

IBM DB2 Query Patroller



Administration Guide

Version 6

SC09-2859-00

IBM DB2 Query Patroller



Administration Guide

Version 6

SC09-2859-00

Before using this information and the product it supports, be sure to read the general information under “Appendix D. Notices” on page 113.

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

Order publications through your IBM representative or the IBM branch office serving your locality or by calling 1-800-879-2755 in the United States or 1-800-IBM-4YOU in Canada.

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

© **Copyright International Business Machines Corporation 1999. All rights reserved.**

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

Contents

About This Book	vii
Who Should Use This Book	vii
Conventions	vii
Related Documentation	viii

Part 1. DB2 Query Patroller Operations 1

Chapter 1. Introduction	3
Syntax Conventions	3
DB2 Query Patroller Components	4
Server	4
Agent	6
Command-Line Interface	6
QueryAdministrator.	6
Tracker	6
QueryEnabler	7
QueryMonitor.	7
DB2 Query Patroller Tables	7
Connections	8
Job Flow	9
Chapter 2. Administration	11
Administration Overview	11
Parameters	11
Environment Variables	11
System Parameters	13
Data Source Parameters	13
Job Queue Parameters	14
User Profile Parameters	14
Administrative Tasks	15
Maintaining Cost Statistics	15
Adding Users	15
Chapter 3. Operation	17
Starting and Stopping DB2 Query Patroller	17
Controlling Job Flow and Execution	17
Releasing and Canceling Jobs	18
Monitoring the System	18
The Control Program (iwm)	18
Chapter 4. Using the Command-Line Interface	21
Command-Line Interface Environment	21

Submit Command (iwm_submit)	22
Control Command (iwm_cmd)	25
Common Parameters	27
Monitoring Jobs	27
Controlling Jobs	29
Monitoring and Controlling Nodes.	30
Listing and Controlling Data Sources	31
Listing and Controlling System Parameters	33

Chapter 5. Error Log Monitoring	35
Running the Log Monitor	35
Log Monitor Configuration File	35
INTERVAL Command	36
LOGFILE Command	36
MAILTO Command	37
INCLUDE and EXCLUDE Commands	37

Chapter 6. Job Accounting 39

Chapter 7. Customizing	43
Exit Analysis	43
Result Destinations	43
Result Destination Variables	44
Piped Result Destinations	44
E-Mail Messages	44

Chapter 8. Configuring the DB2 Query Patroller Server	45
Profiling.	45
Estimating and Monitoring	45
Modifications	46
Configuring the Data Warehouse	46

Chapter 9. Configuring the DB2 Query Patroller Client	47
Configuring QueryEnabler	47
Client Configuration Options.	48

Part 2. The QueryAdministrator Utility 49

Chapter 10. QueryAdministrator	51
About QueryAdministrator	51

Starting QueryAdministrator	51
Stopping QueryAdministrator	52
Displaying Columns	52
Resizing Columns	53

Chapter 11. Data Source Administration	55
Listing Data Sources	55
Editing a Data Source Definition	55

Chapter 12. User Administration	59
Listing User Profiles.	59
Creating a User Profile	59
Editing a User Profile	61
Removing a User Profile	61

Chapter 13. Node Administration	63
Listing All Nodes	63
Changing the Node Status	63

Chapter 14. Job Queue Administration	65
Viewing Job Queues.	65
Creating a Job Queue	65
Editing a Job Queue.	66
Removing a Job Queue.	66

Chapter 15. Result Destination Administration	67
Listing Result Destinations	67
Creating a Result Destination	67
Editing a Result Destination	68
Removing a Result Destination	68

Chapter 16. System Administration	71
Listing System Parameters	71
Editing System Parameters	71

Part 3. The Tracker Utility 73

Chapter 17. Introduction to Tracker	75
Creating Tracker Data on the Database Server	75
Navigating within Tracker.	76
Understanding the Tool Bar	76

Chapter 18. Tracker Reports	79
Understanding the Reports	79
Displaying Data as a Count	80
Displaying Data as a Percentage	80
Total Table Hits Report.	80

Columns Hit in Table X Report	80
Users That Hit Table X Report	80
Columns Hit by User Y Hitting Table X Report	81
Tables Joined with Table X Report	81
Rows Returned by Table Report.	81
Job Activity Over Time Report	81

Chapter 19. Using Tracker	83
Starting Tracker	83
Displaying Job History Data	84
Opening a New Report	84
Changing the Date and Time Range	84
Refreshing Data	86
Report Formats	86
Drilling Down for More Information	87
Drilling Up.	87
Viewing the Drill Path	88
Zooming and Panning	88
Closing a Tracker Report Window	89
Exiting Tracker	89

Part 4. Appendixes 91

Appendix A. Troubleshooting the DB2 Query Patroller Server	93
Distinguishing Between DB2 and DB2 Query Patroller Failures	93
syserr.log	93
DB2 Diagnostic Log	93
Processes	94
DB2 Query Patroller Server	94
DB2 Query Patroller Agent	94
Process Failures	94
Common Server Problems.	95

Appendix B. Troubleshooting DB2 Query Patroller Clients.	97
Troubleshooting QueryEnabler	97
Common QueryEnabler Problems	98
Troubleshooting QueryMonitor and QueryAdministrator.	99
Common QueryMonitor and QueryAdministrator Problems	99
Troubleshooting Tracker	99
Common Tracker Problems	100

Appendix C. DB2 Query Patroller Messages	101
---	------------

Messages	101
Log File	110
Appendix D. Notices	113
Trademarks	114
Trademarks of Other Companies	114

Index	117
Contacting IBM	121

About This Book

This book provides information on administering and operating DB2 Query Patroller. It also includes instructions for using the Tracker utility and guidelines for system configuration and troubleshooting.

Who Should Use This Book

This Guide is intended for those who need to administer, operate, and monitor the DB2 Query Patroller System. The audience generally consists of any person involved in day-to-day operations of DB2 Query Patroller. In many organizations, the system administrator or the database administrator is responsible for these tasks.

The book is organized in four parts. Use the table below to help you quickly find the part of the book you are interested in:

Table 1. How the DB2 Query Patrol Administration Guide is Organized

Part	Description
Part 1, DB2 Query Patroller Operations	This section provides an overview of DB2 Query Patroller and its components, and offers information and procedures for running and maintaining the system.
Part 2, The QueryAdministrator Utility	This section covers all aspects of using the QueryAdministrator utility. The QueryAdministrator utility enables the system administrator to manage the DB2 Query Patroller system parameters, create or delete profiles for DB2 Query Patroller authorized users, manage nodes and result destinations, and manage data sources.
Part 3, The Tracker Utility	The Tracker utility provides reports that display database usage history for queries that have been managed by the DB2 Query Patroller system.
Part 4, Appendixes	The appendixes for this guide include a section on troubleshooting the DB2 Query Patroller system and a section on DB2 Query Patroller error messages.

Conventions

This guide uses these highlighting conventions:

- **Boldface** indicates commands or graphical user interface (GUI) controls such as names of fields, folders, icons, or menu choices.

- *Italics* indicates variables that you should replace with your own value. It is also used to indicate book titles and to emphasize words.
- Monospace indicates examples of text you enter exactly as shown.

Related Documentation

For additional information about DB2 Query Patroller refer to the following documents:

- *DB2 Query Patroller Installation Guide*
- *DB2 Query Patroller User's Guide*

Part 1. DB2 Query Patroller Operations

Chapter 1. Introduction

This part of the book provides information and procedures for running and maintaining the DB2 Query Patroller system. The following chapter provides an overview of the system and its components, and gives you a step by step explanation of how DB2 Query Patroller handles and processes jobs.

Syntax Conventions

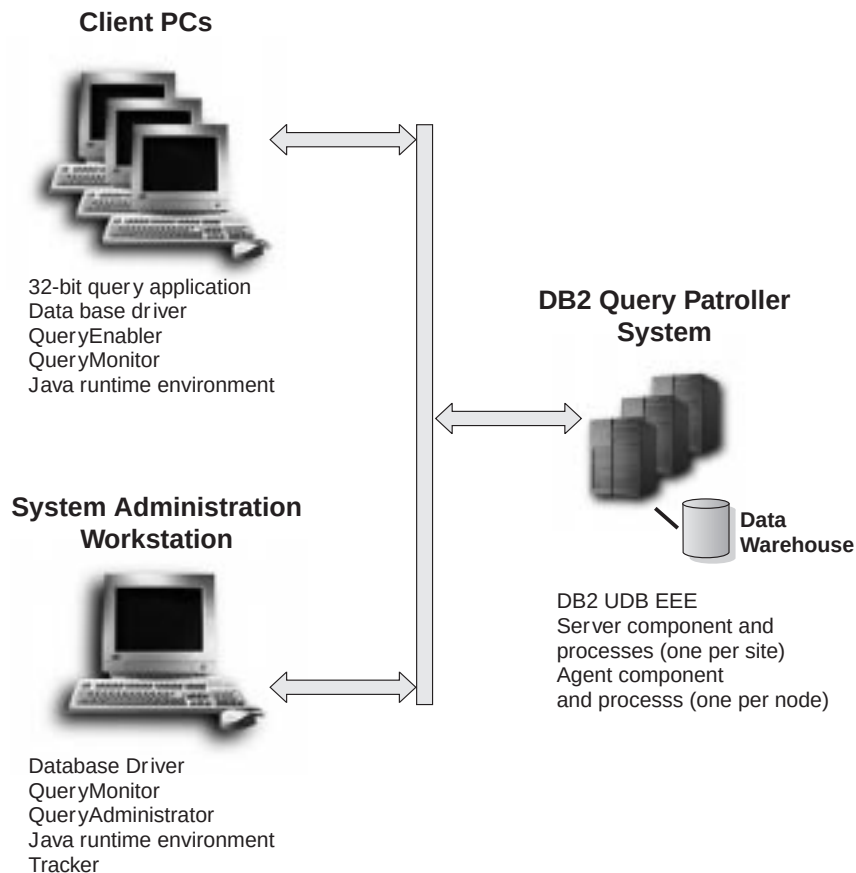
This book uses syntax diagrams as well as the syntax parameters described in the table below. Consider parameters in the table to be case sensitive unless the description of the parameter states otherwise.

Table 2. Syntax Conventions

Convention	Description
{ }	Brace brackets contain multiple choices separated by the pipe () symbol. You can only select one of the values within the brace brackets. For example, the syntax {yes no} indicates that you can select either the yes or no value, but not both.
	The pipe symbol separates multiple choices found within brace brackets.
[]	Square brackets contain optional parameters.
<i>Italics</i>	<i>Italics</i> are used to note a variable parameter that is acting as a place holder. You must replace it with a specific value when you issue the command. For example, <i>node_id</i> represents a variable for the node ID; you must supply a specific node ID when you enter the command.
...	An ellipsis indicates that the previous value may be repeated.

DB2 Query Patroller Components

The DB2 Query Patroller system consists of the DB2 Query Patroller server system, the system administrator's workstation, and client PCs. The diagram below illustrates the architecture of the DB2 Query Patroller system and the product components on each platform:



DB2 Query Patroller consists of the following components:

Server

The server executes on a single database node. The server accepts, analyzes, prioritizes, and schedules database requests and optionally notifies users when their requests have been processed.

The server consists of the following software components:

- **request server**

The request server provides services for requests received from the various DB2 Query Patroller components.

- **cost analyzer**

The cost analyzer component determines the relative cost to execute a query based upon explain plan output. The cost analyzer executes one time per query. The job scheduler uses the calculated cost at a later time to determine when the query should be run.

The processing described in the previous paragraph determines if DB2 Query Patroller should proceed with query execution or put the query on hold for later manual intervention. DB2 Query Patroller assigns a hold status to a query if the computed cost exceeds the user's cost threshold as defined in the User Profile table. (For information about the User Profile table, see "DB2 Query Patroller Tables" on page 7.) The notifier component of DB2 Query Patroller notifies the user in the event that a query has been placed on hold.

- **job scheduler**

The job scheduler schedules the query to a node for execution. DB2 Query Patroller uses a proprietary scheduling technique that takes into account:

- Current number of queries executing on the system
- Cost of all queries currently executing
- Nodes assigned to the DB2 Query Patroller system
- Individual user priorities
- Number of queries executing for each user

- **notifier**

The notifier scans the DB2 Query Patroller tables for newly completed queries at a user-specified time interval. It notifies the user through system mail when the query completes. This component also adds a row to the Job Accounting table for each job.

The following list outlines the query completion status returned to the user through the mail system:

- **Done** (complete)
- **Aborted** (an abnormal termination has occurred)
- **Hold** (the query has been placed on hold)
- **Canceled** (the query was canceled)

- **log monitor**

The server and agent components write error information to a log file. You may want to consider using a product that consolidates system status information, including the DB2 Query Patroller log file. If you do not have such a product, use the DB2 Query Patroller log monitor to send new log entries to interested parties through e-mail.

