axis) **Note:** Compare the results of these two reports over time.
Time spent in queue	Tune Query Patroller thresholds to minimize query execution time	Queries histogram (displaying **Queries Queued -- Total Queue Time** in the **Y-axis**)

Table 9. Uses for historical analysis reports (continued)

Historical data	Use to	Reports to reference
Number of queries processed in a given time unit	Understand data warehouse activity	Queries histogram (displaying **Queries Run -- Number Run** in the **Y-axis**)
Variation in number of queries processed in a given unit of time	Identify periods of high and low system activity to optimize maintenance schedules or to perform dynamic configuration of DB2®	Queries histogram (displaying **Queries Run -- Number Run** in the **Y-axis**)
Query cost	Identify inefficient or problematic queries	Queries (sorted by **Estimated Cost**)
Submitter activity		
Number of queries submitted by a user	Identify heavy users in order to tune submitter thresholds; identify submitters who require education about submitting queries	Submitters (sorted by **Submitter ID**)
Size of queries submitted by a user	Identify submitters of problematic queries; tune submitter thresholds	Queries (sorted by **Estimated Cost**)
Object usage by submitter or application	Determine the optimal location for specific objects depending on which groups of submitters are using the objects	Submitters —> Submitter X —> Tables Hit Submitters —> Submitter X —> Tables Hit —> Table X —> Columns Hit Submitters —> Submitter X —> Tables Hit —> Table X —> Indexes Hit
Object usage		
Database objects hit (tables, columns, indexes)	Identify candidates for materialized query tables, indexes, or multidimensional clustering	Tables Hit Indexes Hit Tables Hit —> Table X —> Columns Hit
Database objects not hit (tables, columns, indexes)	Identify objects to be eliminated or to move to slower access devices	Tables Not Hit Indexes Not Hit Tables Hit —> Table X —> Columns Not Hit

Related tasks:

- "Collecting historical data with Query Patroller" on page 163
- "Enabling collection of historical data" on page 125

Collecting historical data with Query Patroller

You can use Query Patroller's historical analysis functions to collect and analyze data about your data warehouse workload. You can either collect data on a test workload or collect data about your actual production activity. You can use the information you collect about a workload to gain greater understanding of database and object usage, set Query Patroller thresholds, and create query classes.

Prerequisites:

You must decide whether you will collect data about a test workload or your regular production activity. If you want to run a test workload, create a submitter and specify that all queries submitted by the test submitter profile be tracked for historical analysis. All other queries should not be intercepted.

If you are planning to collect data regarding query execution time, you need to ensure that the DB2 timestamp and statement monitor switches are set to 'ON'. If you are planning to collect data about how many rows are returned by queries, you need to ensure that the DB2 statement monitor switch is set to 'ON'.

Procedure:

To collect historical data, use the following method.

Query Patroller Center method:

1. Enable Query Patroller to intercept
2. Start Query Patroller by issuing the **qpstart** command.
3. Enable the collection of historical data
4. Leave all Query Patroller thresholds at their default settings, *or* set the Query Patroller thresholds to unlimited to allow queries to run unfettered. Leaving thesholds at their default settings or setting them to unlimited means that you will not be using the full query management features of Query Patroller, but you will be collecting data about the database workload.
5. Run your test workload, or allow database activity to proceed as normal.
6. Generate historical data to populate the historical analysis tables.
7. Open the Query Patroller Center and use the reports in the Historical Analysis folder to analyze your data.

Related concepts:

- "Query Patroller historical analysis" on page 103
- "Uses for historical analysis reports" on page 161

Related tasks:

- "Enabling Query Patroller to intercept queries" on page 83
- "Enabling collection of historical data" on page 125
- "Generating historical data using Query Patroller" on page 164

Related reference:

- "Historical data collection settings" on page 260

Generating historical data using Query Patroller

Generate historical data when you want to have current information available for historical analysis. Generating historical data runs the SQL Explain facility against the queries that Query Patroller has saved for historical analysis, and makes the information available in the Historical Analysis reports and graphs in the Query Patroller Center.

Other than the **Query Activity Over Time** report, the **Historical Analysis** folder only contains information about the queries for which you have generated historical data. To decide whether you should generate historical data, you can determine when it was last generated, and for which queries.

If SQL Explain was run unsuccessfully on a query, it will not be rerun when historical data is next generated, even if that query falls within the specified time range.

Prerequisites:

You must have DBADM authority.

Procedure:

To generate historical data, use one of the following methods.

Query Patroller Center method:

To generate historical data for all outstanding queries (queries that have not yet had historical data generated for them):

1. From the Query Patroller Center, find the **Historical Analysis** folder in the object tree.
2. Right-click the **Historical Analysis** folder and click **Generate Historical Data** in the pop-up menu. The Generate Historical Data window opens.
3. Click the **Generate data for all outstanding queries** radio button.
4. Click **OK** to generate historical data and close the window.

To generate historical data for queries within a specific time period:

1. From the Query Patroller Center, find the **Historical Analysis** folder in the object tree.
2. Right-click the **Historical Analysis** folder and click **Generate Historical Data** in the pop-up menu. The Generate Historical Data window opens.
3. Click the **Generate data for queries run within a period of time** radio button.
4. Using the **Start date** and **End date** fields, define the period of time for which you want to generate historical data.
5. Click **OK** to generate historical data and close the window.

You can stop the generation of historical data by clicking **Terminate** in the Progress window. The Progress window opens after you click **OK** in the Generate Historical Data window.

Command line method:

Issue the **GENERATE HISTORICAL_DATA** command.

You can stop the generation of historical data by issuing the **GENERATE HISTORICAL_DATA** command using the STOP parameter.

Related concepts:

- "Query Patroller historical analysis" on page 103
- "SQL explain facility" in the *Administration Guide: Performance*

Related tasks:

- "Determining when historical data was last generated" on page 165

Related reference:

- "GENERATE HISTORICAL_DATA" on page 205

Determining when historical data was last generated

The reports and graphs under the **Historical Analysis** folder only contain information about queries that have had historical data generated for them. To determine whether you should generate historical data, you can check:

- when historical data was last generated for all outstanding queries
- which queries have had historical data generated

Prerequisites:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile has the HISTORICAL ANALYSIS privilege with view or edit authority

Procedure:

To determine when historical data was last generated for all outstanding queries, use the following method.

Note: You must have DBADM authority.

Query Patroller Center method:

1. From the Query Patroller Center, find the **Historical Analysis** folder in the object tree.
2. Right-click the **Historical Analysis** folder and click **Generate Historical Data** in the pop-up menu. The **Generate Historical Data** window opens.
3. The date and time that historical data was last generated is displayed in the **Date data was last generated for all outstanding queries** field.

   Note: The **Date data was last generated for all outstanding queries** field in the **Generate Historical Data** window is not updated when you choose to generate data for a specific period of time. It is updated only when you choose to generate historical data for all outstanding queries.
4. Click **Cancel** to close the **Generate Historical Data** window.

To determine which queries have had historical data generated, use the following method.

Note: You must have DBADM authority or be an operator whose profile has the HISTORICAL ANALYSIS privilege with view or edit authority.

Query Patroller Center method:

1. From the Query Patroller Center, find the **Historical Analysis** folder in the object tree.
2. Click the **Historical Analysis** folder. The **Query Activity Over Time** report opens in the contents pane.
3. The **Explain Run** column in the **Query Activity Over Time** report displays whether SQL Explain has been run on a query.
   - If the **Explain Run** column for a query shows a status of **Ran successfully**, historical data has been generated for that query, and will appear in the **Historical Analysis** reports and graphs.
   - If the **Explain Run** column for a query shows a status of **Not yet run**, historical data has not been generated for that query.
   - If the **Explain Run** column for a query shows a status of **Ran unsuccessfully**, historical data has not been generated for that query, and therefore will not appear in any historical analysis reports or graphs. To determine why it was unsuccessful, you can examine the qpuser.log file.
4. Optional: You can sort the queries in the **Query Activity Over Time** report by clicking **Explain Run**.

Related concepts:

- "Query Patroller historical analysis" on page 103

Related tasks:

- "Generating historical data using Query Patroller" on page 164

Viewing historical query details using Query Patroller

Viewing the details of a historical query allows you to see information such as the SQL of the query, the processing time, user information, application information, and package information. By looking at the details of specific historical queries, you can examine information about which queries took a long time to complete and which queries were submitted at busy times.

Prerequisites:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile has the HISTORICAL ANALYSIS privilege with view or edit authority

Procedure:

To view historical query details, use the following method:

Query Patroller Center method:

1. Open the Historical Query Properties notebook.
2. To view the SQL statement and general information about the submitter and query results, click the **General** tab.
3. To view query event time stamps and processing durations, click the **Time** tab.

4. To view package information, click the **Package** tab.
5. To view information about the statement, application, and end user, click the **Other** tab.
6. Click **Close** to close the Historical Query Properties notebook.

Related concepts:

- "Query Patroller historical analysis" on page 103

Related tasks:

- "Viewing result tables using Query Patroller" on page 157
- "Viewing managed query details using Query Patroller" on page 155

Related reference:

- "GET QUERY" on page 210

Viewing index details using Query Patroller

When you are examining Query Patroller historical queries, you might want more information about the indexes that were used by those queries, as well as information about the indexes that were not used. Viewing the index details allows you to see the columns that the index is based on, the index definer, the index type, whether the index is required by the system, and more.

The information displayed in the Index Properties window comes from DB2 system catalog tables. The SYSCAT.INDEXES table is the source for the information in the following fields of the Index Properties window:

- Index name
- Index schema
- Table name
- Table schema
- Index definer
- Index type
- Index defined by user
- Index required by system
- Comments

The DB2 SYSCAT.INDEXCOLUSE table is the source for the information in the **Columns** table in the Index Properties window.

Prerequisites:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile has the HISTORICAL ANALYSIS privilege with view or edit authority

Procedure:

To view index details, use the following method.

Query Patroller Center method:

1. Open the Index Properties window. Index details are displayed in the Index Properties window.
2. Optional: Click on a column heading in the **Columns** table to sort by that heading.
3. Click **Close** to close the Index Properties window.

Related concepts:

- “Query Patroller historical analysis” on page 103

Related tasks:

- “Viewing historical query details using Query Patroller” on page 166

If your system is set up to gather historical data for your queries, you might eventually want to control the accumulation of data and periodically purge data that is no longer useful for your purposes. For information on such maintenance tasks, see Chapter 22, “Managing space,” on page 169.

Chapter 22. Managing space

An effective space management strategy can control the amount of storage used by managed and historical queries and result tables. When you install Query Patroller, default maintenance schedules are set up automatically. You might want to adjust these schedules according to your system workload and characteristics.

This chapter provides information on how to remove managed queries, historical queries, and result tables both manually and on schedule.

Setting Query Patroller maintenance schedules for queries and result tables

Managed queries and result tables consume space, so they should be removed periodically. You can schedule Query Patroller to purge managed queries and result tables at regular intervals, on an ongoing basis.

Prerequisites:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile has the CONFIGURATION privilege with edit authority

Procedure:

To set maintenance schedules for queries and result tables, use one of the following methods.

Query Patroller Center method:

1. Open the Query Patroller System Properties notebook.
2. Click the **Options** tab.
3. Under **Managed query settings**, use the **How long to keep queries** fields to specify how long to keep queries in the Managed Queries folder. To keep queries indefinitely, leave the field blank.
4. Use the **How long to keep result tables** fields to specify how long to keep result tables after the completion of a query. To keep result tables indefinitely, leave the field blank.

   Note: If you delete a managed query or a historical query, the corresponding result table is also deleted. Therefore the length of time specified in the **How long to keep queries** fields should be greater than or equal to the length of time specified in the **How long to keep result tables** fields.
5. Optional: In the **Table space for result tables** type the name of the table space where result tables will be stored. To use the default DB2 table space, leave the field blank.
6. Click **Schedule Purge Job** to open a window where you manually schedule purges of managed queries and result tables. See Scheduling purges of managed queries and result tables.

Note: If you delete a managed query, the corresponding query in the Historical Analysis folder is not deleted. However, if you purge a historical query, and it also exists under Managed Queries, it gets deleted in both places.

7. Click **OK** to set the maintenance schedules.

Command line method:

Issue the **UPDATE QP_SYSTEM** command using the following parameters:

- QUERY_PURGE_PERIOD
- RESULT_PURGE_PERIOD
- RESULT_TABLE_SPACE

Notes:

1. To schedule purges of queries and result tables you must set both the QUERY_PURGE_PERIOD parameter and the RESULT_PURGE_PERIOD parameter to a value other than -1 (the value -1 represents unlimited).
2. When a query is purged, the corresponding result table is also deleted. Therefore, you should set the value of the QUERY_PURGE_PERIOD parameter to be greater than or equal to the value of the RESULT_PURGE_PERIOD parameter. Otherwise, result tables will be purged at the same time as their associated queries, even if the value of the RESULT_PURGE_PERIOD parameter is greater.
3. You can use the **UPDATE QP_SYSTEM** command to schedule purges of queries and result tables only if a valid purge schedule exists. A default schedule for purging queries and result tables is created at installation. You can modify or delete the purge schedule using the Query Patroller Center. You cannot modify or delete the purge schedule using the command line.

Related concepts:

- “Result tables and result sets in Query Patroller” on page 104

Related tasks:

- “Scheduling purges of managed queries and result tables” on page 171

Related reference:

- “UPDATE QP_SYSTEM” on page 254

Managing historical queries

Historical queries are used by Query Patroller to generate historical data. Historical queries consume space, so they should be removed periodically. However, once historical queries have been removed, they can no longer be used for generating historical data. You can schedule Query Patroller to purge historical queries at regular intervals, on an ongoing basis.

Prerequisites:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile has the CONFIGURATION privilege with edit authority

Procedure:

To schedule purges of historical queries, use one of the following methods.

Query Patroller Center method:

1. Open the Query Patroller System Properties notebook.
2. Click the **Options** tab.
3. Under **Historical analysis**, specify the queries that you want to save by selecting the **Only managed queries** or the **All intercepted queries** radio button.
4. Under **Historical analysis**, use the **How long to keep queries** fields to specify how long to keep completed queries in the Historical Analysis view.
5. Click **Schedule Purge Job** to open a window where you manually schedule purges of historical queries. See Scheduling purges of historical queries.

   Note: If you delete a managed query, the corresponding query in the Historical Analysis folder is not deleted. However, if you purge a historical query, and it also exists under Managed Queries, it gets deleted in both places.
6. Click **OK** to schedule purges of historical queries.

Command line method:

Issue the **UPDATE QP_SYSTEM** command using the following parameters:

- QUERIES_TO_SAVE
- HISTORY_PURGE_PERIOD

Note: You can use the **UPDATE QP_SYSTEM** command to schedule purges of historical queries only if a valid purge schedule exists. A default schedule for purging historical queries is created at installation. You can modify or delete the purge schedule using the Query Patroller Center. You cannot modify or delete the purge schedule using the command line.

Related concepts:

- "Query Patroller historical analysis" on page 103

Related tasks:

- "Scheduling purges of historical queries" on page 172

Related reference:

- "UPDATE QP_SYSTEM" on page 254

Scheduling purges of managed queries and result tables

You can schedule purges of managed queries and result tables to run at a specific time, on a recurring basis. For instance, you can schedule purges of managed queries and result tables to run on the last Sunday of each month at 1:00 in the morning.

Note: You can manually set schedules purging managed queries only through the Query Patroller Center. You cannot manually set schedules using the command line.

Prerequisites:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile has the CONFIGURATION privilege with edit authority

Note: If you are an operator, but you do not have edit authority, you can still view the schedule, but you cannot change the schedule.

Procedure:

To schedule purges of managed queries and result tables, use the following method.

Note: The same schedule is used for purging both managed queries and result tables; you do not have to create separate schedules.

Query Patroller Center method:

1. Open the Schedule Managed Queries Purge Job window
2. To specify how often you want to purge managed queries, use the **Interval** and **Frequency** fields. Depending on the interval you select, you may also need to use the **Details** field.
3. To specify when to begin purging managed queries, use the **Date** and **Time** fields.
4. Specify an end date for purging managed queries. If there is no end date, select the **No end date** check box. Otherwise, to specify an end date, use the **Date** and **Time** fields.
5. Optional: To prevent the task from running, select the **Suspend Schedule** check box.
6. Click the **Add** button to add the task to the **Schedule list**.
7. Click **OK** to schedule the task.

Related concepts:

- "Result tables and result sets in Query Patroller" on page 104

Scheduling purges of historical queries

You can schedule purges of historical queries to run at a specific time, on a recurring basis. For instance, you can schedule historical queries to run on the last Sunday of each month at 1:00 in the morning.

Note: You can manually set schedules for purging historical queries only through the Query Patroller Center. You cannot manually set schedules using the command line.

Prerequisites:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile has the CONFIGURATION privilege with edit authority

Note: If you are an operator, but you do not have edit authority, you can still view the schedule, but you cannot change the schedule.

Procedure:

To schedule purges of historical queries, use the following method.

Query Patroller Center method:

1. Open the Schedule Historical Analysis Purge Job window
2. To specify how often you want to purge managed queries, use the **Interval** and **Frequency** fields. Depending on the interval you select, you may also need to use the **Details** field.
3. To specify when to begin purging historical queries, use the **Date** and **Time** fields.
4. Specify an end date for purging historical queries. If there is no end date, select the **No end date** check box. Otherwise, to specify an end date, use the **Date** and **Time** fields.
5. Optional: To prevent the task from running, select the **Suspend Schedule** check box.
6. Click the **Add** button to add the task to the **Schedule list**.
7. Click **OK** to schedule the task.

Related tasks:

- "Setting Query Patroller maintenance schedules for queries and result tables" on page 169
- "Removing historical queries manually" on page 175

Dropping result tables manually using Query Patroller

Drop a result table when you want to free up the space that the result table is consuming. A result table is created when:

- The query submitter's submission preferences specify that the application should be released after a query is submitted
- A query was held and then run
- A query was run in the background

There are several ways to drop result tables:

- You can drop a result table immediately using the Query Patroller Center.
- You can schedule result tables to be dropped using the Query Patroller Center.
- You can drop a result table using the command line.

Prerequisites:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile has the MONITORING privilege with edit authority
- Be the submitter of the query that produced the result table to be dropped

Procedure:

To drop a result table, use any of the following methods.

Query Patroller Center method #1:

1. Open the Managed Query Properties notebook.
2. Click the **Results** tab.
3. Click **Drop Result Table**.

Query Patroller Center method #2:

1. From the Query Patroller Center, expand the **Monitoring** folder to find the **Managed Queries** folder.
2. Click the **Managed Queries** folder. Any managed queries are displayed in the pane on the right side of the window (the contents pane).
3. In the contents pane, right-click the query that you want to drop the result table for, and click **Drop Result Table** in the pop-up menu.
4. Optional: You can drop several result tables at once by holding the **Ctrl** button as you click the queries whose result tables you want to drop. Then right-click and click **Remove** in the pop-up menu.

Query Patroller Center method #3:

You can also schedule result tables to be dropped at a specific time, by Scheduling purges of managed queries and result tables.

Command line method:

Issue the **REMOVE RESULT** command.

Related concepts:

- "Result tables and result sets in Query Patroller" on page 104

Related tasks:

- "Scheduling purges of managed queries and result tables" on page 171
- "Setting Query Patroller maintenance schedules for queries and result tables" on page 169
- "Viewing result tables using Query Patroller" on page 157

Related reference:

- "REMOVE RESULT" on page 236

Removing managed queries manually

Removing a managed query from Query Patroller removes the managed query information from the system. The associated historical query information is *not* removed. If a result table exists for the managed query being removed, the result table is dropped.

Instead of removing managed queries manually, you can also schedule queries to be purged.

Prerequisites:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile has the MONITORING privilege with edit authority

Procedure:

To remove managed queries manually, use one of the following methods.

Query Patroller Center method:

1. From the Query Patroller Center, expand the **Monitoring** folder to find the **Managed Queries** folder.
2. Click the **Managed Queries** folder. Any managed queries are displayed in the pane on the right side of the window (the contents pane).
3. In the contents pane, right-click the query that you want to remove, and click **Remove** in the pop-up menu.
4. Optional: You can drop several queries at once by holding the Ctrl button while selecting the queries you want to drop. Then right-click and click **Remove** in the pop-up menu.

Command line method:

Issue the **REMOVE QUERY_INFO** command.

Related tasks:

- "Setting Query Patroller maintenance schedules for queries and result tables" on page 169
- "Removing historical queries manually" on page 175

Related reference:

- "REMOVE QUERY_INFO" on page 232

Removing historical queries manually

Removing a historical query from Query Patroller removes the historical query information from the system. If it exists, the associated managed query information is also removed. If a result table exists for the historical query being removed, the result table is dropped.

Instead of removing historical queries manually, you can also schedule queries to be purged.

Prerequisites:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile has the HISTORICAL ANALYSIS privilege with edit authority

Procedure:

To remove a historical query manually, use one of the following methods.

Query Patroller Center method:

1. From the Query Patroller Center, expand the **Historical Analysis** folder to find the **Queries** folder.
2. Click the **Queries** folder. Historical queries that fall within the time range specified in the historical analysis toolbar are displayed in the pane on the right

side of the window (the contents pane). Use the historical analysis toolbar to specify the time range you are interested in.

3. In the contents pane, right-click the query that you want to remove, and click **Remove** in the pop-up menu.
4. Optional: You can drop several queries at once by pressing the Ctrl button and selecting the queries you want to drop. Then right-click and click **Remove** in the pop-up menu.

To remove a historical query manually for a particular submitter:

1. From the Query Patroller Center, expand the **Historical Analysis** folder to find the **Submitters** folder. Expand the **Submitters** folder and click on the folder for the particular submitter you want. Expand that submitter's folder to find the **Queries** folder.
2. Click the **Queries** folder. The submitter's historical queries that fall within the time range specified in the historical analysis toolbar are displayed in the pane on the right side of the window (the contents pane). Use the historical analysis toolbar to specify the time range you are interested in.
3. In the contents pane, right-click the query that you want to remove, and click **Remove** in the pop-up menu.
4. Optional: You can drop several queries at once by pressing the Ctrl button and selecting the queries you want to drop. Then right-click and click **Remove** in the pop-up menu.

Command line method:

Issue the **REMOVE QUERY_INFO_HISTORY** command.

Related tasks:

- "Setting Query Patroller maintenance schedules for queries and result tables" on page 169
- "Managing historical queries" on page 170
- "Removing managed queries manually" on page 174

Related reference:

- "REMOVE QUERY_INFO_HISTORY" on page 234

Part 7. Query Patroller tuning and troubleshooting

Chapter 23. Tuning Query Patroller

From time to time you may find that you experience some problems with your query workload. The following table describes some of the problems that you may encounter and suggests some possible causes that you can explore to remedy the situation.

Table 10. List of query management problems and possible causes

Problem	Possible cause
Too many queries are being held.	Maximum query cost for submitter profiles (MAX_COST_ALLOWED) is set too low.
Queries are queued too long or too often.	Maximum number of queries for submitters (MAX_QUERIES_ALLOWED) or the system (MAX_TOTAL_QUERIES) is set too low. If you have query classes defined in your system, you may have set the maximum number of queries. (MAX_QUERIES) too low for one or more of your query classes or you may have created too many query classes.
Small queries are running too slowly.	Submitter's minimum query cost to manage (MIN_COST_TO_MANAGE) is set too low.
Scheduled jobs are not completing overnight.	The setting for the duration of time to run held queries (RUN_HELD_DURATION) is too low.
Performance of time-critical queries is impacted after installing Query Patroller.	Consider allowing certain applications or submitters to bypass Query Patroller interception.
Too many queries are running unmanaged.	Submitter's minimum query cost to manage (MIN_COST_TO_MANAGE) is set too high or certain applications are not being intercepted by Query Patroller that should be intercepted.
Query cost estimates seem inaccurate.	Database statistics not accurate. Ensure that RUNSTATS is run periodically and after all significant database changes.

Related concepts:

- "Query Patroller configuration roadmap" on page 109
- "Query class configuration" on page 116
- "Submitter profile configuration" on page 113

Related reference:

- "RUN HELD_QUERY" on page 240
- "UPDATE QUERY_CLASS" on page 246
- "UPDATE SUBMITTER_PROFILE" on page 251
- "Query Patroller system threshold settings" on page 255

Chapter 24. Using Query Patroller with other DB2 components

Using the DB2 governor with Query Patroller

The main function of Query Patroller is to help database administrators manage queries against a database. The main function of the DB2 governor is to help administrators manage applications running against a database.

The DB2 governor allows you to place limits on resources such as the number of locks, the amount of idle time, and the amount of CPU used by an application. The DB2 governor can be used together with Query Patroller to provide a tremendous amount of administrative control. However, in order to use them together effectively, you must understand how they interact.

Query Patroller is a system of collaborative applications that run against a database. Since the governor can act against these applications the way it does against other applications, there are some guidelines to follow when specifying rules in the governor configuration file.

Specifically, it is important to avoid including the processes used by Query Patroller in the rules of the governor configuration file. Query Patroller uses javaw.exe, java.exe, db2fmp.exe, and qp.exe on Windows and java, db2fmp, and qp on UNIX to carry out its operations. To prevent the governor from acting against Query Patroller, do not include these processes in the governor configuration file. You should also make sure you do not have a general rule that intercepts all applications by default. Instead, explicitly include the list of applications to be intercepted by the DB2 governor.

Note: In addition to rules acting against Query Patroller processes, there may be other rules in the governor configuration file that can cause Query Patroller to be intercepted.

If the processes used by Query Patroller cannot be excluded from interception by the DB2 governor, then you should follow the following guidelines in writing your governor configuration file rules.

- The Query Controller uses javaw.exe and db2fmp.exe on Windows, and java and db2fmp on UNIX. If you must intercept these processes, set the rowssel and rowsread limits to a value that is considerably higher than the larger of the number of records in SYSCAT.DBAUTH where (DBADMAUTH='Y' and GRANTEETYPE='U') and the number of records where (DBADMAUTH='Y' and GRANTEETYPE='G'). If the DB2 governor still intercepts the Query Controller, retry setting the rowssel and rowsread limits with higher values.
- Query Patroller Center uses javaw.exe, java.eseand db2fmp.exe on Windows, and java and db2fmp on UNIX. If you must intercept these processes, set the rowssel and rowsread limits to a value that is considerably higher than the larger of the number of records in SYSCAT.DBAUTH where (DBADMAUTH='Y' and GRANTEETYPE='U') and the number of records where (DBADMAUTH='Y' and GRANTEETYPE='G'). Also, note that if rowssel and rowsread limits are smaller than the number of records in the largest Query Patroller control table, Query Patroller Center or Query Patroller Command Line Processor will be acted upon by the DB2 governor. There is no way to predetermine the maximum number of records in the Query Patroller control tables as they are mostly dynamic. If

desired, increase the rowssel and rowsread limits to values that are higher than the current maximum number of records or set them to unlimited.

Restrictiveness of other limits, including CPU usage and idle time, may also cause the DB2 governor to act against Query Patroller processes depending on the amount of time and resources used by Query Patroller to operate on Query Patroller control tables. Once again, this amount cannot be predetermined as it depends on hardware capability and data size. If desired, increase the limits to higher values to prevent the DB2 governor from acting against Query Patroller.

Effects of DB2 governor actions on Query Patroller processes

If the priority or schedule actions apply to the Query Patroller processes, Query Patroller will continue to run with reduced system resources. However, if a force action applies to a Query Patroller process, the process can be terminated. The force action might terminate a Query Patroller process normally, returning a SQL1224N return code, or it might cause an application error or an abnormal termination of the DARI process (SQL1131N) if the db2fmp process was started before the force was issued. Query Patroller cannot stop the db2fmp process after it has started. The db2fmp process will try to complete execution even after Query Patroller has shut down the database connection that the db2fmp process requires for successful execution.

For more information about the db2fmp process, see the *Application Development Guide: Programming Client Applications.*

Running Query Patroller and the DB2 governor against the same submitter applications

Both Query Patroller and the DB2 governor can be used against the same query submitter applications. For example, a submitting application such as DB2 CLP (db2bp.exe on Windows and db2bp on UNIX) can be listed as an application that is intercepted by Query Patroller as well as included in the governor configuration file.

Query Patroller intercepts queries at submission time, while the DB2 governor intercepts applications at query execution time. Since query submission occurs before query execution, Query Patroller will always intercept queries before the DB2 governor does. This means that if Query Patroller holds or queues a query, the DB2 governor must wait until the query is executed before intercepting the application that submitted the query.

A query that is intercepted by Query Patroller can be executed either by the submitting application or by another application called qprunquery.exe on Windows and qprunquery on UNIX. If the submitter’s submission preferences specify that the submitting application has to wait until the query results are returned before releasing the application, then it is the submitting application that will execute the query. If the submitting application is listed in the DB2 governor configuration file, then the DB2 governor will intercept the submitting application when it executes the query.

If the submitter’s submission preferences specify that the submitting application should be released and the query results sent to a result table, then the query is executed by qprunquery. In this situation, the DB2 governor will only intercept the application if qprunquery is included in the DB2 governor configuration file.

Using Query Patroller with the DB2 connection concentrator

When Query Patroller places a query in the queue, that query blocks the application the entire time it is in the queue, until the query is run.

When the DB2 connection concentrator is not activated, every application gets its own agent to manage the database connection until the application disconnects. When the concentrator is activated, all applications share a pool of agents which are switched between applications on transaction boundaries. This means that if the concentrator is enabled and Query Patroller queues queries, it ties up those agents until the queries are run. This would have the effect of reducing the pool of available agents and affect the performance of DB2 since applications would not be able to connect or execute a request due to their inability to acquire the services of an agent. For this reason, when the connection concentrator is activated, Query Patroller will not queue queries; instead, it will, by default, reject the queries which are supposed to be queued with an sqlcode 29009, reason code 6.

To prevent queries from being rejected when they are chosen to be queued, you can choose to allow Query Patroller to run queries instead of rejecting them when the concentrator is activated by setting the option BLOCK_OPTION at the system level, using the UPDATE QP_SYSTEM command, or at the user level, using the UPDATE SUBMITTER_PROFILE command. By default, BLOCK_OPTION is set to ′reject′ (′R′) specifying that queries are to be rejected rather than queued when the concentrator is activated. To specify that Query Patroller should run queries rather than reject them when the concentrator is activated, set BLOCK_OPTION to ′proceed′ (′P′).

For example, to allow Query Patroller to run queries against the database ″sample″ that would otherwise be rejected when the concentrator is activated, set BLOCK_OPTION option to ′P′ as follows:

```
qp -d sample -u userid -p password "UPDATE QP_SYSTEM USING BLOCK_OPTION 'P'"
```

To allow Query Patroller to run queries submitted under the profile ″STEVED″ that would otherwise be rejected when the concentrator is activated, set BLOCK_OPTION for this profile to ′P′ as follows:

```
qp -d sample -u userid -p password "UPDATE SUBMITTER_PROFILE
FOR USER'STEVED' USING BLOCK_OPTION 'P'"
```

The values for BLOCK_OPTION are stored in the QP_SYSTEM and SUBMITTER_PROFILE tables for the database.

The BLOCK_OPTION setting for QP_SYSTEM is not nullable; the BLOCK_OPTION setting for SUBMITTER_PROFILE is nullable. If the BLOCK_OPTION is set both for QP_SYSTEM and for a user′s submitter profile, the value for the submitter profile takes precedence for that user. For all other users, the BLOCK_OPTION setting for QP_SYSTEM applies. To ensure that the BLOCK_OPTION setting for QP_SYSTEM applies to a particular user, set the BLOCK_OPTION for that user′s SUBMITTER_PROFILE to NULL.

Chapter 25. Query Patroller, Version 8 limitations and restrictions

No interactive mode for query submission:

There is no interactive mode for query submission in Query Patroller Version 8. This means that there is no correlate to the Query Enabler dialog from Query Patroller Version 7. The Query Enabler window allowed the user to schedule a query to run at a later time and to select from the result sets of previously run queries whose SQL statement matched that of the submitted query. In Query Patroller Version 8, the administrator can schedule batches of held queries to run at a later time and users can view or save the results of their queries using the Query Patroller Center.

Incompatible SQL statements and Query Patroller functions:

There are several types of SQL queries which are incompatible with particular Query Patroller functions. These query types are listed in the following table.

Table 11. Query Patroller limitations by SQL statement type

Statement type	Intercept	Manage	Historical analysis	Schedule	Hold	Queue
Non-SELECT statements (UPDATE, INSERT, DELETE)	yes	yes	yes[1]	no	no	yes
Static SQL containing host variables	yes	yes	yes[1]	no	no	no
SQL queries from stored procedures	yes	yes	yes[1]	no	no	no
User-defined functions containing non-zero nested queries	yes	yes	yes[1]	no	no	no
SQL queries containing parameter markers or special registers	yes	yes	yes[1]	no	no	yes
SQL queries containing Declared Global Temporary Tables (DGTT)	yes	yes	no	no	no	yes
SQL queries containing an identity value function (IDENTITY_VAL_LOCAL()) or a sequence value function (seqno())	yes	yes	yes[1]	no	no	no
SQL queries containing an encrypt or decrypt function	yes	yes	yes[1]	no	no	yes

Notes:

1. Historical analysis is only performed on queries with a completion status of ″D″ (done). If a query of this type exceeds a threshold, then it will be assigned a completion status of ″R″ (rejected) rendering the query ineligible for historical analysis.

Quiesce mode:

DB2® Universal Database Version 8 offers a new maintenance mode called *quiesce* mode. When DB2 is in quiesce mode, Query Patroller will be bypassed by all queries.

Potential inconsistency between db2qp.result_info table with the database:

A situation may arise in which there is an inconsistency between the db2qp.result_info table and the result tables in the database. This inconsistency may occur if the table space containing the control tables undergoes a roll forward recovery to a point in time prior to the last update of the result tables which reside on a different table space. For example, if you perform a purge of your result tables, and subsequently restore and perform a roll-forward of the control table space to a point in time prior to the purge, the db2qp.result_info table will contain information that says the result tables still exist when they do not.

UDF provided to select SQL statement text from db2dqp.track_query_info table:

Since the data type for the statement field in the db2dqp.track_query_info table is BLOB, you will not be able to perform a simple SELECT statement to retrieve the values in the table. In order to query this field, you need to use the db2qp.convertToString function to convert the values for retrieval. For example, to select all the values from the statement field in this table, you will enter the following:

```
select db2qp.convertToString(statement) from db2qp.track_query_info
```

Timing considerations for held, queued, and analyzed queries:

If the execution of a query is delayed for any reason, such as when the query is held or queued, and if any tables accessed by the query have been changed or removed between the time that the query was submitted and when the query executes, there may be unexpected results or an error may occur.

If a query is being analyzed and the tables involved in the query have been changed or removed between the time the query was run and when the query is being analyzed, there may be unexpected results and an error may occur.

Command line support for non-English languages:

In order for non-English characters to appear properly during command line usage, Query Patroller commands must be entered in a DB2 CLP window for languages other than English.

Restriction on the generation of historical data with positioned UPDATE or DELETE statements:

Historical data cannot be generated for queries that include positioned UPDATE or positioned DELETE statements.

Query recovery:

On rare occasions, when the status of a queued or running query is changed, Query Patroller may be unable to record the new status immediately. This usually happens in an abnormal termination situation such as one of the following:

- The DB2 server terminates unexpectedly or is shut down with the **db2stop force** command while there are queries that are queued, running, or both.
- The Query Patroller server terminates unexpectedly or is shut down with the **qpstop** *dbname* **force** command while there are queries that are queued, running, or both.
- Both the DB2 server and the Query Patroller server terminate unexpectedly while there are queries that are queued, running, or both.This might occur in the event of a power outage or hardware failure.

Query Patroller server performs an automatic recovery at startup time and at periodic intervals. It checks for queries with a current status of queued or running and checks to see if the status is still accurate. If the current status is accurate, the query is handled normally and, if Query Patroller server was terminated and restarted, internal Query Patroller server data structure is restored. If, however, a query with a status of queued or running is found to no longer exist in DB2 because the DB2 server was terminated and restarted, or the Query Patroller was down and could not update the status of the query, then recovery will be performed on the query. The recovery action taken depends on whether the query was to return the results to a client application or to a DB2 result table:

- A running query that was to return results to a client application will have its status changed to unknown.
- A queued query that was to return results to a client application will have its status changed to aborted.
- A running query that was to return results to a DB2 result table will be automatically re-run.
- A queued query that was to return results to a DB2 result table will be automatically re-queued.

DBCLOB objects not available in Show result dialog:

Due to a JDBC limitation, DBCLOB objects can not be displayed in the Query Patroller **Show result** dialog window. Instead, an empty string will appear in place of DBCLOB objects in the dialog window. This limitation applies only to the Query Patroller Center, and not to the Query Patroller command line.

Apply filters when viewing a large number of queries:

The response time of Query Patroller Center might slow down considerably if you are viewing several hundred managed or historical queries. To alleviate this problem, it is recommended that you apply a filter to the views to reduce the number of queries displayed. For information on how to apply filters in the Query Patroller Center, see the *Query Patroller Guide: Installation, Administration, and Usage* or the Query Patroller information in the DB2 Information Center.

Queries run in the background issue loads:

When a query is run in the background, the results of the query are stored in a result table. Any query that will generate a result table is run by a process called *qprunquery*. This process creates a result table and issues a load from cursor to fill the table with the results of the query. This means that queries that produce result tables are subject to all of the same restrictions as any other load from cursor. For a complete description of these restrictions, see the documentation of the LOAD command in the *DB2 Command Reference*.

During each load performed by *qprunquery*, entries are placed in the db2diag.log file. On UNIX operating systems, one or more messages will be created in a subdirectory under the *INSTANCE*/db2dump directory, where *INSTANCE* is the directory where you installed DB2. On Windows, one or more messages will be created in a subdirectory under the directory specified in the diagpath database configuration parameter. The name of the message file subdirectory is generated based on the details of the load operation. For example, the following is the name of a generated message file subdirectory:

```
qpTbLoad_SAMPLE_349_2003-05-21-16.51.32
```

where:

- `qpTbLoad` specifies that this is a message file from the LOAD command run by **qprunquery**
- `SAMPLE` is the name of the database

- `349` is the query id for which the load was initiated
- `2003-05-21-16.51.32` is the timestamp just before the load is initiated

The message file name contained in this subdirectory would be as follows.

```
qpTbLoad_SAMPLE_349_2003-05-21-16.51.32.MSG.*
```

.

Note: For multipartioned databases, the LOAD command will create more than one message file and append different file extensions to each file name.

The message files are deleted once the load completes successfully. To aid in problem determination, the message files are not deleted if the load fails.

There is a limit to the number of simultaneous loads that can be run in parallel. Exceeding this limit results in an aborted query and an error SQL6555 recorded in the qpdiag.log file. If this error is encountered, you can remedy the situation by changing the range specified by the DB2ATLD_PORTS registry variable, which determines the number of parallel loads permitted at once. To calculate the approximate number of ports required in your system, decide on the maximum number of loads that need to be run at one time, including those issued by *qprunquery* and other load operations. Multiply this number by the number of logical partitions per physical partition in your environment. Add 25% to this amount.

To set the DB2ATLD_PORTS registry variable, issue the following command:

```
db2set DB2ATLD_PORTS=num1:num2
```

where num1< num2

Query Patroller uses a default of 6000 ports in the range 50000–56000; setting DB2ATLD_PORTS will override this value.

Out of memory errors with Query Controller or Query Patroller Center:

When Query Patroller is managing a large number of queries, and the Query Controller or the Query Patroller Center is running, you may receive an out of memory error even when there is sufficient memory available on the machine. In order to be able to use more of the available memory, you can increase the java heap environment variable settings from their default levels.

The environment variables to update are QP_INIT_JAVA_HEAP_SIZE and QP_MAX_JAVA_HEAP_SIZE. If these variables are not set, the default is 32 mb and 512 mb, respectively.

Related concepts:

- "Federated systems" in the *Federated Systems Guide*

Related reference:

- "QUIESCE Command" in the *Command Reference*

Part 8. Appendixes

Appendix A. Query Patroller commands

Query Patroller command line support

Query Patroller commands can be invoked from a command line window, preceded by the keyword qp. The general syntax for entering Query Patroller commands is described below.

Note: For languages other than English, Query Patroller commands must be run from a DB2 CLP window for the correct codepage to apply.

Command syntax:

```
►►──qp──┬──────────────────────────────┬──-d──database──qp-command──────────────►◄
        └──-u──username──-p──password──┘
```

Command parameters:

-u username
Specifies the user ID username under which to connect to the database and execute the Query Patroller command.

-p password
Specifies the password used to authenticate the user name. If the password is omitted, the user is prompted to enter it.

-d database
Identifies the database that the command is being run against.

qp-command
Specifies the Query Patroller command string to be executed.

Examples:

The following command cancels query 854 running against the SAMPLE database: `qp -d sample cancel query 854`

The following command, issued under the username "testuser," lists all of the query classes defined for the TESTDB database: `qp -u testuser -p testpw -d testdb list query_classes`

Usage Notes:

1. To view a list of all Query Patroller commands, enter one of the following commands:

   ```
   qp -d db-name ?
   ```

   or

   ```
   qp -d db-name help
   ```

   where db-name is the name of a database.

   If the client running the Query Patroller command is a remote client, then the user and password options are also required as follows:

   ```
   qp -d db-name -u user_id -p password ?
   ```

or

```
qp -d db-name -u user_id -p password help
```

2. All parameter values whose corresponding SQL types are char or varchar must have single quotes around them. If the parameter values themselves contain a single quote, an escape character is required. For example, the username Mike O’Connell must be specified as ’Mike O″Connell’.
3. On UNIX platforms, if a qp command string contains any single quote, the entire qp command must be surrounded by double quotes. For example, to create a new submitter profile for ″testuser″ you would enter the command: `qp -d wsdb "add submitter_profile 'TESTUSER' using default"`
4. If there is no existing connection, a database connection is established when the command is executed.

Related concepts:

- “Query Patroller components” on page 4

ADD OPERATOR_PROFILE

Adds a new operator profile to the set of Query Patroller operator profiles defined in the OPERATOR_PROFILE table.

Authorization:

You must meet one of the following requirements:

- Have DBADM authority

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line interface.

```
►►──ADD OPERATOR_PROFILE FOR──┬─USER──’username’───┬──────────────────────────►
                              └─GROUP──’groupname’─┘

►──┬─LIKE──┬─USER──’username’───┬─────────────────┬──────────────────────────►◄
   │       └─GROUP──’groupname’─┘                 │
   └─USING──┬─DEFAULT──────────────────────────┬──┘
            │  ┌─────────────────────────┐     │
            │  ▼                         │     │
            └────Setting-keyword──’value’─┴────┘
```

Command parameters:

USER username
: Specifies the username for which the operator profile is created. The user ID username must also exist as a DB2 authorization ID.

GROUP groupname
: Specifies the group name for which the operator profile is created. The groupname must be a DB2 user ID and be an existing ID at the operating system or Kerberos level.

DEFAULT
: Specifies that the operator profile should be created with the default values for all parameters. For any individual parameter that has a default value, the parameter can be set to the default by entering the parameter with DEFAULT as the value. For example, to set the MONITORING privilege for user A to the default value, enter the following:

```
qp -d sample "ADD  OPERATOR_PROFILE FOR USER 'USERA' USING
MONITORING DEFAULT"
```

LIKE USER username
: Specifies that the new operator profile is to have the same settings as the operator profile for USER username.

LIKE GROUP groupname
: Specifies that the new operator profile is to have the same settings as the operator profile for GROUP groupname.

Setting-keyword

CONFIGURATION

Specifies the authority level of the CONFIGURATION privilege to be assigned to the operator.

E Allows the operator to add, edit and view Query Patroller query classes and system settings.

V Allows the operator to view Query Patroller query classes and system settings.

N Renders the operator unable to view or edit Query Patroller query classes and system settings. This is the default value.

HISTDATA

Specifies the authority level of the HISTORICAL DATA privilege to be assigned to the operator.

E Allows the operator to view and remove historical data.

Note: You must have DBADM authority to generate historical data.

V Allows the operator to view historical data.

N Renders the operator unable to view or remove historical data. This is the default value.

MONITORING

Specifies the authority level of the MONITORING privilege to be assigned to the operator.

E Allows the operator to view and manage queries. This includes the ability to monitor, remove and change the status of managed queries, view query details, and delete result tables.

V Allows the operator to view the details of queries managed by Query Patroller.

N Renders the operator unable to view or manage queries. This is the default value.

USERADMIN

Specifies the authority level of the USER ADMINISTRATION privilege to be assigned to the operator.

E Allows the operator to edit and view Query Patroller operator and submitter profiles and other users' submission preferences.

V Allows the operator to view Query Patroller operator and submitter profiles and other users' submission preferences.

N Renders the operator unable to edit or view Query Patroller operator and submitter profiles and other users' submission preferences. This is the default value.

SUSPENDED

Indicates whether or not privileges for this operator profile are suspended.

N Privileges are not suspended for this operator profile. This is the default value.

Y Privileges are suspended for this operator profile.

Examples:

The following command creates an operator profile for the group HELPDESK on the sample database. The helpdesk group will be administering users and managing queries, but they do not need to view or edit system settings or view historical data. The helpdesk group operator profile gives members monitoring privilege with edit authority, and the user administration privilege with edit authority.

```
qp -d sample "ADD OPERATOR_PROFILE FOR GROUP 'HELPDESK' USING HISTDATA 'N'
CONFIGURATION 'N' MONITORING 'E' USERADMIN 'E' SUSPENDED 'N'"
```

Usage notes:

1. Operator profiles do not need to be created for users with DBADM authority on a database. Such users already possess the maximum level of operator privileges, therefore adding operator profiles for them is redundant. It may also be misleading to create an operator profile for a user with DBADM authority since the user can automatically perform all Query Patroller tasks despite any restrictions on the operator privileges associated with the profile.

ADD QUERY_CLASS

Adds a new query class to the list of query classes defined for the database.

Authorization:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile includes CONFIGURATION privilege with edit authority.

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line interface

```
►►──ADD QUERY_CLASS──USING──────────────────────────────────────────────►

►─MAX_QUERIES──max-number-of-queries──MAX_COST──max-query-cost──────────►

►──┬──────────────────────────────┬─────────────────────────────────────►◄
   └─DESCRIPTION──'description'───┘
```

Command parameters:

MAX_QUERIES max-number-of-queries
: The maximum number of queries that can run simultaneously for this query class. When this threshold of running queries is reached, additional queries for this query class are queued until resources become available. The value must be greater than 0 and less than or equal to the value of MAX_TOTAL_QUERIES specified in the Query Patroller system settings.

MAX_COST max-query-cost
: The maximum cost for a single query that this query class will accept. The value must be less than or equal to the value of MAX_TOTAL_COST specified in the Query Patroller System settings.

Note: All query classes defined in a system must have distinct values for MAX_COST.

DESCRIPTION description
: Specifies a description for the query class. This description must be 256 characters or less enclosed in single quotes. This parameter is nullable.

Examples:

The following example creates a new query class for the SAMPLE database. A maximum of 10 queries will be allowed to run simultaneously in this query class, and the maximum size for any one query will be 1000 timerons.

```
qp -d sample "ADD QUERY_CLASS USING MAX_QUERIES 10 MAX_COST 1000
DESCRIPTION 'This query class runs small sales queries.'"
```

Usage Notes:

1. You can create a maximum of 99 query classes in a system.
2. Query classes can be created, removed, or modified while Query Patroller is started. However, these changes will not take effect until Query Patroller has been stopped and restarted.

Related concepts:

- "Query Patroller query classes" on page 101
- "Query class configuration" on page 116

Related tasks:

- "Creating query classes for Query Patroller" on page 128

Related reference:

- "DB2 Query Patroller control tables" on page 263
- "REMOVE QUERY_CLASS" on page 231
- "GET QUERY_CLASS" on page 211
- "LIST QUERY_CLASSES" on page 218
- "UPDATE QUERY_CLASS" on page 246
- "Query Patroller command line support" on page 191
- "qpstart - Start Query Patroller" on page 228
- "qpstop - Stop Query Patroller" on page 229

ADD SUBMISSION_PREFERENCES

Creates a submission preferences file for a specified submitter.

Authorization:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile includes the USER ADMINISTRATION privilege with edit authority
- Be the owner of the submission preferences file being created.

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line interface.

```
►►──ADD SUBMISSION_PREFERENCES──┬──────────────────────┬──────────────────────────►
                                └─FOR USER──username─┘

►──┬─LIKE USER──username──────────────────────┬─────────────────────────────────►◄
   └─USING──┬─DEFAULT─────────────────────────┤
            │  ┌──────────────────────────┐   │
            │  ▼                          │   │
            └────parameter-keyword──value─┴───┘
```

Command parameters:

USER username
: Specifies the username of the submitter for whom the submission preferences files is created. If no username is specified, the current login ID is used. The user ID *username* must also exist as a DB2 authorization ID.

LIKE USER username
: Specifies that the new submission preferences file is to have the same settings as the submission preferences for USER *username*.

DEFAULT
: Specifies that this user uses the default submission preferences which are the submission preferences assigned to PUBLIC.

parameter-keyword value
: The following parameters can be set using this command:

SUBMITTER_PROFILE_NAME profile-name
SUBMITTER_PROFILE_TYPE profile-type
: Specifies the group profile to use to establish submitter settings, such as thresholds, if the submitter has no profile of type user. The value of SUBMITTER_PROFILE_TYPE must be ’group’. If multiple group profiles exist for this submitter and no group profile is specified here, queries will be submitted using the group profile with the lowest resource thresholds.

RESULT_EXCEEDED_ACTION
Specifies the action to take when the query results to be stored in a result table exceed the maximum result rows specified in the submitter’s profile.

’A’ Specifies that no results are stored in the result table if the result set is longer than the limit defined in the submitter’s profile. This option is the default.

’T’ Specifies that a truncated set of results is stored in the result table if the result set is longer than the limit defined in the submitter’s profile.

RESULT_ACCESSIBILITY
Specifies whether the result table containing query results will be available to more users than just the submitter.

’O’ Specifies that the result table is accessible by the DB2 IDs listed in the value of the OTHER_GRANTEES parameter. The DB2 IDs listed must have access to the database where the query was submitted.

’S’ Specifies that access to the result table is restricted to the query submitter. This option is the default.

OTHER_GRANTEES grantees
Specifies the DB2 user IDs or group IDs that can access the result table. Up to 1024 alphanumeric characters are accepted. Multiple IDs must be separated with commas.

RESULT_DESTINATION result-destination-id
Specifies whether the submitting application will wait for query results to return, or will be freed up for further activity.

’A’ Specifies that the application that submitted the query will wait for the result set to return while Query Patroller manages the query. When this option is selected, the application that submitted the query may become unresponsive until the result set is returned. This option is the default.

’T’ Specifies that the result set is stored in a DB2 table. When the query is submitted, the application that submitted the query becomes free for further processing.

EMAIL_ADDRESSES email-addresses
Specifies the e-mail address or addresses to receive notification regarding queries submitted by this submitter.

Note: This notification only applies if e-mail notification is enabled in the QP_SYSTEM settings.

The value for this parameter may be up to 1024 characters. Multiple e-mail addresses must be separated by a comma.

Related concepts:

- “Query Patroller query submission preferences” on page 147

Related tasks:

- “Setting query submission preferences for another submitter” on page 147
- “Setting your own query submission preferences” on page 299

Related reference:

- "DB2 Query Patroller control tables" on page 263
- "GET SUBMISSION_PREFERENCES" on page 212
- "LIST SUBMISSION_PREFERENCES" on page 219
- "UPDATE SUBMISSION_PREFERENCES" on page 248
- "Query Patroller command line support" on page 191
- "REMOVE SUBMISSION_PREFERENCES" on page 238

ADD SUBMITTER_PROFILE

Adds a new submitter profile to the SUBMITTER_PROFILE table.

Authorization:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile includes the USER ADMINISTRATION privilege with edit authority.

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line interface

```
►►──ADD SUBMITTER_PROFILE──────────────────────────────────────────────────►

►─FOR──┬─USER──'username'───┬──┬─LIKE──┬─USER──'username'───┬──────────┬──────►◄
       └─GROUP──'groupname'─┘  │       └─GROUP──'groupname'─┘          │
                               └─USING──┬─DEFAULT──────────────────────┬┘
                                        │  ┌─────────────────────────┐ │
                                        │  ▼                         │ │
                                        └──profile-parameter──'value'─┴─┘
```

Command parameters:

USER username
: Specifies the name of the user associated with the profile. The user ID *username* must also exist as a DB2 authorization ID.

GROUP groupname
: Specifies the name of the group associated with the profile. The group specified must be a DB2 group ID and exist as a group at the operating system level or Kerberos level.

USING DEFAULT
: Specifies that the submitter profile should be created with the default values for all parameters. For any individual parameter that has a default value, the parameter can be set to the default by entering the parameter with DEFAULT as the value. For example, to set the PRIORITY for USERA to the default value, enter the following:

```
qp -d sample "ADD SUBMITTER_PROFILE FOR USER 'USERA' USING PRIORITY
DEFAULT"
```

LIKE USER username
: Specifies that the new submitter profile is to have the same settings as the submitter profile for USER *username*.

LIKE GROUP groupname
: Specifies that the new submitter profile is to have the same settings as the submitter profile for GROUP *groupname*.

profile-parameter

Specifies the parameter values to be assigned to the profile. The following parameters can be set:

PRIORITY priority

Specifies the priority level assigned to queries submitted under this profile. The higher the value for this parameter, the higher the priority that is assigned to the submitter's queries.

- Values must be an integer between 0 and 999 inclusive
- The default value is 500

MAX_QUERIES_ALLOWED max-queries

Specifies the maximum number of queries that a submitter is able to run simultaneously. Queries submitted after this limit is reached are placed in a queued state until other submitted queries complete. When creating a submitter profile for a group, note that the value set for this parameter applies to each user. For example, if this value were set to 10 for Group A, then each user belonging to Group A has the authority to run 10 queries simultaneously.

- A value of ″-1″ indicates that users with this profile can have unlimited queries running simultaneously (up to the value of MAX_TOTAL_QUERIES specified in the QP_SYSTEM table.)
- The default value for this parameter is 100.

MAX_RESULT_ROWS max-number-of-result-rows

Specifies the maximum number of result rows that will be stored in a result table for a single query submitted under this profile. Only queries whose results are to be stored in a result table are subject to this limit.

- A value of ″-1″ indicates that users with this profile can store results with as many rows as necessary to accomodate the complete result set.
- The default value for this parameter is 1,000,000 rows.

MAX_COST_ALLOWED max-query-cost

Specifies the maximum query cost for a submitter under this profile. If the estimated cost of a query submitted under this profile exceeds this value, the query is placed in a held state.

- A value of ″-1″ indicates that users with this profile can run queries of any size (up to the value of MAX_TOTAL_COST specified in the QP_SYSTEM table.)
- The default value for this parameter is 10,000.

MIN_COST_TO_MANAGE min-query-cost

Specifies the minimum cost of a query managed by Query Patroller. Queries whose estimated cost is lower than this value are not managed by Query Patroller. A query that falls below this minimum cost is still tracked for historical analysis, provided the value of the QUERIES_TO_SAVE parameter in the QP_SYSTEM table is set to A (all queries). The default value is 1000.

ACCOUNT_ID account-id

Specifies an alphanumeric ID to use for account tracking purposes. Up to 128 characters will be accepted. You can use this parameter to sort submitters into logical groupings to track usage costs. This parameter is nullable.

SUSPENDED Y/N

Specifies whether or not a submitter is prohibited from submitting queries. The default value for this parameter is ″N″. The character value for this parameter must be enclosed in single quotes.

INTERCEPT Y/N

Specifies that Query Patroller is to intercept or manage queries submitted by this submitter. If queries are not intercepted, Query Patroller will not evaluate the cost of the query or track it for historical analysis. The default value for this parameter is ″Y″. The character value for this parameter must be enclosed in single quotes.

Examples:

The following example adds a submitter profile for the MARKETING group to use the SALES database. Because users in this group tend to submit large queries, the profile defined for this group allows queries with a large estimated cost and allows a large number of result rows to be stored in result tables. Users in this group also tend to submit fewer queries at once, so the value for MAX_QUERIES_ALLOWED is set to 100. The remaining settings for this profile are left as default values.

```
qp -d sales "ADD SUBMITTER_PROFILE FOR GROUP 'MARKETING' USING
MAX_QUERIES_ALLOWED 100 MAX_RESULT_ROWS 1000000000 MAX_COST_ALLOWED
1000000"
```

Usage Notes:

1. To print or view all of the settings for a particular submitter profile, use the GET SUBMITTER_PROFILE command.
2. To list all of the submitter profiles currently defined for a database, use the LIST SUBMITTER_PROFILES command.

Related concepts:

- “Submitter Properties — Overview”--*Help: Query Patroller*
- “Query Patroller submitter profiles” on page 137
- “Scenario: Managing query submitter needs by configuring submitter profiles” on page 89
- “Submitter profile configuration” on page 113

Related tasks:

- “Creating submitter profiles for users and groups” on page 141

Related reference:

- “DB2 Query Patroller control tables” on page 263
- “REMOVE SUBMITTER_PROFILE” on page 239
- “GET SUBMITTER_PROFILE” on page 213
- “LIST SUBMITTER_PROFILES” on page 220
- “UPDATE SUBMITTER_PROFILE” on page 251
- “Query Patroller command line support” on page 191

CANCEL QUERY

Cancels the specified query.

Authorization:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile includes MONITORING privilege with edit authority
- Be the submitter of the query to be canceled.

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line interface

```
►►──CANCEL QUERY──query-id──────────────────────────────────────────►◄
```

Command parameters:

QUERY query-id
Specifies the ID of the query to be canceled.

Related reference:

- "GET QUERY" on page 210
- "LIST QUERIES" on page 215
- "Query Patroller command line support" on page 191

GENERATE HISTORICAL_DATA

Collects data from the TRACK_QUERY_INFO table, runs SQL Explain facility against this data, and populates the QUERY_ANALYSIS table with the results. These results are used to generate the Historical Analysis reports and graphs in the Query Patroller Center.

Authorization:

You must meet one of the following requirements:

- Have DBADM authority

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line interface

```
►►──GENERATE HISTORICAL_DATA──┬────────────────────────────────────────┬──────────►◄
                              ├─STOP───────────────────────────────────┤
                              └─FROM START──start-time──END──end-time──┘
```

Command parameters:

STOP Stops the generation of historical data. When this option is specified, an update is made to the QP_SYSTEM control table indicating that a stop command has been issued. When the historical data generator is running, it checks the QP_SYSTEM table every 20 queries to see if a stop command has been issued. If so, generation of historical data will stop.

FROM START *start-time* **END** *end-time*
Specifies that historical data should be generated for database activity between *start-time* and *end-time*. The format for the *start-time* and *end-time* values is *'YYYY–MM–DD HH24:MI:SS'* where:

- YYYY represents a 4–digit year value
- MM represents a 2–digit month value from 1–12 (for example, January is represented by '01')
- DD represents a 2–digit day of the month value from 1–31
- HH24 represents an hour of the day value from 00–24

  Note: If the value for the hour is '24' the minutes and seconds values must be '00'.
- MI represents a 2–digit minutes value between 00–59
- SS represents a 2–digit seconds value between 00–59.

Usage Notes:

1. The results of this command can be viewed in the Query Patroller Center historical analysis reports and graphs.

2. It is strongly recommended that you run the historical data generator during periods of minimal database usage. Running this command during these off-peak hours minimizes the risk of a performance impact on the database.
3. It is recommended that you run the GENERATE_HISTORICAL_DATA command on a regular basis to reduce the number of queries that data is collected for at one time.

Related reference:

- "UPDATE QP_SYSTEM" on page 254
- "Historical data collection settings" on page 260
- "Query Patroller command line support" on page 191
- "REMOVE QUERY_INFO_HISTORY" on page 234

FILE RESULT

Sends the results of a specified query to a specified file.

Authorization:

You must meet one of the following requirements:

- Have DBADM authority
- Be the submitter who submitted the query
- Be granted access to the results of queries submitted by this user (specified in the submitter's submission preferences)

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line interface.

```
►►──FILE RESULT FOR──QUERY──query-id──TO──filepath─────────────────────────────►◄
```

Command parameters:

QUERY query-id

Specifies the ID of the query whose results are to be saved to a file.

Note: Only results that are stored in result tables can be saved.

TO filepath

The fully qualified path name for the file that is the destination for the saved query results.

Usage Notes:

1. When saving or filing the results of a query, a BLOB column value will not be saved. Instead it will be replaced with the keyword, ″BLOB″. A CLOB value will be truncated if it is greater than 32K in size.

Related reference:

- "Query Patroller command line support" on page 191

GET OPERATOR_PROFILE

Retrieves the settings for a specified Query Patroller operator profile.

Authorization:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile includes the USER ADMINISTRATION privilege with view or edit authority.

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line interface

```
►►──GET OPERATOR_PROFILE FOR──┬─USER──'username' ──┬──────────────────────────►◄
                              └─GROUP──'groupname' ─┘
```

Command parameters:

USER username
Specifies the name of the user whose operator profile is to be retrieved. The user ID *username* must also exist as a DB2 authorization ID.

GROUP groupname
Specifies the name of the group whose operator profile is to be retrieved. This group name must also exist as a DB2 authorization ID.

Examples:

The following command retrieves the profile settings for user jsmith for the TESTDB database:

```
qp -d testdb "GET OPERATOR_PROFILE FOR USER 'JSMITH'"
```

Related concepts:

- "Query Patroller operators" on page 133

GET QP_SYSTEM

Retrieves the system settings for a Query Patroller enabled database.

Authorization:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile includes the CONFIGURATION privilege with edit or view authority.

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line interface.

```
►►──GET QP_SYSTEM──────────────────────────────────────────────────────►◄
```

Related concepts:

- "Query Patroller System Properties — Overview"--*Help: Query Patroller*

GET QUERY

Retrieves the details of a specified query.

Authorization:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile includes the MONITORING privilege with view or edit authority
- Be the submitter of the query to be retrieved.

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line interface

```
►►──GET QUERY──query-id──┬─────────────────┬───────────────────────────────►◄
                         └─SHOW FULL QUERY─┘
```

Command parameters:

query-id
Specifies the ID of the query to be retrieved.

SHOW FULL QUERY
Specifies that the full query text is to be displayed or printed. By default, only the first 1KB of query text is returned unless this keyword is specified.

Related reference:

- “CANCEL QUERY” on page 204
- “Query Patroller command line support” on page 191

GET QUERY_CLASS

Retrieves the settings for a specified Query Patroller query class.

Authorization:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile includes CONFIGURATION privilege with view or edit authority.

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line interface

```
►►──GET QUERY_CLASS──query-class-id────────────────────────────────────────►◄
```

Command parameters:

query-class-id
The ID for the query class to be retrieved.

Related concepts:

- "Query Patroller query classes" on page 101

GET SUBMISSION_PREFERENCES

Retrieves the submission preferences for a specified Query Patroller user.

Authorization:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile includes the USER ADMINISTRATION privilege with at least view authority
- Be the owner of the submission preferences being retrieved.

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line interface.

```
►►──GET SUBMISSION_PREFERENCES──┬──────────────────────────┬──────────────────────►◄
                                └─FOR──USER──'username'──┘
```

Command parameters:

FOR username
The name of the user whose submission preferences are to be retrieved. The user ID *username* must also exist as a DB2 authorization ID. If this is not specified, then the submission preferences associated with the current user are returned.

Usage Notes:

1. If the specified username does not have any submission preferences defined, this command returns the values for the default PUBLIC submission preferences.

Related reference:

- “DB2 Query Patroller control tables” on page 263
- “LIST SUBMISSION_PREFERENCES” on page 219
- “UPDATE SUBMISSION_PREFERENCES” on page 248
- “ADD SUBMISSION_PREFERENCES” on page 198
- “REMOVE SUBMISSION_PREFERENCES” on page 238

GET SUBMITTER_PROFILE

Retrieves the setting details for a specified Query Patroller submitter profile.

Authorization:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile includes the USER ADMINISTRATION privilege with edit or view authority.

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line interface.

```
►►──GET SUBMITTER_PROFILE FOR──┬─USER──’username’──┬───────────────────────────►◄
                               └─GROUP──’groupnam’─┘
```

Command parameters:

USER username
: Specifies the name of the user whose submitter profile is to be retrieved. The user ID *username* must also exist as a DB2 authorization ID.

GROUP groupname
: Specifies the name of the group whose submitter profile is to be retrieved.The group name must also exist as a DB2 authorization ID.