Agent

The agent executes on one or more database nodes and processes database requests. On a single-processor or non-clustered SMP machine, the agent and server components run on the same machine. On an MPP machine, or clustered SMP machines, the server component runs on one node and the agent(s) may run on several nodes, including the same node as the server component.

The DB2 Query Patroller agent consists of the following software components:

- **node manager**

The node manager executes any work scheduled to its node by the job scheduler.

- **executor**

The executor component runs SQL statements and system commands.

Command-Line Interface

The command-line interface enables the system administrator to monitor and control DB2 Query Patroller from a command-line prompt. This feature also enables you to submit and monitor jobs from the command line. When combined with shell scripts or languages such as Perl, awk, and REXX, you can use the command-line interface as an application programming interface (API) to DB2 Query Patroller.

QueryAdministrator

QueryAdministrator provides an interface for administering the DB2 Query Patroller system. QueryAdministrator is a Java application that runs on Windows 32-bit operating systems.

QueryAdministrator enables the system administrator to manage the DB2 Query Patroller system parameters, create or delete profiles for DB2 Query Patroller users, manage nodes, result destinations, data sources and job queues.

Tracker

The Tracker utility provides reports that display a database usage history for queries that have been managed by the DB2 Query Patroller system. For example, Tracker allows you to determine which tables and columns have been accessed most frequently, which tables have returned the most result rows, and which jobs have been completed within a specific time period.

QueryEnabler

QueryEnabler places queries submitted through an ODBC-compliant query tool under the management of the DB2 Query Patroller system. With QueryEnabler, you can wait for the results to return, or you can have the results returned at a later time. This frees your workstation to submit other queries or perform other work. QueryEnabler runs on Windows 32-bit operating systems.

QueryMonitor

QueryMonitor provides an interface for monitoring queries that have been submitted to the DB2 Query Patroller system. QueryMonitor is a Java application that runs on Windows 32-bit operating systems.

QueryMonitor enables the user to monitor query status, view job details, cancel jobs, submit new jobs, drop result tables, and resubmit completed jobs.

DB2 Query Patroller Tables

The following list explains the tables used by DB2 Query Patroller. The DB2 Query Patroller installation creates these tables in the database.

- **Job**

Various components of DB2 Query Patroller use the Job table to monitor and track each job under the control of DB2 Query Patroller. The various DB2 Query Patroller components maintain this table. Users can monitor their jobs, cancel jobs, and drop temporary result tables through the QueryMonitor component. System administrators can monitor any job, change a job's status, manage system and node parameters, and maintain user profiles using QueryAdministrator. The notifier component of DB2 Query Patroller can automatically purge obsolete job information after a period of time specified by the system administrator.

- **User Profile**

The User Profile table stores the user profile information for the DB2 Query Patroller users. A User Profile record must exist in this table before the corresponding user can log into the DB2 Query Patroller system. The system administrator maintains this table using QueryAdministrator.

- **Node Information**

The Node Information table contains information concerning the status of each node in the system. The server and agent components maintain this table. The system administrator can alter the status of any node using the command-line interface or QueryAdministrator.

- **System Parameters**

The System Parameters table contains system-wide information that DB2 Query Patroller uses to control processing. The system administrator maintains this table using the command-line interface or QueryAdministrator.

- **Result Destinations**

Normally, the database stores query results as tables. The Result Destinations table contains the descriptions of alternate destinations for query results. See “Result Destinations” on page 43 for additional information about alternate destinations. The system administrator maintains this table using the command-line interface or QueryAdministrator.

- **Data Sources**

The Data Sources table contains information that DB2 Query Patroller uses to control processing for each database instance. Presently, DB2 Query Patroller controls only one database. The system administrator maintains this table using the command-line interface or QueryAdministrator.

- **Job Queues**

The Job Queues table defines the job queues for each data source (database instance). Queues receive job assignments based on the job’s cost. Each queue can have different limits on the number of jobs run. The system administrator maintains this table using the command-line interface.

- **Job Accounting**

When job accounting status is active, the notifier component of DB2 Query Patroller adds one row to the job accounting table each time a job completes, either successfully or unsuccessfully. The information in this table is used by the Tracker utility.

- **Result Tables**

Each time DB2 Query Patroller executes a job containing a SQL SELECT statement, it normally creates a temporary table containing the results of the query. QueryEnabler retrieves results from these tables and returns the results to the originating query application. QueryEnabler and QueryMonitor can be used to drop result tables. If result tables are not dropped by the user after retrieval, the notifier component of DB2 Query Patroller can automatically purge result tables after a period of time specified by the system administrator.

Connections

The various DB2 Query Patroller components communicate with each other through either the DB2 Query Patroller tables or socket connections. All of the Windows-based components communicate with DB2 Query Patroller through the DB2 Query Patroller tables. The command-line interface, agent component,

and some parts of the server component communicate with the DB2 Query Patroller request server through sockets. To reduce the number of TCP socket connections required, the `iwm_net` process runs on the server node and on the agent nodes. The `iwm_net` processes enable the processes on one node to communicate with each other using TCP/IP sockets and concentrate all communication between nodes to a single outgoing socket connection.

Job Flow

This section describes the flow of a job from the time a query is submitted to DB2 Query Patroller until the time that notification of job completion is received.

1. Job Submission

Queries are typically submitted with 32-bit ODBC query applications. QueryEnabler intercepts these queries and, with user intervention, submits the query to DB2 Query Patroller as a job. Jobs can also be submitted through QueryMonitor and the command-line interface. When you submit a job, the DB2 Query Patroller component used to submit the job saves the SQL statement, along with the user ID, date and time the job is scheduled to be run, and the maximum cost permitted. The User Profile information provides the maximum cost. The component used to submit the job also adds this information to the Job table and assigns a job number.

2. Cost Analysis

After a user submits a job to DB2 Query Patroller, the server component estimates the cost to execute the query. The server component uses the cost estimate from DB2.

If the computed cost is greater than the maximum cost (obtained from the User Profile when the user submitted the job), the job is put on hold. Otherwise, the job runs as soon as the scheduled date and time passes and sufficient system resources are available. If the job is on hold, the DB2 Query Patroller system administrator must release or cancel the job. For jobs submitted through the command-line interface, cost analysis is performed as part of job submission, not as a separate step.

3. Job Scheduling

The server component's job scheduler activates periodically to scan for new jobs to be run. It processes jobs that are ready to run in order of priority. Jobs of equal priority are processed in the order that they were submitted. To determine if a job can run, the scheduler looks to see if the number of queries already running has reached the limit set in the User Profile. The scheduler also looks to see if the total number of queries running or the cost of the new job plus the total cost of any running jobs has reached the limits set for the data source and the system. Next, based on the job's cost, the scheduler checks for an associated job queue and

determines the limit on the number of jobs for the queue. If a job cannot be scheduled due to system limits, the scheduler bypasses the job and processes the next ready job. The scheduler processes any jobs skipped during a scheduling cycle the next time it activates.

If a job can be scheduled, it must be assigned to a node. The job scheduler selects a node based on the limits set for the data source and on the CPU and disk-utilization statistics being gathered by the node managers. The node selected is the one with the fewest number of assigned jobs. If two or more nodes are running the same number of jobs, the job scheduler selects the node with the lowest CPU utilization.

4. Job Execution

The node manager of each agent activates periodically to scan for new jobs scheduled to be run on that node. (For more information, see “DB2 Query Patroller Components” on page 4.) The node manager creates a new process for each job to run the executor. Normally, for a job containing a SQL SELECT statement, the executor component opens a cursor to fetch the result rows. When the database returns the first row, DB2 Query Patroller creates the result table. The executor component uses an insert cursor to efficiently create the result table. When each job completes, the executor component sends the completion status and execution statistics back to the server component to be recorded in the job table.

In addition to running jobs newly scheduled for the node, each time the node manager activates it also collects CPU and disk utilization statistics and checks for jobs to be canceled or that have aborted. Jobs may be canceled any time from submission until they have been scheduled. Database connections for running jobs are removed when a job is canceled.

5. Notification

The server component’s notifier activates periodically to scan for jobs that complete successfully or unsuccessfully, or for any jobs with a held status. This component notifies users of these events through e-mail and adds a row to the Job Accounting table for each completed job when the job accounting status is active. The notifier component also purges expired job table entries and result tables.

Chapter 2. Administration

This chapter provides information about DB2 Query Patroller parameters and system administration tasks.

Administration Overview

You can perform system administration tasks for DB2 Query Patroller through the command-line interface or QueryAdministrator. System administration tasks include setting various DB2 Query Patroller parameters to control its operation, maintaining the cost statistics, and adding users to the system.

Parameters

The following information outlines the DB2 Query Patroller parameters.

Environment Variables

The system administrator can set environment variables to alter the behavior of DB2 Query Patroller. To have an effect, the environment variables must be set for the iwm account before starting DB2 Query Patroller. The *DB2 Query Patroller Installation Guide* provides information about setting DB2 Query Patroller environment variables.

- **IWM_DISKMON**

Set the IWM_DISKMON environment variable to the file system where the result sets are created. Each agent node can monitor the space available in a different file system. When available disk space goes below a system threshold, no more jobs will be submitted to that node. If IWM_DISKMON is not set, disk space is not monitored and creation of result tables will fail when the file system becomes full.

- **IWM_EXIT_AN**

DB2 Query Patroller invokes the exit analysis program if you set IWM_EXIT_AN to the path name of the executable program before starting the server processes. The exit can be a shell script or any other executable. DB2 Query Patroller invokes the exit under the iwm user ID. DB2 Query Patroller passes the job owner's user ID as the only argument to the exit. The SQL statement can be read from STDIN. If the exit routine terminates with an exit code of zero, and without having written anything to STDOUT, DB2 Query Patroller continues to process the job normally. If the exit routine terminates with a non-zero exit code or writes anything to STDOUT, the job ends abnormally and sends a message to STDOUT.

- **IWM_INTERVAL**

Set the IWM_INTERVAL environment variable to the number of seconds that each server and agent process sleeps between searches for work. The default value is 20, which can be appropriate on a system with a moderate load. On a system with a light load, use a smaller value, such as 5, so DB2 Query Patroller will be more responsive to new requests.

- **IWMLOG**

Set the IWMLOG environment variable to the directory into which DB2 Query Patroller writes its log files. The default value is \$IWM_RUNTIME/log.

- **IWM_LOGMON**

If you set IWM_LOGMON, the iwm startup process will start the log monitor when it starts the server processes, and stop the log monitor when the server processes stop. The IWM_LOGMON environment variable should be set to the path name of the configuration file.

- **IWM_MAIL**

You can set IWM_MAIL to cause the notifier component to use the program named to format e-mail messages. If you do not set this variable, DB2 Query Patroller uses \$IWM_RUNTIME/bin/iwm_mail.sh.

- **IWM_NET**

IWM_NET must be set to the TCP/IP address and port of the server node, unless the command is run on the server node or one of the agent nodes. Use the format *address:port*, where *address* represents either the dotted notation for the address or a name resolvable in the /etc/hosts file or through DNS, and *port* represents either the port number or a name resolvable in the /etc/services file.

- **IWM_NOEXPLAIN**

If you want to ensure that all jobs run and that none are disqualified due to their estimated cost, disable cost analysis for all DB2 Query Patroller jobs by setting the IWM_NOEXPLAIN environment variable to any non-null value. Setting this variable prevents DB2 Query Patroller from performing a cost analysis on queries.

- **IWM_RES_TBLSPC**

Set the IWM_RES_TBLSPC environment variable to the name of the tablespace that will hold result tables. Each node may specify a different name, or multiple nodes may share a name. If this variable is left unset, the result tables will be placed into tablespaces determined by DB2.

- **IWM_RUNTIME**

Set the IWM_RUNTIME environment variable to the path where the DB2 Query Patroller software is installed.

- **IWM_SERVER**

Set the IWM_server environment variable to the TCP port and IP address of the DB2 Query Patroller server component. Use the format *address:port*. The address represents either the dotted notation for the address or a name resolvable in the /etc/hosts file or through the Domain Name System (DNS). The port represents either the port number or a name resolvable in the /etc/services file.

System Parameters

The System Parameters table stores the following parameters. These global parameters apply to all of DB2 Query Patroller. You can display and change these parameter values through the command-line interface and QueryAdministrator.

- **Job Cost Limit**

This parameter indicates the maximum total cost allowed for all running jobs.

- **Job Count Limit**

This parameter indicates the maximum total number of jobs allowed to be running.

- **Job Purge Days**

Setting this parameter to a non-zero value informs DB2 Query Patroller to purge information about a job after a defined number of days have passed.

- **Results Purge Days**

Setting this parameter to a non-zero value informs DB2 Query Patroller to drop result tables after the defined number of days have elapsed since job completion. A value of zero will retain all result tables.

- **Job Accounting Status**

This parameter indicates if rows should be added to the Job Accounting table when jobs complete. This parameter needs to be enabled to run the Tracker component.