Related concepts:

- “Query Patroller submitter profiles” on page 137

Related tasks:

- “Viewing query submitter profiles”--*Help: Query Patroller*

Related reference:

- “ADD SUBMITTER_PROFILE” on page 201
- “REMOVE SUBMITTER_PROFILE” on page 239
- “LIST SUBMITTER_PROFILES” on page 220
- “UPDATE SUBMITTER_PROFILE” on page 251
- “Query Patroller command line support” on page 191

LIST OPERATOR_PROFILES

Lists the settings for all of the Query Patroller operator profiles or a specified number of operators profiles for a database. The list is sorted alphabetically by profile name.

Authorization:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile includes the USER ADMINISTRATION privilege with view or edit authority.

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line interface.

```
>>──LIST OPERATOR_PROFILES──┬────────────────────────────────────┬──────────><
                            └─SHOW──┬─ALL──────────────────────┬─┘
                                    └─number-of-profiles-to-list─┘
```

Command parameters:

SHOW ALL
: Specifies that all of the operator profiles currently defined for the database are to be listed. This is the default behavior.

SHOW number-of-profiles-to-list
: Specifies the maximum number of operator profiles to be listed. Value must be a positive integer.

Usage Notes:

1. If the LIST OPERATOR_PROFILES command is entered without any parameters, all of the operator profiles for the system are displayed.

Related concepts:

- "Query Patroller System Properties — Overview"--*Help: Query Patroller*

LIST QUERIES

Lists the queries for a specified database. This list is displayed in descending order by query id.

Authorization:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile includes the MONITORING privilege with edit or view authority
- Be the submitter who owns the queries being listed.

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see the information about the Query Patroller command line interface.

```
►►──LIST QUERIES──┬──────────────────────────────────────┬────────────────────────►
                  ├─FOR──┬─USER ALL────────────────────┬─┤
                  │      ├─USER──’username’────────────┤ │
                  │      └─QUERY_CLASS──query-class-id─┘ │
                  ├─WITH STATUS──’status’────────────────┤
                  ├─CREATED BETWEEN──start──AND──stop────┤
                  ├─COMPLETED BETWEEN──start──AND──stop──┤
                  └─FROM──query-id1──TO──query-id2───────┘

►──┬───────────────────────────────────────┬──────────────────────────────────────►◄
   └─SHOW──┬─ALL─────────────────────────┬─┘
           └─number-of-queries-to-list───┘
```

Command parameters:

FOR USER ALL
Specifies that managed queries for all users are listed.

FOR USER username
Specifies that queries submitted by a particular submitter are listed. If the USER parameter is not specified, the default is to list the queries belonging to the current user.

FOR QUERY_CLASS query-class-id
Specifies that queries running in the query class identified by *query-class-id* are listed.

WITH STATUS status
Specifies that queries with a particular status are listed. The values for this parameter may be any of the following:

A Specifies that aborted queries are listed.

C Specifies that cancelled queries are listed.

D Specifies that completed ("done") queries are listed.

U Lists the queries whose status is unknown.

H Specifies that held queries are listed.

J Specifies that rejected queries are listed.

L Specifies that released queries are listed.

Q Specifies that queued queries are listed.

R Specifies that running queries are listed.

CREATED BETWEEN start AND stop
Specifies that queries created between the specified start and stop time are listed. The format for the start and stop time is '*YYYY-MM-DD HH24:MI:SS*' where:

- YYYY represents a 4–digit year value
- MM represents a 2–digit month value from 1–12 (for example, January is represented by '01')
- DD represents a 2–digit day of the month value from 1–31
- HH24 represents an hour of the day value from 00–24

  Note: If the value for the hour is '24' the minutes and seconds values must be '00'.
- MI represents a 2–digit minutes value between 00–59
- SS represents a 2–digit seconds value between 00–59.

COMPLETED BETWEEN start AND stop
Specifies that queries completed between the specified start and stop time are listed. The format for the start and stop time is '*YYYY-MM-DD HH24:MI:SS*' where:

- YYYY represents a 4–digit year value
- MM represents a 2–digit month value from 1–12 (for example, January is represented by '01')
- DD represents a 2–digit day of the month value from 1–31
- HH24 represents an hour of the day value from 00–24

  Note: If the value for the hour is '24' the minutes and seconds values must be '00'.
- MI represents a 2–digit minutes value between 00–59
- SS represents a 2–digit seconds value between 00–59.

FROM query-id1 TO query-id2
Specifies that queries with query ids between query-id1 and query-id2 are listed.

SHOW ALL
Specifies that all queries matching the specified criteria are displayed.

SHOW number-of-queries-to-list
Specifies the maximum number of queries matching the specified criteria to list. This number must be a positive integer.

Usage Notes:

1. If the LIST QUERIES command is entered without any parameters, a list of all managed queries issued by the current user is returned.

2. A submitter without DBADM authority or without an operator profile with MONITORING privilege is able to list only the queries owned by that submitter.

Related concepts:

- “Managed query status” on page 153

Related tasks:

- “Viewing managed query details using Query Patroller” on page 155

Related reference:

- “GET QUERY” on page 210
- “Query Patroller command line support” on page 191

LIST QUERY_CLASSES

Lists the settings for the Query Patroller query classes defined for a particular database.

Authorization:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile includes the CONFIGURATION privilege with edit or view authority.

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line interface.

```
►►──LIST QUERY_CLASSES──┬──────────────────────────────────────────┬──────────►◄
                        └─SHOW──┬─ALL─────────────────────────┬─┘
                                └─number-of-query-classes-to-list─┘
```

Command parameters:

ALL Specifies that all of the query classes currently defined for the database are to be listed.

number-of-query-classes-to-list
Specifies the maximum number of query classes to be listed. This value must be a positive integer.

Usage Notes:

1. If the LIST QUERY_CLASSES command is entered without any parameters, all of the query classes for the system are displayed.

Related concepts:

- "Query Patroller query classes" on page 101

LIST SUBMISSION_PREFERENCES

Lists the submission preferences for all Query Patroller submitters or a specified number of submitters for a database.

Authorization:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile includes the USER ADMINISTRATION privilege with edit or view authority.

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line interface.

```
►►──LIST SUBMISSION_PREFERENCES──────────────────────────────────────────────►

►──┬───────────────────────────────────────────────────┬──────────────────►◄
   └─SHOW──┬─ALL──────────────────────────────────────┤
           └─number-of-submission-preferences-to-list─┘
```

Command parameters:

ALL Specifies that all of the submission preferences currently defined for the database are listed.

number-of-submission-preferences-to-list
Specifies the maximum number of submission preferences that are listed. This value must be a positive integer.

Usage Notes:

1. If the LIST SUBMISSION_PREFERENCES command is entered without any parameters, all of the submission preferences for the system are displayed.

Related concepts:

- "Query Patroller query submission preferences" on page 147

LIST SUBMITTER_PROFILES

Lists the settings for all of the Query Patroller submitter profiles or a specified number of submitter profiles for a database.

Authorization:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile includes the USER ADMINISTRATION privilege with edit or view authority.

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line interface.

```
►►──LIST SUBMITTER_PROFILES──────────────────────────────────────────────────►

►──┬──────────────────────────────────────────────┬──────────────────────────►◄
   └─SHOW──┬─ALL──────────────────────────────────┤
           └─number-of-submitter-profiles-to-list─┘
```

Command parameters:

ALL Specifies that all of the submitter profiles currently defined for the database are to be listed.

number-of-submitter-profiles-to-list
Specifies the maximum number of submitter profiles to be listed. This value must be a positive integer.

Usage Notes:

1. If the LIST SUBMITTER_PROFILES command is entered without any parameters, all of the submitter profiles for the system are displayed.

Related concepts:

- "Query Patroller query submission preferences" on page 147

qpcenter - Start Query Patroller Center

Starts the Query Patroller Center. **qpcenter** can be issued from an operating system command prompt.

Authorization:

None.

Required connection:

Database.

Command syntax:

```
►►──qpcenter──┬──────────────────────────────┬──-d──database──────────────────────►◄
              └──-u──username────-p──password──┘
```

Command parameters:

-u username
Specifies the username user name under which to connect to the database.

-p password
The password used to authenticate the user name. If the password is omitted, the user is prompted to enter it.

-d database
Identifies the database with which to establish a connection.

Related tasks:

- "Installing the Query Patroller client tools using the DB2 Setup wizard (UNIX)" on page 34
- "Installing the Query Patroller client tools using the DB2 Setup Wizard (Windows)" on page 69

Related reference:

- "qpstart - Start Query Patroller" on page 228

qpmigrate - Migrate Query Patroller data and settings

Migrates Query Patroller Version 7 data and settings to the Query Patroller Version 8 system.

Authorization:

- You need DBADM authority to run the **qpmigrate** command.

Required connection:

None. This command establishes a database connection.

Command syntax:

```
►►──qpmigrate──┬──────────┬──┬────────────────────────────────────────────┬──────►
               └─NOPROMPT─┘  └─USER──user_name──┬───────────────────────┬─┘
                                                └─PASSWORD──password─┘

►─DB──db_name──┬─?──────────────────────────────────┬─────────────────────────►◄
               ├─HELP───────────────────────────────┤
               └─CLEANUP──┬──────────────────────┬──┘
                          └─KEEP_RESULT_TABLES─┘
```

Command parameters:

Note: The following command parameter key words are *not* case sensitive.

[NOPROMPT]
: This parameter specifies that no prompt will be given for a password (when USER is specified and PASSWORD is not specified). An error will be returned instead. If this option is used with the CLEANUP option, then the clean up operation will proceed immediately without displaying the prompt ″Are you sure you want to do this?″ for confirmation. If the NOPROMPT parameter is specified, then this must be the first parameter. This parameter is optional.

[USER *user_name* **]**
: This parameter specifies the name under which **qpmigrate** is run.

[PASSWORD *password***]**
: This parameter specifies the corresponding password of the above user. This parameter is required if USER is specified. If USER is specified but PASSWORD is not specified, then the program will prompt you to enter the PASSWORD (unless NOPROMPT is specified, in which case an error will be returned).

DB *db_name*
: This parameter specifies the name of the database on which you want to run the **qpmigrate** command. This parameter is required, except when displaying the **qpmigrate** command syntax.

?
: This parameter specifies that the **qpmigrate** command syntax is to be displayed. The **qpmigrate** command syntax is also displayed if no parameter is specified.

HELP
: This parameter specifies that the **qpmigrate** command syntax is to be displayed. The **qpmigrate** command syntax is also displayed if no parameter is specified.

CLEANUP [KEEP_RESULT_TABLES]

This parameter permanently removes the DB2 Query Patroller Version 7 control tables. After a normal migration using the **qpmigrate** command, it is suggested that you use the DB2 Query Patroller Version 8 system for a sufficient period of time to verify that the migration was successful. When you are certain that the migration was successful, and you no longer need the DB2 Query Patroller Version 7 data, then you can use this parameter to remove the DB2 Query Patroller Version 7 tables and free up disk space. CLEANUP will also remove the DB2 Query Patroller Version 7 result tables by default. If you want to keep the result tables, use the KEEP_RESULT_TABLES parameter.

Related concepts:

- "Automatic migration of Query Patroller server with the DB2 Setup wizard" on page 79
- "Installation tasks overview" on page 7

Related tasks:

- "Setting up Query Patroller server manually" on page 77
- "Migrating Query Patroller server manually" on page 80

Related reference:

- "qpsetup - Set up Query Patroller server" on page 224

qpsetup - Set up Query Patroller server

The **qpsetup** command can do the following:

- Set up a Query Patroller server on a specified DB2 database.
- Replace Query Patroller control tables on a specified control table space.

1. Set up a Query Patroller server on a specified DB2 database:

   Authorization:

   - You require SYSADM authority to run the **qpsetup** command.

   Required connection:

   None. This command establishes a database connection.

   Command Syntax:

```
>>--qpsetup--+------------------------------------------+--DB--db-name------------------------>
             '--USER--user-name--PASSWORD--password--'

>--+------------------------------------------------------------------------------------+----><
   +--CONTROL_TABLESPACE--tablespace-name--+------------------------------------------------+--+
   |                                       +--CONTROL_DBPARTITIONGROUP--db-partition-group-name--+
   |                                       |                       .-,---------------------.  |
   |                                       |                       V                       |  |
   |                                       +--CONTROL_DBPARTITIONNUM----db-partition-group-number-+
   |                                       +--CONTROL_PATH--tablespace-path-------------------+
   |                                       +--CONTROL_DMS--+-YES-+----------------------------+
   |                                       |               '-NO--'                            |
   |                                       +--CONTROL_DMS_CONTAINER--+-FILE---+---------------+
   |                                       |                         '-DEVICE-'               |
   |                                       '--CONTROL_DMS_NUMPAGES--number-of-4K-pages--------'
   '--RESULT_TABLESPACE--tablespace-name--+-------------------------------------------------+-'
                                          +--RESULT_DBPARTITIONGROUP--db-partition-group-name--+
                                          |                      .-,---------------------.   |
                                          |                      V                       |   |
                                          +--RESULT_DBPARTITIONNUM----db-partition-group-number-+
                                          +--RESULT_PATH--tablespace-path--------------------+
                                          +--RESULT_DMS--+-YES-+-----------------------------+
                                          |              '-NO--'                             |
                                          +--RESULT_DMS_CONTAINER--+-FILE---+----------------+
                                          |                        '-DEVICE-'                |
                                          +--RESULT_DMS_NUMPAGES--number-of-4K-pages---------+
                                          +--REPLACE-----------------------------------------+
                                          '--HELP--------------------------------------------'
```

   Command parameters:

   USER *user-name*
   : This parameter specifies the name under which **qpsetup** is run.

   PASSWORD *password*
   : This parameter specifies the corresponding password of the above mentioned user. This parameter is required if the USER parameter is specified. The above mentioned user will be prompted for a password if the user does not supply one.

   DB *db-name*
   : This parameter specifies the name of the database on which you want to set up DB2 Query Patroller. This parameter is required.

   CONTROL_TABLESPACE *tablespace-name*
   : This parameter specifies the name of the table space on which you want to create DB2 Query Patroller control tables. This parameter is required. If the table space does not exist, then it will be created. For increased performance and high availability, it is recommended that the control tablespace be in a single database partition group and at the same database partition where the DB2 Query Patroller server is launched.

CONTROL_DBPARTITIONGROUP *db-partition-group-name*
This parameter specifies the name of the database partition group on which you want to create the control table space. This parameter is optional. If the specified database partition group does not exist, then it will be created. If it is not specified, then the table space will be created on the default database partition group, IBMDEFAULTGROUP. This parameter is ignored if the control table space already exists. For increased performance and high availability, it is recommended that the control tablespace be in a single database partition group and on the same database partition where the DB2 Query Patroller server is launched.

CONTROL_DBPARTITIONNUM *db-partition-group-number*
This parameter specifies the database partition numbers on which you want to create the control database partition group. This parameter is optional. If it is not specified, then the database partition group will be created on all database partitions. This parameter will be ignored if the CONTROL_DBPARTITIONGROUP parameter is not specified, or if the database partition group already exists.

CONTROL_PATH *tablespace-path*
This parameter specifies the path of the table space containers for the control table space. This parameter is required if the control table space needs to be created. It will be ignored otherwise.

CONTROL_DMS YES | NO
This parameter specifies if the control table space to be created is a DMS table space or an SMS table space This parameter is optional. The default value is NO. This parameter is ignored if the control table space already exists.

CONTROL_DMS_CONTAINER FILE | DEVICE
This parameter specifies if the container type is FILE or DEVICE. This parameter is optional. The default value is FILE. This parameter is ignored if the control table space already exists and if the CONTROL_DMS parameter is not specified or has NO as its value.

CONTROL_DMS_NUMPAGES *number-of–4K-pages*
This parameter specifies the number of 4K pages to be created for the control table space. This parameter is required if the CONTROL_DMS parameter value is YES. This parameter is ignored if the control table space already exists.

RESULT_TABLESPACE *tablespace-name*
This parameter specifies the name of the table space on which you want to create the result tables. This parameter is required. If the table space does not exist, it will be created.

RESULT_DBPARTITIONGROUP *db-partition-group-name*
This parameter specifies the name of the database partition group on which you want to create the result table space. This parameter is optional. If it is not specified, then the table space will be created on the default database partition group, IBMDEFAULTGROUP. This parameter is ignored if the result table space already exists.

RESULT_DBPARTITIONNUM *db-partition-group-number*
This parameter specifies the database partition numbers on which you want to create the result database partition group. This parameter is optional. If it is not specified, then the database partition group will be created on all database partitions. This parameter will be ignored if the

RESULT_DBPARTITIONGROUP parameter is not specified, or if the specified database partition group already exists.

RESULT_PATH *tablespace-path*
This parameter specifies the path of the table space containers for the result table space. This parameter is required if the result table space needs to be created. This parameter will be ignored otherwise.

RESULT_DMS YES | NO
This parameter specifies if the result table space to be created is a DMS table space or an SMS table space. This parameter is optional. The default value is NO. This parameter is ignored if the result table space already exists.

RESULT_DMS_CONTAINER FILE | DEVICE
This parameter specifies if the container type is FILE or DEVICE. This parameter is optional. The default value is FILE. This parameter is ignored if the result table space already exists and if the RESULT_DMS parameter is not specified or has NO as its value.

RESULT_DMS_NUMPAGES *number-of–4K-pages*
This parameter specifies the number of 4K pages to be created for the result table space. This parameter is required if the RESULT_DMS parameter value is YES. This parameter is ignored if the result table space already exists.

REPLACE
This parameter specifies whether or not the DB2 Query Patroller control tables are to be replaced with new ones, if they already exist. This parameter is optional.

HELP This parameter specifies that the **qpsetup** command syntax is to be displayed.

2. Replace Query Patroller control tables on a specified control table space:

Authorization:

- You require DBADM authority to make specifications on existing table spaces using the **qpsetup** command.

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: If the table space does not exist, it will NOT be created and an error message will be returned.

```
►►──qpsetup──CONTROL_TABLES_ONLY──┬─────────────────────────────────────┬──────►
                                  └─USER──user-name──PASSWORD──password─┘

►─DB──db-name──CONTROL_TABLESPACE──tablespace-name──┬─────────┬──────────────►◄
                                                    └─REPLACE─┘
```

Command parameters:

CONTROL_TABLES_ONLY
This parameter specifies that **qpsetup** will only create the DB2 Query Patroller control tables, without creating table spaces nor binding packages.

USER *user-name*
This parameter specifies the name under which **qpsetup** is run.

PASSWORD *password*
This parameter specifies the corresponding password of the above user. This parameter is required if USER is specified. The above mentioned user will be prompted for a password if the user does not supply one.

DB *db-name*
This parameter specifies the name of the database on which you want to setup DB2 Query Patroller. This parameter is required.

CONTROL_TABLESPACE *tablespace-name*
This parameter specifies name of the table space on which you want to create DB2 Query Patroller control tables. This parameter is required. If the table space does not exist, then an error message will be returned.

REPLACE
This parameter specifies whether or not the DB2 Query Patroller control tables are to be replaced with new ones, if they already exist. This parameter is optional.

Usage notes:

1. For each database that you want to use Query Patroller with, you need to run the **qpsetup** command. For each database, this will create a set of Query Patroller control database objects, such as control tables, views and triggers associated with the tables, as well as user defined functions and procedures required for QP execution. The control tables contain information such as configuration settings, user profiles, and historical query data.

Related tasks:

- "Setting up Query Patroller server manually" on page 77
- "Migrating Query Patroller server manually" on page 80

Related reference:

- "qpmigrate - Migrate Query Patroller data and settings" on page 222

qpstart - Start Query Patroller

Starts Query Patroller. **qpstart** can be issued from an operating system command prompt.

Authorization:

You must be the owner of the instance containing the database that you want to run Query Patroller on.

Required connection:

Database.

Command syntax:

```
►►──qpstart──db-name────────────────────────────────────────────────────────►◄
```

Command parameters:

db-name
Specifies the name of the database for which you want Query Patroller to manage queries.

Usage notes:

- Start DB2 before starting Query Patroller.
- The configuration parameter *dyn_query_mgmt* must be set to ENABLE for the database for which you want Query Patroller to manage queries.

Related tasks:

- "Starting Query Patroller" on page 83
- "Stopping Query Patroller" on page 84

Related reference:

- "qpstop - Stop Query Patroller" on page 229

qpstop - Stop Query Patroller

Stops Query Patroller. **qpstop** can be issued from an operating system command prompt.

Authorization:

You must have DBADM authority.

Required connection:

Database.

Command syntax:

```
►►──qpstop──db-name──┬───────┬──────────────────────────────────────────►◄
                     └─force─┘
```

Command parameters:

db-name
Specifies the name of the database for which you want Query Patroller to stop managing queries.

force
Specifies that active queries should be forced. Forced queries will be in an inconsistent state until Query Patroller is restarted and query recovery is completed.

Related tasks:

- “Starting Query Patroller” on page 83
- “Stopping Query Patroller” on page 84

Related reference:

- “qpstart - Start Query Patroller” on page 228

REMOVE OPERATOR_PROFILE

Deletes the specified operator profile from the set of Query Patroller operator profiles.

Authorization:

You must meet one of the following requirements:

- Have DBADM authority

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line interface.

```
►►──REMOVE OPERATOR_PROFILE FOR──┬─USER──'username'───┬──────────────────────────►◄
                                 └─GROUP──'groupname'─┘
```

Command parameters:

USER username
Specifies the name of the user whose operator profile is to be deleted. The user ID *username* must also exist as a DB2 authorization ID.

GROUP groupname
Specifies the name of the group whose operator profile is to be deleted. This group name must also exist as a DB2 authorization ID.

Examples:

The following command removes the operator profile for ″sdiniro″ in the SAMPLE database:

```
qp -d sample "REMOVE OPERATOR_PROFILE FOR USER 'SDINIRO'"
```

Related tasks:

- "Removing operator profiles for users and groups"--*Help: Query Patroller*

REMOVE QUERY_CLASS

Removes a query class definition from the Query Patroller QUERY_CLASS control table.

Authorization:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile includes the CONFIGURATION privilege with edit authority.

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line interface

```
►►──REMOVE QUERY_CLASS──query-class-id─────────────────────────────────►◄
```

Command parameters:

query-class-id
The identifier for the query class to be removed.

Examples:

The following example removes query class 5 from the list of query classes defined for the TESTDB database:

```
qp -d testdb "REMOVE QUERY_CLASS 5"
```

Usage Notes:

1. Query classes can be created, removed, or modified while Query Patroller is started. However, these changes will not take effect until Query Patroller has been stopped and restarted.

Related reference:

- “DB2 Query Patroller control tables” on page 263
- “ADD QUERY_CLASS” on page 196
- “GET QUERY_CLASS” on page 211
- “LIST QUERY_CLASSES” on page 218
- “UPDATE QUERY_CLASS” on page 246
- “Query Patroller command line support” on page 191
- “qpstart - Start Query Patroller” on page 228
- “qpstop - Stop Query Patroller” on page 229

REMOVE QUERY_INFO

Deletes the information for a particular query or set of queries from the MANAGE_QUERY_INFO control tables. This means that the information is not available in the Managed query folders in the Query Patroller Center interface or through the GET QUERY or LIST QUERY commands.

Authorization:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile includes the MONITORING privilege with edit authority.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line interface

```
►►──REMOVE QUERY_INFO──┬─FOR QUERY──query-id──────────────┬────────────────────────►◄
                       ├─FOR ALL──────────────────────────┤
                       └─OLDER THAN──value──┬─MINUTES─┬───┘
                                            ├─HOURS───┤
                                            └─DAYS────┘
```

Command parameters:

FOR QUERY query-id
Specifies the ID of the query whose information is removed.

FOR ALL
Specifies that all query information in the MANAGE_QUERY_INFO control table should be deleted.

OLDER THAN value time-unit
Indicates that queries older than the specified time are deleted from the MANAGE_QUERY_INFO table.

time-units

MINUTES
Queries older than the specified number of minutes are deleted.

HOURS
Queries older than the specified number of hours are deleted.

DAYS Queries older than the specified number of days are deleted.

Usage Notes:

1. When a query is deleted, any related result table or result set information is deleted.
2. Queries in Queued or Running state cannot be removed.

Related reference:

- “DB2 Query Patroller control tables” on page 263
- “System maintenance settings” on page 258
- “Query Patroller command line support” on page 191
- “REMOVE QUERY_INFO_HISTORY” on page 234

REMOVE QUERY_INFO_HISTORY

Deletes the information for a particular query or set of queries from the TRACK_QUERY_INFO control tables. This means that, once this command is run, the information is not available in the historical analysis views in the Query Patroller Center interface.

When the information for a query has been deleted from the TRACK_QUERY_INFO control table, it is also deleted from the MANAGE_QUERY_INFO table, and any result set information corresponding to that query is deleted from the RESULT_INFO table. This means that the query is not viewable from the managed queries views in the Query Patroller Center, and result tables for the query are not available.

Authorization:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile includes the HISTORICAL DATA privilege with edit authority.

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line interface

```
►►──REMOVE QUERY_INFO_HISTORY──┬─FOR QUERY──query-id───────┬──────────────►◄
                               ├─FOR ALL───────────────────┤
                               └─OLDER THAN──value──┬─DAYS──┬┘
                                                    └─YEARS─┘
```

Command parameters:

FOR QUERY query-id
Specifies the ID of the query whose information is removed.

FOR ALL
Specifies that all query information in the TRACK_QUERY_INFO control table should be deleted.

OLDER THAN value time-unit
Indicates that historical information for queries older than the specified time is deleted from the TRACK_QUERY_INFO table.

time-units

DAYS Historical information for queries older than the specified number of days is deleted.

YEARS
Historical information for queries older than the specified number of years is deleted.

Usage Notes:

1. When a query is deleted from the TRACK_QUERY_INFO table, any corresponding managed query in the MANAGE_QUERY_INFO table, any result table or any result information is also deleted.

Related reference:

- "DB2 Query Patroller control tables" on page 263
- "Historical data collection settings" on page 260
- "Query Patroller command line support" on page 191
- "REMOVE QUERY_INFO" on page 232

REMOVE RESULT

Drops the DB2 table containing the result set for a specified query or set of queries.

Authorization:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile includes MONITORING privilege with edit authority
- Be the submitter of the query or queries that generated the results.

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line interface.

```
►►──REMOVE RESULT──┬─FOR QUERY──query-id─────────────────┬──────────────────────►◄
                   ├─FOR ALL─────────────────────────────┤
                   ├─FOR USER──'username' ───────────────┤
                   └─OLDER THAN──value──┬─MINUTES─┬──────┘
                                        ├─HOURS───┤
                                        └─DAYS────┘
```

Command parameters:

QUERY query-id
The ID of the query that generated the results to be deleted.

ALL Indicates that all of the result sets for the specified database are to be deleted.

USER username
Specifies that all of the result sets for the queries submitted by USER username are to be deleted.

OLDER THAN value time-unit
Indicates that result tables older than the specified time are deleted.

time-units

MINUTES
Result tables older than the specified number of minutes are deleted.

HOURS
Result tables older than the specified number of hours are deleted.

DAYS Result tables older than the specified number of days are deleted.

Examples:

The following example drops the DB2 table containing the results for query 958 executed on the SAMPLE database:

```
qp -d sample "REMOVE RESULT FOR QUERY 958"
```

Related concepts:

- "Result tables and result sets in Query Patroller" on page 104

Related reference:

- "System maintenance settings" on page 258
- "Query Patroller command line support" on page 191
- "SHOW RESULT" on page 242
- "FILE RESULT" on page 207

REMOVE SUBMISSION_PREFERENCES

Deletes the submission preferences file for a particular Query Patroller submitter.

Authorization:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile includes the USER ADMINISTRATION privilege with edit authority
- Be the owner of the submission preferences to be removed.

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line processor

```
►►──REMOVE SUBMISSION_PREFERENCES FOR──┬─USER──'username' ──┬──────────────────►◄
                                       └─GROUP──'groupname' ─┘
```

Command parameters:

USER username
Specifies the name of the user whose submission preferences file is to be deleted. The user ID *username* must also exist as a DB2 authorization ID.

GROUP groupname
Specifies the name of the group whose submission preferences file is to be deleted. This group name must also exist as a DB2 authorization ID.

Related concepts:

- "Query Submission Preferences — Overview"--*Help: Query Patroller*
- "Query Patroller query submission preferences" on page 147

Related tasks:

- "Removing query submission preferences for submitters"--*Help: Query Patroller*

Related reference:

- "DB2 Query Patroller control tables" on page 263
- "GET SUBMISSION_PREFERENCES" on page 212
- "LIST SUBMISSION_PREFERENCES" on page 219
- "UPDATE SUBMISSION_PREFERENCES" on page 248
- "Query Patroller command line support" on page 191
- "ADD SUBMISSION_PREFERENCES" on page 198

REMOVE SUBMITTER_PROFILE

Deletes a specified submitter profile from the Query Patroller SUBMITTER_PROFILE control tables.

Authorization:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile includes the USER ADMINISTRATION privilege with edit authority.

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line processor

```
►►──REMOVE SUBMITTER_PROFILE FOR──┬─USER──'username'───┬──────────────────────►◄
                                  └─GROUP──'groupname'─┘
```

Command parameters:

USER username
Specifies the name of the user whose submitter profile is to be deleted. The user ID *username* must also exist as a DB2 authorization ID.

GROUP groupname
Specifies the name of the group whose submitter profile is to be deleted. This group name must also exist as a DB2 authorization ID.

Examples:

The following command removes the submitter profile in the SALES database for the group ″managers″:

```
qp -d sales "REMOVE SUBMITTER_PROFILE FOR GROUP 'MANAGERS'"
```

Related tasks:

- "Removing submitter profiles for users and groups"--*Help: Query Patroller*

Related reference:

- "DB2 Query Patroller control tables" on page 263
- "ADD SUBMITTER_PROFILE" on page 201
- "GET SUBMITTER_PROFILE" on page 213
- "LIST SUBMITTER_PROFILES" on page 220
- "UPDATE SUBMITTER_PROFILE" on page 251
- "Query Patroller command line support" on page 191

RUN HELD_QUERY

Releases a held query. This means that the query runs as soon as resources are available.

Authorization:

You must meet one of the following requirements:
- Have DBADM authority
- Be an operator whose profile includes the MONITORING privilege with edit authority.

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line interface.

```
►►──RUN HELD_QUERY──query-id────────────────────────────────────────────►◄
```

Command parameters:

query-id
Specifies the ID of the held query to be run.

Related concepts:
- "Managed query status" on page 153

Related tasks:
- "Changing the status of queries using Query Patroller" on page 154
- "Running held queries at a scheduled time" on page 158

Related reference:
- "CANCEL QUERY" on page 204
- "Held query handling settings" on page 256
- "Query Patroller command line support" on page 191

RUN IN BACKGROUND QUERY

Runs a query that was previously submitted. The query may be running or queued. Query Patroller stops the execution of the query and gives control back to the client while the query is resubmitted. The results of the rerun query are stored in a result table.

Authorization:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile includes the MONITORING privilege with edit authority
- Be the submitter who submitted the query originally.

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line interface.

```
►►──RUN IN BACKGROUND QUERY──query-id────────────────────────────────────►◄
```

Command parameters:

query-id
Specifies the ID of the query to be run.

Related concepts:

- "Managed query status" on page 153

Related tasks:

- "Changing the status of queries using Query Patroller" on page 154
- "Running held queries at a scheduled time" on page 158

Related reference:

- "Query Patroller command line support" on page 191

SHOW RESULT

Displays the results for a specified query.

Authorization:

You must meet one of the following requirements:

- Have DBADM authority
- Be the submitter who submitted the query
- Be granted access to the results of queries submitted by this user (specified in the submitter's submission preferences)

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line interface.

```
►►──SHOW RESULT FOR──QUERY──query-id──┬──────────────────────────┬──────────►◄
                                      └─┬─WITH────┬─COLUMN NAMES─┘
                                        └─WITHOUT─┘
```

Command parameters:

QUERY query-id
Specifies the ID of the query whose results are to be displayed.

Note: Only results that are stored in result tables can be viewed.

WITH/WITHOUT COLUMN NAMES
Specifies whether or not column names should be displayed in the query results. If no option is specified, column names will be displayed by default.

Examples:

The following command displays the results for query 88 with column names:

```
qp -d sample "show result for query 88"
```

Related concepts:

- "Result tables and result sets in Query Patroller" on page 104

Related tasks:

- "Viewing result tables using Query Patroller" on page 157

Related reference:

- "REMOVE RESULT" on page 236
- "Query Patroller command line support" on page 191
- "FILE RESULT" on page 207

UPDATE OPERATOR_PROFILE

Updates a specified Query Patroller operator profile.

Authorization:

You must meet one of the following requirements:
- Have DBADM authority

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line interface.

```
►►──UPDATE OPERATOR_PROFILE FOR──┬─USER──'username'───┬──────────────────────────►
                                 └─GROUP──'groupname'─┘

►─USING──┬─DEFAULT──────────────────────────┬───────────────────────────────────►◄
         │   ┌────────────────────────────┐ │
         │   ▼                            │ │
         └───Setting-keyword──'value'─────┴─┘
```

Command parameters:

USER username
Specifies the username of the operator profile to be modified. The user ID *username* must also exist as a DB2 authorization ID.

GROUP groupname
Specifies the group name of the operator profile to be modified.This group name must also exist as a DB2 authorization ID.

Setting-keyword value
The following operator profile parameters can be set using this command:

DEFAULT
Specifies that the operator profile should be created with the default values for all parameters. For any individual parameter that has a default value, the parameter can be set to the default by entering the parameter with DEFAULT as the value.For example, to change the CONFIGURATION privilege for USERA to the default value, enter the following:

```
qp -d sample "UPDATE OPERATOR_PROFILE FOR USER 'USERA' USING
CONFIGURATION DEFAULT"
```

CONFIGURATION
Specifies the authority level of the CONFIGURATION privilege to be assigned to the operator.

E Allows the operator to edit and view Query Patroller query class and system settings.

V Allows the operator to view Query Patroller query class and system settings.

N Renders the operator unable to view or edit Query Patroller query class and system settings. This is the default value for this parameter.

HISTDATA
Specifies the authority level of the HISTORICAL DATA privilege to be assigned to the operator.

E Allows the operator to view and remove historical data.

Note: You must have DBADM authority to generate historical data.

V Allows the operator to view historical data.

N Renders the operator unable to view or remove historical data. This is the default value for this parameter.

MONITORING
Specifies the authority level of the MONITORING privilege to be assigned to the operator.

E Allows the operator to view and manage queries. This includes the ability to monitor, remove, and change the status of managed queries, view query details, and delete result tables.

V Allows the operator to view the details of queries managed by Query Patroller.

N Renders the operator unable to view or manage queries. This is the default value for this parameter.

USERADMIN
Specifies the authority level of the USER ADMINISTRATION privilege to be assigned to the operator.

E Allows the operator to edit and view Query Patroller operator and submitter profiles and submission preferences.

V Allows the operator to view Query Patroller operator and submitter profiles and submission preferences.

N Renders the operator unable to edit or view Query Patroller operator and submitter profiles and submission preferences. This is the default value for this parameter.

SUSPENDED
Indicates whether or not privileges for this operator profile are to be suspended.

N Privileges are not suspended for this operator profile. This is the default value for this parameter.

Y Privileges are suspended for this operator profile.

Examples:

The following command updates the operator profile for the HELPDESK group, giving operators with this profile the ability to change the status of held queries run against the PRODUCTION database:

```
qp -d production "UPDATE OPERATOR_PROFILE FOR GROUP 'HELPDESK' USING
MONITORING 'E'"
```

Related concepts:

- "Query Patroller operator profiles" on page 133

Related tasks:

- "Changing operator profiles for users and groups"--*Help: Query Patroller*

Related reference:

- "DB2 Query Patroller control tables" on page 263
- "ADD OPERATOR_PROFILE" on page 193
- "REMOVE OPERATOR_PROFILE" on page 230
- "GET OPERATOR_PROFILE" on page 208
- "LIST OPERATOR_PROFILES" on page 214
- "Query Patroller command line support" on page 191

UPDATE QUERY_CLASS

Updates the setting details for a specified Query Patroller query class.

Authorization:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile includes CONFIGURATION privilege with edit authority.

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line interface.

```
>>──UPDATE QUERY_CLASS──query-class-id──USING─────────────────────────────────>

>──┬─────────────────────────────────────┬──┬────────────────────────────┬────>
   └─MAX_QUERIES──max-number-of-queries─┘  └─MAX_COST──max-query-cost─┘

>──┬──────────────────────────────────┬───────────────────────────────────────><
   └─DESCRIPTION──'description'──┘
```

Command parameters:

query-class-id
Specifies the ID for the query class whose setting details are to be updated.

MAX_QUERIES max-number-of-queries
Specifies the maximum number of queries that can be running simultaneously for this query class. When this threshold of running queries is reached, additional queries for this query class will be queued until resources become available. The value must be greater than or equal to 0 and less than or equal to the value of MAX_TOTAL_QUERIES specified in the Query Patroller system settings.

MAX_COST max-query-cost
The maximum cost for a single query that this query class will accept. The value must be greater than 0 and less than or equal to the value of MAX_TOTAL_COST specified in the Query Patroller System settings. This value must be distinct for every query class defined in the system.

DESCRIPTION description
Optionally specifies a text description for the query class. This parameter is nullable.

Examples:

The following command updates query class 8 in the TESTDB database so that it will accept up to 50 queries:

```
qp -d testdb "UPDATE QUERY_CLASS 8 USING MAX_QUERIES 50"
```

Usage Notes:

1. Query classes can be created, removed, or modified while Query Patroller is started. However, these changes will not take effect until Query Patroller has been stopped and restarted.

Related concepts:

- “Query class configuration” on page 116

Related tasks:

- “Changing query classes for Query Patroller”--*Help: Query Patroller*

Related reference:

- “DB2 Query Patroller control tables” on page 263
- “ADD QUERY_CLASS” on page 196
- “REMOVE QUERY_CLASS” on page 231
- “GET QUERY_CLASS” on page 211
- “LIST QUERY_CLASSES” on page 218
- “Query Patroller command line support” on page 191
- “qpstart - Start Query Patroller” on page 228
- “qpstop - Stop Query Patroller” on page 229

UPDATE SUBMISSION_PREFERENCES

Updates submission preferences for a specified submitter.

Authorization:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile includes the USER ADMINISTRATION privilege with edit authority
- Be the owner of the profile associated with the submission preferences being updated.

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line interface.

```
►►──UPDATE SUBMISSION_PREFERENCES──┬──────────────────────┬──────────────────►
                                   └─FOR USER──username─┘

►─USING──┬─DEFAULT───────────────────────────┬────────────────────────────────►◄
         │ ┌─────────────────────────────┐ │
         │ ▼                               │ │
         └─parameter-keyword──value──┴─┘
```

Command parameters:

USER username
: Specifies the username of the submitter whose submission preferences are being updated. If no username is specified, the current login ID is used. The user ID *username* must also exist as a DB2 authorization ID.

DEFAULT
: Specifies that this user uses the default submission preferences which are the submission preferences assigned to PUBLIC.

parameter-keyword value
: The following parameters can be set using this command:

SUBMITTER_PROFILE_NAME profile-name
SUBMITTER_PROFILE_TYPE profile-type
: Specifies the group profile to use to establish submitter settings, such as thresholds, if the submitter has no profile of type user. If multiple group profiles exist for this submitter and no group profile is specified here, queries will be submitted using the group profile with the lowest resource thresholds.

RESULT_EXCEEDED_ACTION
: Specifies the action to take when the query results to be stored in a result table exceed the maximum result rows specified in the submitter's profile.

′A′ Specifies that no results are stored in the result table if the result set is longer than the limit defined in the submitter′s profile. This option is the default.

′T′ Specifies that a truncated set of results is stored in the result table if the result set is longer than the limit defined in the submitter′s profile.

RESULT_ACCESSIBILITY
Specifies whether the result table containing query results will be available to more users than just the submitter.

′O′ Specifies that the result table is accessible by the DB2 IDs listed in the value of the OTHER_GRANTEES parameter. The DB2 IDs listed must have access to the database where the query was submitted.

′S′ Specifies that access to the result table is restricted to the query submitter. This option is the default.

OTHER_GRANTEES grantees
Specifies the DB2 user IDs or group IDs that can access the result table. Up to 1024 alphanumeric characters are accepted. Multiple IDs must be separated with commas.

RESULT_DESTINATION result-destination-id
Specifies whether the submitting application will wait for query results to return, or will be freed up for further activity.

′A′ Specifies that the application that submitted the query will wait for the result set to return while Query Patroller manages the query. When this option is selected, the application that submitted the query may become unresponsive until the result set is returned. This option is the default.

′T′ Specifies that the result set is stored in a DB2 table. When the query is submitted, the application that submitted the query becomes free for further processing.

EMAIL_ADDRESSES email-addresses
Specifies the e-mail address or addresses to receive notification regarding queries submitted by this submitter.

Note: This notification only applies if e-mail notification is enabled in the QP_SYSTEM settings.

The value for this parameter may be up to 1024 characters. Multiple e-mail addresses must be separated by a comma.

Examples:

A Query Patroller user wants to allow his team members to view the results of his queries against the TEAMDB database. To do this, he updates his submission preferences to make his query results accessible to users ″JSMITH″ and ″AWONG″ with the following command:

```
qp -d teamdb "UPDATE SUBMISSION_PREFERENCES FOR USER 'BJONES' USING
RESULT_ACCESSIBILITY 'O' OTHER_GRANTEES 'JSMITH, AWONG'"
```

This command grants users ’JSMITH’ and ’AWONG’ access to result tables that are created by ’BJONES’ after the execution of the command. These users will not have access to result tables created prior to the execution of the UPDATE SUBMISSION_PREFERENCES command.

Related concepts:

- “Query Patroller query submission preferences” on page 147

Related tasks:

- “Setting query submission preferences for another submitter” on page 147
- “Setting your own query submission preferences” on page 299

UPDATE SUBMITTER_PROFILE

Updates a submitter profile in the SUBMITTER_PROFILE table.

Authorization:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile includes the USER ADMINISTRATION privilege with edit authority.

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line interface.

```
►►──UPDATE SUBMITTER_PROFILE──────────────────────────────────────────────────►

►─FOR──┬─USER──'username' ──┬─USING──┬─DEFAULT───────────────────────┬──────────►◄
       └─GROUP──'groupname' ─┘        └─profile-parameter──'value' ───┘
```

Command parameters:

USER username
: Specifies the name of the user to be associated with the profile. The user ID *username* must also exist as a DB2 authorization ID.

GROUP groupname
: Specifies the name of the group to be associated with the profile. The group specified must also exist as a DB2 authorization ID.

DEFAULT
: Specifies that the submitter profile should be updated with the default values for all parameters. For any individual parameter that has a default value, the parameter can be set to the default by entering the parameter with DEFAULT as the value. For example, to set the MIN_COST_TO_MANAGE parameter for USERA to the default value, enter the following:

```
qp -d sample "UPDATE SUBMITTER_PROFILE FOR USER 'USERA' USING
MIN_COST_TO_MANAGE DEFAULT"
```

profile-parameter
: Specifies the parameter values to be assigned to the profile. The following parameters can be set:

PRIORITY priority
: Specifies the priority level assigned to queries submitted under this profile.

- Values must be an integer between 0 and 999 inclusive
- The default value is 500

MAX_QUERIES_ALLOWED max-queries

Specifies the maximum number of queries that a submitter is able to run simultaneously. Queries submitted after this limit is reached are placed in a queued state until other submitted queries complete. When creating a submitter profile for a group, note that the value set for this parameter applies to each user. For example, if this value were set to 10 for Group A, then each user belonging to Group A has the authority to run 10 queries simultaneously.

- A value of ″-1″ indicates that users with this profile can have unlimited queries running simultaneously (up to the value of MAX_TOTAL_QUERIES specified in the QP_SYSTEM table.)
- The default value for this parameter is 100.

MAX_RESULT_ROWS max-number-of-result-rows

Specifies the maximum number of result rows that will be stored in a result table for a single query submitted under this profile. Only queries whose results are to be stored in a result table are subject to this limit.

- A value of ″-1″ indicates that users with this profile can store results with as many rows as necessary to accomodate the complete result set.
- The default value for this parameter is 1,000,000 rows.

MAX_COST_ALLOWED max-query-cost

Specifies the maximum query cost for a submitter under this profile. If the estimated cost of a query submitted under this profile exceeds this value, the query is placed in a held state.

- A value of ″-1″ indicates that users with this profile can run queries of any size (up to the value of MAX_TOTAL_COST specified in the QP_SYSTEM table.)
- The default value for this parameter is 10,000.

MIN_COST_TO_MANAGE min-query-cost

Specifies the minimum cost of a query to be managed by Query Patroller. Queries whose estimated cost is lower than this value will not be managed by Query Patroller. A query that falls below this minimum cost will still be tracked for historical analysis, provided the value of the QUERIES_TO_SAVE parameter in the QP_SYSTEM table is set to A (all queries). The default value is 1000.

ACCOUNT_ID account-id

Specifies an alphanumeric ID to use for account tracking purposes. Up to 128 characters will be accepted. You can use this parameter to sort submitters into logical groupings to track usage costs.

SUSPENDED Y/N

Specifies whether or not a submitter is prohibited from submitting queries. The default value for this parameter is ″N″.

INTERCEPT Y/N

Specifies that Query Patroller is to intercept or manage queries submitted by this submitter. If queries are not intercepted, Query Patroller will not evaluate the cost of the query or track it for historical analysis. The default value for this parameter is ″Y″.

Examples:

The following example suspends the privileges for the user "jsmith". After the execution of this command, ths submitter will not be permitted to submit queries against the SALES database.

```
qp -d sales "UPDATE SUBMITTER_PROFILE FOR USER 'JSMITH' USING SUSPENDED
'Y'"
```

Related tasks:

- "Changing submitter profiles for users and groups"--*Help: Query Patroller*

Related reference:

- "DB2 Query Patroller control tables" on page 263
- "ADD SUBMITTER_PROFILE" on page 201
- "REMOVE SUBMITTER_PROFILE" on page 239
- "GET SUBMITTER_PROFILE" on page 213
- "LIST SUBMITTER_PROFILES" on page 220
- "Query Patroller command line support" on page 191

UPDATE QP_SYSTEM

Updates the Query Patroller system settings for a particular database. This command updates entries in the QP_SYSTEM control table.

Authorization:

You must meet one of the following requirements:
- Have DBADM authority
- Be an operator whose profile includes the CONFIGURATION privilege with edit authority.

Required connection:

None. This command establishes a database connection.

Command syntax:

Note: For information about entering Query Patroller commands using the command line interface and general syntax for Query Patroller commands, see Query Patroller command line interface

```
►►──UPDATE QP_SYSTEM USING──┬─DEFAULT─────────────────────┬──────────────►◄
                            │  ┌─────────────────────────┐ │
                            │  ▼                         │ │
                            └───Setting-keyword──'value'──┴─┘
```

Command parameters:

DEFAULT
Resets the entries in the QP_SYSTEM control table to the default values.

Setting-keyword value
Specifies the parameter to be updated and the value that is to be assigned to that parameter. You can update the following system setting categories and their associated parameters using this command:
- system thresholds
  - MAX_TOTAL_QUERIES
  - MAX_TOTAL_COST
- held query handling
  - RUN_HELD_QUERIES
  - RUN_HELD_DURATION
- query intercept
  - INTERCEPT_APPLICATIONS
  - INCLUDE_APPLICATIONS
  - EXCLUDE_APPLICATIONS
- system maintenance
  - QUERY_PURGE_PERIOD
  - RESULT_PURGE_PERIOD
  - RESULT_TABLE_SPACE
- historical data collection
  - QUERIES_TO_SAVE
  - CAPTURE_REJECTED_QUERY_INFO
  - HISTORY_PURGE_PERIOD
- E-mail notification
  - EMAIL_ENABLE

  - EMAIL_SERVER
  - SEND_DESIGNATED
  - DESIGNATED_EMAIL_ADDRESS

Usage Notes:

1. To view or print the list of system settings, use the GET QP_SYSTEM command.
2. For information about different system setting parameters and their accepted values, see the following setting descriptions:
   - Query Patroller system threshold settings
   - Held query handling settings
   - Query interception settings
   - System maintenance settings
   - Historical data collection settings
   - E-mail notification settings

Related tasks:

- "Enabling Query Patroller to intercept queries" on page 83
- "Enabling collection of historical data" on page 125

Related reference:

- "GET QP_SYSTEM" on page 209
- "Query Patroller system threshold settings" on page 255
- "Held query handling settings" on page 256
- "Query interception settings" on page 257
- "Historical data collection settings" on page 260
- "E-mail notification settings" on page 261
- "Query Patroller command line support" on page 191

Query Patroller system threshold settings

These parameters specify the system-wide thresholds for the number of queries that can run simultaneously, and the maximum workload cost of all simultaneous queries.

Command syntax:

Note: These parameters are part of the UPDATE QP_SYSTEM command.

system threshold settings:

```
|──┬───────────────────────────────────────────┬──────────────────────────►
   └─MAX_TOTAL_QUERIES──max-number-of-queries──┘

►──┬──────────────────────────────────────┬───────────────────────────────|
   └─MAX_TOTAL_COST──max-total-query-cost─┘
```

Parameters Descriptions:

MAX_TOTAL_QUERIES max-number-of-queries
The maximum number of queries that can run simultaneously for the

entire database. When this threshold of running queries is reached, additional queries are placed in a queued state where they wait until sufficient resources become available.

- This parameter may be set to ″-1″ to allow an unlimited number of queries to run simultaneously.
- The default setting for this parameter is –1 (unlimited).
- This value must not be less than the MAX_QUERIES value for any query classes defined in the QUERY_CLASS control table.

.

MAX_TOTAL_COST max-total-query-cost

Represents the maximum workload cost threshold, in timerons. If the cost of an incoming query causes the aggregate cost of all running queries to exceed this value, then the incoming query is placed in a queued state where it waits until sufficient resources become available.

- If you type a floating point number, the value must be in decimal or exponential notation.
- This parameter may be set to ″-1″ to allow an unlimited workload cost.
- The default setting for this parameter is –1 (unlimited) .
- This value must not be less than the MAX_COST value for any query classes defined in the QUERY_CLASS control table.

Related reference:

- "GET QUERY" on page 210
- "LIST QUERIES" on page 215
- "Query Patroller command line support" on page 191

Held query handling settings

These parameters specify whether held queries are to be run on schedule, and how long held queries should be allowed to run.

Command syntax:

Note: These parameters are part of the UPDATE QP_SYSTEM command.

Held query handling settings:

```
├─┬──────────────────────────────┬─┬─────────────────────────────────────────────────────────────┬────────┤
  └─RUN_HELD_QUERIES──┬─'Y'─┬──┘ └─RUN_HELD_DURATION──run-held-queries-duration──time-unit─┘
                      └─'N'─┘
```

Parameter Descriptions:

RUN_HELD_QUERIES

Y Indicates that held queries are to be run at a scheduled time. If no schedule already exists, a default schedule is created.

Note: The Query Patroller command line interface cannot be used to create, update, view, or delete schedules. These tasks must be performed through the Query Patroller Center interface.

N Indicates that held queries are not to be run according to a schedule. This is the default value.

RUN_HELD_DURATION run-held-queries-duration time-unit
Specifies the length of time, in seconds, that held queries are to run. Once this specified period of time has passed, no more held queries will be permitted to run. All remaining held queries will be held until the next scheduled start time for running held queries.

time-unit

MINUTES
Held queries are to run for the specified number of minutes.

HOURS
Held queries are to run for the specified number of hours.

- This number must be an integer greater than or equal to 0.
- The default time is 8 hours.

Related reference:

- "GET QUERY" on page 210
- "LIST QUERIES" on page 215
- "Query Patroller command line support" on page 191

Query interception settings

These parameters specify the applications whose queries are intercepted by Query Patroller.

Command syntax:

Note: These parameters are part of the UPDATE QP_SYSTEM command.

```
>>-+--------------------------------------------------------------------------+-><
   '-INTERCEPT_APPLICATION--+-'A'-------------------------------------------+-'
                            +-'I' --INCLUDE_APPLICATIONS--'list-of-applications' -+
                            '-'E' --EXCLUDE_APPLICATIONS--'list-of-applications' -'
```

Parameter Descriptions:

INTERCEPT_APPLICATION

A Specifies that queries from all applications are intercepted by Query Patroller. This option is the default.

I Specifies that only queries from the applications specified in the INCLUDE_APPLICATIONS parameter are intercepted by Query Patroller.

E Specifies that queries from all applications except those specified in the EXCLUDE_APPLICATIONS parameter are intercepted by Query Patroller.

INCLUDE_APPLICATIONS
Specifies the executable filenames of applications that are intercepted by Query Patroller, separated by commas. Up to 1024 characters are accepted. All other applications will not be intercepted.

Notes:

1. This setting only applies if the INTERCEPT_APPLICATION parameter is set to ″I″.
2. The filename values are case-sensitive.

EXCLUDE_APPLICATIONS

Specifies the executable filenames of applications that are not intercepted by Query Patroller, separated by commas. Up to 1024 characters are accepted. All other applications will be intercepted.

Notes:

1. This setting only applies if the INTERCEPT_APPLICATION parameter is set to ″E″.
2. The filename values are case-sensitive.

Related reference:

- "GET QUERY" on page 210
- "LIST QUERIES" on page 215
- "Query Patroller command line support" on page 191

System maintenance settings

These parameters specify how often queries and result tables are purged from the system.

Command syntax:

Note: These parameters are part of the UPDATE QP_SYSTEM command.

System maintenance settings:

```
|--+-------------------------------------------------------------+------------->
   '-QUERY_PURGE_PERIOD--time-to-keep-queries--+-MINUTES-+-'
                                               +-HOURS---+
                                               '-DAYS----'

>--+-------------------------------------------------------------+------------->
   '-RESULT_PURGE_PERIOD--time-to-keep-results--+-MINUTES-+-'
                                                +-HOURS---+
                                                '-DAYS----'

>--+----------------------------------------------------+--------------------|
   '-RESULT_TABLE_SPACE--result-table-space-name-'
```

Parameters Descriptions:

QUERY_PURGE_PERIOD time-to-keep-queries time-unit

Indicates how long, in seconds, a query will remain in the MANAGE_QUERY_INFO table after the query completes. When this time expires, and when a scheduled purge job is run, the query is removed from the MANAGE_QUERY_INFO control table. This means that the query is no longer available through the Query Patroller Managed Query view, or through the GET QUERY or LIST QUERY commands.

time-unit

MINUTES
Queries older than the specified number of minutes are removed.

HOURS
Queries older than the specified number of hours are removed.

DAYS Queries older than the specified number of days are removed.

- The value for this parameter must be an integer.
- A value of 0 means that all queries are removed from the MANAGE_QUERY_INFO table whenever the scheduled purge job runs.

  Note: The Query Patroller command line interface cannot be used to create, update, view, or delete schedules. These tasks must be performed through the Query Patroller Center interface. If no schedule is created, a default schedule is used.
- If this parameter is not set, or is set to DEFAULT, then the queries will be kept for one week from the time they are run.
- A value of −1 means that the queries are never deleted.
- If the query is in a done state, then its information is still retained in the TRACK_QUERY_INFO table. If the query is in a cancelled, aborted, unknown or rejected state, then it is not stored in TRACK_QUERY_INFO table. Queries that are in a held state are not removed by this command, but by the REMOVE QUERY_INFO command.

RESULT_PURGE_PERIOD time-to-keep-results time-unit
Indicates how long to keep the result table after a query completes. When the next scheduled purge job runs, result tables that are older than this are dropped.

time-unit

MINUTES
Results older than the specified number of minutes are deleted.

HOURS
Results older than the specified number of hours are deleted.

DAYS Results older than the specified number of days are deleted.

- The value for this parameter must be an integer.
- A value of 0 means all result tables are dropped whenever scheduled purge job runs.

  Note: The Query Patroller command line interface cannot be used to create, update, view, or delete schedules. These tasks must be performed through the Query Patroller Center interface. If no schedule is created, a default schedule is used.
- If this parameter is not set, or is set to DEFAULT, then the result tables will be kept for 604800 seconds (one week) from the time they are created.
- A value of −1 means that the result tables are never dropped.

RESULT_TABLE_SPACE result-table-space-name

Specifies the name of the table space that will hold result tables.

- Up to 128 alphanumeric characters are accepted.
- If no value is specified, DB2 will determine which table space to use.

Usage Notes:

1. The same schedule applies to both the purging of queries and the dropping of result sets.
2. When a query is deleted from the MANAGE_QUERY_INFO table, any result tables or any result information for this query is also deleted. Because of this, RESULT_PURGE_PERIOD must be less than or equal to the value for QUERY_PURGE_PERIOD.

Historical data collection settings

These parameters specify what type of queries are tracked for historical data collection, and how long that information is retained in the Query Patroller TRACK_QUERY_INFO control table. Once this information is deleted, it is no longer available in the Query Patroller Center Historical Analysis reports and graphs.

Command syntax:

Note: These parameters are part of the UPDATE QP_SYSTEM command.

Historical data collection settings:

```
├──┬────────────────────────────┬──┬────────────────────────────────────────┬──┬──────────────────────────────────────────┬──────────┤
   └─QUERIES_TO_SAVE──┬─'M'─┬──┘  └─CAPTURE_REJECTED_QUERY_INFO──┬─'Y'─┬──┘  └─HISTORY_PURGE_PERIOD──value──┬─HOURS─┬─┘
                      └─'A'─┘                                    └─'N'─┘                                  └─DAYS──┘
```

Parameters Descriptions:

QUERIES_TO_SAVE

M Specifies that historical analysis will involve only those queries managed by Query Patroller. This option is the default.

A Specifies that historical analysis will involve all queries intercepted by Query Patroller, including queries smaller than a submitter's minimum cost to manage setting.

CAPTURE_REJECTED_QUERY_INFO

Y Specifies that data for rejected queries should be captured. Since historical analysis is only performed on completed queries, the data collected for rejected queries will only be viewable in the Managed Query reports.

N Specifies that data for rejected queries should not be captured.

HISTORY_PURGE_PERIOD value time-unit

Indicates how long to retain completed queries for historical analysis reports and graphs.

time-units

HOURS

Historical data for queries older than the specified number of hours are deleted.

DAYS Historical data for queries older than the specified number of days are deleted.

- The value for this parameter must be an integer.
- A value of 0 means all queries are removed from the TRACK_QUERY_INFO table whenever the scheduled purge job runs.

  Note: The Query Patroller command line interface cannot be used to create, update, view, or delete schedules. These tasks must be performed through the Query Patroller Center interface. If no schedule is created, a default schedule is used.
- If this parameter is not set or set to DEFAULT, this value is set to one year.
- A value of −1 means that the historical data are never purged.

Usage Notes:

1. When the Shistorical information for a query is deleted from the TRACK_QUERY_INFO table, the corresponding entry in the MANAGE_QUERY_INFO table, any result tables, and any result information are also deleted. Because of this, HISTORY _PURGE_PERIOD must be greater than or equal to the value for QUERY_PURGE_PERIOD.

Related reference:

- "GET QUERY" on page 210
- "LIST QUERIES" on page 215
- "Query Patroller command line support" on page 191

E-mail notification settings

These parameters specify whether submitters will be notified when query results have been stored in result tables, or when there has been an error in processing a query that would have generated a result table. The e-mail notification settings also specify whether nofication will be sent to a designate in cases where no submitter e-mail is specified.

Command syntax:

Note: These parameters are part of the UPDATE QP_SYSTEM command.

E-mail notification settings:

```
├──┬────────────────────────┬──┬───────────────────────────────┬──────────────────────►
   └─EMAIL_ENABLE──┬─'Y'─┬──┘  └─EMAIL_SERVER──'email-server'──┘
                   └─'N'─┘

►──┬───────────────────────────────────────────────────────────────────────────────┬──────┤
   └─SEND_DESIGNATED──┬─'Y'──DESIGNATED_EMAIL_ADDRESS──'designated-email-address'─┬─┘
                      └─'N'───────────────────────────────────────────────────────┘
```

Parameter Descriptions:

EMAIL_ENABLE

Y Specifies that an e-mail message is sent to notify a submitter when a query completes and the results have been stored in a result table, or when an error occurs in processing a query that would have generated a result table. If a query completes and the results have not been stored in result table, then no e-mail message is sent.

N Specifies that no e-mail message is sent to notify a submitter when a query completes. This is the default setting.

EMAIL_SERVER
Specifies the host name or IP address of the appropriate STMP e-mail server. Up to 256 characters are accepted.

SEND_DESIGNATED

Y Specifies that e-mail is to be sent to the designated address if the e-mail address field is blank in the submitter's Query Submission Preferences dialog.

N Specifies that no notification messages are sent if there is no submitter e-mail address specified in the submission preferences for the submitter of the query. This is the default setting.

DESIGNATED_EMAIL_ADDRESS designated-email-address
Specifies the e-mail address to send notification to if the results of a completed query are stored in a results table, and there is no e-mail address specified in the query submitter's submission preferences. The value for this parameter can be up to 256 characters.

Related reference:

- "GET QUERY" on page 210
- "LIST QUERIES" on page 215
- "Query Patroller command line support" on page 191

Appendix B. Query Patroller control tables

DB2 Query Patroller control tables

The Query Patroller control tables are created in the target database during the DB2 Query Patroller setup. The control tables contain information DB2 Query Patroller requires to process queries. It is the information in these tables that can be viewed and updated through the Query Patroller Center and the command line interface.

The table schema for the Query Patroller control tables is DB2QP.

There are three types of control tables for Query Patroller:

- Profile tables
- Query information tables
- System settings information tables.

Profile tables

Operator profiles (OPERATOR_PROFILE) control table

Contains a row for every operator profile that is defined.

Table 12. OPERATOR_PROFILE control table

Column Name	Data Type	Nullable	Description
NAME	VARCHAR(128)	No	Username of operator profile. Together with TYPE column, forms the primary key for the table.
TYPE	CHAR(1)	No	Type of profile: • ’U’=User • ’G’=Group Default value is ’U’. Together with NAME column, forms the primary key for the table.
SUSPENDED	CHAR(1)	No	Status of operator privileges • ’Y’= Privileges suspended • ’N’=Privileges not suspended Default value is ’N’.
CONFIGURATION	CHAR(1)	No	CONFIGURATION privilege authority level • ’E’=Edit privilege • ’V’=View privilege • ’N’=No privilege Default value is ’N’.
MONITORING	CHAR(1)	No	MONITORING privilege authority level • ’E’=Edit privilege • ’V’=View privilege • ’N’=No privilege Default value is ’N’.
USERADMIN	CHAR(1)	No	USER ADMINISTRATION privilege authority level • ’E’=Edit privilege • ’V’=View privilege • ’N’=No privilege Default value is ’N’.

Table 12. OPERATOR_PROFILE control table (continued)

Column Name	Data Type	Nullable	Description
MONITORING	CHAR(1)	No	MONITORING privilege authority level • ’E’=Edit privilege • ’V’=View privilege • ’N’=No privilege Default value is ’N’.
HISTDATA	CHAR(1)	No	HISTORICAL DATA privilege authority level • ’E’=Edit privilege • ’V’=View privilege • ’N’=No privilege Default value is ’N’.
RESERVE	BLOB(64K)		

Submitter profiles (SUBMITTER_PROFILE) control table

Contains a row for every submitter profile that is defined.

Table 13. SUBMITTER_PROFILE control table

Column Name	Data Type	Nullable	Description
ID	INTEGER	No	System-generated ID number for submitter profile. Primary key for the table.
NAME	VARCHAR(128)	No	Name of user ID associated with the submitter profile
TYPE	CHAR(1)	No	Type of profile: • ’U’=User • ’G’=Group Default value is ’U’.
SUSPENDED	CHAR(1)	No	Status of submitter privileges: • ’Y’= Privileges suspended • ’N’=Privileges not suspended Default value is ’N’.
INTERCEPT	CHAR(1)	No	Query interception setting: • ’Y’= Intercept queries from submitter • ’N’=Do not intercept queries from submitter. Default value is ’Y’.
PRIORITY	SMALLINT	No	Priority level of queries in the queue. Value must be between 0 and 999 inclusive, with 999 as the highest priority, and 0 as the lowest. Default value is 500.
MAX_QUERIES_ALLOWED	INTEGER	No	Maximum queries allowed to run concurrently from this submitter without queuing up. Value of −1= unlimited. Default value is 100.
MAX_COST_ALLOWED	REAL	No	Maximum cost of query from this submitter without being held. Value of −1= unlimited. Default value is 10 000 000.
MIN_COST_TO_MANAGE	REAL	No	Minimum cost of a query from this submitter in order to be managed. Must be greater than or equal to 0. Default value is 15 000.
MAX_RESULT_ROWS	BIGINT	No	Maximum result rows stored in result table for a single query from this submitter. (Relevant to scheduled queries only.) Value of −1= unlimited. Default value is 1 000 000.
ACCOUNT_ID	VARCHAR(128)		Account ID for submitter profile. Used for chargeback function.
RESERVE	BLOB(64K)		This column is reserved for future use and should not be modified.