Data Source Parameters

The Data Sources table stores parameters that apply to a specific database instance. You can display and change these parameters' values through the command-line interface and QueryAdministrator.

- **Data Source Status**

As indicated by this status, new jobs either will be accepted normally, held, or rejected.

- **Job Cost Limit**

This parameter indicates the maximum total cost allowed for all jobs running against the data source.

- **Job Count Limit**

This parameter indicates the maximum total number of jobs allowed to run against the data source.

- **Node CPU Limit**

If the CPU utilization of a node exceeds this value, no additional jobs are scheduled to run on the node.

- **Node Disk Limit**

If the number of available bytes of disk for a node is less than this value, no additional jobs are scheduled to run on the node.

Job Queue Parameters

The Job Queue table stores the following parameters. You can display and change these parameters' values using the QueryAdministrator.

- **Maximum Cost**

DB2 Query Patroller assigns jobs to the queue that has the smallest cost limit that is not less than the cost of the job.

- **Maximum Jobs**

This parameter indicates the maximum number of jobs allowed to run in the queue.

User Profile Parameters

The User Profile table stores the following parameters. You can display and change these parameter values through the QueryAdministrator.

- **Cost Analysis Required**

This parameter indicates whether the user can bypass cost analysis when submitting a job. Bypassing cost analysis makes DB2 Query Patroller less able to effectively manage the system workload.

- **Job Cost Limit**

Any jobs with an estimated cost greater than this value receive a hold status. The DB2 Query Patroller system administrator must either release or cancel jobs with a hold status. The job cost is measured in timeron units.

- **Job Count Limit**

This parameter indicates the maximum number of jobs allowed to run for a specific user.

- **Priorities**

Each user has one of three priority levels defined: low, normal, and high.

Administrative Tasks

The following subsections provide information about performing the system administrator tasks of maintaining cost statistics and adding users to the system.

Maintaining Cost Statistics

If the cost analyzer uses the database catalog to determine the relative cost of a full table scan, **RUNSTATS** should periodically be run to keep the catalog entries current. The catalog reflects the status of a table as of the most recent **RUNSTATS**. If it has never been run, the catalog will be empty.

Adding Users

Each DB2 Query Patroller user must be defined in the User Profile table, and must have a system account on the database server system with the **CONNECT** database privilege.

The task of adding users includes defining an e-mail address to which DB2 Query Patroller can send notifications. Mail can be sent to other mail packages like CC-Mail and Microsoft Exchange, given the proper configuration of the mail system. (The scope of this document does not address mail system configuration.) If not set up properly, the notification message bounces back to the iwm account. The mail for this user should be monitored, either directly or by forwarding it to another user.

Chapter 3. Operation

You can perform DB2 Query Patroller operations through a combination of the command-line interface and QueryAdministrator. This chapter provides information about the following operations:

- “Starting and Stopping DB2 Query Patroller”.
- “Controlling Job Flow and Execution”.
- “Releasing and Canceling Jobs” on page 18.
- “Monitoring the System” on page 18.
- “The Control Program (iwm)” on page 18.

It also provides control program, command syntax, and parameter descriptions.

Starting and Stopping DB2 Query Patroller

The control program (iwm) starts and stops DB2 Query Patroller. When starting DB2 Query Patroller, you should start the server node first, followed by the agent nodes. When stopping DB2 Query Patroller, you should stop the agent nodes first, followed by the server node. You must run iwm on each node for these activities to occur.

You can use a number of methods to pause DB2 Query Patroller without fully stopping the system. For instance, you can stop one or more of the agent nodes using the iwm program or QueryAdministrator. You can also make one or more nodes quiescent using the command-line interface or the QueryAdministrator and then later reactivate the node. You can alter the system or data source parameters to prevent additional jobs from being scheduled or even alter the data source status to prevent new jobs from being submitted.

Controlling Job Flow and Execution

You can control the number of jobs, where they run, and the load put on the database by altering the system, data source, and job queue parameters. See “Parameters” on page 11 for parameter descriptions.

Releasing and Canceling Jobs

If a user submits a query to DB2 Query Patroller and DB2 Query Patroller estimates the job's cost to be greater than the user's limit, then the job receives a hold status. In order for a job with a hold status to run, the system administrator must release the job using QueryAdministrator or the command-line interface. While a job is in the DB2 Query Patroller system, either the system administrator or the job's owner may cancel the job. A list of the jobs with a hold status or those jobs with some other status can be obtained through the command-line interface or QueryMonitor. See the *DB2 Query Patroller User's Guide* for more information about viewing a job's status.

Monitoring the System

In addition to monitoring job flow, the status of DB2 Query Patroller, and the database software, you need to monitor the host system itself. You can use the command-line interface to perform some monitoring and automated operations by using shell scripts, tools such as awk, and languages such as Perl. However, to be alerted to severe problems, monitor the system error log file produced by the server and agent components.

The Control Program (iwm)

The DB2 Query Patroller control program (iwm) starts, stops, and displays the status of DB2 Query Patroller processes. The following diagram provides the syntax for iwm:

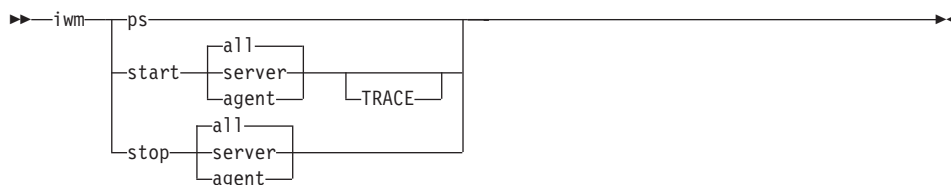


Table 3. iwm Parameter Descriptions

Parameter	Description
ps	This parameter displays a list of all DB2 Query Patroller processes running on this machine. The list displays in ps format. For an active agent node, you should see the iwm_net and iwm_nodemgr processes listed. You may also see other processes if the node has jobs running. For an active server node, you should see the iwm_net, iwm_server, iwm_remote, iwm_sched, and several iwm_local processes listed. If the log monitor is active, you should see the iwm_logmon process listed.
start {all server agent} [trace]	This parameter starts DB2 Query Patroller processes. The option defines the set of processes to start. The all option specifies that both the server and agent processes are to be started. The server option starts only the server processes. The agent option starts only the agent processes. Only one node per system should be started with the all or server options.
stop [all server agent]	This parameter directs DB2 Query Patroller to stop the requested DB2 Query Patroller process. If you do not specify any options, all is the default.

Chapter 4. Using the Command-Line Interface

This chapter provides information about using the DB2 Query Patroller command-line interface.

The command-line interface for DB2 Query Patroller enables the system administrator to monitor and control DB2 Query Patroller from an operating system shell. The command-line interface also enables end users to submit and monitor jobs from the system prompt. When combined with shell scripts or languages such as Perl, awk, and REXX, you can use the command-line interface as an application programming interface (API) to DB2 Query Patroller.

The command-line interface consists of the following two commands:

- Submit command (**iwm_submit**)
- Control command (**iwm_cmd**)

You can execute these commands on any machine with a TCP/IP connection to the server node.

Command-Line Interface Environment

The command-line interface requires the following items for successful execution:

- The user's login shell must have the following environment variables defined:
 - IWM_RUNTIME must be set to the path name into which DB2 Query Patroller was installed.
 - IWM_NET must be set to the TCP/IP address and port of the server node, unless the command is run on the server node or one of the agent nodes. Use the format *address:port*, where *address* represents either the dotted notation for the address or a name resolvable in the */etc/hosts* file or through DNS, and *port* represents either the port number or a name resolvable in the */etc/services* file.
 - PATH must contain \$IWM_RUNTIME/bin
- The user must have an entry in the User Profile table.

Submit Command (iwm_submit)

The submit command (**iwm_submit**) enables users to submit jobs to DB2 Query Patroller. Although SQL queries are usually submitted through a 32-bit ODBC query application (via QueryEnabler) or through QueryMonitor, you can use **iwm_submit** to submit SQL commands, and system commands.

The following diagram provides the syntax for the **iwm_submit** command:

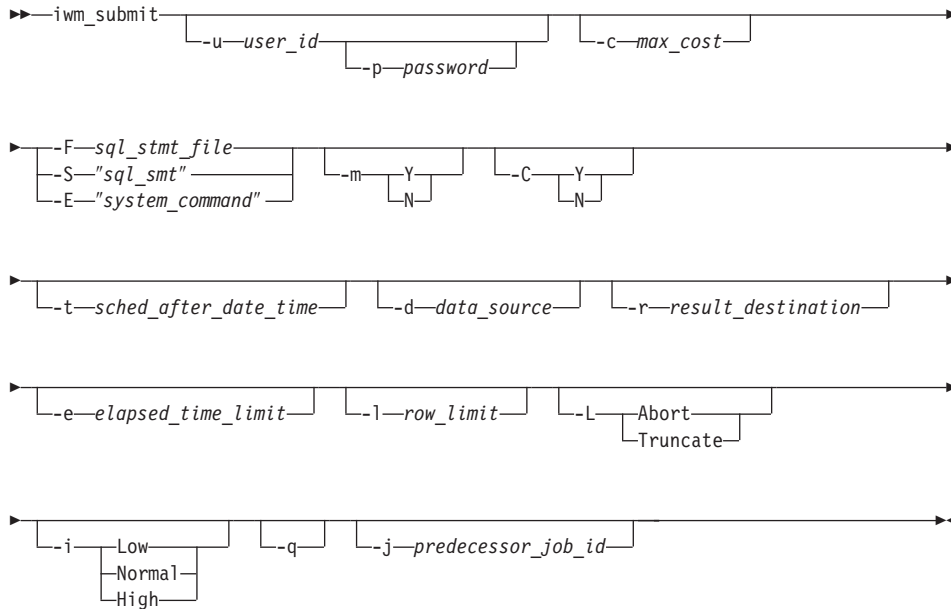


Table 4. iwm_submit Parameter Descriptions

Parameter	Description
-u user_id	User identification. This parameter enables the submitting user to specify a different owner for the job. This gives the DB2 Query Patroller system administrator the ability to submit queries or jobs on behalf of end users. The user ID must be a valid system and database ID, and a registered DB2 Query Patroller user. The command iwm_submit executes under the submitting user's environment. The user ID must conform to the system user identification syntax. The job owner defaults to the submitting user.

Table 4. `iwm_submit` Parameter Descriptions (continued)

Parameter	Description
<code>-p password</code>	User password. If you provide the <code>-u</code> parameter and the submitting user is neither root nor the DB2 Query Patroller system administrator, this password parameter must also be specified. The password must be the user's database password.
<code>-c max_cost</code>	Cost threshold. This option enables the user to lower the cost threshold from what is defined in the DB2 Query Patroller user profile. Only root and the DB2 Query Patroller system administrator can increase the cost threshold with this option. The cost threshold must be a numeric value. By default, DB2 Query Patroller uses the cost threshold defined in the DB2 Query Patroller database.
<code>-F sql_stmt_file</code>	Specifies a path name containing a single SQL statement. The <code>sql_stmt_file</code> must be a valid file path name. The file contents are immediately placed in the DB2 Query Patroller database.
<code>-S "sql_stmt"</code>	Specifies a single SQL command enclosed in quotes. The SQL statement is immediately placed in the DB2 Query Patroller database.
<code>-E "system_command"</code>	Specifies an entire system command including any optional arguments. The command is executed from the user's home directory, and its source directory must be available in the search path.
<code>-m {Y N}</code>	Mail Notification. This parameter specifies whether or not the user should be notified through e-mail when the job completes or receives a hold status. By default, the user receives notification upon completion of the submitted job.
<code>-C {Y N}</code>	Cost analysis override. If the user's DB2 Query Patroller profile allows queries to be submitted without cost analysis, this parameter determines whether the cost analysis should be performed. The option must be either Y or N. Y directs <code>iwm_submit</code> to perform cost analysis; N indicates no cost analysis. By default, DB2 Query Patroller performs cost analysis on SQL queries. DB2 Query Patroller ignores this option for system commands.