Query submission preferences (SUBMISSION_PREFERENCES) control table

Contains a row for every submission preferences file that is defined.

Table 14. SUBMISSION_PREFERENCES control table

Column Name	Data Type	Nullable	Description
USER_NAME	VARCHAR(128)	No	User ID of owner of preferences. Primary key for table.
PROFILE_ID	INTEGER		The submitter profile to be used if user does not have their own profile. Foreign key references SUBMITTER_PROFILE table on delete cascade.
RESULT_DESTINATION	CHAR(1)	No	Location to return query results: • ’A’=return results to application • ’T’=create result table to store results Default value is ’A’.
RESULT_EXCEEDED_ACTION	CHAR(1)	No	Action to take when query results exceed MAX_RESULT_ROWS for submitter[1]: • ’A’=abort • ’T’=return truncated results Default value is ’A’.
RESULT_ACCESSIBILITY	CHAR(1)	No	Accessibility of query result tables: • ’S’=submitter • ’O’= other users Default value is ’S’.
OTHER_GRANTEES	VARCHAR(1024)	No[2]	List of user IDs that can access results of queries from this submitter. Multiple values separated by a comma.
EMAIL_ADDRESSES	VARCHAR(1024)		E-mail addresses for notification regarding queries from this submitter. Multiple values separated by a comma. Only applies if e-mail notification is enabled in QP_SYSTEM table.
RESERVE	BLOB(64K)		

Notes:

1. This column is only applicable to scheduled queries, where results are returned to a result table and not to the submitting application.
2. This column is not null if the value for RESULT_ACCESS_TYPE = ’O’.

Query information tables

Managed query information (MANAGE_QUERY_INFO) control table

Contains a row for every query managed by Query Patroller. Entries in this table are deleted after the period of time specified in the QUERY_PURGE_PERIOD column of the QP_SYSTEM table or when deleted manually.

Table 15. MANAGE_QUERY_INFO control table

Column Name	Data Type	Nullable	Description
ID	INTEGER	No	Query ID. Foreign key references TRACK_QUERY_INFO table on delete cascade. Primary key for table.

Table 15. MANAGE_QUERY_INFO control table (continued)

Column Name	Data Type	Nullable	Description
STATUS	CHAR(1)	No	Status of query[1]: • ’H’=Held • ’Q’=Queued • ’R’=Running
QUERY_CLASS_ID	SMALLINT		Query class that query is assigned to run in[2].
USER_MAX_COST_ALLOWED	REAL		Query submitter’s MAX_COST_ALLOWED from SUBMITTER_PROFILE table.
APPLICATION_HANDLE	BIGINT		ID of the application that the query is submitted from.
MAX_RESULT_ROWS	BIGINT		Query submitter’s MAX_RESULT_ROWS from SUBMITTER_PROFILE table.
TIME_UPDATED	TIMESTAMP	No	Time that this record was last updated. Default value is the current timestamp.
SESSION_AUTH_ID	VARCHAR(128)	No	DB2 session authorization ID.
SESSION_AUTH_TYPE	CHAR(1)	No	ID type of the SESSION_AUTH_ID: • ’U’=USER • ’G’=GROUP • (future: ’R’=ROLE)
MESSAGE_RETURNED	VARCHAR(1024)		DB2 message returned after execution[3].
RESERVE	BLOB(64K)		This column is reserved for future use and should not be modified.

Notes:

1. This is the status of the query prior to completion. Once the query has completed, the final status is recorded in the COMPLETION_STATUS field of the TRACK_QUERY_INFO table.
2. If the query ran under the default query class, the value for QUERY_CLASS_ID is 0.
3. The MESSAGE_RETURNED field is usually empty if the query completes successfully.

Query result information (RESULT_INFO) control table

Contains a row for every query with results stored in a results table. Entries in this table are deleted after the period of time specified in the RESULT_PURGE_PERIOD column of the QP_SYSTEM table or when deleted manually.

Table 16. RESULT_INFO control table

Column Name	Data Type	Nullable	Description
QUERY_ID	INTEGER	No	Query ID. Foreign key references MANAGE_QUERY_INFO table on delete cascade. Primary key for table.
STATUS	CHAR(1)	No	Status of query results: • ’D’=Dropped • ’E’=Exists • ’N’=Does not exist • ’P’=Purged • ’T’=Truncated
OWNER	VARCHAR(128)		User ID that owns results. This is the schema of the result set.
RESULT_TABLE	VARCHAR(128)		Database table containing result set.
RESULT_SELECT	CLOB(2MB)		The SELECT statement issued to retrieve results.

Table 16. RESULT_INFO control table (continued)

Column Name	Data Type	Nullable	Description
RESERVE	BLOB(64K)		This column is reserved for future use and should not be modified.

Historical analysis (QUERY_ANALYSIS) control table

Contains data generated by historical analysis using DB2 Explain data. This table is populated when historical data generation is performed. Records are deleted from this table when the corresponding entry in the TRACK_QUERY_INFO table is deleted.

Note: A single query might have multiple entries in this table depending on the type of Explain data that exists for the query.

Table 17. QUERY_ANALYSIS control table

Column Name	Data Type	Nullable	Description
QUERY_ID	INTEGER	No	Query ID. Foreign key references TRACK_QUERY_INFO table on delete cascade.
STATEMENT_TYPE	CHAR(2)	No	Descriptive label for type of query. • ′S′=Select • ′D′=Delete • ′DC′=Delete where current of cursor • ′I′=Insert • ′U′=Update • ′UC′=Update where current of cursor
OBJECT_TYPE	CHAR(1)		Type of data recorded: • ′C′=Column • ′I′=Index
OPERATOR_TYPE	CHAR(1)		Reserved for future enhancement.
OBJECT_SCHEMA	VARCHAR(128)		The schema for the table index. Applicable when OBJECT_TYPE=′I′.
OBJECT_NAME	VARCHAR(128)		The name of the table column or index.
TABLE_SCHEMA	VARCHAR(128)	No	The schema of the table that the column or index belongs to.
TABLE_NAME	VARCHAR(128)	No	Name of the table that the column or index belongs to.
RESERVE	BLOB(64K)		This column is reserved for future use and should not be modified.

Historical query information (TRACK_QUERY_INFO) control table

Contains a row for every query managed by Query Patroller. For queries intercepted but not managed by Query Patroller, an entry is stored in this table only if the QUERIES_TO_SAVE field in the QP_SYSTEM table has a value of ′A′ (track all queries).

Entries in this table are deleted after the period of time specified in the HISTORY_PURGE_PERIOD column of the QP_SYSTEM table or when deleted manually.

Table 18. TRACK_QUERY_INFO control table

Column Name	Data Type	Nullable	Description
ID	INTEGER	No	ID number for the query. Primary key for the table.

Table 18. TRACK_QUERY_INFO control table (continued)

Column Name	Data Type	Nullable	Description
TYPE	SMALLINT	No	Type of statement: • 0x0001= Select • 0x0002=Modify
COMPLETION_STATUS	CHAR(1)	No	Status that query completed with: • ′A′=Aborted • ′C′=Cancelled • ′D′=Done • ′N′=Not completed • ′U′=Unknown Default value is ′N′.
MANAGED	CHAR(1)	No	Query Patroller managing query:: • ′Y′=yes • ′N′=no Default value is ′Y′.
EXPLAIN_RUN	CHAR(1)	No	Status of EXPLAIN run: • ′F′=Failed • ′N′=Not run • ′S′=Ran successfully Default value is ′N′.
QUERY_PRIORITY	SMALLINT		Priority of submitter that submitted query.
STMT_ATTRIBUTES	INTEGER	No	Bitmap with the following bits: • 0x00000001 - static SQL • 0x00000002 - host variable/parameter marker • 0x00000004 - special register • 0x00000008 - DGTT[1] • 0x00000010 - identity/sequence value • 0x00000020 - result set not allowed • 0x00000040 - session variable • 0x00010000 - queuing not allowed Default value is 0.
NESTING_LEVEL	INTEGER	No	Nesting level of the query. Default value is 0.
ROUTINE_ID	INTEGER		Routine unique identifier.
PARENT_QUERY_ID	INTEGER		ID of the immediate parent query.
PACKAGE_SCHEMA	VARCHAR(128)		
PACKAGE_NAME	VARCHAR(128)		
PACKAGE_VERSION	VARCHAR(128)		
SECTION_ENTRY_NUMBER	INTEGER		
PROFILE_ID	INTEGER	No	Submitter profile used for query.
RESULT_ROWS	BIGINT		The number of rows in the returned result set[2].
EXECUTION_TIME_SECONDS	BIGINT		The seconds portion of the query execution time[3].
EXECUTION_TIME_MILLI_SECONDS	BIGINT		The milliseconds portion of the query execution time[3].
SYSTEM_TIME_SECONDS	BIGINT		The seconds portion of the total system processor time for the query[3]. System time represents the time spent in system calls.
SYSTEM_TIME_MILLI_SECONDS	BIGINT		The milliseconds portion of the total system processor time for the query[3]. System time represents the time spent in system calls.
USER_TIME_SECONDS	BIGINT		The seconds portion of the total user processor time for the query[3]. User time represents the time spent executing database manager code.

Table 18. TRACK_QUERY_INFO control table (continued)

Column Name	Data Type	Nullable	Description
USER_TIME_MILLI_SECONDS	BIGINT		The milliseconds portion of the total user processor time for the query[3]. User time represents the time spent executing database manager code.
ESTIMATED_COST	REAL		The estimated cost of the query in timerons.
TIME_CREATED	TIMESTAMP	No	Time when the query was submitted. Default value is the current timestamp.
TIME_STARTED	TIMESTAMP		Query start time.
TIME_COMPLETED	TIMESTAMP		Query completion time.
TIME_RELEASED	TIMESTAMP		Reserved for future enhancement.
USER_ID	VARCHAR(128)	No	User ID from DB2.
USER_TYPE	CHAR(1)	No	Type of profile: • ’U’=User • ’G’=Group
STMT_AUTH_ID	VARCHAR(128)	No	Statement authorization ID from DB2.
STMT_AUTH_TYPE	CHAR(1)	No	Type of statement authorization: • ’U’=User • ’G’=Group
ACCOUNT_ID	VARCHAR(128)		Chargeback account ID of submitter
APPLICATION	VARCHAR(128)		Name of submitting application.
APPLICATION_HOST	VARCHAR(255)		Host name of the machine that submitted the query.
CLIENT_USER_ID	VARCHAR(255)		The client user ID set by the application using the sqleseti API.
CLIENT_ACCOUNT_ID	VARCHAR(255)		The client account ID set by the application using the sqleseti API.
CLIENT_APPLICATION	VARCHAR(255)		The client application name set by the application using the sqleseti API.
CLIENT_WORKSTATION	VARCHAR(255)		The client workstation name set by the application using the sqleseti API.
REASON_HELD	VARCHAR(255)		Reserved for future enhancement.
REASON_QUEUED	VARCHAR(255)		Reserved for future enhancement.
ENVIRONMENT_VALUES	BLOB(64K)		Compilation environment values.
STATEMENT	BLOB(2MB)		Text of query.[4]
RESERVE	BLOB(64K)		This column is reserved for future use and should not be modified.

Notes:

1. DGTT= Declared Global Temporary Table.
2. For queries whose results are returned to the client application, this information is only available when the statement monitor switch is set to ’ON’. For queries whose results are returned to a result table, this information is available even when the statement monitor switch is set to ’OFF’.
3. Available only when the timestamp and statement monitor switches are set to ’ON’.
4. Since the data type for this field is BLOB, you will not be able to perform a simple SELECT statement to retrieve the values in the table. In order to query this field, you need to use the db2qp.convertToString function to convert the values for retrieval. For example, to select the statement field from this table, you will enter the following:

   ```
   select db2qp.convertToString(statement) from db2qp.track_query_info
   ```

System settings information tables

Query Patroller system (QP_SYSTEM) control table

The QP_SYSTEM table contains a single entry that stores all of the settings for the database.

Table 19. QP_SYSTEM control table

Column Name	Data Type	Nullable	Description
QUERIES_TO_SAVE	CHAR(1)	No	Queries to be tracked for Historical Analysis: • ’M’=Managed queries only • ’A’=All queries Default value is ’M’.
QUERY_PURGE_PERIOD	INTEGER	No	Length of time to keep queries (in seconds). Value of −1 =unlimited. Default value is 604800 (one week).
RESULT_PURGE_PERIOD	INTEGER	No	Length of time to keep query results (in seconds). Value of −1 =unlimited. Default value is 604800 (one week).
HISTORY_PURGE_PERIOD	INTEGER	No	Length of time to keep historical data for queries (in seconds). Value of −1 =unlimited. Default value is 31536000 (365 days).
MAX_TOTAL_QUERIES	INTEGER	No	Maximum number of managed queries that can run concurrently. Value of −1 =unlimited. Default value is −1 (unlimited).
MAX_TOTAL_COST	DOUBLE	No	Maximum total cost of all concurrent managed queries. Value of −1 =unlimited. Default value is −1.
MAX_QUERY_COST	FLOAT	No	Reserved for future enhancement. Default value is −1.
RESULT_TABLE_SPACE	VARCHAR(128)		Table space to store result tables. If no value is specified, the default table space is used.
REJECT_HIGH_COST_QUERY	CHAR(1)	No	Reserved for future enhancement. Default value is ’N’.
RUN_HELD_QUERIES	CHAR(1)	No	Run held queries on schedule: • ’N’=Queries not run • ’Y’=Queries run. • (future: ’C’=Cancel) Default value is ’N’.
RUN_HELD_DURATION	INTEGER	No	Period of time over which held queries are run, in seconds. Value of −1 =unlimited. Default value is 28800.
EMAIL_ENABLE	CHAR(1)	No	Allow e-mail messages to be sent to submitters regarding query completion or errors. • ’N’=E-mail not enabled • ’Y’=E-mail enabled Default value is ’N’.
EMAIL_SERVER	VARCHAR(256)		The host name or IP address of the SMTP e-mail server.

Table 19. QP_SYSTEM control table (continued)

Column Name	Data Type	Nullable	Description
SEND_DESIGNATED	CHAR(1)		E-mail is to be sent to the designated address if no e-mail address is specified in submitter's submission preferences: • 'Y'=Yes • 'N'=No Default value is 'N'.
DESIGNATED_EMAIL_ADDRESS	VARCHAR(256)		E-mail address that will receive notification messages in cases where no e-mail address is specified in submitter's submission preferences, and SEND_DESIGNATED='Y'.
INTERCEPT_STATIC_SQL	CHAR(1)	No	Reserved for future enhancements. Default value is 'Y'.
INTERCEPT_APPLICATION	CHAR(1)	No	Intercept applications: • 'A'=All applications • 'I'=Intercept only applications listed in INCLUDE_APPLICATIONS • 'E'=Intercept all but the applications listed in EXCLUDE_APPLICATIONS Default value is 'A'.
TIME_HIST_GENERATOR_LAST_RUN	TIMESTAMP		The date and time that historical data was last generated..
INCLUDE_APPLICATIONS	VARCHAR(1024)		List of applications to intercept, separated by a comma.
EXCLUDE_APPLICATIONS	VARCHAR(1024)		List of applications not to intercept, separated by a comma.
RESERVE	BLOB(64K)		This column is reserved for future use and should not be modified.

Query classes (QUERY_CLASS) control table

The QUERY_CLASSES table contains a row for every query class that is defined for the database.

Table 20. QUERY_CLASS control table

Column Name	Data Type	Nullable	Description
ID	SMALLINT	No	Query class ID. Primary key for table.
MAX_QUERIES	INTEGER	No	Maximum number of queries that can run concurrently in the query class. Value of −1 =unlimited.
MAX_COST	REAL	No	Maximum cost of queries running in this class, in timerons[1].
DESCRIPTION	VARCHAR(256)		Text description of query class.
RESERVE	BLOB(64K)		This column is reserved for future use and should not be modified.

Notes:

1. No two query classes defined in the same system can have identical values for this field.

Schedule information (SCHEDULE) control table

The SCHEDULE table contains a row for every schedule that is defined for a database.

Table 21. SCHEDULE control table

Column Name	Data Type	Nullable	Description
ID	INTEGER	No	System-generated ID for schedule. Primary key for table.
TYPE	CHAR(1)	No	Type of schedule: • ′H′=Purge historical data • ′Q′=Purge query or result table • ′R′=Run held query
SUSPENDED	CHAR(1)	No	Schedule is suspended: • ′N′=No • ′Y′=Yes Default value is ′N′.
START_DATE	TIMESTAMP	No	Start date for schedule.
END_DATE	TIMESTAMP		End date for schedule.
INTERVAL_UNIT	INTEGER		Interval unit for schedule
INTERVAL	INTEGER		Frequency of schedule
INTERVAL_DETAIL	INTEGER		
NEXT_START_TIME	TIMESTAMP		Next date and time that scheduled job is to run.
RESERVE	BLOB(64K)		This column is reserved for future use and should not be modified.

Related reference:

- “ADD OPERATOR_PROFILE” on page 193
- “ADD QUERY_CLASS” on page 196
- “ADD SUBMITTER_PROFILE” on page 201
- “REMOVE OPERATOR_PROFILE” on page 230
- “REMOVE QUERY_CLASS” on page 231
- “REMOVE SUBMITTER_PROFILE” on page 239
- “UPDATE OPERATOR_PROFILE” on page 243
- “UPDATE QUERY_CLASS” on page 246
- “UPDATE SUBMITTER_PROFILE” on page 251
- “UPDATE QP_SYSTEM” on page 254
- “UPDATE SUBMISSION_PREFERENCES” on page 248
- “REMOVE QUERY_INFO” on page 232
- “REMOVE QUERY_INFO_HISTORY” on page 234
- “ADD SUBMISSION_PREFERENCES” on page 198
- “REMOVE SUBMISSION_PREFERENCES” on page 238

Appendix C. Query Patroller messages

DQP Messages

DQP0001E The database name must be specified in the command.

Explanation: The command syntax requires that the database name be specified.

User Response: See the Query Patroller documentation for a detailed description of the command.

Specify the database name and reissue the command.

DQP0002E Query Patroller server is unable to connect to database *databasename*.

Explanation: Query Patroller was not able to establish a connection to the *databasename* database.

User Response: Ensure that the database name is correct and that the database manager is running.

See the qpdiag.log file for possible causes of this message. If the problem persists, contact IBM Support.

DQP0003E Unable to start the Query Patroller server.

Explanation: Query Patroller cannot be started.

User Response: See the qpdiag.log file for possible causes of this message. If the problem persists, contact IBM Support.

DQP0004E Unable to stop the Query Patroller server.

Explanation: The Query Patroller server cannot be stopped.

User Response: Verify that the Query Patroller server was actually running when you attempted to stop it. See the qpdiag.log file for possible causes of this message. If the problem persists, contact IBM Support.

DQP0005E Query Patroller configuration cannot be found for database *databasename*.

Explanation: The configuration cannot be found for database *databasename*.

User Response: Issue the qpstop command to stop Query Patroller.

DQP0006E Unable to find the Java Properties file *filename*.

Explanation: The required properties file *filename* cannot be found.

User Response: Check the sqllib/msg/<locale>/qp/ directory to determine if the Properties files exist.

If the problem persists, contact IBM Support.

DQP0007E An internal error *error* has occurred.

Explanation: An internal error occurred during processing.

User Response: See the qpdiag.log file for possible causes of this message. If the problem persists, contact IBM Support.

DQP0008E User *username* does not have a submitter profile.

Explanation: Each user is required to have a submitter profile.

User Response: Request that the administrator define a submitter profile for the user.

DQP0009E File *filename* cannot be opened.

Explanation: An error occurred while attempting to open the specified file.

User Response: Ensure that the file exists and that its permissions are correct.

See the qpdiag.log file for possible causes of this message. If the problem persists, contact IBM Support.

DQP0010E User *user-ID* does not have an effective submitter profile.

Explanation: The user *user-ID* does not have an effective Query Patroller submitter profile. This may occur due to one or more of the following reasons:

- The user may not have a submitter profile.
- The user and/or group submitter profiles(s) belonging to the user may have been suspended.

User Response: Request that the database administrator create a submitter profile or have the submitter profile reactivated.

DQP0011E The submission preferences dialog cannot be displayed because neither *property-file1* **nor** *property-file2* **was found.**

Explanation: The properties file cannot be found. As a result, the submission preferences dialog cannot be displayed.

User Response: Verify that the files exist.

DQP0012E The database *databasename* **has not been setup for use with Query Patroller.**

Explanation: To use Query Patroller on a database, Query Patroller needs to be installed on the server that contains the database. During installation, the databases whose queries are to be managed by Query Patroller are chosen and Query Patroller control tables and stored procedures are created in this database. If you need to manage queries of additional databases using Query Patroller, then you must run the qpsetup command for each database.

User Response: Request that an administrator install Query Patroller on the server containing the database or, if this has already been done, request that the administrator run the qpsetup command to setup Query Patroller for the database.

If the qpsetup command has been run for the database and the problem persists, check that the file db2qp exists in INSTPATH/function on UNIX or the file db2qp.dll exists on INSTPATH\function on Windows, where INSTPATH is the DB2 instance directory.

DQP0019E SYSADM authority is required to issue *command-name* **command.**

Explanation: The command requires SYSADM authority.

User Response: Request that the system administrator run the command.

DQP0020E Function not supported. Reason code = *reason-code***.**

Explanation: The statement cannot be processed because it violates a restriction as indicated by the following reason code:

1 The connection codepage cannot be set because the functionality is not yet available.

2 The restriction is unknown.

User Response: The action corresponding to the reason code is:

1 Update the Query Patroller client to a newer version.

2 If the problem persists, contact IBM Support.

DQP0021E The DB2 server and the Query Patroller server are at a different product level.

Explanation: The DB2 server and the Query Patroller server must be on the same product level.

User Response: Ensure that your DB2 server and Query Patroller server are at the same product level. If you have applied a FixPak on the DB2 server before installing Query Patroller server, ensure that you reapply the FixPak.

See the Query Patroller documentation for a description of the product prerequisites.

DQP0406E Unable to start the Query Controller as the maximum number of *maxnumber* **Query Controllers is already running.**

Explanation: There has been an attempt to start more than the maximum number of allowed Query Controllers.

User Response: Use one of the Query Controllers that is currently running.

DQP0408E DBADM authority is required to run the command *commandname***.**

Explanation: The command *command-name* requires DBADM authority.

User Response: Request that the database administrator grant DBADM authority and reissue the command.

DQP0409E The Query Patroller system settings do not exist.

Explanation: Query Patroller cannot be run when the system settings do not exist.

User Response: Issue the qpsetup command to create the Query Patroller system settings.

See the Query Patroller documentation for a detailed description of the qpsetup command.

DQP0410E The submission preferences for the PUBLIC group do not exist.

Explanation: Query Patroller cannot be run when the submission preferences for the PUBLIC group do not exist.

User Response: Recreate the submission preferences for the PUBLIC group and issue the qpstart command to start Query Patroller.

DQP0412I The database connection has been reestablished.

Explanation: The database connection has been reestablished.

User Response: No action required.

DQP0413E The database connection has been lost. Query Patroller is terminated.

Explanation: The database connection has been lost. As a result, Query Patroller has terminated.

User Response: Reestablish the database connection and issue the qpstart command to restart Query Patroller.

DQP0414E Query Patroller was not stopped because there are active queries.

Explanation: Query Patroller was not stopped because there are queued or active queries.

User Response: Reissue the command using the FORCE option. If the active queries are forced, they will be in an inconsistent state until Query Patroller is restarted and the query recovery is complete.

DQP0415I The estimated cost of the query exceeds the maximum allowed for the user's submitter profile.

Explanation: The query will be held.

User Response: Request that the database administrator run or cancel the query.

DQP0416E The estimated cost of the query exceeds the maximum query cost for the system.

Explanation: The query will be held.

User Response: Request that the database administrator run or cancel the query.

DQP0417E The number of queries running has reached the maximum allowed for the system.

Explanation: The query will be placed in the queue.

User Response: The query will run automatically when the number of queries running drops below the maximum allowed for the system.

DQP0418E The number of queries running for user *username* **is the maximum number allowed for the submitter profile.**

Explanation: The query will be placed in the queue.

User Response: The query will run automatically when the number of queries running drops below the maximum allowed for the system.

DQP0419E The number of queries running under query class *queryclassnumber* **is the maximum allowed for the query class.**

Explanation: The query will be placed in the queue.

User Response: The query will run automatically when the number of queries running drops below the maximum allowed for the query class.

DQP0420I The sum of the estimated cost and the current system workload exceeds the maximum system workload allowed.

Explanation: The sum of the estimated cost of the query submitted and the current system workload is greater than the maximum system workload allowed.

User Response: Increase the maximum system workload allowed, if desired.

DQP0421I The estimated cost of the query exceeds the maximum system workload allowed.

Explanation: The query is being held because its estimated cost exceeds the maximum system workload allowed.

User Response: Request that the administrator release the query so that it can run, or increase the maximum system workload allowed.

DQP0422E DB2 Query Patroller license cannot be found.

Explanation: A valid license key for DB2 Query Patroller cannot be found or it has expired.

User Response: Install a license key for the fully entitled version of the product. You can obtain a license key for the product by contacting your IBM representative or authorized dealer.

DQP0423E Query Patroller is already started.

Explanation: Unable to start Query Patroller as it is already running.

User Response: To restart Query Patroller, issue the qpstop command and then issue qpstart.

DQP0424E No start Query Patroller command was issued.

Explanation: The start Query Patroller command has not been processed. It must be processed before a stop Query Patroller command is issued.

User Response: Issue a start Query Patroller command, qpstart, and resubmit the current command.

DQP1001E The date or time specified is not valid.

Explanation: The date or time specified is in an incorrect format.

User Response: Enter the correct format for the date and time.

DQP1002E The start date and time specified must occur before the end date and time.

Explanation: The specified start date and time must precede the end date and time.

User Response: Specify a start date and time that occur before the end date and time.

DQP1003E The operation was aborted because it would violate the consistency of the date. SQLSTATE = ***sqlstate*****.**

Explanation: The Query Patroller control tables have triggers to protect the consistency of the data contained in them. Such a trigger has been activated through an action that has been attempted. The operation was aborted due to a configuration consistency constraint violation.

User Response: Examine the qpschema.sql file to find the trigger that signaled the SQLSTATE. Based on this examination, make the necessary adjustments to prevent the trigger from aborting the action again.

DQP1004E A memory allocation error occurred.

Explanation: There was not enough memory to continue processing.

User Response: Ensure that your system has sufficient memory.

See the Query Patroller documentation for the system memory requirements. If the problem persists, contact IBM Support.

DQP1005E The time range must not be greater than two years.

Explanation: The time range specified is greater than two years.

User Response: Specify a time range that is within the next two years and reissue the command.

DQP1006E SQL statement ***sqlstmt*** **failed with SQLCODE** ***SQLCODE*****.**

Explanation: The SQL statement failed.

User Response: Check the qpdiag.log file for possible causes of this message. If the problem persists, contact IBM Support.

DQP1008E SQL data type ***datatype*** **is not supported.**

Explanation: The SQL data type is not supported.

User Response: See the Query Patroller documentation for information about supported SQL data types.

DQP1009E Result destination is too large to view.

Explanation: The Query Patroller Center and Query Patroller Command Line Processor cannot return your results because the results exceed the maximum size limit of the control table.

User Response: Resubmit the query and ensure that the results are returned to your application rather than stored in a result table.

If your query generated a result table because it was run in the background, resubmit the query with your submission preferences set to wait until results are returned.

If your query generated a result table because it was held and then run, ask an administrator to raise the query cost of your submitter profile so that you can resubmit the query without having it held.

Alternatively, you can select directly from the result table, but the names of the columns in the result table do not match those of your query and the result table contains an additional column, A0000.

DQP1010E File ***filename*** **cannot be accessed.**

Explanation: The specified file cannot be accessed.

User Response: Ensure that the file exists and that the file permissions are correct.

DQP1011E User ***username*** **is not authorized to issue the command.**

Explanation: The command cannot run because the user does not possess the necessary authorization level.

User Response: Check the Query Patroller documentation for the required authorization. Request that the database administrator grant the user the required authority and reissue the command.

DQP1012E File ***filename*** **does not exist.**

Explanation: The specified file does not exist.

User Response: Ensure that the file exists. If the problem persists, contact IBM Support.

DQP1023E An error occurred while saving to the file *filename*.

Explanation: Possible causes of this message include:

- The file does not have the correct file permissions.
- The file name does not conform to the file system convention.

User Response: Possible solutions include:

- Verify the file permissions.
- Specify the correct file name.

DQP1024W Creation, change, or removal of a query class will not take effect until the Query Patroller server is restarted.

Explanation: You have just created, changed, or deleted a query class. This will change the behavior of how queries are processed by the Query Patroller server. This change in behavior will not occur until the next time Query Patroller is started.

User Response: Request that an administrator restart the Query Patroller server by issuing the qpstop command followed by the qpstart command on the computer where the Query Patroller server is running.

DQP1025W The update will not take effect until the Query Patroller server is restarted.

Explanation: The settings will not take effect until the Query Patroller server is restarted.

User Response: Restart the Query Patroller server to have the changes take effect.

DQP1026W The update was successful. However, communication with the Query Patroller server has failed.

Explanation: The update was successful. However, the Query Patroller server cannot be contacted to reflect the changes.

User Response: Restart the Query Patroller server to see the reflected changes.

DQP1028E The network operation did not complete within a reasonable amount of time. The command cannot be completed.

Explanation: The network operation did not complete within a reasonable amount of time. The command cannot be completed.

User Response: Reissue the command when the network is not as busy. If problem persists, contact the system administrator to analyze if there is a network error.

DQP2020E The path *path* is invalid.

Explanation: The specified path is invalid.

User Response: Verify the path and reissue the command.

DQP2101I Operator profile for user *username* was added successfully.

Explanation: An operator profile was created for the user.

User Response: No action required.

DQP2102I Operator profile for group *groupname* was added successfully.

Explanation: An operator profile was created for the group.

User Response: No action required.

DQP2103I Operator profile for user *username* was updated successfully.

Explanation: An existing user operator profile was updated.

User Response: No action required.

DQP2104I Operator profile for group *groupname* was updated successfully.

Explanation: An existing group operator profile was updated.

User Response: No action required.

DQP2105I Operator profile for user *username* was removed successfully.

Explanation: A user operator profile was removed.

User Response: No action required.

DQP2106I Operator profile for group *groupname* was removed successfully.

Explanation: A group operator profile was removed.

User Response: No action required.

DQP2107E Operator profile for user *username* does not exist.

Explanation: The user operator profile is not defined.

User Response: Ensure that the user name specified is correct and reissue the command.

DQP2108E Operator profile for group *groupname* does not exist.

Explanation: The group operator profile is not defined.

User Response: Ensure that the group name specified is correct and reissue the command.

DQP2109E Operator profile for user *username* already exists.

Explanation: The user operator profile already exists.

User Response: Ensure that the user name specified is unique and reissue the command.

DQP2110E Operator profile for group *groupname* already exists.

Explanation: The group operator profile already exists.

User Response: Ensure that the group name specified is unique and reissue the command.

DQP2111I There are no operator profiles defined.

Explanation: There are no operator profiles defined to display.

User Response: No action required.

DQP2112I Query class *queryclassID* was added successfully.

Explanation: A new query class was defined.

User Response: No action required.

DQP2113I Query class *queryclassID* was updated successfully.

Explanation: An existing query class was updated.

User Response: No action required.

DQP2114I Query class *queryclassID* was removed successfully.

Explanation: A query class was removed.

User Response: No action required.

DQP2115E Query class *queryclassID* does not exist.

Explanation: The query class does not exist.

User Response: Ensure that the query class ID specified is correct and reissue the command.

DQP2116E The query class with the same maximum query cost value already exists.

Explanation: The query class already exists.

User Response: Specify a query class that does not already exist and reissue the command.

DQP2117I There are no query classes defined.

Explanation: There are no existing query classes. There is no information to list.

User Response: No action required.

DQP2118I Submitter profile for user *username* was added successfully.

Explanation: A new user submitter profile was created for the user.

User Response: No action required.

DQP2119I Submitter profile for group *groupname* was added successfully.

Explanation: A new group submitter profile was created for the group.

User Response: No action required.

DQP2120I Submitter profile for user *username* was updated successfully.

Explanation: An existing user submitter profile was updated.

User Response: No action required.

DQP2121I Submitter profile for group *groupname* was updated successfully.

Explanation: An existing group submitter profile was updated.

User Response: No action required.

DQP2122I Submitter profile for user *username* was removed successfully.

Explanation: A user submitter profile was removed.

User Response: No action required.

DQP2123I Submitter profile for group *groupname* was removed successfully.

Explanation: A group submitter profile was removed.

User Response: No action required.

DQP2124E Submitter profile for user *username* does not exist.

Explanation: The user submitter profile does not exist.

User Response: Specify an existing user name and reissue the command.

DQP2125E Submitter profile for group *groupname* does not exist.

Explanation: The group submitter profile does not exist.

User Response: Specify an existing group name and reissue the command.

DQP2126E Submitter profile for user *username* already exists.

Explanation: The user submitter profile exists.

User Response: Specify a unique user name and reissue the command.

DQP2127E Submitter profile for group *groupname* already exists.

Explanation: The group submitter profile exists.

User Response: Specify a unique group name and reissue the command.

DQP2128I No submitter profiles exist.

Explanation: There are no submitter profiles. There is no information to list.

User Response: No action required.

DQP2129I The settings for the Query Patroller system were updated successfully.

Explanation: The settings for the Query Patroller system were updated successfully.

User Response: No action required.

DQP2130E Query *queryID* does not exist.

Explanation: The query ID specified does not exist.

User Response: Specify an existing query ID and reissue the command.

DQP2131I No queries matching the input criteria exist.

Explanation: There is no information to list as there are no queries that match the input criteria.

User Response: Modify the input criteria, if necessary.

DQP2132I New submission preferences for user *username* were added.

Explanation: New user submission preferences were created for the user.

User Response: No action required.

DQP2133I Submission preferences for user *username* were updated successfully.

Explanation: Existing submission preferences were updated.

User Response: No action required.

DQP2134I Submission preferences for user *username* were removed. This user will use the default submission preferences.

Explanation: An individual's submission preferences were removed.

User Response: No action required.

DQP2135E Submission preferences for user *username* already exist.

Explanation: The user submission preferences already exist.

User Response: Specify a unique user name and reissue the command.

DQP2136I All result tables were removed successfully.

Explanation: All of the result tables were dropped.

User Response: No action required.

DQP2137I There are no existing result tables to drop.

Explanation: There are no result tables to drop.

User Response: No action required.

DQP2138I The result tables belonging to user *username* were removed successfully.

Explanation: The result tables were dropped.

User Response: No action required.

DQP2139I There are no result tables belonging to user *username*. Nothing was removed.

Explanation: There are no result tables to drop.

User Response: No action required.

DQP2140I Result tables for query *queryID* were removed successfully.

Explanation: The result tables were dropped.

User Response: No action required.

DQP2141I Result table for query *queryID* does not exist. Nothing was removed.

Explanation: The result table for the specified query does not exist.

User Response: No action required.

DQP2142E Query *queryID* is not in the correct state.

Explanation: The query is not in the correct state to execute the command.

User Response: Verify the state of the query and reissue the command.

DQP2143E Invalid time unit *timeunit* specified for parameter *parametername*.

Explanation: The time unit specified for the parameter is invalid.

User Response: See the Query Patroller documentation for a detailed description of the command.

Specify the correct time unit and reissue the command.

DQP2144E The parameter *parametername* specified is not correct.

Explanation: The parameter specified is not correct.

User Response: Check the Query Patroller documentation for a detailed description of the command. Correct the syntax and reissue the command.

DQP2145E The parameter *parametername* must be specified.

Explanation: The command syntax specified is not correct.

User Response: Check the Query Patroller documentation for a detailed description of the command. Correct the syntax and reissue the command.

DQP2146E An incorrect value *value* was specified for the parameter *parametername*.

Explanation: An incorrect value was specified for the parameter.

User Response: Check the Query Patroller documentation for a detailed description of the command. Correct the parameter value and reissue the command.

DQP2147E Missing value for parameter *parametername*.

Explanation: A required parameter value is missing from the command.

User Response: Check the Query Patroller documentation for a detailed description of the command. Include the necessary parameter value and reissue the command.

DQP2148E Unexpected keyword *keyword*, expected keyword might be *keyword*.

Explanation: A keyword that does not apply to the command was encountered.

User Response: Check the Query Patroller documentation for a detailed description of the command. Correct the syntax and reissue the command.

DQP2149E Unexpected end of command, expected value might include *value*.

Explanation: A syntax error in the command was detected.

The command cannot be processed.

User Response: See the Query Patroller documentation for a detailed description of the command.

Correct the syntax and reissue the command.

DQP2150E The parameter *parametername* cannot be specified more than once.

Explanation: The parameter *parametername* is specified more than once.

User Response: Check the Query Patroller documentation for a detailed description of the command. Correct the syntax and reissue the command.

DQP2151E There is no default value for parameter *parametername*.

Explanation: The parameter does not have a default value.

User Response: Check the Query Patroller documentation for a detailed description of the command. Specify a valid parameter value and reissue the command.

DQP2152E The value of parameter *parametername* cannot be null.

Explanation: A value must be specified for parameter *parametername*.

User Response: Check the Query Patroller documentation for a detailed description of the command. Provide a value for the parameter and reissue the command.

DQP2153E The value of parameter *parametername* must be surrounded by single quotation marks.

Explanation: The parameter must be surrounded by single quotation marks.

User Response: Check the Query Patroller documentation for a description of how to run a command. Surround the parameter value with single quotation marks and reissue the command.

DQP2154E The query is no longer running and the query status is unknown.

Explanation: The query is not running, but its final status cannot be determined. It may have completed successfully or failed. This is an abnormal situation which may have been caused by the following scenarios:

- The Query Patroller server crashed, or was shut down with the FORCE option, terminated because of a power outage, or its connection to DB2 was forced off while this query was running.
- The DB2 server crashed, was shut down with the force option, or terminated because of a power outage, while this query was queued or running.
- The Query Patroller server did not respond within an expected time frame when DB2 attempted to report query completion status.

User Response: Examine the client application that submitted the query for the query status and results. Resubmit the query if necessary.

DQP2155E User *username* is suspended.

Explanation: The user is not authorized to submit queries, nor perform any Query Patroller commands.

User Response: Request that the database administrator reactivate the user submitter profile.

DQP2156E Communication with the Query Patroller server has timed out.

Explanation: The DB2 server cannot connect to or communicate with the Query Patroller server within the expected time frame.

User Response: The Query Patroller server may be temporarily overloaded, leading to a slower than normal response time. Check the system workload and ensure that the Query Patroller system is tuned properly for optimal performance. Resubmit the query if necessary.

If the problem persists, contact IBM Service.

DQP2157E Submission preferences for user *username* do not exist.

Explanation: There are no existing submission preferences for the user.

User Response: Create submission preferences for the user or specify a user name that has existing submission preferences.

DQP2158E The submission preferences belonging to user *username* cannot be copied onto itself.

Explanation: The submission preferences cannot be copied onto itself.

User Response: Specify a different username and reissue the command.

DQP2159I The query information for query *queryID* has been removed.

Explanation: There is no existing information related to the *queryID* query.

User Response: No action required.

DQP2160I The historical information for query *queryID* has been removed.

Explanation: There is no existing historical information related to the *queryID* query.

User Response: No action required.

DQP2161I All query information has been removed.

Explanation: The query information has been removed.

User Response: No action required.

DQP2162I All historical query information has been removed.

Explanation: The historical query information has been removed.

User Response: No action required.

DQP2163I Query *queryID* has been canceled.

Explanation: The query has been canceled.

User Response: No action required.

DQP2164I The query information for *number* queries older than *timeunit* has been removed.

Explanation: The information related to queries older than *timeunit* was removed.

User Response: No action required.

DQP2165I The historical query information for *number* queries older than *timeunit* has been removed.

Explanation: The historical information related to queries older than *timeunit* was removed.

User Response: No action required.

DQP2166E The query *queryID* cannot be canceled. The query has already been completed, canceled, or aborted.

Explanation: The query *queryID* cannot be canceled because the query has already been completed, canceled, or aborted.

User Response: No action required.

DQP2167E Unexpected keyword *keyword*, end of command is expected.

Explanation: A keyword that does not apply to the command was encountered.

User Response: Check the Query Patroller documentation for a detailed description of the command. Correct the syntax and reissue the command.

DQP2168I The query information for all the specified queries has been removed.

Explanation: The query information has been removed.

User Response: No action required.

DQP2169I The historical information for all the specified queries has been removed.

Explanation: The historical information for the queries has been removed.

User Response: No action required.

DQP2170I *Number* result tables older than *timeunit* have been dropped.

Explanation: The result tables have been removed because they were older than *timeunit*.

User Response: No action required.

DQP2171I Query Patroller has processed *number* queries and historical data was generated for *number* queries successfully

Explanation: Historical data was generated for the number of queries identified.

User Response: No action required.

DQP2172I The query class update may take a long time. Currently queued and newly submitted queries will remain queued until the process is completed.

Explanation: The query class update may take a long time to complete.

User Response: No action required.

DQP2173E A query class update is in progress. Another update cannot be performed at this time.

Explanation: A query class update cannot be performed while another query class update is in progress.

User Response: Reissue the command after the current update is complete.

DQP2174I The maximum number of query classes has been exceeded.

Explanation: The maximum number of query classes has been reached. New query classes cannot be created.

User Response: Decrease the number of existing query classes.

DQP2175I A query class with the same maximum query cost value already exists or the maximum number of query classes has been exceeded.

Explanation: The maximum number of query classes has been reached. New query classes cannot be created.

Query classes must have a unique query cost value.

User Response: Decrease the number of existing query classes and ensure that all query classes have a unique query cost value.

DQP2176E The length of the keyword *keyword* exceeds the maximum length allowed.

Explanation: The length of the specified keyword has exceeded the maximum length allowed.

User Response: Ensure that the keyword is valid and reissue the command.

DQP2178E An invalid user name or password was entered.

Explanation: An invalid user name or password was entered.

User Response: Reissue the command with the correct user name and password.

DQP2179I Historical query information does not exist.

Explanation: Unable to remove historical information for some queries as the information does not exist.

User Response: No action required.

DQP2180W Query *queryid* is now running in the background.

Explanation: The specified query is now running in the background.

User Response: No action required.

DQP2181E The query was aborted because the number of rows in the result set is larger than the number of rows that can be accommodated in the result table.

Explanation: Query Patroller attempted to create a result table for this query because it was either run in the background or held and then released from held state. The submission preferences of this submitter stated that in the event the result set is longer than the maximum allowed, no result should be returned.

Query Patroller aborted the query when it detected that the number of rows returned by the query would exceed the maximum size of a result table specified in the submitter profile under which this query was submitted.

User Response: Modify the query to reduce the number of rows it returns or request that an administrator or operator increase the number of rows allowed for a result table. If the query was run in the background, try waiting for the result instead of releasing the application.

DQP2182I Held query *queryid* has been run.

Explanation: The specified query is running in the background.

User Response: No action required.

DQP2183E Query *queryid* cannot be run in the background.

Explanation: The specified query cannot be run in the background.

User Response: See the Query Patroller documentation for the class of queries that can be run in the background.

DQP2184E Result tables older than *time-unit* do not exist.

Explanation: The result tables do not exist. Nothing was removed.

User Response: No action required.

DQP2185I Query information older than *time-value* does not exist.

Explanation: The query information cannot be removed as it does not exist.

User Response: No action required.

DQP2186I Historical query information older than *time-value* does not exist.

Explanation: The historical query information cannot be removed as it does not exist.

User Response: No action required.

DQP2187W Submission preferences for user *username* were added successfully. However, communication with the Query Patroller server has failed. The changes have not been reflected on the Query Patroller server.

Explanation: The new submission preferences for the user were added successfully. However, the Query Patroller server cannot be contacted to reflect the changes.

User Response: Restart the Query Patroller server to see the reflected changes.

DQP2188W Submission preferences for user *username* were updated successfully. However, communication with the Query Patroller server has failed. The changes have not been reflected on the Query Patroller server.

Explanation: The submission preferences for the user were updated successfully. However, the Query Patroller server cannot be contacted to reflect the changes.

User Response: Restart the Query Patroller server to see the reflected changes.

DQP2189W Submitter profile for user *username* was added successfully. However, communication with the Query Patroller server has failed. The changes have not been reflected on the Query Patroller server.

Explanation: The submitter profile for the user was added successfully. However, the Query Patroller server cannot be contacted to reflect the changes.

User Response: Restart the Query Patroller server to see the reflected changes.

DQP2190W Submitter profile for user *username* was updated successfully. However, communication with the Query Patroller server has failed. The changes have not been reflected on the Query Patroller server.

Explanation: The submitter profile for the user was updated successfully. However, the Query Patroller server cannot be contacted to reflect the changes.

User Response: Restart the Query Patroller server to see the reflected changes.

DQP2191W Submitter profile for group *groupname* was added successfully. However, communication with the Query Patroller server has failed. The changes have not been reflected on the Query Patroller server.

Explanation: The submitter profile for the group was added successfully. However, the Query Patroller server cannot be contacted to reflect the changes.

User Response: Restart the Query Patroller server to see the reflected changes.

DQP2192W Submitter profile for group *groupname* was updated successfully. However, communication with the Query Patroller server has failed. The changes have not been reflected on the Query Patroller server.

Explanation: The new submitter profile for the group was updated successfully. However, the Query Patroller server cannot be contacted to reflect the changes.

User Response: Restart the Query Patroller server to see the reflected changes.

DQP2193W The settings for the Query Patroller system were updated successfully. However, communication with the Query Patroller server has failed. The changes have not been reflected on the Query Patroller server.

Explanation: The settings for the Query Patroller system were updated successfully. However, the Query Patroller server cannot be contacted to reflect the changes.

User Response: Restart the Query Patroller server to see the reflected changes.

DQP2194W Submission preferences for user *username* were removed successfully. However, communication with the Query Patroller server has failed. The changes have not been reflected on the Query Patroller server.

Explanation: The submission preferences for the user were removed successfully. However, the Query Patroller server cannot be contacted to reflect the changes.

User Response: Restart the Query Patroller server to see the reflected changes.

DQP2195W Submitter profile for user *username* was removed successfully. However, communication with the Query Patroller server has failed. The changes have not been reflected on the Query Patroller server.

Explanation: The submitter profile for the user was removed successfully. However, the Query Patroller server cannot be contacted to reflect the changes.

User Response: Restart the Query Patroller server to see the reflected changes.

DQP2196W Submitter profile for group *groupname* was removed successfully. However, communication with the Query Patroller server has failed. The changes have not been reflected on the Query Patroller server.

Explanation: The submitter profile for the group was removed successfully. However, the Query Patroller server cannot be contacted to reflect the changes.

User Response: Restart the Query Patroller server to see the reflected changes.

DQP2197I Query *queryID* cannot be removed.

Explanation: The query information cannot be removed.

User Response: No action required.

DQP2198E Unable to display result sets for query *queryID*. The result sets do not exist.

Explanation: The result sets for the query do not exist. This error might occur when the query is a non-SELECT statement, the result sets were manually dropped, or when a client application is specified as the result destination.

User Response: No action required.

DQP2199W The purge period specified for managed queries is less than the purge period specified for the result tables. The associated result table will be deleted when the managed query is deleted.

Explanation: The purge period specified for managed queries is less than the purge period specified for the result tables. When a managed query is deleted, the associated result table is also deleted; therefore the purge period for the result tables is ignored.

User Response: Set the purge period for the managed queries to be greater than or equal to the purge period for the result tables.

DQP2200W The purge period specified for historical queries is less than the purge period specified for the managed queries. The associated managed query will be deleted when the historical query is deleted.

Explanation: When a historical query is deleted the associated managed query is also deleted, therefore, the purge period for the managed queries is ignored.

User Response: Set the purge period for the historical queries to be greater than or equal to the purge period for the managed queries.

DQP22202I The historical analysis data generator was stopped by a user. Query Patroller processed *number* queries and historical data was generated for *number* queries successfully before it was stopped.

Explanation: The GENERATE HISTORICAL_DATA STOP command was issued in a separate process. Historical data was generated for the number of queries identified.

User Response: No action required.

DQP2203I An attempt to stop the generation of historical data is in progress.

User Response: No action required.

DQP2204I Query Patroller is not generating historical data at this time.

Explanation: Query Patroller is not able to issue the GENERATE HISTORICAL_DATA STOP command as historical data is not generating data at this time.

User Response: No action required.

DQP2209W Releasing this query will cause the system to run a workload that exceeds the maximum system workload cost.

Explanation: The query that is being released has a cost that exceeds the system workload cost. Once it has been release from held state, Query Patroller will queue it until there are no other queries queued.

This does not guarantee that there will be no other queries running at the same time as this query.

User Response: No action required.

DQP2211E Unable to run Historical Analysis Data Generator as it is currently running.

Explanation: Only one Historical Analysis Data Generator can run on a database at one time.

User Response: Wait until the currently running Historical Analysis Data Generator completes before trying to run it again. Or stop the currently running Historical Analysis Data Generator by issuing the GENERATE HISTORICAL_DATA STOP Query Patroller command.

DQP2505I The database partition group *dbpartitiongroupname* does not exist. The qpsetup command will attempt to create a new one.

Explanation: The specified database partition group does not exist.

User Response: The qpsetup command will attempt to create a new database partition group.

DQP2506E One or more Query Patroller control tables already exist.

Explanation: One or more Query Patroller control tables were found. The qpsetup command cannot proceed.

User Response: Verify if the control tables are valid and complete. If you wish to replace the control tables with new control tables, reissue the qpsetup command using the REPLACE option.

DQP2507E The table space *tablespacename* does not exist. The qpsetup command cannot proceed.

Explanation: The specified table space name does not exist.

User Response: Ensure that the table space name is correct and reissue the command.

DQP2508E The schema *schemaname* already exists. The command *commandname* cannot proceed.

Explanation: The specified schema name already exists.

User Response: Specify a unique schema name and reissue the command.

DQP2516E Failed to bind package *package-name*, reason *reason*.

Explanation: An attempt to bind a package failed.

User Response: If the failure occurred due to an SQLCODE, see the Message Reference documentation for more information. If the problem persists, contact IBM Support.

DQP2518E The Query Patroller control tables do not exist.

Explanation: The Query Patroller control tables cannot be found in the database specified.

User Response: Issue the qpsetup command to create the Query Patroller control tables.

DQP2519E Failed to update the DB2QP.QP_SYSTEM table with the result table space information. Reason = *reason*.

Explanation: The qpsetup command attempted to update the QP_SYSTEM table with the result table space, however the update failed.

User Response: Analyze the reason, correct the problem, and reissue the command.

If the problem persists, contact IBM Support.

DQP2520W The schema *schema-name* already exists. The qpsetup command will attempt to create Query Patroller control tables using this schema.

Explanation: The schema already exists in the specified database. The qpsetup command will attempt to create Query Patroller control tables in this schema. DB2 tables existing in this schema will not be replaced unless the REPLACE option is specified in the qpsetup command.

User Response: If any existing tables have the same name as the Query Patroller control tables, move them to a different schema.

DQP2521E The table space container *container-path* is already in use.

Explanation: The specified table space container path is already in use by another application.

User Response: Verify the path and reissue the command.

DQP2522E The qpsetup command failed. All database objects created by this command were cleaned up.

Explanation: The qpsetup command failed. All database objects, such as table spaces, tables, functions, and procedures, were cleaned up.

User Response: Refer to the qpsetup.log file for possible causes of this message. If the problem persists, contact IBM Support.

DQP2523I The qpsetup command completed successfully.

Explanation: The qpsetup command completed successfully.

User Response: No action required.

DQP2526I Package *package-name* was bound successfully.

Explanation: Query Patroller bound the package successfully.

User Response: No action required.

DQP2604I The cleanup of the Query Patroller Version 7 control tables has completed successfully.

Explanation: The Query Patroller migration cleanup completed successfully and all Version 7 Query

Patroller tables, view, and triggers have been dropped.

User Response: No action required.

DQP2605E The cleanup of the Query Patroller Version 7 control tables has failed.

Explanation: The Query Patroller migration tool encountered fatal error while cleaning the Version 7 Query Patroller database objects. The cleanup did not complete successfully.

User Response: Refer to the qpmigrate.log file for possible causes of this message.

DQP2606W The cleanup of the Query Patroller Version 7 control tables has completed with warnings.

Explanation: The Query Patroller migration cleanup encountered warnings while deleting the Version 7 Query Patroller tables. However, it may have completed successfully. There may have been errors migrating users and/or parts of the system configuration.

User Response: Refer to the qpmigrate.log file for possible causes of this message.

DQP2607I Migration of Query Patroller control tables from Version 7 to Version 8 has completed successfully.

Explanation: The Query Patroller migration tool completed successfully.

User Response: No action required.

DQP2608E Migration of Query Patroller control tables from Version 7 to Version 8 has failed.

Explanation: The Query Patroller Migration tool encountered fatal errors while migrating the database. The migration did not complete successfully.

User Response: Refer to the qpmigrate.log file for possible causes of this message.

DQP2609W Migration of Query Patroller control tables from Version 7 to Version 8 has completed with warnings.

Explanation: The Query Patroller Migration tool encountered warnings while migrating the database, however, the migration may have completed successfully. There may have been errors migrating users and/or parts of the system configuration.

User Response: Refer to the qpmigrate.log file for possible causes of this message.

DQP2610E Stop Query Patroller server before proceeding.

Explanation: The Query Patroller server must be stopped before the Query Patroller migration tool can be run.

User Response: Issue the qpstop command to stop the Query Patroller server.

DQP2611W User profile for user *username* has not been migrated. SQLCODE = *SQLCODE*.

Explanation: The user has not been migrated.

User Response: Refer to the SQLCODE for possible causes of this message.

DQP2612I Migration of Query Patroller user and group profiles has completed successfully.

Explanation: The data from the Version 7 Query Patroller table IWM003_USER_PROF has been migrated to the Version 8 Query Patroller tables SUBMITTER_PROFILE, OPERATOR_PROFILE, and SUBMISSION_PREFERENCES.

User Response: No action required.

DQP2613W Migration of Query Patroller user and group profiles has completed with warnings.

Explanation: The Query Patroller migration tool encountered warnings while migrating from Version 7 Query Patroller table IWM003_USER_PROF to the Version 8 Query Patroller tables SUBMITTER_PROFILE, OPERATOR_PROFILE, and SUBMISSION_PREFERENCES.

User Response: Refer to the qpmigrate.log for a list of users that were not migrated. If the user name conflict was expected, then no action is required. Otherwise, you may copy the data manually into the table.

DQP2614W Migration of Query Patroller user and group profiles has failed.

Explanation: There was an error reading data from the Version 7 Query Patroller tables and the migration of user and group profiles information could not continue. The Version 7 Query Patroller tables may be corrupt or there was an I/O error while reading from the Version 7 Query Patroller tables.

User Response: Ensure that all Version 7 Query Patroller tables are present with valid data.

DQP2615I Migration of the Query Patroller system configuration has completed successfully.

Explanation: The data from Version 7 Query Patroller table IWM003_JOB_QUEUE and IWM003_SYS_PARMS have been migrated to the Version 8 Query Patroller tables QUERY_CLASS and QP_SYSTEM successfully.

User Response: No action required.

DQP2616W Migration of the Query Patroller system configuration has completed with warnings.

Explanation: The Query Patroller migration tool encountered warnings while migrating from Version 7 Query Patroller tables IWM003_JOB_QUEUE and IWM003_SYS_PARMS to the Version 8 Query Patroller tables QUERY_CLASS and QP_SYSTEM. It is possible that various job queues already exist in the table.

User Response: Refer to the qpmigrate.log file for a list of job queues that were not migrated. If the job queue conflict was expected, then no action is needed. Otherwise, you may copy the data manually into the table.

DQP2617E Migration of the Query Patroller system configuration has failed.

Explanation: There was an error reading data from the Version 7 Query Patroller tables and the migration of system configuration information could not continue. It is possible that the Version 7 Query Patroller tables are corrupt or there was an I/O error while reading from the Version 7 Query Patroller tables.

User Response: Ensure that all the Version 7 Query Patroller tables are present with valid data.

DQP2625W Query ID *queryid* failed to migrate because an entry with an identical maximum query cost of *querycost* already exists.

Explanation: Query Patroller migration found an entry in the QUERY_CLASS control table with a maximum cost that is identical to the query class that is attempting to be migrated. Entries with identical maximum costs are not permitted.

User Response: If the maximum cost conflict was expected, then no action is required. Otherwise, change the maximum cost of the query class to migrate and manually insert it into the QUERY_CLASS control table.

DQP3000E The maximum number of queries for this query class must not be greater than that of the Query Patroller System.

Explanation: The maximum number of queries may be set for a query class and for the Query Patroller system itself. No query class can exist with a maximum of queries larger than that of the Query Patroller system.

User Response: Lower the maximum number of queries for the query class or increase the maximum number of queries for the Query Patroller system before increasing the maximum number of queries for the query class.

DQP3001E The maximum query cost for this query class must not be greater than the maximum workload cost of the Query Patroller System.

Explanation: No query class can exist with a maximum query cost larger than the maximum workload cost of the Query Patroller system.

User Response: Lower the maximum query cost for the query class or increase the maximum workload cost for the Query Patroller system before increasing the maximum query cost for the query class.

DQP3002E One or more query classes exist with the maximum number of queries greater than the value specified for the Query Patroller System.

Explanation: The maximum number of queries may be set for a query class and for the Query Patroller system itself. No query class can exist with a maximum number of queries larger than that of the Query Patroller system.

User Response: Increase the maximum number of queries for the Query Patroller system or decrease the maximum number of queries for each query class that is preventing the system maximum from being lowered.

DQP3003E One or more query classes exist with a maximum query cost greater than the value specified for the maximum workload cost of the Query Patroller System.

Explanation: No query class can exist with a maximum query cost larger than the maximum workload cost of the Query Patroller system.

User Response: Increase the maximum workload cost for the Query Patroller system or decrease the maximum query cost for each query class that is preventing the system maximum from being lowered.

DQP3010E The public submitter profile cannot be removed.

Explanation: The public submitter profile cannot be removed.

User Response: No action required.

DQP3011E A user submitter profile name cannot be specified.

Explanation: The command does not accept a user submitter profile name to be specified. By default, the user's own submitter profile would be used automatically, if it exists. If it does not exist, the user can specify a group submitter profile that the user belongs to.

User Response: Reissue the command and specify a group submitter profile, or do not specify a submitter profile.

DQP3012E The public submission preferences cannot be removed.

Explanation: The public submission preferences cannot be removed.

User Response: No action required.

Appendix D. Query Patroller graphical user interface

Logging on to the Query Patroller Center

Before you can use the Query Patroller Center, you need to log on. The Login window opens automatically when you launch the Query Patroller Center.

Procedure:

To log on to the Query Patroller Center, use one of the following methods.

Control Center method:

1. Open the Login window
2. Specify the login credentials you want to use by selecting either **Default** or **Different user ID**. By default, you log on to the Query Patroller Center using the same ID that you are using to sign onto the operating system.
3. If you are not using the default ID, type different values in the **User ID** and **Password** fields.
4. Click **OK** to open the Query Patroller Center.

Start Menu method:

1. Open the Login window
2. In the **Database** field, select the database you want to work with. If the database you want to work with is not listed, click **Update Database List** to open a window where you can add a cataloged database to the list.
3. Specify the login credentials you want to use by selecting either **Default** or **Different user ID**. By default, you log on to the Query Patroller Center using the same operating system ID you are currently using.
4. If you are not using the default ID, type different values in the **User ID** and **Password** fields.
5. Click **OK** to open the Query Patroller Center.

Command line method:

Issue the **qpcenter** command.

Related tasks:

- "Updating the list of databases in Query Patroller" on page 124

Getting started with the Query Patroller Center

Use the Query Patroller Center to monitor and control the flow of queries against a database. By using the Query Patroller Center, you can ensure that users and queries are prioritized appropriately, that query workloads are balanced across the system, and that excessively large queries are caught before they are run.

The Query Patroller Center also provides historical analysis tools and reports. The historical analysis functions allow you to examine various aspects of database use over time.

From the Query Patroller Center you can also open other centers and tools to help you monitor a database.

Query Patroller authorities and privileges:

Administrators
The Query Patroller administrator is the person who has DBADM authority for the database.

Operators
If you are an operator, the Query Patroller administrator will assign a subset of administrator privileges to you, and you will have access to a subset of Query Patroller Center functions. The four operator privileges are:

- User administration
- Monitoring
- Configuration
- Historical analysis

Query submitters
If you are a Query Patroller submitter (and not a Query Patroller administrator or operator), you will see a different view of the Query Patroller Center. You will have access to your own managed queries, as well as your own query submission preferences.

Query Patroller tasks:

Depending on your Query Patroller authority and privileges, you may be able to perform some or all of these tasks.

Setting thresholds and schedules

- Setting query thresholds for the Query Patroller system
- Running held queries at a scheduled time
- Setting Query Patroller maintenance schedules for queries and result tables

User administration tasks

- Creating submitter profiles for users and groups
- Creating operator profiles for users and groups
- Changing submitter profiles for users and groups
- Changing operator profiles for users and groups
- Suspending or restoring submitter privileges for users and groups
- Suspending or restoring operator privileges for users and groups
- Removing submitter profiles for users and groups
- Removing operator profiles for users and groups
- Setting query submission preferences for another submitter

Monitoring tasks

- Canceling queries by Changing the status of queries using Query Patroller
- Running held queries by Changing the status of queries using Query Patroller
- Dropping result tables manually using Query Patroller

Configuration tasks

- Setting query thresholds for the Query Patroller system

- Creating query classes for Query Patroller
- Changing query classes for Query Patroller
- Removing query classes for Query Patroller

Historical analysis tasks

- Generating historical data using Query Patroller
- Managing historical queries
- Viewing historical query details using Query Patroller

Submitter tasks

- Canceling queries by Changing the status of queries using Query Patroller
- Running queries in the background by Changing the status of queries using Query Patroller
- Setting your own query submission preferences
- Viewing result tables using Query Patroller

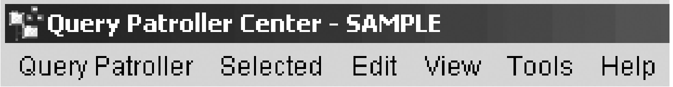
Accessibility

- Keyboard shortcuts and accelerators
- ASCII values for Alt-Gr key combinations

The Query Patroller Center interface:

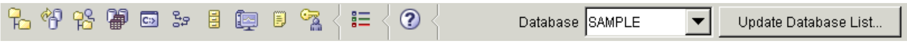
The Query Patroller Center interface has the following elements that help you control and analyze query flow.

Menu bar



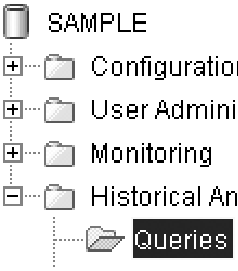
Use the menu bar to work with objects in the Query Patroller Center, open other administration centers and tools, and access online help.

Query Patroller Center toolbar



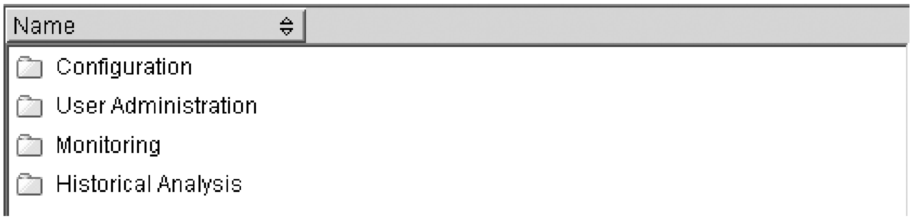
Use the toolbar icons below the menu bar to access other administration centers and tools. These functions can also be selected in the **Tools** menu. Use the **Database** drop-down menu and the **Update Database List** button to determine the database that will be monitored by the Query Patroller Center. If you do not have the administration tools (for example, the Control Center) installed, the Query Patroller Center toolbar will only contain the Legend and Help icons, the **Database** drop-down menu and the **Update Database List** button.

Object tree



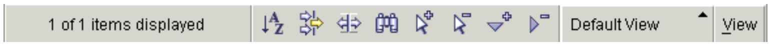
Use the object tree to display and work with database objects, users, queries, and historical analysis reports. Depending on your privileges, you might not be able to view all the objects in the object tree.