Table 4. iwm_submit Parameter Descriptions (continued)

Parameter	Description
-t <i>sched_after_date_time</i>	Schedule after date/time. This parameter specifies the date and time after which the job will run. The date/time format is <i>YYYYMMDDHHMMSS</i> . The parameter default is the present date and time.
-d <i>data_source</i>	Data source. Specifies the data source to run the job on. The only supported data source is IWM. This is the default value.
-r <i>result_destination</i>	Result Destination. This parameter specifies the name of an alternate destination where results are sent instead of being written to a table in the database.
-e <i>elapsed_time_limit</i>	Elapsed time limit. This option enables the user to set a limit for the elapsed time the job will be permitted to run. If the time limit is exceeded, DB2 Query Patroller aborts the job. The time limit value is in seconds. By default, DB2 Query Patroller will not impose an elapsed time limit on the job.
-l <i>row_limit</i>	Result row limit. This option enables the user to set a limit for the number of rows returned by a SQL SELECT statement. If the row limit is exceeded, the action taken by DB2 Query Patroller depends on the setting of the -L parameter. By default, DB2 Query Patroller will not impose a result row limit.
-L {Abort Truncate}	Result row limit action. This option enables the user to specify the action taken by DB2 Query Patroller should the job return more rows than the result row limit. If it is not specified or set to Abort, DB2 Query Patroller aborts the job. Otherwise, if it is set to Truncate, DB2 Query Patroller truncates the result table at the row limit.
-i {Low Normal High}	User Priority. This parameter enables the user to select the priority of the job. The priority value can be low, normal, or high. Only the first letter is significant. These values may also be specified as 0, 1, or 2. The user's DB2 Query Patroller profile defines the actual priority levels that correspond to these values. By default, DB2 Query Patroller assigns the user's normal priority to the job.
-q	Quiet mode. This parameter directs DB2 Query Patroller to print only the job number upon successful job submission. This may be useful if the iwm_submit command is used from within a shell script.

Table 4. `iwm_submit` Parameter Descriptions (continued)

Parameter	Description
<code>-j predecessor_job_id</code>	Predecessor job number. This parameter specifies a job number of a previously submitted DB2 Query Patroller job. The new job cannot be scheduled until the predecessor job completes.

Control Command (`iwm_cmd`)

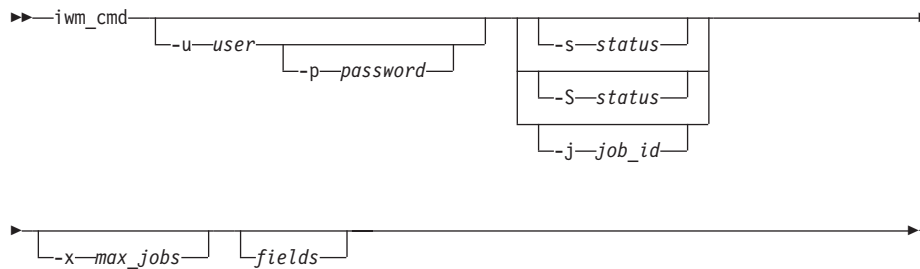
The control command (**`iwm_cmd`**) enables the system administrator to monitor and control DB2 Query Patroller from the operating system shell. It also enables end users to monitor their jobs.

Specifically, the system administrator and end user can use **`iwm_cmd`** when:

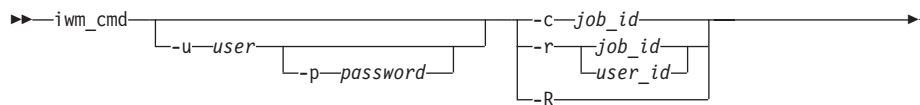
- Monitoring jobs
- Controlling jobs
- Monitoring and controlling nodes
- Listing and controlling data sources
- Listing and controlling system parameters

Each function of **`iwm_cmd`** has a different syntax. The following diagrams provide the complete syntax for the **`iwm_cmd`**:

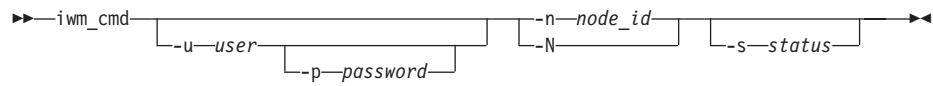
- Monitoring jobs:



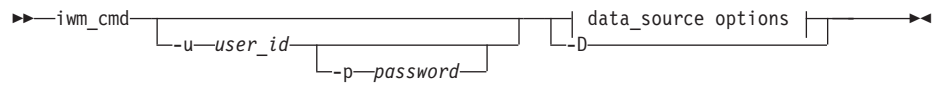
- Controlling jobs:



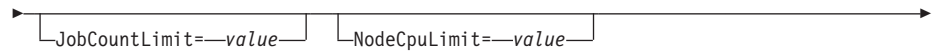
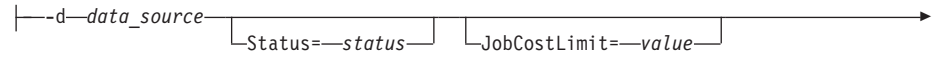
- Monitoring and controlling nodes:



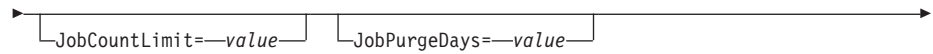
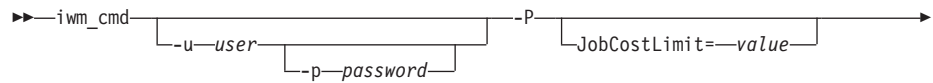
- Listing and controlling data sources:



data_source options:



- Listing and controlling system parameters:



Common Parameters

Table 5. iwm_cmd Common Parameter Descriptions

Parameter	Description
-u <i>user_id</i>	User identification. This parameter provides either the DB2 Query Patroller user ID for the user whose jobs will be monitored or controlled, or the DB2 Query Patroller system administrator's ID (iwm). By default, DB2 Query Patroller uses the current system user ID.
-p <i>password</i>	User password. If the user provides the -u parameter and the current user is not the DB2 Query Patroller system administrator, the password parameter must also be specified. The password must be the user's database password.

Monitoring Jobs

iwm_cmd allows users to monitor their jobs and the system administrator to monitor the status of all incomplete jobs. DB2 Query Patroller provides an interface to make it easy for programs invoking **iwm_cmd** to process information about each job. If a user invokes **iwm_cmd** without any parameters, all jobs for that user display.

Syntax:

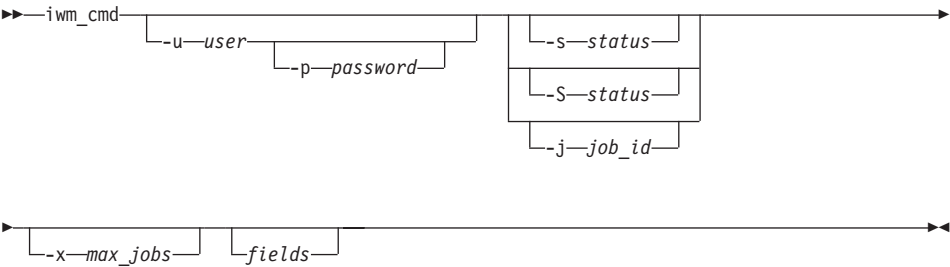


Table 6. iwm_cmd Parameters for Monitoring Jobs

Parameter	Description
-s <i>status</i>	User job status. This parameter specifies that jobs belonging to the user with the given status should display. The status must be one of the following: • Q - Queued • S - Scheduled • H - Held • R - Running • U - Unfinished • D - Done • A - Aborted • C - Canceled • F - Finished Only the first letter is significant. It can be in either uppercase or lowercase. The unfinished status includes any queued, scheduled, held, or running jobs. The finished status includes any done, aborted, or canceled jobs.
-S <i>status</i>	Unfinished job status. This parameter specifies that all DB2 Query Patroller jobs with the given status should display. The status must be one of the following: • Q - Queued • S - Scheduled • H - Held • R - Running • U - Unfinished Only the first letter is significant. It can be in either uppercase or lowercase. The Unfinished status includes queued, scheduled, held, or running jobs. This parameter may only be used by the DB2 Query Patroller system administrator.
-j <i>job_id</i>	Job identifier. This parameter specifies that only the requested job should display. If someone other than the DB2 Query Patroller system administrator uses this parameter, the user must be the owner of the job.
-x <i>max_jobs</i>	Maximum job count. This parameter specifies the maximum number of jobs to be listed. If this parameter is not specified, a maximum of 100 jobs display. If max_jobs contains the value 0, no limit is applied.

Table 6. iwm_cmd Parameters for Monitoring Jobs (continued)

Parameter	Description
<i>fields</i>	Job fields. There are two job fields: Job and Status. You can enter these fields together or separately. This parameter is case sensitive. If you provide a field parameter, the jobs do not display in an easy-to-read format. Instead, the values of the fields are written to STDOUT in a format that other programs can easily process. A horizontal tab character separates each field from the next. Newline characters separate each job's fields from the next.

Controlling Jobs

iwm_cmd allows users to cancel their jobs. It also allows the system administrator to cancel or release a job.

Syntax:

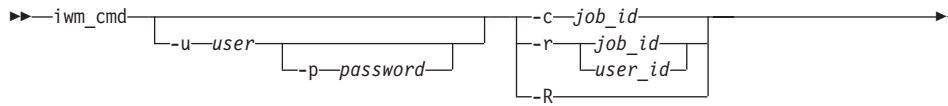


Table 7. iwm_cmd Parameters for Controlling Jobs

Parameter	Description
-c <i>job_id</i>	Cancel a job. This parameter specifies that the given job should be canceled. A job need not have finished in order for it to be canceled. If the user is not the DB2 Query Patroller system administrator, the job must be owned by the user.
-r <i>job_id</i>	Release a job. This parameter specifies that the given job should be released. The job must currently have a status of held. This parameter may only be used by the DB2 Query Patroller system administrator.
-r <i>user_id</i>	Release all jobs for a user. This parameter specifies that all jobs owned by a user with a current status of held should be released. This parameter may only be used by the DB2 Query Patroller system administrator.

Table 7. iwm_cmd Parameters for Controlling Jobs (continued)

Parameter	Description
-R	Release all jobs. This parameter specifies that all Held jobs should be released. This parameter may only be used by the DB2 Query Patroller system administrator.

Monitoring and Controlling Nodes

iwm_cmd enables the DB2 Query Patroller system administrator to monitor and control the status and activity of agent nodes.

Syntax:

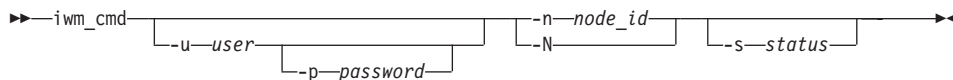


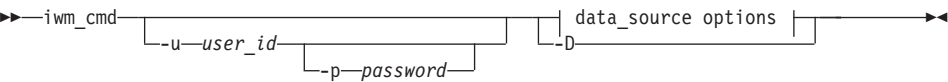
Table 8. iwm_cmd Parameters for Monitoring and Controlling Nodes

Parameter	Description
-n <i>node_id</i>	Node identifier. This parameter specifies that the status of the given node should be displayed or be altered.
-N	All nodes. This parameter specifies that the status of all agent nodes should be displayed or be altered.
-s <i>status</i>	Node status. This parameter specifies that the status of the node(s) should be altered and provides the new requested status. The status must be from the following list. Only the first letter is significant. It can be either uppercase or lowercase. • Active. The server component can schedule new jobs to run on the node. • Quiesced. Jobs already scheduled to run on the node continue to run, but no more will be scheduled. • Inactive. Jobs already scheduled to run on the node continue to run, but no more will be scheduled to run on the node. After all previously scheduled jobs complete, the node manager process running on the node will terminate. • Force. The node manager process running on the node will terminate immediately without waiting for jobs already running on the node to complete.

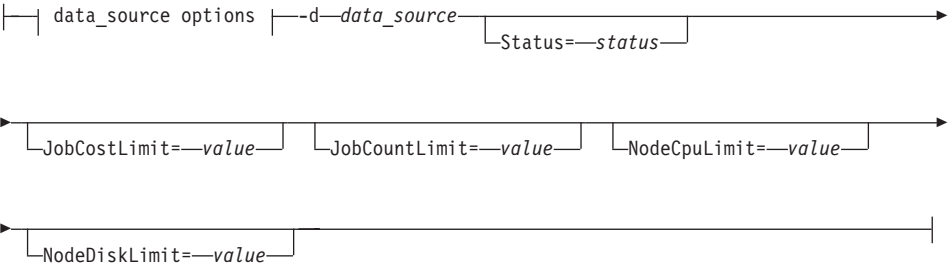
Listing and Controlling Data Sources

iwm_cmd enables the DB2 Query Patroller system administrator to list and change the data source parameters for one or all data sources.

Syntax:



data_source options:



iwm_cmd enables the DB2 Query Patroller system administrator to list and change the data source parameters for one or all data sources.

Table 9. iwm_cmd Parameters for Listing and Controlling Data Sources

Parameter	Description
-d <i>data_source</i>	Data source name. This parameter specifies the name of the data source to be listed or altered. In the current release, the only valid value is IWM.