Contents pane



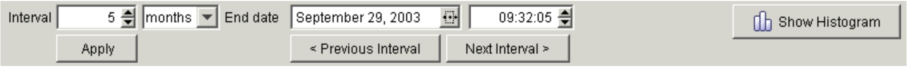
Use the contents pane to display and work with system, user, query, and historical analysis objects. The contents pane displays those objects that make up the contents of the object that is selected in the object tree.

Contents pane toolbar


1 of 1 items displayed

Use the toolbar below the contents pane to tailor the view of objects and information in the contents pane to suit your needs. These toolbar functions can also be selected in the **Edit** menu and the **View** menu.

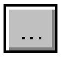
Historical analysis toolbar


Interval 5 months End date September 29, 2003 09:32:05

Use the historical analysis toolbar to display and work with the time intervals you are interested in, and to move between the graphical view and the reports view. For more information about this toolbar, see Getting started with the Query Patroller historical analysis interface.

Accessing custom controls with the keyboard

You can use the keyboard to access controls found on the graphical user interface. The following two controls are unique to DB2. Here is how you access them using the keyboard:

- 

  To access this button using the keyboard, use the Tab key until the button is selected then press the Spacebar.
- The date entry field

To access this feature using the keyboard, use the Tab key until the entry field is selected, then type the date in the entry field.

Getting started with the Query Patroller historical analysis interface

Use the historical analysis functions of the Query Patroller Center to examine various aspects of data warehouse use over time.

The historical analysis reports include reports on queries, tables, indexes, and submitters, for example:

- Query Activity over Time report: Use this report to determine such things as peak database usage and which queries are most costly.
- Tables Not Hit report: Use this report to determine which tables to eliminate.
- Submitters report: Use this report to determine information about submitters, such as which submitters are using the most resources.

Authorities and privileges:

To generate historical data, you must meet the following requirement:

- Have DBADM authority

To perform historical analysis tasks that modify Query Patroller data, such as removing historical queries manually, you must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile has the HISTORICAL ANALYSIS privilege with edit authority

To perform historical analysis tasks that involve viewing Query Patroller data, such as filtering tables, you must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile has the HISTORICAL ANALYSIS privilege with edit or view authority

Historical analysis tasks:

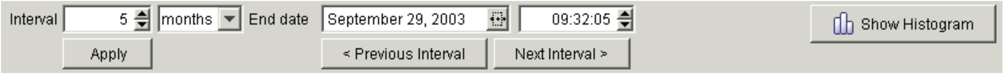
Depending on your Query Patroller authority and privileges, you may be able to perform some or all of these tasks:

- Generating historical data using Query Patroller
- Viewing historical query details using Query Patroller
- Filtering tables for historical analysis using Query Patroller
- Managing historical queries

The Query Patroller Center historical analysis interface:

There are two main types of views for Query Patroller historical analysis data, a table view and a graphical view.

Historical analysis toolbar:



Use the historical analysis toolbar to display and work with the time intervals you are interested in, and to move between the graphical view and the table view.

Use the **Interval** and **End date** fields to establish a time range that determines which items appear in the folders under the **Historical Analysis** folder and in the historical analysis contents pane.

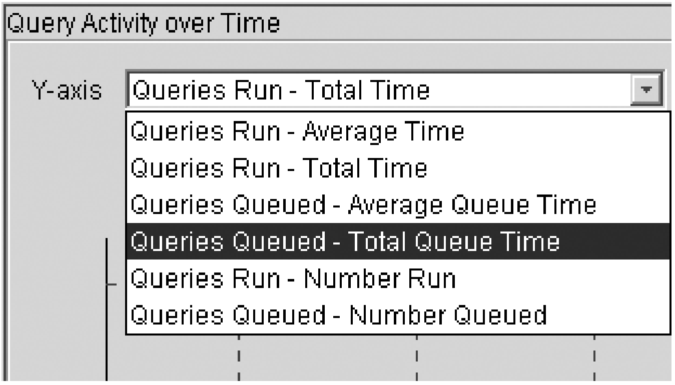
When you have specified an interval and an end date, you must click the **Apply** push button to refresh the view to show the historical analysis information with those settings.

Use the **Previous Interval** and **Next Interval** push buttons to move through the historical data displayed in the contents pane. Clicking **Previous Interval** moves the end date back by the interval specified in the **Interval** fields. For example, if the interval value is 5, the unit of time is days, and the end date is May 22, 2003, clicking **Previous Interval** will update the end date to May 17, 2003. Clicking **Next Interval** advances the end date by the interval specified in the **Interval** fields. For example, if the interval value is 5, the unit of time is days, and the end date is May 22, 2003, clicking **Next Interval** updates the end date to May 27, 2003. Clicking the **Previous Interval** and **Next Interval** push buttons automatically refreshes the information displayed in the contents pane and the items under the **Historical Analysis** folder.

When you are looking at the graphical view of a report, you can use the **Show Table** push button on the far right of the toolbar to switch to the table view. When you are looking at the table view, the **Show Table** push button becomes the **Show Graph** push button, so you can use it to switch back to the graphical view.

For historical analysis data in the **Queries** folder only, there is a histogram view instead of a graph. The **Show Graph** push button becomes the **Show Histogram** push button. In the histogram view, you can double-click the individual histogram bars to obtain information at a finer level. For example, if you are looking at a report that covers three months of queries, the report shows three histogram bars. Double-clicking one of the histogram bars displays the data for the weeks that comprise that month. If you have drilled down to obtain information at a finer level, you can then drill up to obtain information at a higher level by double-clicking with the right mouse button on any of the histogram bars. For example, if you have drilled down from a report showing three months of queries to a report showing four weeks of queries, double-click with the right mouse button to drill back up to the histogram that displays months.

Changing the information displayed in the historical analysis histogram:


Query Activity over Time
Queries Run - Average Time
Queries Run - Total Time
Queries Queued - Average Queue Time
Queries Queued - Total Queue Time
Queries Run - Number Run
Queries Queued - Number Queued

Y-axis Queries Run - Total Time

Use the Y-axis drop-down menu to specify the type of information you want to display about historical queries in the histogram. You can choose from the following six options:

- Queries Run - Average Time
- Queries Run - Total Time
- Queries Queued - Average Queue Time
- Queries Queued - Total Queue Time
- Queries Run - Number Run
- Queries Queued - Number Queued

Related concepts:

- “Query Patroller historical analysis” on page 103
- “Uses for historical analysis reports” on page 161

Related tasks:

- “Collecting historical data with Query Patroller” on page 163
- “Enabling collection of historical data” on page 125

Filtering tables for historical analysis using Query Patroller

When you are using the Query Patroller Center to look at historical analysis reports, there might be a significant number of rows in the Tables Hit and Tables Not Hit reports. Use the Filter window to display only the tables that meet conditions based on the name of the table and the schema.

Prerequisites:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile has the HISTORICAL ANALYSIS privilege with view or edit authority

Procedure:

To filter tables for historical analysis:

Query Patroller Center method:

1. Open the Filter notebook.
2. Optional: On the **Locate** tab, select an operator for the table name comparison, and enter a value for the table name to be compared to.
3. Optional: Select an operator for the table schema comparison, and enter a value for the table schema to be compared to.
4. To indicate that the filter should show only tables that match both comparisons you have entered, click the **Meet all conditions** radio button. To indicate that the filter should show tables that match either the table name comparison or the table schema comparison, click the **Meet any conditions** radio button.
5. Optional: To view the WHERE clause that is generated by the table name and table schema comparisons you have entered, click the **Details** tab.
6. Optional: The Filter notebook will open automatically when a folder is selected and the number of objects listed exceeds the value specified in the **Object count** field, if you have selected the **Automatically display filter when object count is exceeded** check box. To adjust the number of objects that will cause the Filter notebook to open automatically, enter a new value in the **Object count** field.
7. To enable the filter, select the **Enable filter** check box, then click **OK** to close the Filter notebook.
8. Optional: If you do not want to enable the filter, clear the **Enable filter** check box, then click **OK** to close the Filter notebook. The values you have entered in the Filter notebook will be retained, and will appear when you reopen the notebook.

The filter that you define will affect the number of objects that you see in the object tree and the number of reports that you see in the contents pane.

Related concepts:

- "Query Patroller historical analysis" on page 103
- "Getting started with the Query Patroller historical analysis interface" on page 295

Appendix E. Submitter tasks

Setting your own query submission preferences

The default query submission preferences are called PUBLIC. If you do not have your own submission preferences, by default you will use the settings specified in the PUBLIC submission preferences. If you do not want to use the values specified in the PUBLIC submission preferences, you can set different submission preferences for yourself.

Prerequisites:

You must meet one of the following requirements:
- Have DBADM authority
- Be an operator whose profile has the USER ADMINISTRATION privilege with edit authority
- Be the submitter to whom the query submission preferences belong

Procedure:

To set your query submission preferences, use one of the following methods.

Query Patroller Center method:
1. Open the Query Submission Preferences window.
2. If you belong to multiple group profiles, use the **Submitter profile to use** field to select which group profile you want to submit your queries under. If you have a submitter profile whose type is user, this field shows the submitter's user ID. You cannot change this value.

   If you lack a submitter profile whose type is user, yet belong to multiple group submitter profiles, this field lists the group submitter profiles. Select the group submitter profile that you want to submit your queries with. The Select automatically entry applies the profile with the lowest resource limits.
3. Specify what will happen after a query is submitted:
   - To specify that the application that submitted the query will wait for the result set to return while Query Patroller manages the query, select the **Wait until the results are returned** radio button. This option is the default.

     When this option is selected, the application that submitted the query can become unresponsive until the result set is returned.
   - To specify that the result set will be stored in a DB2 table and the application that submitted the query will become free for further processing, select the **Release the application and retrieve the results from a result table** radio button.
4. Specify the access levels you want to use for your result tables:
   - To specify that only the submitter can access the result tables, select the **Restrict access to the submitter** radio button.
   - To make the result table accessible to specific DB2 users select the **Grant access to other users or groups** radio button. Type the names of specific users or groups in the associated field, separated by commas. The DB2 users listed must have access to the database where the query was submitted.

5. Specify how to handle result sets that are longer than the maximum allowed:
   - To specify that the incomplete results should not be stored in a result table, select the **Do not return any results** radio button.
   - To specify that a truncated set of results should be stored in a result table, select the **Return a truncated set of results** radio button.
6. In the **E-mail address** field, type the e–mail address to be used for sending notifications to the submitter. Notifications can be sent to the address or addresses specified when a query completes or if a query encounters an error, but only in cases where a result table is created.

   Note: A result table is created for submitters when:
   - The **Release the application** option is specified in the Query Submission Preferences window.
   - The **Wait until results are returned** option is specified in the Query Submission Preferences window, yet the query was placed in a held state before completing.
   - The status of a query is changed to **Run query in the background**.
7. Click **OK** to set preferences for query submission.

Command line method:

To create new query submission preferences, issue the **ADD SUBMISSION_PREFERENCES** command.

To change existing query submission preferences, issue the **UPDATE SUBMISSION_PREFERENCES** command.

Related concepts:

- "Query Patroller query submission preferences" on page 147
- "Query Submission Preferences — Overview"--*Help: Query Patroller*

Related tasks:

- "Removing query submission preferences for submitters"--*Help: Query Patroller*

Monitoring your queries

By viewing the properties of a query that has been managed by Query Patroller, you can see details such as information about the submitter of the query, the processing time, and the result table.

Prerequisites:

You must meet one of the following requirements:

- Have DBADM authority
- Be an operator whose profile has the MONITORING privilege with edit or view authority
- Be the submitter of the query

Procedure:

To view the details of a query:

Query Patroller Center method:

1. Open the Managed Query Properties notebook.
2. To view general information about the query, click the **General** tab.
   a. To view the SQL of the query in a new window, click **View SQL in Separate Window**.
   b. To view the profile of the submitter who submitted the query, click **Submitter Properties**. You must have DBADM authority or be an operator whose profile has the USER ADMINISTRATION privilege with edit or view authority to open the Submitter Properties window.
   c. If you want to change the status of the query, for example to cancel it, click **Change Status**.
3. To view query execution information and result table details, click the **Results** tab.
   a. To display the result table for the query, click **Show Results**.
   b. To save the result table for the query, click **Save Results**.
   c. To delete the result table for the query, click **Drop Result Table**.
4. To view query event timestamps and processing durations, click the **Time** tab.
5. To view authorization IDs, application information, and user information, click the **Other** tab.
6. Click **Close** to close the Managed Query Properties notebook.

Command line method:

Issue the **GET QUERY** command.

Canceling your queries

Depending on your authority level, you may change the status of queries managed by Query Patroller in several ways: cancel a query, release a held query, or run a query in the background.

Canceling queries
Cancel a query if you realize after it has been submitted that it contains an error, or that its cost is too high. For example, you might receive notification that your query is being held because its cost exceeds the maximum amount of system resource that is allowed for any one of your queries. Canceling a query places it in the canceled state.

Releasing queries from the held state
Release a held query if you decide that a particular query should run, even though the query exceeds the submitter's maximum query cost. Releasing a held query places the query in the running state or the queued state, depending on the current system workload.

Running queries in the background
Run a query in the background if your query submission preferences specify that you will wait until the results of a query are returned, but you want to use your client application while that particular query runs. Running a query in the background places the query in the running state or the queued state, depending on the current system workload.

Prerequisites:

- To cancel a query, you must meet one of the following requirements:
  – Have DBADM authority

- Be an operator whose profile has the MONITORING privilege with edit authority
  - Be the submitter of the query
- To release a held query, you must meet one of the following requirements:
  - Have DBADM authority
  - Be an operator whose profile has the MONITORING privilege with edit authority
- To run a query in the background, you must:
  - Be the submitter of the query

Procedure:

To change the status of a query:

Query Patroller Center method:

1. Open the Change Query Status window.
2. To cancel the query, click **Cancel query**.
3. To run the query, click **Release query from held state**.
4. To regain control of the submitting application, click **Run query in the background**. Query Patroller stops the execution of the query and resubmits it. The results of the query will be returned to a result table.
5. Click **OK** to change the status of the query as you have indicated and close the Change Query Status window.

Command line method:

To cancel the query, issue the **CANCEL QUERY** command.

To run the query, issue the **RUN HELD_QUERY** command.

To rerun the query in the background, issue the **RUN IN BACKGROUND QUERY** command.

Appendix F. DB2 Universal Database technical information

DB2 documentation and help

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

- DB2 Information Center
  - Topics
  - Help for DB2 tools
  - Sample programs
  - Tutorials
- Downloadable PDF files, PDF files on CD, and printed books
  - Guides
  - Reference manuals
- Command line help
  - Command help
  - Message help
  - SQL state help
- Installed source code
  - Sample programs

You can access additional DB2 Universal Database™ technical information such as technotes, white papers, and Redbooks™ online at ibm.com®. Access the DB2 Information Management software library site at www.ibm.com/software/data/pubs/.

DB2 documentation updates

IBM® may periodically make documentation FixPaks and other documentation updates to the DB2 Information Center available. If you access the DB2 Information Center at http://publib.boulder.ibm.com/infocenter/db2help/, you will always be viewing the most up-to-date information. If you have installed the DB2 Information Center locally, then you need to install any updates manually before you can view them. Documentation updates allow you to update the information that you installed from the *DB2 Information Center CD* when new information becomes available.

The Information Center is updated more frequently than either the PDF or the hardcopy books. To get the most current DB2 technical information, install the documentation updates as they become available or go to the DB2 Information Center at the www.ibm.com site.

DB2 Information Center

The DB2® Information Center gives you access to all of the information you need to take full advantage of DB2 family products, including DB2 Universal Database™, DB2 Connect™, DB2 Information Integrator and DB2 Query Patroller™. The DB2 Information Center also contains information for major DB2 features and components including replication, data warehousing, and the DB2 extenders.

The DB2 Information Center has the following features if you view it in Mozilla 1.0 or later or Microsoft® Internet Explorer 5.5 or later. Some features require you to enable support for JavaScript™:

Flexible installation options

You can choose to view the DB2 documentation using the option that best meets your needs:

- To effortlessly ensure that your documentation is always up to date, you can access all of your documentation directly from the DB2 Information Center hosted on the IBM® Web site at http://publib.boulder.ibm.com/infocenter/db2help/
- To minimize your update efforts and keep your network traffic within your intranet, you can install the DB2 documentation on a single server on your intranet
- To maximize your flexibility and reduce your dependence on network connections, you can install the DB2 documentation on your own computer

Search

You can search all of the topics in the DB2 Information Center by entering a search term in the **Search** text field. You can retrieve exact matches by enclosing terms in quotation marks, and you can refine your search with wildcard operators (*, ?) and Boolean operators (AND, NOT, OR).

Task-oriented table of contents

You can locate topics in the DB2 documentation from a single table of contents. The table of contents is organized primarily by the kind of tasks you may want to perform, but also includes entries for product overviews, goals, reference information, an index, and a glossary.

- Product overviews describe the relationship between the available products in the DB2 family, the features offered by each of those products, and up to date release information for each of these products.
- Goal categories such as installing, administering, and developing include topics that enable you to quickly complete tasks and develop a deeper understanding of the background information for completing those tasks.
- Reference topics provide detailed information about a subject, including statement and command syntax, message help, and configuration parameters.

Show current topic in table of contents

You can show where the current topic fits into the table of contents by clicking the **Refresh / Show Current Topic** button in the table of contents frame or by clicking the **Show in Table of Contents** button in the content frame. This feature is helpful if you have followed several links to related topics in several files or arrived at a topic from search results.

Index You can access all of the documentation from the index. The index is organized in alphabetical order by index term.

Glossary

You can use the glossary to look up definitions of terms used in the DB2 documentation. The glossary is organized in alphabetical order by glossary term.

Integrated localized information

The DB2 Information Center displays information in the preferred language set in your browser preferences. If a topic is not available in your preferred language, the DB2 Information Center displays the English version of that topic.

For iSeries™ technical information, refer to the IBM eServer™ iSeries information center at www.ibm.com/eserver/iseries/infocenter/.

Related tasks:

- "Updating the DB2 Information Center installed on your computer or intranet server" on page 312

DB2 Information Center installation scenarios

Different working environments can pose different requirements for how to access DB2® information. The DB2 Information Center can be accessed on the IBM® Web site, on a server on your organization's network, or on a version installed on your computer. In all three cases, the documentation is contained in the DB2 Information Center, which is an architected web of topic-based information that you view with a browser. By default, DB2 products access the DB2 Information Center on the IBM Web site. However, if you want to access the DB2 Information Center on an intranet server or on your own computer, you must install the DB2 Information Center using the DB2 Information Center CD found in your product Media Pack. Refer to the summary of options for accessing DB2 documentation which follows, along with the three installation scenarios, to help determine which method of accessing the DB2 Information Center works best for you and your work environment, and what installation issues you might need to consider.

Summary of options for accessing DB2 documentation:

The following table provides recommendations on which options are possible in your work environment for accessing the DB2 product documentation in the DB2 Information Center.

Internet access	Intranet access	Recommendation
Yes	Yes	Access the DB2 Information Center on the IBM Web site, or access the DB2 Information Center installed on an intranet server.
Yes	No	Access the DB2 Information Center on the IBM Web site.
No	Yes	Access the DB2 Information Center installed on an intranet server.
No	No	Access the DB2 Information Center on a local computer.

Scenario: Accessing the DB2 Information Center on your computer:

Tsu-Chen owns a factory in a small town that does not have a local ISP to provide him with Internet access. He purchased DB2 Universal Database™ to manage his inventory, his product orders, his banking account information, and his business expenses. Never having used a DB2 product before, Tsu-Chen needs to learn how to do so from the DB2 product documentation.

After installing DB2 Universal Database on his computer using the typical installation option, Tsu-Chen tries to access the DB2 documentation. However, his browser gives him an error message that the page he tried to open cannot be found. Tsu-Chen checks the installation manual for his DB2 product and discovers that he has to install the DB2 Information Center if he wants to access DB2 documentation on his computer. He finds the *DB2 Information Center CD* in the media pack and installs it.

From the application launcher for his operating system, Tsu-Chen now has access to the DB2 Information Center and can learn how to use his DB2 product to increase the success of his business.

Scenario: Accessing the DB2 Information Center on the IBM Web site:

Colin is an information technology consultant with a training firm. He specializes in database technology and SQL and gives seminars on these subjects to businesses all over North America using DB2 Universal Database. Part of Colin's seminars includes using DB2 documentation as a teaching tool. For example, while teaching courses on SQL, Colin uses the DB2 documentation on SQL as a way to teach basic and advanced syntax for database queries.

Most of the businesses at which Colin teaches have Internet access. This situation influenced Colin's decision to configure his mobile computer to access the DB2 Information Center on the IBM Web site when he installed the latest version of DB2 Universal Database. This configuration allows Colin to have online access to the latest DB2 documentation during his seminars.

However, sometimes while travelling Colin does not have Internet access. This posed a problem for him, especially when he needed to access to DB2 documentation to prepare for seminars. To avoid situations like this, Colin installed a copy of the DB2 Information Center on his mobile computer.

Colin enjoys the flexibility of always having a copy of DB2 documentation at his disposal. Using the **db2set** command, he can easily configure the registry variables on his mobile computer to access the DB2 Information Center on either the IBM Web site, or his mobile computer, depending on his situation.

Scenario: Accessing the DB2 Information Center on an intranet server:

Eva works as a senior database administrator for a life insurance company. Her administration responsibilities include installing and configuring the latest version of DB2 Universal Database on the company's UNIX® database servers. Her company recently informed its employees that, for security reasons, it would not provide them with Internet access at work. Because her company has a networked environment, Eva decides to install a copy of the DB2 Information Center on an intranet server so that all employees in the company who use the company's data warehouse on a regular basis (sales representatives, sales managers, and business analysts) have access to DB2 documentation.

Eva instructs her database team to install the latest version of DB2 Universal Database on all of the employee's computers using a response file, to ensure that each computer is configured to access the DB2 Information Center using the host name and the port number of the intranet server.

However, through a misunderstanding Migual, a junior database administrator on Eva's team, installs a copy of the DB2 Information Center on several of the employee computers, rather than configuring DB2 Universal Database to access the DB2 Information Center on the intranet server. To correct this situation Eva tells Migual to use the **db2set** command to change the DB2 Information Center registry variables (DB2_DOCHOST for the host name, and DB2_DOCPORT for the port number) on each of these computers. Now all of the appropriate computers on the network have access to the DB2 Information Center, and employees can find answers to their DB2 questions in the DB2 documentation.

Installing the DB2 Information Center using the DB2 Setup wizard (UNIX)

DB2 product documentation can be accessed in three ways: on the IBM Web site, on an intranet server, or on a version installed on your computer. By default, DB2 products access DB2 documentation on the IBM Web site. If you want to access the DB2 documentation on an intranet server or on your own computer, you must install the documentation from the *DB2 Information Center CD*. Using the DB2 Setup wizard, you can define your installation preferences and install the DB2 Information Center on a computer that uses a UNIX operating system.

Prerequisites:

This section lists the hardware, operating system, software, and communication requirements for installing the DB2 Information Center on UNIX computers.

- **Hardware requirements**

  You require one of the following processors:
  - PowerPC (AIX)
  - HP 9000 (HP-UX)
  - Intel 32–bit (Linux)
  - Solaris UltraSPARC computers (Solaris Operating Environment)
- **Operating system requirements**

  You require one of the following operating systems:
  - IBM AIX 5.1 (on PowerPC)
  - HP-UX 11i (on HP 9000)
  - Red Hat Linux 8.0 (on Intel 32–bit)
  - SuSE Linux 8.1 (on Intel 32–bit)
  - Sun Solaris Version 8 (on Solaris Operating Environment UltraSPARC computers)

  Note: The DB2 Information Center runs on a subset of the UNIX operating systems on which DB2 clients are supported. It is therefore recommended that you either access the DB2 Information Center from the IBM Web site, or that you install and access the DB2 Information Center on an intranet server.
- **Software requirements**
  - The following browser is supported:
    - Mozilla Version 1.0 or greater
- The DB2 Setup wizard is a graphical installer. You must have an implementation of the X Window System software capable of rendering a graphical user interface for the DB2 Setup wizard to run on your computer. Before you can run the DB2 Setup wizard you must ensure that you have properly exported your display. For example, enter the following command at the command prompt:

  ```
  export DISPLAY=9.26.163.144:0.
  ```
- **Communication requirements**

– TCP/IP

Procedure:

To install the DB2 Information Center using the DB2 Setup wizard:

1. Log on to the system.
2. Insert and mount the DB2 Information Center product CD on your system.
3. Change to the directory where the CD is mounted by entering the following command:

   ```
   cd /cd
   ```

   where */cd* represents the mount point of the CD.
4. Enter the **./db2setup** command to start the DB2 Setup wizard.
5. The IBM DB2 Setup Launchpad opens. To proceed directly to the installation of the DB2 Information Center, click **Install Product**. Online help is available to guide you through the remaining steps. To invoke the online help, click **Help**. You can click **Cancel** at any time to end the installation.
6. On the **Select the product you would like to install** page, click **Next**.
7. Click **Next** on the **Welcome to the DB2 Setup wizard** page. The DB2 Setup wizard will guide you through the program setup process.
8. To proceed with the installation, you must accept the license agreement. On the **License Agreement** page, select **I accept the terms in the license agreement** and click **Next**.
9. Select **Install DB2 Information Center on this computer** on the **Select the installation action** page. If you want to use a response file to install the DB2 Information Center on this or other computers at a later time, select **Save your settings in a response file**. Click **Next**.
10. Select the languages in which the DB2 Information Center will be installed on **Select the languages to install** page. Click **Next**.
11. Configure the DB2 Information Center for incoming communication on the **Specify the DB2 Information Center port** page. Click **Next** to continue the installation.
12. Review the installation choices you have made in the **Start copying files** page. To change any settings, click **Back**. Click **Install** to copy the DB2 Information Center files onto your computer.

You can also install the DB2 Information Center using a response file.

The installation logs db2setup.his, db2setup.log, and db2setup.err are located, by default, in the /tmp directory.

The db2setup.log file captures all DB2 product installation information, including errors. The db2setup.his file records all DB2 product installations on your computer. DB2 appends the db2setup.log file to the db2setup.his file. The db2setup.err file captures any error output that is returned by Java, for example, exceptions and trap information.

When the installation is complete, the DB2 Information Center will be installed in one of the following directories, depending upon your UNIX operating system:

- AIX: /usr/opt/db2_08_01
- HP-UX: /opt/IBM/db2/V8.1
- Linux: /opt/IBM/db2/V8.1

- Solaris Operating Environment: /opt/IBM/db2/V8.1

Related tasks:

- "Installing the DB2 Information Center using the DB2 Setup wizard (Windows)" on page 309

Installing the DB2 Information Center using the DB2 Setup wizard (Windows)

DB2 product documentation can be accessed in three ways: on the IBM Web site, on an intranet server, or on a version installed on your computer. By default, DB2 products access DB2 documentation on the IBM Web site. If you want to access the DB2 documentation on an intranet server or on your own computer, you must install the DB2 documentation from the *DB2 Information Center CD*. Using the DB2 Setup wizard, you can define your installation preferences and install the DB2 Information Center on a computer that uses a Windows operating system.

Prerequisites:

This section lists the hardware, operating system, software, and communication requirements for installing the DB2 Information Center on Windows.

- **Hardware requirements**

  You require one of the following processors:

  - 32-bit computers: a Pentium or Pentium compatible CPU
- **Operating system requirements**

  You require one of the following operating systems:

  - Windows 2000
  - Windows XP

  Note: The DB2 Information Center runs on a subset of the Windows operating systems on which DB2 clients are supported. It is therefore recommended that you either access the DB2 Information Center on the IBM Web site, or that you install and access the DB2 Information Center on an intranet server.
- **Software requirements**
  - The following browsers are supported:
    - Mozilla 1.0 or greater
    - Internet Explorer Version 5.5 or 6.0 (Version 6.0 for Windows XP)
- **Communication requirements**
  - TCP/IP

Restrictions:

- You require an account with administrative privileges to install the DB2 Information Center.

Procedure:

To install the DB2 Information Center using the DB2 Setup wizard:

1. Log on to the system with the account that you have defined for the DB2 Information Center installation.

2. Insert the CD into the drive. If enabled, the auto-run feature starts the IBM DB2 Setup Launchpad.
3. The DB2 Setup wizard determines the system language and launches the setup program for that language. If you want to run the setup program in a language other than English, or the setup program fails to auto-start, you can start the DB2 Setup wizard manually.

   To start the DB2 Setup wizard manually:

   a. Click **Start** and select **Run**.
   b. In the **Open** field, type the following command:

      ```
      x:\setup.exe /i 2-letter language identifier
      ```

      where *x:* represents your CD drive, and *2-letter language identifier* represents the language in which the setup program will be run.
   c. Click **OK**.
4. The IBM DB2 Setup Launchpad opens. To proceed directly to the installation of the DB2 Information Center, click **Install Product**. Online help is available to guide you through the remaining steps. To invoke the online help, click **Help**. You can click **Cancel** at any time to end the installation.
5. On the **Select the product you would like to install** page, click **Next**.
6. Click **Next** on the **Welcome to the DB2 Setup wizard** page. The DB2 Setup wizard will guide you through the program setup process.
7. To proceed with the installation, you must accept the license agreement. On the **License Agreement** page, select **I accept the terms in the license agreement** and click **Next**.
8. Select **Install DB2 Information Center on this computer** on the **Select the installation action** page. If you want to use a response file to install the DB2 Information Center on this or other computers at a later time, select **Save your settings in a response file**. Click **Next**.
9. Select the languages in which the DB2 Information Center will be installed on **Select the languages to install** page. Click **Next**.
10. Configure the DB2 Information Center for incoming communication on the **Specify the DB2 Information Center port** page. Click **Next** to continue the installation.
11. Review the installation choices you have made in the **Start copying files** page. To change any settings, click **Back**. Click **Install** to copy the DB2 Information Center files onto your computer.

You can install the DB2 Information Center using a response file. You can also use the **db2rspgn** command to generate a response file based on an existing installation.

For information on errors encountered during installation, see the db2.log and db2wi.log files located in the ’My Documents’\DB2LOG\ directory. The location of the ’My Documents’ directory will depend on the settings on your computer.

The db2wi.log file captures the most recent DB2 installation information. The db2.log captures the history of DB2 product installations.

Related tasks:

- “Installing the DB2 Information Center using the DB2 Setup wizard (UNIX)” on page 307

Invoking the DB2 Information Center

The DB2 Information Center gives you access to all of the information that you need to use DB2 products for Linux, UNIX, and Windows operating systems such as DB2 Universal Database, DB2 Connect, DB2 Information Integrator, and DB2 Query Patroller.

You can invoke the DB2 Information Center from one of the following places:

- Computers on which a DB2 UDB client or server is installed
- An intranet server or local computer on which the DB2 Information Center installed
- The IBM Web site

Prerequisites:

Before you invoke the DB2 Information Center:

- *Optional*: Configure your browser to display topics in your preferred language
- *Optional*: Configure your DB2 client to use the DB2 Information Center installed on your computer or intranet server

Procedure:

To invoke the DB2 Information Center on a computer on which a DB2 UDB client or server is installed:

- From the Start Menu (Windows operating system): Click **Start → Programs → IBM DB2 → Information → Information Center**.
- From the command line prompt:
  - For Linux and UNIX operating systems, issue the **db2icdocs** command.
  - For the Windows operating system, issue the **db2icdocs.exe** command.

To open the DB2 Information Center installed on an intranet server or local computer in a Web browser:

- Open the Web page at http://<host-name>:<port-number>/, where <host-name> represents the host name and <port-number> represents the port number on which the DB2 Information Center is available.

To open the DB2 Information Center on the IBM Web site in a Web browser:

- Open the Web page at publib.boulder.ibm.com/infocenter/db2help/.

Related concepts:

- "DB2 Information Center" on page 303

Related tasks:

- "Displaying topics in your preferred language in the DB2 Information Center" on page 312
- "Invoking contextual help from a DB2 tool" on page 319
- "Updating the DB2 Information Center installed on your computer or intranet server" on page 312
- "Invoking message help from the command line processor" on page 320
- "Invoking command help from the command line processor" on page 320
- "Invoking SQL state help from the command line processor" on page 321

Updating the DB2 Information Center installed on your computer or intranet server

The DB2 Information Center available from http://publib.boulder.ibm.com/infocenter/db2help/ will be periodically updated with new or changed documentation. IBM may also make DB2 Information Center updates available to download and install on your computer or intranet server. Updating the DB2 Information Center does not update DB2 client or server products.