Table 9. iwm_cmd Parameters for Listing and Controlling Data Sources (continued)

Parameter	Description
Status= <i>status</i>	Data source status. This parameter specifies that the status of the data source should be altered and provides the new status. The status must be one of the following: • Active. Jobs will be scheduled normally. • Hold. All incoming jobs will be held, but any jobs already in the queue will be allowed to proceed. • Inactive. All requests to submit a new job will be rejected. • ReleaseAll. All jobs in the data source with a status of held will be released and the data source status will be set to active. • HoldAll. All jobs in the data source that are not yet running will be held. The status of the data source will be set to hold.
JobCostLimit= <i>value</i>	Alter job cost limit. This parameter specifies that the limit of the total estimated cost of running jobs against this data source should be altered. The new value must be at least zero.
JobCountLimit= <i>value</i>	Alter job count limit. This parameter specifies that the limit of the total number of jobs running against this data source should be altered. The new value must be at least zero. If zero, scheduling of new jobs is disabled.
NodeCpuLimit= <i>value</i>	Alter node CPU limit. This parameter specifies that the per-node CPU utilization maximum for scheduling new jobs against this data source should be altered. The new value must be between 0 and 100. If CPU utilization is not being collected, the value of this parameter is ignored.
NodeDiskLimit= <i>value</i>	Alter node disk space limit. This parameter specifies that the per-node disk space minimum for scheduling new jobs against this data source should be altered. The value is in bytes and must be at least zero. If disk space utilization is not being collected, the value of this parameter is ignored.
-D	List data sources. This parameter specifies that summary information on all data sources should display.

Listing and Controlling System Parameters

iwm_cmd enables the DB2 Query Patroller system administrator to list and change system parameters.

Syntax:

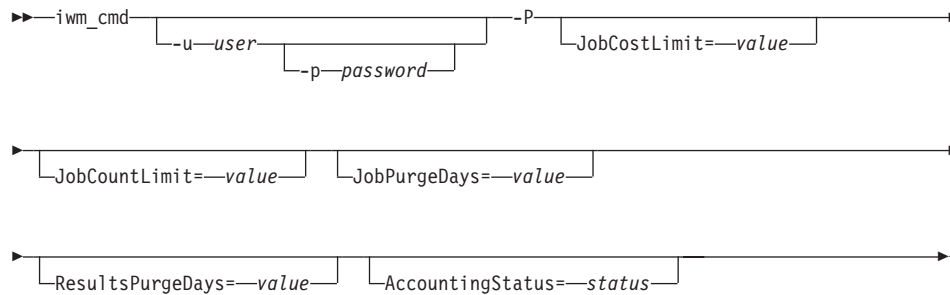


Table 10. iwm_cmd Parameters for Listing and Controlling System Parameters

Parameter	Description
-P	System parameters. This parameter specifies whether or not the values of the DB2 Query Patroller system parameters should be displayed or altered.
JobCostLimit= <i>value</i>	Alter job cost limit. This parameter specifies that the limit of the total estimated cost of running jobs should be altered. The new value must be at least zero.
JobCountLimit= <i>value</i>	Alter job count limit. This parameter specifies that the limit of the total number of running jobs should be altered. The new value must be at least zero. If zero, scheduling of new jobs is disabled.
JobPurgeDays= <i>value</i>	Alter job information purging. This parameter specifies that the purge age for DB2 Query Patroller jobs should be altered. The new value must be between 0 and 999. If zero, automatic purging of job information is disabled. If not zero, the value of JobPurgeDays must be equal to or greater than the value of ResultsPurgeDays.
ResultsPurgeDays= <i>value</i>	Alter result table purging. This parameter specifies that the purge age for result tables should be altered. The new value must be between 0 and 999. If zero, automatic purging of result tables information is disabled. If the value of JobPurgeDays does not equal zero, it must be equal to or greater than the value of ResultsPurgeDays.

Table 10. iwm_cmd Parameters for Listing and Controlling System
Parameters (continued)

Parameter	Description
AccountingStatus= <i>status</i>	Alter job accounting status. This parameter specifies that the status of the DB2 Query Patroller job accounting feature should be altered. The status must be either active or inactive. If Active, DB2 Query Patroller inserts a row into the job accounting table as each job completes, aborts, or is canceled.

Chapter 5. Error Log Monitoring

This chapter provides information about running the log monitor and using the log monitor configuration file.

To be alerted to a severe problem, you should monitor the system error log file produced by the server and agent components.

The Log Monitor program periodically activates and examines the log file for new entries. The filter criteria qualify any new messages defined to the log monitor in its configuration file. The program then sends qualified messages through system e-mail to a list of recipients defined in the configuration file.

Running the Log Monitor

If you set the IWM_LOGMON environment variable, the iwm start-up program starts the log monitor when it starts the server processes. The control program also stops the log monitor when the server processes stop. The IWM_LOGMON environment variable must be set to the path name of the configuration file.

To start the log monitor outside iwm, execute the following command:

iwm_logmon -c *configuration_file*

After initialization, the log monitor becomes a background process. To stop the log monitor, use an operating system command to stop the process.

Log Monitor Configuration File

The log monitor configuration file must contain one or more configuration command lines. These commands define the log file to be monitored, the wake-up interval, the list of message recipients, and the filter criteria.

Each line in the configuration file may be empty or may contain a comment or configuration command. Comment lines start with a pound sign (#). One or more white space characters can precede the comment indicator or the first character in the configuration command. You can also enter comments at the end of each configuration command line when preceded by a pound sign.

INTERVAL Command

Syntax:

►►—INTERVAL—*seconds*—————►►

The INTERVAL command accepts one parameter: the number of seconds that the log monitor should wait between checking the log file for new messages. If you do not specify an INTERVAL command, the parameter defaults to 30 seconds. If you specify multiple INTERVAL commands, log monitor uses the last-specified command.

LOGFILE Command

Syntax:

►►—LOGFILE—*file_name*—
 *—
 └─TRUNCATE─┐
 └─CONTINUE─┘—————►►

The LOGFILE command defines the log file name and the action that should be taken when the log monitor starts. You must first specify the log file name parameter, which may be specified as an asterisk (*) if you want to use the default log file name. If you set the IWMLOG environment variable, the default log file name is \$IWMLOG/syserr.log. If you do not set the IWMLOG environment variable, the default log file name is \$IWM_RUNTIME/log/syserr.log.

The action to take is specified by TRUNCATE, CONTINUE, or by omitting the parameter altogether. If you specify TRUNCATE, DB2 Query Patroller removes all messages from the log file when the log monitor initializes. If you specify CONTINUE, DB2 Query Patroller sends only messages added to the log file after the log monitor starts. If you omit the action parameter, DB2 Query Patroller sends all messages already in the log file, as well as any new messages.

If you specify multiple LOGFILE commands, DB2 Query Patroller uses the last command. If you do not specify a LOGFILE command, the log monitor monitors the default log file and any messages already in the log file at start-up are sent.

MAILTO Command

Syntax:

►►MAILTO—address◄◄

The MAILTO command defines the recipients of messages. Each MAILTO command must include one or more e-mail addresses. You must specify at least one MAILTO command.

INCLUDE and EXCLUDE Commands

Syntax:

►►INCLUDE—test1—
EXCLUDE—test2—
test3—◄◄

Each *test* is one of: **SEV**[!]=code, **NUM**[!]=number[:number], **PGM**[!]=program, or **ALL**.

The INCLUDE and EXCLUDE commands define the message filtering criteria. The log monitor tests each new message against the filter commands. Recipients receive those messages that match an INCLUDE command or do not match any filter commands. The log monitor tests each message against the filter commands in the order specified in the configuration file. The first command that matches the message on all tests specified causes the message to be included or excluded.

You cannot specify the ALL test with any other test. It should appear only on the last INCLUDE or EXCLUDE command in the configuration file because it matches all messages, causing all subsequent filter commands to be ignored.

Of the other tests (SEV, NUM, and PGM), you can specify up to three on each filter command, but you can only specify one of each type on each command. If you specify more than one test on a command, all tests must match the message for the command to apply. A comparison operator and comparison value must follow each test type name. The operator must either be the equal sign (=) or an exclamation mark followed by an equal sign (!=). The equal sign means that in order to match, the test must be equal to the value. The exclamation mark and equal sign specifies that the test must not be equal to the value. The following list explains the filter test types.

- **SEV** (*message severity*)

The severity comparison value must be one of the following letters: V (Verbose), W (Warning), I (Informational), or E (Error). For example, to test

for error messages, use SEV=E; to test for non-verbose messages use SEV!=V. For additional information about severity codes, see “Appendix C. DB2 Query Patroller Messages” on page 101.

- **NUM** (*message number*)

The message number comparison value must either be a single one to five digit number or a range of message numbers specified by two numbers separated by a colon (:). The values correspond to the message numbers described in “Appendix C. DB2 Query Patroller Messages” on page 101. For example, to test for message 100, use **NUM=100**; to test for all messages greater than 100, specify **NUM!=0:100**.

- **PGM** (*program name*)

The program name comparison must be the name of the program that generated the message. For example, to test for messages generated by the iwm_analyze program, use **PGM=iwm_analyze**; to test for messages not generated by the iwm_sqlexec program, use **PGM!=iwm_sqlexec**.

Chapter 6. Job Accounting

This chapter provides information about the job accounting process and the columns in the Job Accounting table.

If you set the job accounting status system parameter to be active, the notifier component adds a row to the Job Accounting table each time a job completes, either successfully or unsuccessfully.

This information is used by the Tracker utility to provide reports that display a database usage history for queries managed by the DB2 Query Patroller system. For example, this information will be used by Tracker to create reports showing which tables and columns have been accessed most frequently, which tables have returned the most result rows, and which jobs have been completed within a specific time frame.

The following list explains each of the columns in the Job Accounting table.

- **ACCOUNT_ID**

The ACCOUNT_ID column represents the accounting identifier associated with the user from the User Profile table.

- **COMPLETION_STATUS**

The COMPLETION_STATUS column contains a single letter code indicating how the job completed. Values are D for done (normal completion), C for canceled, or A for aborted.

- **DATA_SOURCE**

The DATA_SOURCE column identifies the database instance against which the query was run. Currently, the only supported data source identifier is IWM.

- **DTIME_COMPLETED**

The DTIME_COMPLETED column identifies the date and time that the job was completed.

- **DTIME_CREATED**

The DTIME_CREATED column identifies the date and time that the user submitted the job to DB2 Query Patroller.

- **DTIME_SCHEDULED**

The DTIME_SCHEDULED column identifies the date and time after which the user scheduled the job to be run.

- **DTIME_STARTED**

The DTIME_STARTED column identifies the date and time that the job started running. If the job was canceled or aborted before the job started running, this column contains a NULL value.

- **ELAPSED_TIME**

The ELAPSED_TIME column identifies the elapsed time in seconds from when the job started running until it completed.

- **ELAPSED_TIME_LIMIT**

The ELAPSED_TIME_LIMIT column identifies the maximum number of elapsed seconds a job will be permitted to run.

- **ERROR_REASON**

The ERROR_REASON column identifies the reason why the job did not complete successfully.

- **ESTIMATED_COST**

The ESTIMATED_COST column identifies the estimated cost for the job in accounting units, if one was derived. This column contains the value of the estimated cost for the job multiplied by the cost factor from the Data Source table.

- **JOB_ID**

The JOB_ID column contains a number that identifies the job.

- **JOB_PRIORITY**

The JOB_PRIORITY column contains the priority level number assigned to the job.

- **JOB_TEXT**

The JOB_TEXT column contains either the job's SQL statement or its system command.

- **JOB_TYPE**

The JOB_TYPE column contains either the letter S for SQL jobs or the letter C for jobs containing system commands.

- **NODE_ID_EXECUTED**

The NODE_ID_EXECUTED column identifies the host name of the node where the job was run.

- **NODE_ID_SUBMITTED**

The NODE_ID_SUBMITTED column identifies the node from which the job was submitted.

- **PREDECESSOR_JOB**

If the job was assigned a predecessor job (a job which must have completed before this job could run), the PREDECESSOR_JOB column contains the predecessor's job identifier. If no predecessor was assigned, this column contains a NULL value.

- **RESULT_DESTINATION**

If the user selected an alternative result destination, the `RESULT_DESTINATION` column contains the identifier of the destination. If the results were to be written to a result table created in the database, this column contains a `NULL` value.

- **RESULT_ROWS**

The `RESULT_ROWS` column contains the number of database rows that a job selected, updated, deleted, or inserted.

- **RESULT_ROWS_LIMIT**

The `RESULT_ROWS_LIMIT` column identifies the maximum number of rows a SQL `SELECT` statement is allowed to return.

- **RESULT_TABLE_OWNER**

The `RESULT_TABLE_OWNER` column contains the database identifier of the owner of the result table, if it was created.

- **RESULT_TABLE_NAME**

The `RESULT_TABLE_NAME` column contains the table name of the result table, if it was created.

- **SUBMIT_SOURCE**

The `SUBMIT_SOURCE` column contains the name of the program that submitted the job.

- **SUBMITTER_USER_ID**

The `SUBMITTER_USER_ID` column contains the system identifier of the submitter of the job. This identifier could be different from the job's owner if one user submitted the job on behalf of another user.

- **SYSTEM_TIME**

The `SYSTEM_TIME` column contains the number of CPU seconds of system time consumed by the job.