Prerequisites:

You must have access to a computer that is connected to the Internet.

Procedure:

To update the DB2 Information Center installed on your computer or intranet server:

1. Open the DB2 Information Center hosted on the IBM Web site at: http://publib.boulder.ibm.com/infocenter/db2help/
2. In the Downloads section of the welcome page under the Service and Support heading, click the **DB2 Universal Database documentation** link.
3. Determine if the version of your DB2 Information Center is out of date by comparing the latest refreshed documentation image level to the documentation level you have installed. The documentation level you have installed is listed on the DB2 Information Center welcome page.
4. If a more recent version of the DB2 Information Center is available, download the latest refreshed *DB2 Information Center* image applicable to your operating system.
5. To install the refreshed *DB2 Information Center* image, follow the instructions provided on the Web page.

Related tasks:

- "Copying files from the DB2 HTML Documentation CD to a Web server" in the *Quick Beginnings for DB2 Personal Edition*

Related reference:

- "DB2 PDF and printed documentation" on page 313

Displaying topics in your preferred language in the DB2 Information Center

The DB2 Information Center attempts to display topics in the language specified in your browser preferences. If a topic has not been translated into your preferred language, the DB2 Information Center displays the topic in English.

Procedure:

To display topics in your preferred language in the Internet Explorer browser:

1. In Internet Explorer, click the **Tools** —> **Internet Options** —> **Languages...** button. The Language Preferences window opens.

2. Ensure your preferred language is specified as the first entry in the list of languages.
   - To add a new language to the list, click the **Add...** button.

     Note: Adding a language does not guarantee that the computer has the fonts required to display the topics in the preferred language.
   - To move a language to the top of the list, select the language and click the **Move Up** button until the language is first in the list of languages.
3. Refresh the page to display the DB2 Information Center in your preferred language.

To display topics in your preferred language in the Mozilla browser:

1. In Mozilla, select the **Edit —>** **Preferences —>** **Languages** button. The Languages panel is displayed in the Preferences window.
2. Ensure your preferred language is specified as the first entry in the list of languages.
   - To add a new language to the list, click the **Add...** button to select a language from the Add Languages window.
   - To move a language to the top of the list, select the language and click the **Move Up** button until the language is first in the list of languages.
3. Refresh the page to display the DB2 Information Center in your preferred language.

DB2 PDF and printed documentation

The following tables provide official book names, form numbers, and PDF file names. To order hardcopy books, you must know the official book name. To print a PDF file, you must know the PDF file name.

The DB2 documentation is categorized by the following headings:

- Core DB2 information
- Administration information
- Application development information
- Business intelligence information
- DB2 Connect information
- Getting started information
- Tutorial information
- Optional component information
- Release notes

The following tables describe, for each book in the DB2 library, the information needed to order the hard copy, or to print or view the PDF for that book. A full description of each of the books in the DB2 library is available from the IBM Publications Center at www.ibm.com/shop/publications/order

Core DB2 information

The information in these books is fundamental to all DB2 users; you will find this information useful whether you are a programmer, a database administrator, or someone who works with DB2 Connect, DB2 Warehouse Manager, or other DB2 products.

Table 22. Core DB2 information

Name	Form Number	PDF File Name
IBM DB2 Universal Database Command Reference	SC09-4828	db2n0x81
IBM DB2 Universal Database Glossary	No form number	db2t0x81
IBM DB2 Universal Database Message Reference, Volume 1	GC09-4840, not available in hardcopy	db2m1x81
IBM DB2 Universal Database Message Reference, Volume 2	GC09-4841, not available in hardcopy	db2m2x81
IBM DB2 Universal Database What's New	SC09-4848	db2q0x81

Administration information

The information in these books covers those topics required to effectively design, implement, and maintain DB2 databases, data warehouses, and federated systems.

Table 23. Administration information

Name	Form number	PDF file name
IBM DB2 Universal Database Administration Guide: Planning	SC09-4822	db2d1x81
IBM DB2 Universal Database Administration Guide: Implementation	SC09-4820	db2d2x81
IBM DB2 Universal Database Administration Guide: Performance	SC09-4821	db2d3x81
IBM DB2 Universal Database Administrative API Reference	SC09-4824	db2b0x81
IBM DB2 Universal Database Data Movement Utilities Guide and Reference	SC09-4830	db2dmx81
IBM DB2 Universal Database Data Recovery and High Availability Guide and Reference	SC09-4831	db2hax81
IBM DB2 Universal Database Data Warehouse Center Administration Guide	SC27-1123	db2ddx81
IBM DB2 Universal Database SQL Reference, Volume 1	SC09-4844	db2s1x81
IBM DB2 Universal Database SQL Reference, Volume 2	SC09-4845	db2s2x81
IBM DB2 Universal Database System Monitor Guide and Reference	SC09-4847	db2f0x81

Application development information

The information in these books is of special interest to application developers or programmers working with DB2 Universal Database (DB2 UDB). You will find information about supported languages and compilers, as well as the documentation required to access DB2 UDB using the various supported programming interfaces, such as embedded SQL, ODBC, JDBC, SQLJ, and CLI. If you are using the DB2 Information Center, you can also access HTML versions of the source code for the sample programs.

Table 24. Application development information

Name	Form number	PDF file name
IBM DB2 Universal Database Application Development Guide: Building and Running Applications	SC09-4825	db2axx81
IBM DB2 Universal Database Application Development Guide: Programming Client Applications	SC09-4826	db2a1x81
IBM DB2 Universal Database Application Development Guide: Programming Server Applications	SC09-4827	db2a2x81
IBM DB2 Universal Database Call Level Interface Guide and Reference, Volume 1	SC09-4849	db2l1x81
IBM DB2 Universal Database Call Level Interface Guide and Reference, Volume 2	SC09-4850	db2l2x81
IBM DB2 Universal Database Data Warehouse Center Application Integration Guide	SC27-1124	db2adx81
IBM DB2 XML Extender Administration and Programming	SC27-1234	db2sxx81

Business intelligence information

The information in these books describes how to use components that enhance the data warehousing and analytical capabilities of DB2 Universal Database.

Table 25. Business intelligence information

Name	Form number	PDF file name
IBM DB2 Warehouse Manager Standard Edition Information Catalog Center Administration Guide	SC27-1125	db2dix81
IBM DB2 Warehouse Manager Standard Edition Installation Guide	GC27-1122	db2idx81
IBM DB2 Warehouse Manager Standard Edition Managing ETI Solution Conversion Programs with DB2 Warehouse Manager	SC18-7727	iwhe1mstx80

DB2 Connect information

The information in this category describes how to access data on mainframe and midrange servers using DB2 Connect Enterprise Edition or DB2 Connect Personal Edition.

Table 26. DB2 Connect information

Name	Form number	PDF file name
IBM Connectivity Supplement	No form number	db2h1x81
IBM DB2 Connect Quick Beginnings for DB2 Connect Enterprise Edition	GC09-4833	db2c6x81
IBM DB2 Connect Quick Beginnings for DB2 Connect Personal Edition	GC09-4834	db2c1x81
IBM DB2 Connect User's Guide	SC09-4835	db2c0x81

Getting started information

The information in this category is useful when you are installing and configuring servers, clients, and other DB2 products.

Table 27. Getting started information

Name	Form number	PDF file name
IBM DB2 Universal Database Quick Beginnings for DB2 Clients	GC09-4832, not available in hardcopy	db2itx81
IBM DB2 Universal Database Quick Beginnings for DB2 Servers	GC09-4836	db2isx81
IBM DB2 Universal Database Quick Beginnings for DB2 Personal Edition	GC09-4838	db2i1x81
IBM DB2 Universal Database Installation and Configuration Supplement	GC09-4837, not available in hardcopy	db2iyx81
IBM DB2 Universal Database Quick Beginnings for DB2 Data Links Manager	GC09-4829	db2z6x81

Tutorial information

Tutorial information introduces DB2 features and teaches how to perform various tasks.

Table 28. Tutorial information

Name	Form number	PDF file name
Business Intelligence Tutorial: Introduction to the Data Warehouse	No form number	db2tux81
Business Intelligence Tutorial: Extended Lessons in Data Warehousing	No form number	db2tax81
Information Catalog Center Tutorial	No form number	db2aix81
Video Central for e-business Tutorial	No form number	db2twx81
Visual Explain Tutorial	No form number	db2tvx81

Optional component information

The information in this category describes how to work with optional DB2 components.

Table 29. Optional component information

Name	Form number	PDF file name
IBM DB2 Cube Views Guide and Reference	SC18–7298	db2aax81
IBM DB2 Query Patroller Guide: Installation, Administration and Usage Guide	GC09–7658	db2dwx81
IBM DB2 Spatial Extender and Geodetic Extender User's Guide and Reference	SC27-1226	db2sbx81
IBM DB2 Universal Database Data Links Manager Administration Guide and Reference	SC27-1221	db2z0x82
DB2 Net Search Extender Administration and User's Guide **Note:** HTML for this document is *not* installed from the HTML documentation CD.	SH12-6740	N/A

Release notes

The release notes provide additional information specific to your product's release and FixPak level. The release notes also provide summaries of the documentation updates incorporated in each release, update, and FixPak.

Table 30. Release notes

Name	Form number	PDF file name
DB2 Release Notes	See note.	See note.
DB2 Installation Notes	Available on product CD-ROM only.	Not available.

Note: The Release Notes are available in:

- XHTML and Text format, on the product CDs
- PDF format, on the PDF Documentation CD

In addition the portions of the Release Notes that discuss *Known Problems and Workarounds* and *Incompatibilities Between Releases* also appear in the DB2 Information Center.

To view the Release Notes in text format on UNIX-based platforms, see the Release.Notes file. This file is located in the DB2DIR/Readme/*%L* directory, where `%L` represents the locale name and `DB2DIR` represents:

- For AIX operating systems: /usr/opt/db2_08_01
- For all other UNIX-based operating systems: /opt/IBM/db2/V8.1

Related tasks:

- "Printing DB2 books from PDF files" on page 318
- "Ordering printed DB2 books" on page 318
- "Invoking contextual help from a DB2 tool" on page 319

Printing DB2 books from PDF files

You can print DB2 books from the PDF files on the *DB2 PDF Documentation* CD. Using Adobe Acrobat Reader, you can print either the entire book or a specific range of pages.

Prerequisites:

Ensure that you have Adobe Acrobat Reader installed. If you need to install Adobe Acrobat Reader, it is available from the Adobe Web site at www.adobe.com

Procedure:

To print a DB2 book from a PDF file:

1. Insert the *DB2 PDF Documentation* CD. On UNIX operating systems, mount the DB2 PDF Documentation CD. Refer to your *Quick Beginnings* book for details on how to mount a CD on UNIX operating systems.
2. Open index.htm. The file opens in a browser window.
3. Click on the title of the PDF you want to see. The PDF will open in Acrobat Reader.
4. Select **File** → **Print** to print any portions of the book that you want.

Related concepts:

- "DB2 Information Center" on page 303

Related tasks:

- "Ordering printed DB2 books" on page 318

Related reference:

- "DB2 PDF and printed documentation" on page 313

Ordering printed DB2 books

If you prefer to use hardcopy books, you can order them in one of three ways.

Procedure:

Printed books can be ordered in some countries or regions. Check the IBM Publications website for your country or region to see if this service is available in your country or region. When the publications are available for ordering, you can:

- Contact your IBM authorized dealer or marketing representative. To find a local IBM representative, check the IBM Worldwide Directory of Contacts at www.ibm.com/planetwide
- Phone `1-800-879-2755` in the United States or `1-800-IBM-4YOU` in Canada.

- Visit the IBM Publications Center at http://www.ibm.com/shop/publications/order. The ability to order books from the IBM Publications Center may not be available in all countries.

At the time the DB2 product becomes available, the printed books are the same as those that are available in PDF format on the *DB2 PDF Documentation CD*. Content in the printed books that appears in the *DB2 Information Center CD* is also the same. However, there is some additional content available in DB2 Information Center CD that does not appear anywhere in the PDF books (for example, SQL Administration routines and HTML samples). Not all books available on the DB2 PDF Documentation CD are available for ordering in hardcopy.

Note: The DB2 Information Center is updated more frequently than either the PDF or the hardcopy books; install documentation updates as they become available or refer to the DB2 Information Center at http://publib.boulder.ibm.com/infocenter/db2help/ to get the most current information.

Related tasks:

- "Printing DB2 books from PDF files" on page 318

Related reference:

- "DB2 PDF and printed documentation" on page 313

Invoking contextual help from a DB2 tool

Contextual help provides information about the tasks or controls that are associated with a particular window, notebook, wizard, or advisor. Contextual help is available from DB2 administration and development tools that have graphical user interfaces. There are two types of contextual help:

- Help accessed through the **Help** button that is located on each window or notebook
- Infopops, which are pop-up information windows displayed when the mouse cursor is placed over a field or control, or when a field or control is selected in a window, notebook, wizard, or advisor and F1 is pressed.

The **Help** button gives you access to overview, prerequisite, and task information. The infopops describe the individual fields and controls.

Procedure:

To invoke contextual help:

- For window and notebook help, start one of the DB2 tools, then open any window or notebook. Click the **Help** button at the bottom right corner of the window or notebook to invoke the contextual help.

  You can also access the contextual help from the **Help** menu item at the top of each of the DB2 tools centers.

  Within wizards and advisors, click on the Task Overview link on the first page to view contextual help.
- For infopop help about individual controls on a window or notebook, click the control, then click **F1**. Pop-up information containing details about the control is displayed in a yellow window.

Note: To display infopops simply by holding the mouse cursor over a field or control, select the **Automatically display infopops** check box on the **Documentation** page of the Tool Settings notebook.

Similar to infopops, diagnosis pop-up information is another form of context-sensitive help; they contain data entry rules. Diagnosis pop-up information is displayed in a purple window that appears when data that is not valid or that is insufficient is entered. Diagnosis pop-up information can appear for:

– Compulsory fields.

– Fields whose data follows a precise format, such as a date field.

Related tasks:

- "Invoking the DB2 Information Center" on page 311
- "Invoking message help from the command line processor" on page 320
- "Invoking command help from the command line processor" on page 320
- "Invoking SQL state help from the command line processor" on page 321

Invoking message help from the command line processor

Message help describes the cause of a message and describes any action you should take in response to the error.

Procedure:

To invoke message help, open the command line processor and enter:

```
? XXXnnnnn
```

where *XXXnnnnn* represents a valid message identifier.

For example, `? SQL30081` displays help about the SQL30081 message.

Related tasks:

- "Invoking contextual help from a DB2 tool" on page 319
- "Invoking the DB2 Information Center" on page 311
- "Invoking command help from the command line processor" on page 320
- "Invoking SQL state help from the command line processor" on page 321

Invoking command help from the command line processor

Command help explains the syntax of commands in the command line processor.

Procedure:

To invoke command help, open the command line processor and enter:

```
? command
```

where *command* represents a keyword or the entire command.

For example, `? catalog` displays help for all of the CATALOG commands, while `? catalog database` displays help only for the CATALOG DATABASE command.

Related tasks:

- “Invoking contextual help from a DB2 tool” on page 319
- “Invoking the DB2 Information Center” on page 311
- “Invoking message help from the command line processor” on page 320
- “Invoking SQL state help from the command line processor” on page 321

Invoking SQL state help from the command line processor

DB2 Univerrsal Database returns an SQLSTATE value for conditions that could be the result of an SQL statement. SQLSTATE help explains the meanings of SQL states and SQL state class codes.

Procedure:

To invoke SQL state help, open the command line processor and enter:

```
? sqlstate or ? class code
```

where *sqlstate* represents a valid five-digit SQL state and *class code* represents the first two digits of the SQL state.

For example, `? 08003` displays help for the 08003 SQL state, and `? 08` displays help for the 08 class code.

Related tasks:

- “Invoking the DB2 Information Center” on page 311
- “Invoking message help from the command line processor” on page 320
- “Invoking command help from the command line processor” on page 320

DB2 tutorials

The DB2® tutorials help you learn about various aspects of DB2 Universal Database. The tutorials provide lessons with step-by-step instructions in the areas of developing applications, tuning SQL query performance, working with data warehouses, managing metadata, and developing Web services using DB2.

Before you begin:

You can view the XHTML versions of the tutorials from the Information Center at http://publib.boulder.ibm.com/infocenter/db2help/.

Some tutorial lessons use sample data or code. See each tutorial for a description of any prerequisites for its specific tasks.

DB2 Universal Database tutorials:

Click on a tutorial title in the following list to view that tutorial.

Business Intelligence Tutorial: Introduction to the Data Warehouse Center
Perform introductory data warehousing tasks using the Data Warehouse Center.

Business Intelligence Tutorial: Extended Lessons in Data Warehousing
Perform advanced data warehousing tasks using the Data Warehouse Center.

Information Catalog Center Tutorial

Create and manage an information catalog to locate and use metadata using the Information Catalog Center.

Visual Explain Tutorial

Analyze, optimize, and tune SQL statements for better performance using Visual Explain.

DB2 troubleshooting information

A wide variety of troubleshooting and problem determination information is available to assist you in using DB2® products.

DB2 documentation

Troubleshooting information can be found throughout the DB2 Information Center, as well as throughout the PDF books that make up the DB2 library. You can refer to the ″Support and troubleshooting″ branch of the DB2 Information Center navigation tree (in the left pane of your browser window) to see a complete listing of the DB2 troubleshooting documentation.

DB2 Technical Support Web site

Refer to the DB2 Technical Support Web site if you are experiencing problems and want help finding possible causes and solutions. The Technical Support site has links to the latest DB2 publications, TechNotes, Authorized Program Analysis Reports (APARs), FixPaks and the latest listing of internal DB2 error codes, and other resources. You can search through this knowledge base to find possible solutions to your problems.

Access the DB2 Technical Support Web site at http://www.ibm.com/software/data/db2/udb/winos2unix/support

DB2 Problem Determination Tutorial Series

Refer to the DB2 Problem Determination Tutorial Series Web site to find information on how to quickly identify and resolve problems you might encounter while working with DB2 products. One tutorial introduces you to the DB2 problem determination facilities and tools available, and helps you decide when to use them. Other tutorials deal with related topics, such as ″Database Engine Problem Determination″, ″Performance Problem Determination″, and ″Application Problem Determination″.

See the full set of DB2 problem determination tutorials on the DB2 Technical Support site at http://www.ibm.com/software/data/support/pdm/db2tutorials.html

Related concepts:

- "DB2 Information Center" on page 303

Accessibility

Accessibility features help users with physical disabilities, such as restricted mobility or limited vision, to use software products successfully. The following list specifies the major accessibility features in DB2® Version 8 products:

- All DB2 functionality is available using the keyboard for navigation instead of the mouse. For more information, see "Keyboard input and navigation" on page 323.

- You can customize the size and color of the fonts on DB2 user interfaces. For more information, see “Accessible display.”
- DB2 products support accessibility applications that use the Java™ Accessibility API. For more information, see “Compatibility with assistive technologies.”
- DB2 documentation is provided in an accessible format. For more information, see “Accessible documentation.”

Keyboard input and navigation

Keyboard input

You can operate the DB2 tools using only the keyboard. You can use keys or key combinations to perform operations that can also be done using a mouse. Standard operating system keystrokes are used for standard operating system operations.

For more information about using keys or key combinations to perform operations, see Keyboard shortcuts and accelerators: Common GUI help.

Keyboard navigation

You can navigate the DB2 tools user interface using keys or key combinations.

For more information about using keys or key combinations to navigate the DB2 Tools, see Keyboard shortcuts and accelerators: Common GUI help.

Keyboard focus

In UNIX® operating systems, the area of the active window where your keystrokes will have an effect is highlighted.

Accessible display

The DB2 tools have features that improve accessibility for users with low vision or other visual impairments. These accessibility enhancements include support for customizable font properties.

Font settings

You can select the color, size, and font for the text in menus and dialog windows, using the Tools Settings notebook.

For more information about specifying font settings, see Changing the fonts for menus and text: Common GUI help.

Non-dependence on color

You do not need to distinguish between colors in order to use any of the functions in this product.

Compatibility with assistive technologies

The DB2 tools interfaces support the Java Accessibility API, which enables you to use screen readers and other assistive technologies with DB2 products.

Accessible documentation

Documentation for DB2 is provided in XHTML 1.0 format, which is viewable in most Web browsers. XHTML allows you to view documentation according to the display preferences set in your browser. It also allows you to use screen readers and other assistive technologies.

Syntax diagrams are provided in dotted decimal format. This format is available only if you are accessing the online documentation using a screen-reader.

Related concepts:

- "Dotted decimal syntax diagrams" on page 324

Dotted decimal syntax diagrams

Syntax diagrams are provided in dotted decimal format for users accessing the Information Center using a screen reader.

In dotted decimal format, each syntax element is written on a separate line. If two or more syntax elements are always present together (or always absent together), they can appear on the same line, because they can be considered as a single compound syntax element.

Each line starts with a dotted decimal number; for example, 3 or 3.1 or 3.1.1. To hear these numbers correctly, make sure that your screen reader is set to read out punctuation. All the syntax elements that have the same dotted decimal number (for example, all the syntax elements that have the number 3.1) are mutually exclusive alternatives. If you hear the lines 3.1 USERID and 3.1 SYSTEMID, you know that your syntax can include either USERID or SYSTEMID, but not both.

The dotted decimal numbering level denotes the level of nesting. For example, if a syntax element with dotted decimal number 3 is followed by a series of syntax elements with dotted decimal number 3.1, all the syntax elements numbered 3.1 are subordinate to the syntax element numbered 3.

Certain words and symbols are used next to the dotted decimal numbers to add information about the syntax elements. Occasionally, these words and symbols might occur at the beginning of the element itself. For ease of identification, if the word or symbol is a part of the syntax element, it is preceded by the backslash (\) character. The * symbol can be used next to a dotted decimal number to indicate that the syntax element repeats. For example, syntax element *FILE with dotted decimal number 3 is given the format 3 * FILE. Format 3* FILE indicates that syntax element FILE repeats. Format 3* * FILE indicates that syntax element * FILE repeats.

Characters such as commas, which are used to separate a string of syntax elements, are shown in the syntax just before the items they separate. These characters can appear on the same line as each item, or on a separate line with the same dotted decimal number as the relevant items. The line can also show another symbol giving information about the syntax elements. For example, the lines 5.1*, 5.1 LASTRUN, and 5.1 DELETE mean that if you use more than one of the LASTRUN and DELETE syntax elements, the elements must be separated by a comma. If no separator is given, assume that you use a blank to separate each syntax element.

If a syntax element is preceded by the % symbol, this indicates a reference that is defined elsewhere. The string following the % symbol is the name of a syntax fragment rather than a literal. For example, the line 2.1 %OP1 means that you should refer to separate syntax fragment OP1.

The following words and symbols are used next to the dotted decimal numbers:

- ? means an optional syntax element. A dotted decimal number followed by the ? symbol indicates that all the syntax elements with a corresponding dotted decimal number, and any subordinate syntax elements, are optional. If there is only one syntax element with a dotted decimal number, the ? symbol is displayed on the same line as the syntax element, (for example 5? NOTIFY). If there is more than one syntax element with a dotted decimal number, the ? symbol is displayed on a line by itself, followed by the syntax elements that are optional. For example, if you hear the lines 5 ?, 5 NOTIFY, and 5 UPDATE, you know that syntax elements NOTIFY and UPDATE are optional; that is, you can choose one or none of them. The ? symbol is equivalent to a bypass line in a railroad diagram.
- ! means a default syntax element. A dotted decimal number followed by the ! symbol and a syntax element indicates that the syntax element is the default option for all syntax elements that share the same dotted decimal number. Only one of the syntax elements that share the same dotted decimal number can specify a ! symbol. For example, if you hear the lines 2? FILE, 2.1! (KEEP), and 2.1 (DELETE), you know that (KEEP) is the default option for the FILE keyword. In this example, if you include the FILE keyword but do not specify an option, default option KEEP will be applied. A default option also applies to the next higher dotted decimal number. In this example, if the FILE keyword is omitted, default FILE(KEEP) is used. However, if you hear the lines 2? FILE, 2.1, 2.1.1! (KEEP), and 2.1.1 (DELETE), the default option KEEP only applies to the next higher dotted decimal number, 2.1 (which does not have an associated keyword), and does not apply to 2? FILE. Nothing is used if the keyword FILE is omitted.
- * means a syntax element that can be repeated 0 or more times. A dotted decimal number followed by the * symbol indicates that this syntax element can be used zero or more times; that is, it is optional and can be repeated. For example, if you hear the line 5.1* data area, you know that you can include one data area, more than one data area, or no data area. If you hear the lines 3*, 3 HOST, and 3 STATE, you know that you can include HOST, STATE, both together, or nothing.

  Notes:

  1. If a dotted decimal number has an asterisk (*) next to it and there is only one item with that dotted decimal number, you can repeat that same item more than once.
  2. If a dotted decimal number has an asterisk next to it and several items have that dotted decimal number, you can use more than one item from the list, but you cannot use the items more than once each. In the previous example, you could write HOST STATE, but you could not write HOST HOST.
  3. The * symbol is equivalent to a loop-back line in a railroad syntax diagram.
- + means a syntax element that must be included one or more times. A dotted decimal number followed by the + symbol indicates that this syntax element must be included one or more times; that is, it must be included at least once and can be repeated. For example, if you hear the line 6.1+ data area, you must include at least one data area. If you hear the lines 2+, 2 HOST, and 2 STATE, you know that you must include HOST, STATE, or both. Similar to the * symbol, the + symbol can only repeat a particular item if it is the only item with that dotted decimal number. The + symbol, like the * symbol, is equivalent to a loop-back line in a railroad syntax diagram.

Related reference:

- “How to read the syntax diagrams” in the *SQL Reference, Volume 2*

Common Criteria certification of DB2 Universal Database products

DB2 Universal Database is being evaluated for certification under the Common Criteria at evaluation assurance level 4 (EAL4). For more information about Common Criteria, see the Common Criteria web site at: http://niap.nist.gov/cc-scheme/.

Notices

IBM may not offer the products, services, or features discussed in this document in all countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country/region or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country/region where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product, and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information that has been exchanged, should contact:

IBM Canada Limited
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario
L6G 1C7
CANADA

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems, and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements, or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious, and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information may contain sample application programs, in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (*your company name*) (*year*). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _*enter the year or years*_. All rights reserved.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both, and have been used in at least one of the documents in the DB2 UDB documentation library.

ACF/VTAM
AISPO
AIX
AIXwindows
AnyNet
APPN
AS/400
BookManager
C Set++
C/370
CICS
Database 2
DataHub
DataJoiner
DataPropagator
DataRefresher
DB2
DB2 Connect
DB2 Extenders
DB2 OLAP Server
DB2 Information Integrator
DB2 Query Patroller
DB2 Universal Database
Distributed Relational
  Database Architecture
DRDA
eServer
Extended Services
FFST
First Failure Support Technology
IBM
IMS
IMS/ESA

iSeries
LAN Distance
MVS
MVS/ESA
MVS/XA
Net.Data
NetView
OS/390
OS/400
PowerPC
pSeries
QBIC
QMF
RACF
RISC System/6000
RS/6000
S/370
SP
SQL/400
SQL/DS
System/370
System/390
SystemView
Tivoli
VisualAge
VM/ESA
VSE/ESA
VTAM
WebExplorer
WebSphere
WIN-OS/2
z/OS
zSeries

The following terms are trademarks or registered trademarks of other companies and have been used in at least one of the documents in the DB2 UDB documentation library:

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Query Patroller Glossary

A

administrator. A person responsible for administrative tasks such as access authorization and content management. Administrators can also grant levels of authority to users.

B

bypass. Refers to when a query is allowed to run without being managed by Query Patroller.

C

charge-back account. An account used for tracking computing resource usage by business departments or projects. Examples of charge-back accounts are ″Marketing Department″ or ″Summer catalog project″.

cost. The estimated total resource usage that is necessary to run the access plan for a statement (or the elements of a statement). Cost is derived from a combination of processor cost (in number of instructions) and I/O (in number of seeks and page transfers). See also "timeron" on page 332.

E

elapsed queued time. In Query Patroller, the length of time that elapses between the time a query is created and the time a query starts to run. See also "elapsed total time."

elapsed total time. In Query Patroller, the combined total of the elapsed queued time and the elapsed run time for a query. See also "elapsed queued time" and "execution time."

execution time. The elapsed run time of a query. This is the time between the start and the end of the query execution. See also "user time" on page 332 and "system time" on page 332.

H

held query. A query that has been prevented from running by Query Patroller because its estimated cost is higher than a Query Patroller threshold. The query is held until released from the held state by Query Patroller itself or by a user with sufficient authority, such as an administrator. See also "intercepted query" and "managed query."

historical analysis. In Query Patroller, the activity of analyzing past usage of a data warehouse. This is done for reasons including: management reporting, tuning, and optimizing Query Patroller thresholds.

I

intercepted query. A query that Query Patroller intercepts from another application and that may be either managed or not managed by Query Patroller. Managed queries are a subset of intercepted queries. See also "managed query" and "held query."

M

managed query. A query that is managed by Query Patroller. See also "intercepted query" and "held query."

O

operator. In Query Patroller, a person who has a subset of administrator authorities as defined in their operator profile.

P

profile. In Query Patroller, defines a set of characteristics for each operator or submitter. See also “operator” on page 331 and “submitter.”

Q

queue priority. In Query Patroller, a numerical value that represents the priority assigned to a query when a query is submitted.

queued time. See “elapsed queued time” on page 331.

query. A request for information from the database based on specific conditions.

query class. In Query Patroller, a mechanism that allows queries to be grouped and run according to their size. Query classes allow the flow of queries on a database to be controlled, ensuring that resources are shared among queries in the different size groupings (query classes).

Query Controller. The server component of the Query Patroller system, which works in conjunction with the DB2 server to manage queries against the database.

Query Patroller Center. The graphical user interface for administering Query Patroller, enabling you to manage queries against databases. You can use the Query Patroller Center to monitor and control the flow of queries against a database, administer users, and view historical query information.

query status. In Query Patroller, each of several states that a managed query may pass through. See also “operator” on page 331 and “managed query” on page 331.

query submission preferences. In Query Patroller, a set of characteristics for a submitter that the submitter can view and update. See also “submitter.”

R

result destination. In Query Patroller, where the results of a query are returned. The result destination can be either the application through which the query was submitted or a result table. See also “result table.”

result set. The set of rows that a query or a stored procedure returns.

result table. In Query Patroller, a table created to store the result set of a query.

S

submitter. In the Query Patroller environment, a user given access to submit queries.

system time. In UNIX, represents the time spent in system calls. See also “user time.”

T

timeron. A unit of measurement that is used to give a rough relative estimate of the resources, or cost, required by the database server to execute two plans for the same query. The resources calculated in the estimate include weighted processor and I/O costs. See also “cost” on page 331.

total time. See “elapsed total time” on page 331.

U

user time. In UNIX, represents the time spent executing database manager code. See also “system time.”

Index

Contacting IBM

In the United States, call one of the following numbers to contact IBM:

- 1-800-IBM-SERV (1-800-426-7378) for customer service
- 1-888-426-4343 to learn about available service options
- 1-800-IBM-4YOU (426-4968) for DB2 marketing and sales

In Canada, call one of the following numbers to contact IBM:

- 1-800-IBM-SERV (1-800-426-7378) for customer service
- 1-800-465-9600 to learn about available service options
- 1-800-IBM-4YOU (1-800-426-4968) for DB2 marketing and sales

To locate an IBM office in your country or region, check IBM's Directory of Worldwide Contacts on the web at http://www.ibm.com/planetwide

Product information

Information regarding DB2 Universal Database products is available by telephone or by the World Wide Web at http://www.ibm.com/software/data/db2/udb

This site contains the latest information on the technical library, ordering books, product downloads, newsgroups, FixPaks, news, and links to web resources.

If you live in the U.S.A., then you can call one of the following numbers:

- 1-800-IBM-CALL (1-800-426-2255) to order products or to obtain general information.
- 1-800-879-2755 to order publications.

For information on how to contact IBM outside of the United States, go to the IBM Worldwide page at www.ibm.com/planetwide

Part Number: CT2TCNA

Printed in Ireland

(1P) P/N: CT2TCNA

GC09-7658-01

Spine information:

IBM® DB2 Query Patroller™

DB2 Query Patroller Guide

Version 8.2