- **USER_ID**

The `USER_ID` column contains the system identifier of the job owner.

- **USER_TIME**

The `USER_TIME` column contains the number of CPU seconds of user time consumed by the job.

Chapter 7. Customizing

This chapter provides information about performing cost analysis prior to an exit, result destinations, and e-mail message format.

Exit Analysis

DB2 Query Patroller can invoke a customer-written exit program just prior to performing cost analysis for a query. The exit program can examine the SQL statement and prevent DB2 Query Patroller from executing the statement. DB2 Query Patroller invokes the exit program for all jobs containing a SELECT, INSERT, UPDATE, or DELETE statement unless the user bypassed cost analysis. (The user can only bypass cost analysis if the system administrator sets the Cost Analysis Required option to N in the user's profile.)

DB2 Query Patroller invokes the exit analysis program if you set IWM_EXIT_AN to the path name of the executable program before starting the server processes. The exit program can be a shell script or any other executable. DB2 Query Patroller invokes the exit program under the iwm user ID. DB2 Query Patroller passes the job owner's user ID as the only argument to the exit. The SQL statement can be read from STDIN. If the exit routine terminates with an exit code of zero and without having written anything to STDOUT, DB2 Query Patroller continues to process the job normally. If the exit routine terminates with a non-zero exit code or writes anything to STDOUT, the job aborts. The job abort message includes the message written to STDOUT.

Result Destinations

Normally, the database stores query results as tables. The Result Destinations table contains the descriptions of any available alternative destinations for query results. These alternative destinations can designate that the results should be written to a file or piped to a process. If written to a file or pipe, the results are output in comma-separated variable format.

When the results are to be written to a file, the destination description designates the file name. When the results are to be piped, the description designates the name of the executable to which the results will be piped, along with any command-line parameters.

Result Destination Variables

For either files or piped destinations, the description can contain the following variables. The run-time values replace these variables before the query executes.

- **\$(JOB)**

A seven-digit job number replaces this variable.

- **\$(USER)**

The user ID of the job's owner replaces this variable.

- **\$(HOME)**

The path name of the home directory of the job's owner replaces this variable.

Piped Result Destinations

The pipe process can be a shell script or any other executable. DB2 Query Patroller invokes this process under the job owner's user ID, but with the environment variables set as they were when agent started. The process can read the results from STDIN unless the system administrator specifies the \$(PIPE) variable symbol in the result destination description, in which case it can be read from a named pipe created by the DB2 Query Patroller SQL executor. If the process terminates before reading all of the results, the job aborts. To create a result file on the machine named "remote", you can use the following destination description:

```
rsh remote "cat > job$(JOB)_results"
```

E-Mail Messages

Unless you set the IWM_MAIL environment, the DB2 Query Patroller notifier component uses the shell script \$IWM_RUNTIME/bin/iwm_mail.sh to format e-mail messages. You can also create your own script to format the messages and then set the IWM_MAIL environment variable to the name of your script. If you want to create your own formatting script, use the iwm_mail.sh script supplied by IBM as a guide.

The formatting script executes each time the notifier needs to send e-mail to a user about a job that has either completed normally, aborted, been canceled, or been put on hold. The script receives the job details using STDIN and must format and write the message body to STDOUT. The iwm_mail.sh script supplied by IBM is written as a KornShell script. You may write a customized formatting script in any language.

Chapter 8. Configuring the DB2 Query Patroller Server

This chapter provides information and outlines approaches for configuring the DB2 Query Patroller server for best performance.

There are no specific guidelines for setting the DB2 Query Patroller configuration parameters, because each data warehouse has its own set of unique attributes, such as size of the database, the complexity of the queries, the number of users supported, and the availability of hardware resources. This section describes an approach that can be taken to configure the DB2 Query Patroller server.

Profiling

If feasible, profile the set of queries that will access the data warehouse through DB2 Query Patroller by submitting each query in isolation. This method provides valuable statistics, such as the cost estimate of the query, the time for the query to be executed, and the size of the answer set. Additional information, such as the behavior patterns of the end users and hardware resources, help you determine the appropriate setting for DB2 Query Patroller configuration parameters.

Estimating and Monitoring

Once you have profiled the set of queries that will access the database, the behaviour patterns of end users, and the hardware resources, apply what you estimate to be the proper parameter configuration. In this initial phase, monitor the data warehouse during peak hours to determine if the data warehouse is being under utilized or is performing poorly in an attempt to process more queries than it can manage. If DB2 Query Patroller is configured too conservatively, the queries may not be submitted to the data warehouse even though resources are available. On the other hand, if DB2 Query Patroller configuration is too liberal, too many queries may be running against the data warehouse, causing the system to waste valuable resources paging or switching contexts.

Modifications

It is important to gather information from DB2 Query Patroller in order to determine the most effective configuration. The process of gathering this information must be performed systematically. Typically, you should modify only one parameter at a time. After each modification, observe the system to determine the specific impact of the modification.

Configuring the Data Warehouse

DB2 Query Patroller stores valuable information in DB2 tables. This information can be accessed to help configure the data warehouse. Since the DB2 Query Patroller schema resides in the same DB2 database as the data warehouse, the data warehouse should be configured to account for the DB2 requirements of DB2 Query Patroller.

Run the following queries to measure the DB2 Query Patroller requirements. These queries should be executed against the DB2 Query Patroller schema directly through DB2 and not through the DB2 Query Patroller client interface:

1. To determine the maximum number of queries currently running against DB2 Query Patroller and the cost of the queries:

```
SELECT COUNT(*), SUM(STATIC_COST) FROM IWM003_JOB_TABLE  
WHERE JOB_STATUS='R'
```
2. To determine how many jobs are in the queue state waiting to run, but are held up due to a resource limitation:

```
SELECT COUNT(*), SUM(STATIC_COST) FROM IWM003_JOB_TABLE  
WHERE JOB_STATUS='R' AND TIMESTAMP(DTIME_START_AFTER) <  
CURRENT_TIMESTAMP
```
3. To check the load on the DB2 Query Patroller nodes to determine their individual workloads:

```
SELECT NODE_ID, SCHEDULED_JOBS, CPU_UTILIZATION FROM  
IWM003_NODE_TABLE WHERE NODE_STATUS='ACTIVE'
```

Note: The number of scheduled jobs includes both the active number of jobs and jobs waiting to run.

You can run other queries against the DB2 Query Patroller schema to derive additional information. When accessing the DB2 Query Patroller schema, keep in mind that when DB2 Query Patroller is running it is constantly accessing these tables to update status and retrieve information. Access to these tables must be shared with DB2 Query Patroller. If any of these tables is locked indefinitely, DB2 Query Patroller will appear to be hung as it waits for the database lock to be freed.

Chapter 9. Configuring the DB2 Query Patroller Client

This chapter provides hints and tips for configuring DB2 Query Patroller Client applications.

Configuring QueryEnabler

The typical DB2 Query Patroller Client requires two DB2 Query Patroller components to be installed on their desktop. The QueryEnabler and the QueryMonitor. The QueryMonitor allows the user to manage queries. The QueryEnabler is the component of DB2 Query Patroller that traps the ODBC query.

From the perspective of the ODBC 32-bit query application, QueryEnabler is an ODBC driver. From the DB2 perspective, QueryEnabler is an ODBC application. When you are configuring an ODBC query application to access a DB2 data warehouse through QueryEnabler, keep this distinction in mind to avoid confusion between the DB2 data source name and QueryEnabler data source name. Configuring QueryEnabler involves the following tasks:

1. Define and configure a DB2 data source name for the DB2 data warehouse. You can do this using DB2's Client Configuration Assistant.
2. Define a DB2 Query Patroller data source using the ODBC Data Source Administrator. You will provide the previously defined DB2 data source name referencing the target data warehouse as the final data source.

Note: You should use a naming convention which easily distinguishes a DB2 data source name from a DB2 Query Patroller data source name.

If a connection can not be established for the ODBC query application using the DB2 Query Patroller defined data source name, a connection using the DB2 data source name should be attempted. Once the DB2 connection is successful, try a connection to DB2 Query Patroller once again.

Note: Other DB2 Query Patroller client-side tools, such as QueryMonitor, use the DB2 alias to access the data warehouse. If you have specified the DB2 Query Patroller ODBC data source name with these tools, the QueryEnabler scheduling window will open when you submit a query.

Client Configuration Options

The DB2 Query Patroller client has several configuration options that control the operational semantics of the queries that it submits. These configuration options can be modified in the file DB2IWM.INI in the Windows install directory. The configuration options are:

- **MaxResultRows**

This option specifies the maximum number of rows that will be returned in the answer set. If this value is set to 0, the complete answer set will be returned.

- **MaxElapsedTime**

This option specifies the maximum number of seconds the query will be permitted to run against the DB2 Query Patroller server. If this value is set to 0 or -1 the query will always run to completion.

- **n-tier**

This option is provided to allow queries to be posted to the DB2 Query Patroller server without requiring user intervention. This environment variable can be set to one of two values:

- **RUN**

When the DB2_IWM_SUBMITWAIT variable is set to **RUN**, queries will be submitted to the DB2 Query Patroller server in a Submit and Wait mode. A user dialog is not required. The RUN value has a time out option: RUN: <timeout>. This option is in units of seconds. If this option is not set, QueryEnabler will wait until the query has been processed before returning control to the query application.

- **CHECK**

When the DB2_IWM_SUBMITWAIT variable is set to CHECK, it will behave similarly to RUN, but in this case, DB2 Query Patroller will check to determine if an answer set exists for the query. If an answer set does exist, that result set will be returned, as opposed to the same query being executed again. The CHECK value has a time out option: CHECK: <timeout>. This option is in units of seconds. If this option is not set, QueryEnabler will wait until the query has been processed before returning control to the query application.

Note: The default value for the **n-tier** option is **OFF**.

Part 2. The QueryAdministrator Utility

Chapter 10. QueryAdministrator

This chapter introduces the QueryAdministrator utility. It provides procedures for starting and stopping QueryAdministrator along with general information on how to use the QueryAdministrator main window.

Note: The procedures in this portion of the manual assume that you have already installed QueryAdministrator according to the instructions in the *DB2 Query Patroller Installation Guide*.

About QueryAdministrator

QueryAdministrator is a utility that provides an easy-to-use interface for administering the DB2 Query Patroller system. QueryAdministrator is a Java application that runs on Windows 32-bit operating systems.

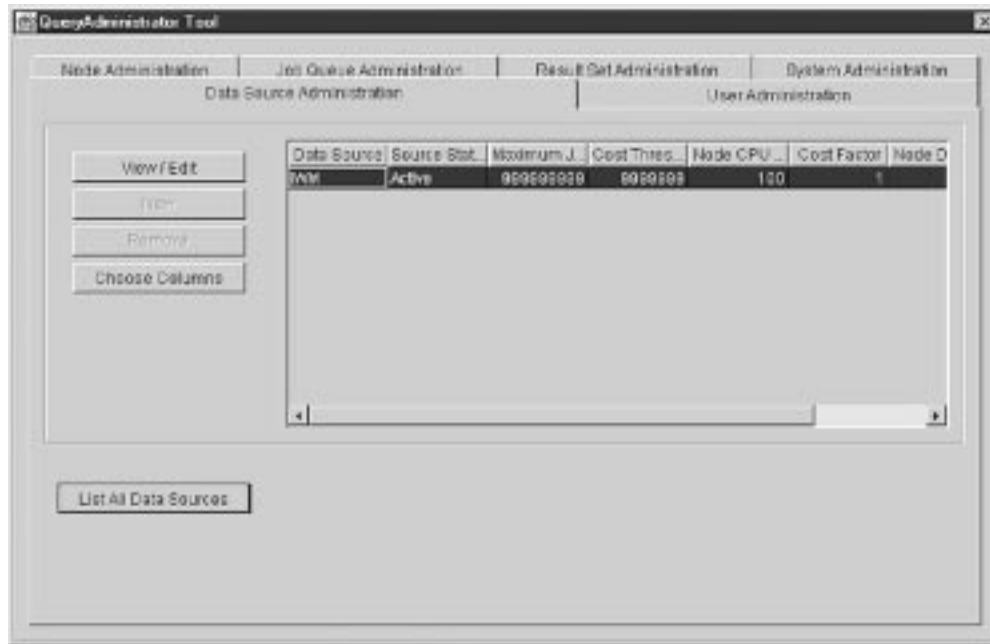
QueryAdministrator enables the system administrator to manage the DB2 Query Patroller system parameters, create or delete profiles for DB2 Query Patroller users, manage nodes, result destinations, and manage data sources.

Starting QueryAdministrator

Use the following procedure to start QueryAdministrator:

1. Select **Start > Programs > DB2 > QueryAdministrator**.
The QueryAdministrator Login window displays.
2. Enter the user name, password, and the database alias.
3. Click **OK**.

The QueryAdministrator main window opens:



Stopping QueryAdministrator

To stop QueryAdministrator, close the window.

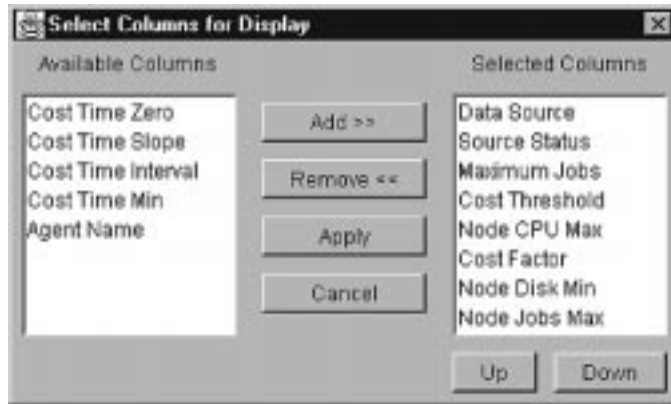
Displaying Columns

When viewing information on any of the tabs in the QueryAdministrator main window, you can choose the columns you wish to have displayed.

To choose the columns to display, complete the following steps:

1. Click **Choose Columns**.

The Select Columns for Display window opens:



2. To add a column to the display, do the following:
 - a. Select the column to be added from the **Available Columns** list.
 - b. Click **Add>>** to add the column to the **Selected Columns** list.
 - c. Select the newly added column in the **Selected Columns** list.
 - d. Click **Up** or **Down** to move the newly added column to the position where you want it to display.
3. To remove a column from the display select the column to be removed from the Selected Columns list and click **<<Remove**.
4. Click **Apply** when you have finished adding and removing columns.

Any changes you make to the columns displayed are retained for the current session only.

Resizing Columns

Whenever you are viewing information on a tab in the QueryAdministrator main window you can resize any column to a preferred width.

To resize a column, complete the following steps:

1. Move the mouse pointer to the vertical line forming the right edge of the column heading. The mouse pointer changes to a double arrow.
2. Left-click and hold the mouse button on the vertical line; move the line to resize the column.
3. Release the mouse button.

Any changes you make to the columns displayed are retained for the current session only.

Chapter 11. Data Source Administration

This chapter describes the procedures for listing and editing data sources.

Listing Data Sources

Use the following procedure to list data sources:

1. In the QueryAdministrator main window, select the **Data Source Administration** tab.
2. Click **List All Data Sources** to list all data sources.

Each data source is listed in tabular format on the Data Source Administration page.

Note: DB2 Query Patroller only supports a single DB2 data source.

Editing a Data Source Definition

Use the following procedure to edit a data source definition:

1. On the Data Source Administration page, select a data source.
2. Click **View / Edit**. The Detailed Data Source Information window opens:

QueryAdministrator - Detailed Information for Data Source IWM

Data Source	IWM	Source Status	Active
Maximum Jobs	999999999	Cost Threshold	9999999
Node Disk Min	0	Node CPU Max	100
Node Jobs Max	999	Cost Factor	1
Cost Time Zero	0	Cost Time Slope	0
Cost Time Interval	0	Cost Time Min	0
Agent Name	iwm_sqlexec		

OK Cancel

3. Edit each field that you want to change. See the table below for a description of each field. For example, if you want to put the data source on hold or make the data source inactive, change the **Status** field to the appropriate value.

Note: Once a data source has been created, you cannot change its name.

4. Click **OK** after all new values have been entered.

Table 11. Fields to Edit a Data Source

Field	Description
Source	Provides the name of the data source. Only one data source, which must be named IWM, is recognized by DB2 Query Patroller.
Type	Indicates the DBMS type: UDB
Status	Provides the status of the data source: • Active indicates that the data source is accepting and scheduling new jobs. • Held indicates that the data source is accepting new jobs and placing them on hold. Queued, scheduled, and running jobs will be allowed to finish. • Inactive indicates that the data source is not accepting new jobs, no additional jobs will be scheduled, but running jobs will be allowed to finish.
Max Jobs	Indicates the maximum number of jobs that can be running simultaneously for this data source. Once this threshold of running jobs is reached, no more jobs will be scheduled for this data source. Max Jobs must be greater than or equal to 0.
Cost Threshold	Indicates the total cost threshold for jobs running simultaneously for this data source. Once this cost threshold for running jobs is reached, no more jobs will be scheduled for this data source. Cost Threshold must be greater than 0.
Node Disk Min	If disk utilization is being monitored for a node, jobs will not be scheduled to that node if the number of bytes available is less than this value. Node Disk Min must be greater than or equal to 0.
Node CPU Max	If CPU utilization is being monitored for a node, jobs will not be scheduled to that node if the CPU utilization is greater than this value. Node CPU Max must be an integer from 0 to 100.

Table 11. Fields to Edit a Data Source (continued)

Field	Description
Node Jobs Max	Indicates the maximum number of jobs that can be simultaneously scheduled on each node. Jobs will not be scheduled to a node if the total number already scheduled on that node is equal to or greater than this maximum. Node Jobs Max must be an integer from 1 to 9999.
Cost Factor	Provides the accounting multiplier for the estimated cost of a job, which is used when writing a job's estimated cost in the accounting table. Since the units for estimated cost are megabytes scanned, the units for the cost factor are dollars per megabyte. Cost Factor must be greater than or equal to 0.
Cost Time Zero	Provides the estimated time (in seconds) for a zero-cost query. If zero is entered, no time estimate is provided for a zero-cost query. Cost Time Zero must be greater than or equal to 0.
Cost Time Slope	If Cost Time Slope is greater than zero, the time estimate in seconds is computed as: static cost * cost time slope + cost time interval. If Cost Time Slope is zero, no time estimate is provided. Cost Time Slope must be greater than or equal to 0.
Cost Time Interval	See the description for Cost Time Slope.
Cost Time Min.	Provides a minimum estimated time. If Cost Time Min. is greater than zero and greater than the estimated time, then this minimum is used instead of the estimated time. If Cost Time Min. is zero, then the estimated time is set to Cost Time Zero as if the static cost were zero. Cost Time Min. must be greater than or equal to zero.
Agent Name	This field is set by default to: iwm_sqlexec and cannot be modified.

Chapter 12. User Administration

This chapter describes the procedures for creating, editing, and removing user profiles. All users submitting queries through the DB2 Query Patroller system must have a user profile.

Listing User Profiles

Use the following procedure to list all user profiles:

- 1. In the QueryAdministrator main window, select the **User Administration** tab.
 - 2. Click **List All Users** to list all user profiles.
- Each user profile is listed in a table on the User Administration page.

Creating a User Profile

Use the following procedure to create a user profile:

- 1. In the QueryAdministrator main window, select the **User Administration** tab.
 - 2. Click **Create User**.
- A user profile window opens:

The screenshot shows a window titled "QueryAdministrator - New User Data". It contains the following fields and controls:

- User ID:
- Type:
- Cost Analysis:
- Authority Level:
- Maximum Queries:
- Low Priority:
- Normal Priority:
- High Priority:
- User Threshold:
- Account ID:
- Email Addr:
- Buttons: OK, Cancel

- 3. Enter a value in each of the fields listed in the table below:

4. Click **OK** after all values have been entered.

Table 12. Fields to Define a User Profile

Field	Description
User ID	Provides the ID for the user or group. The user ID for the administrative user must be iwm.
Type	Indicates whether the profile is for a user or a group.
Cost Analysis	Provides a setting for cost analysis: • Do Cost Analysis indicates that cost analysis is performed on all jobs for this user or group. • Don't Do Cost Analysis indicates that the user can determine whether cost analysis is to be performed on specific jobs.
Auth Level	Provides the authority level for the user or group: • Not Allowed indicates that this user is not allowed to use DB2 Query Patroller. • User indicates that this user has normal authority to submit queries and manage the status of those jobs in the DB2 Query Patroller system. • Operator indicates that the user has operator authority. • Administrator indicates that this user has administrative authority to alter the DB2 Query Patroller configuration as well as operator authority. The user ID for the administrative user must be iwm.
Max Queries	Indicates the maximum number of the user's jobs that DB2 Query Patroller will run simultaneously. Max Queries must be an integer from 0 to 99.
Low Priority	Provides the priority assigned to a job when the job is submitted on low priority. Low Priority must be an integer from 0 to 999.
Normal Priority	Provides the priority assigned to a job when the job is submitted on normal priority. Normal Priority must be an integer from 0 to 999.
High Priority	Provides the priority assigned to a job when the job is submitted on high priority. High Priority must be an integer from 0 to 999.
User Threshold	Provides the threshold, which if exceeded by a job, causes the job to be placed on hold.
Account ID	Provides a user-defined accounting ID that is written in the Job Accounting table.

Table 12. Fields to Define a User Profile (continued)

Field	Description
Email Addr	Provides an address for notification when a job completes. If Email Addr is null, the user ID is used as the e-mail address.

Editing a User Profile

Use the following procedure to edit a user profile:

1. On the User Administration page, select a user profile.
2. Click **View / Edit**. The Detailed User Profile window opens.
3. Edit each field that you wish to change. See Table 12 on page 60 for a description of each field.

Note: Once a user profile has been created, you cannot change the **User ID** and **Type**.

4. Click **OK** after all new values have been entered.

Removing a User Profile

Use the following procedure to remove a user profile:

1. On the **User Administration** page, select a user profile.
2. Click **Remove**.
3. Click **Yes** to verify that you want to remove the selected user profile.

Note: User iwm cannot be deleted.

Chapter 13. Node Administration

This chapter describes the procedures for listing nodes and changing node status.

Listing All Nodes

Use the following procedure to list all nodes:

1. In the QueryAdministrator main window, select the **Node Administration** tab.
2. Click **List All Nodes**. Each node is listed in tabular format on the Node Administration page.

Changing the Node Status

Use the following procedure to change the node status:

1. On the Node Administration page, select a node.
2. Click **View / Edit**.

The Detailed Node Information window opens. (See the table below for a description of the fields in the Detailed Node Information window.)

Node ID	mohawk	Node Status	Active
Status Requested	Active	DateTime Last Status	February 23, 1999 5:04:46 PM EST
Scheduled Jobs	0	CPU Utilization	0
Disk Available	513,433,344	Node Manager PID	76734

3. Select the new status in the **Requested Status** field.

Note: **Requested Status** is the only field in the Detailed Node Information window that can be changed; all other fields display values that have been supplied by DB2 Query Patroller.

4. Click **OK**.

Table 13. Fields in the Detailed Node Information Window

Field	Description
Node ID	Provides the ID for the node.
Current Status	Contains the current node status • Active indicates that the node is able to run jobs. • Inactive indicates that the node is not available to DB2 Query Patroller. • Quiescing indicates that the node is in transition to the quiescent state. Running jobs will complete but no new jobs will be scheduled. • Quiesced indicates that the node is quiescent. The node is available to DB2 Query Patroller but no new jobs are being scheduled.
Requested Status	Indicates what the node status will be changed to: • Active indicates that the node will be made active. • Inactive indicates that the node will be made inactive. Running jobs will complete and no new jobs will be scheduled. • Force indicates that the node will be made inactive immediately. Running jobs are terminated immediately and no new jobs will be scheduled. • Quiesced indicates that the node will be made quiescent. Running jobs will complete.
Date/Time Last Status	Indicates the date and time the status was last changed.
Scheduled Jobs	Provides the number of jobs scheduled to run plus the number of jobs running on this node.
CPU Utilization	Provides the CPU utilization of the node as a percentage (0 - 100). If CPU utilization information is not being collected, the value is -1.
Disk Available	Indicates the bytes available in the file system where results are created. If disk utilization is not being monitored, the value is -1.
Node Manager PID	Indicates the process ID of the node_mgr process.

Chapter 14. Job Queue Administration

This chapter describes the procedures for creating, editing, and removing job queues. This chapter is divided into the following sections:

- “Viewing Job Queues”.
- “Creating a Job Queue”.
- “Editing a Job Queue” on page 66.
- “Removing a Job Queue” on page 66.

Viewing Job Queues

Use the following procedure to view the job queues:

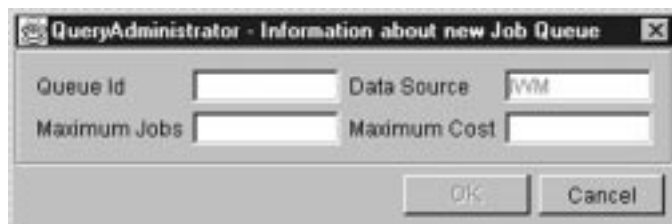
1. In the QueryAdministrator main window, select the **Job Queues** tab.
2. Click **View / Edit** to list the system job queues. Each job queue is listed in tabular format on the Job Queues page.

Creating a Job Queue

Use the following procedure to create a job queue:

1. In the QueryAdministrator main window, select the **Job Queues** tab.
2. Click **New**.

An empty job queue information window opens. (See the table below for a description of the fields in the in the job queue information window.)



Field	Value
Queue Id	
Data Source	JWM
Maximum Jobs	
Maximum Cost	

3. Enter a value in the fields listed in the table below:
4. Click **OK** after the values have been entered.

Table 14. Caption. Description

Field	Description
Queue Id	Provides an ID for the job queue. This value must be unique.
Maximum Jobs	Maximum number of jobs allowed to run in the queue.
Data Source	The only data source supported is IWM. This field cannot be altered.
Maximum Cost	Indicates the maximum query cost that will be accepted by the job queue. DB2 Query Patroller assigns jobs to the queue that has the smallest cost limit that is not less than the cost of the job.

Editing a Job Queue

Use the following procedure to edit a job queue:

1. On the Job Queues page, select a job queue.
2. Click **View / Edit**. A detailed job queue window opens.
3. Edit the queue ID, the maximum jobs value, or the maximum cost value.
4. Click **OK**.

Removing a Job Queue

Use the following procedure to remove a job queue:

1. On the Job Queues page, select the job queue you want to remove.
2. Click **Remove**.
3. Click **Yes** to verify that you want to remove the selected job queue.

Chapter 15. Result Destination Administration

This chapter describes the procedures for creating, editing, and removing result set destinations. This chapter is divided into the following sections:

- “Listing Result Destinations”.
- “Creating a Result Destination”.
- “Editing a Result Destination” on page 68.
- “Removing a Result Destination” on page 68.

Listing Result Destinations

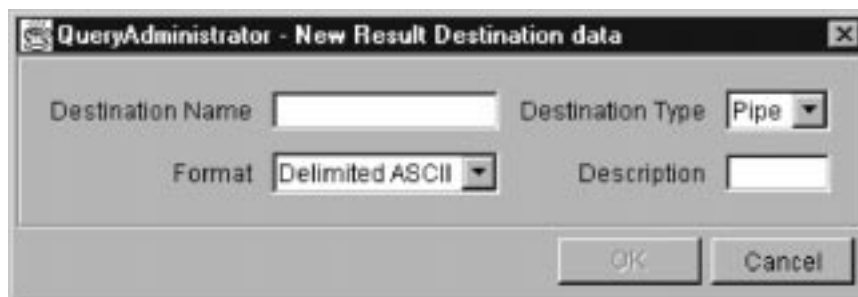
Use the following procedure to list result destinations:

1. In the QueryAdministrator main window, select the **Result Set Administration** tab.
2. Click **List All Result Sets** to list all result destinations. Each result destination is listed in tabular format on the Result Set Administration page.

Creating a Result Destination

Use the following procedure to create a result destination:

1. In the QueryAdministrator main window, select the **Result Set Administration** tab.
2. Click **New**. An empty result destination window opens.



3. Enter a value in each of the fields listed in the table below.

4. Click **OK** after all values have been entered.

Table 15. Fields to Define a Result Destination

Field	Description
Name	Provides the name for the result destination. QueryEnabler displays these names in the Result Destinations field when a user schedules a new query.
Type	Indicates the type of destination: • File indicates that the destination is a file. • Pipe indicates that the destination is an executable program.
Format	The only format supported is delimited ASCII.
Description	For a file, the description provides the name of the file. For a pipe, the description provides the command line for the executable program. Descriptions for both types may contain any of the following substitution variables: • \$(USER) is the UNIX user ID. • \$(HOME) is the user's home directory. • \$(JOB) is the DB2 Query Patroller job number. In addition, for a pipe, the \$(PIPE) variable represents a named pipe. Instead of writing the data to the command's STDIN, DB2 Query Patroller writes the data to the named pipe.

Editing a Result Destination

Use the following procedure to edit a result destination:

1. On the Result Set Administration page, select a result destination.
2. Click **View / Edit**. The Detailed Result Destination Information window opens.
3. Edit the destination type, the description, or both.
4. Click **OK**.

Note: The Destination Name cannot be changed. Only the type and description of the destination can be changed.

Removing a Result Destination

Use the following procedure to remove a result destination:

1. On the Result Set Administration page, select the result destination you want to remove.
2. Click **Remove**.
3. Click **Yes** to verify that you want to remove the selected result destination.

Chapter 16. System Administration

This chapter describes the procedures for listing and editing system parameters.

Listing System Parameters

Use the following procedure to list the system parameters:

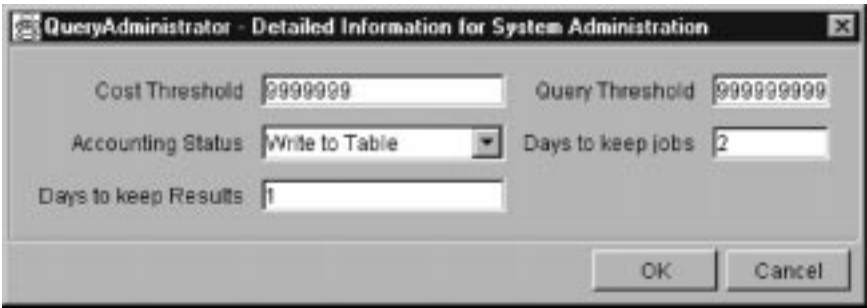
- 1. In the QueryAdministrator main window, select the **System Administration** tab.
- 2. Click **List System Administration** to list the system parameters.

Editing System Parameters

Use the following procedure to edit system parameters:

- 1. In the QueryAdministrator main window, select the **System Administration** tab.
- 2. Click **List System Administration** to list the system parameters.
- 3. Click **View / Edit**.

The Detailed System Information window opens:



- 4. Enter a value in each of the fields listed in the following table:
- 5. Click **OK** after all values have been entered.

Table 16. System Parameters

Field	Description
Cost Threshold	If the total estimated cost of all running queries is greater than the cost threshold, no more queries will be scheduled to run on any node.

Table 16. System Parameters (continued)

Field	Description
Query Threshold	If the total count of running queries is greater than the query threshold, no more queries will be scheduled to run on any node.
Accounting Status	Indicates whether information is written to the job accounting table for use by Tracker utility. Write to Table indicates that accounting is on; Do Not Write to Table indicates that accounting is off.
Days to Keep Job	Indicates the number of days to retain an entry in the job table after the job completes. Use 0 to retain job entries indefinitely.
Days to Keep Results	Indicates the number of days to retain result sets after the job completes. Use 0 to retain result sets indefinitely.
CPU Threshold	This field is not used. The Node CPU Max field on the Data Source Administration page defines the CPU threshold for each node.

Part 3. The Tracker Utility

Chapter 17. Introduction to Tracker

The Tracker application provides reports that display a database usage history for queries that have been managed by the DB2 Query Patroller system. For example, Tracker allows you to determine which tables and columns have been accessed most frequently, which tables have returned the most result rows, and which jobs have been completed within a specific time period.

Creating Tracker Data on the Database Server

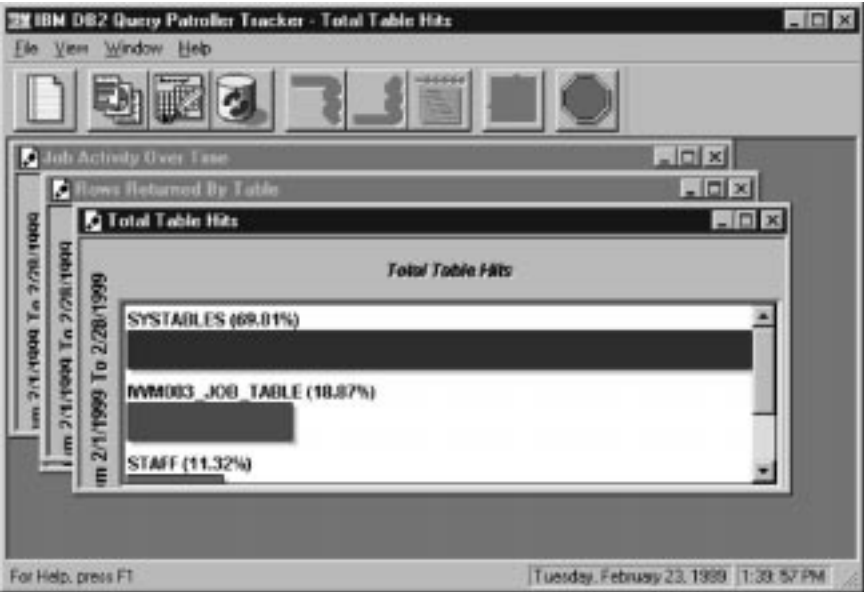
The information displayed by Tracker is gathered by DB2 Query Patroller running on your database server. DB2 Query Patroller collects job accounting data, which is then analyzed by `iwm_tracker`. The `iwm_tracker` program, the back-end parser for DB2 Query Patroller, analyzes data for successfully completed job runs. Each time `iwm_tracker` runs, it examines any job accounting entries added since its last run.

Before running Tracker for the first time, you must set up `iwm_tracker` as follows:

1. In DB2 Query Patroller, turn on job accounting data collection. Use the QueryAdministrator utility, or the command line interface to set the Accounting Status system parameter to active. This must be completed before jobs are submitted. Information will be gathered for the Tracker utility only after the Accounting Status system parameter is set to active.
2. Determine when the `iwm_tracker` program runs. Choose a time period that is appropriate for the query volume and usage pattern of your database. You might want to run `iwm_tracker` at night or on the weekend, when the database load is light.
3. Run `iwm_tracker` at least once to ensure that data is available the first time you run the Tracker utility.

Navigating within Tracker

The Tracker screens display in a multiple document interface (MDI) format. The main, or parent, window contains one or more child windows. The child windows cannot be moved outside the parent window. The Tracker parent window and examples of child windows are illustrated below:



The child window displayed in the foreground is the current (active) window. You can designate any child window as active by clicking on the window.

Understanding the Tool Bar

The tool bar of the parent window contains push buttons that provide shortcuts for performing Tracker functions, which are also available as menu items. Each push button contains an icon, which is a graphical representation of the function performed by clicking the push button.

Table 17. Tracker Tool Bar Push Buttons and Descriptions

Push Button	Description
	New Report push button: Opens a report in a new window using the current report format and date range.

Table 17. Tracker Tool Bar Push Buttons and Descriptions (continued)

Push Button	Description
	Change Report Format push button: Changes the format of the report in the active window and all subsequent windows, including the next time you start up Tracker.
	Set Date Range push button: Sets the date range and time for the report displayed in the active window and all subsequent windows, the altered format will be retained the next time you start up Tracker.
	Refresh Data push button: Updates the data in the active window by rerunning the query that produced the report.
	Drill Down push button: Drills down to the next level of reports available from this report. Drill Down is active only when a drill-down report is available.
	Drill Up push button: Drills up to the report at the previous level. Drill Up is active only after you have drilled down to a lower-level report.
	Drill Path push button: Displays the path you took to display the report in the active window. Drill Path is active only after you have drilled down to a lower-level report.
	Zoom +/- push button: Zooms in and out on the window. This push button toggles to display the Pan push button after being selected. You can zoom in and out on either a pie chart or a Job Activity over Time report.
	Pan push button: Pans on the contents of the window. This push button toggles to display the Zoom push button after being selected. You can pan on either a pie chart or a Job Activity over Time report.

Chapter 18. Tracker Reports

This chapter describes the reports available in Tracker including: the Total Table Hits report and the drill-down reports beneath it, the Rows Returned by Table report, and the Job Activity over Time report.

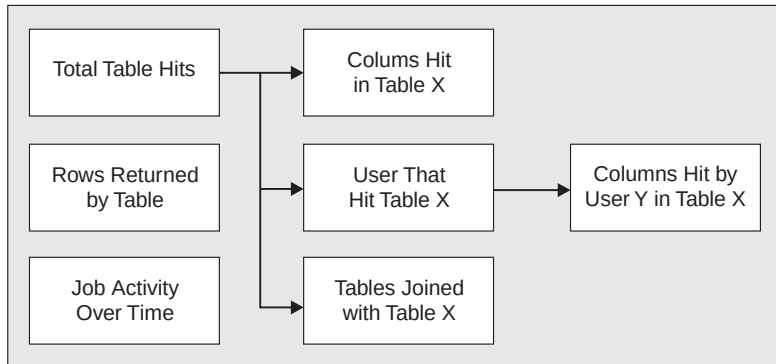
Understanding the Reports

The Tracker query utility allows you to display database usage history information through predefined reports. These reports provide you with job history information from DB2 Query Patroller if DB2 Query Patroller has been set up to collect job accounting information.

When you enter the Tracker application, you choose an initial report for Tracker to display. You can choose to display the following reports:

- Total Table Hits
- Rows Returned by Table
- Job Activity Over Time

After a report displays, you can drill down to display more specific information if additional reports are available. The reports and drill paths you can choose are shown in the following diagram:



The Total Table Hits report and the Rows Hit by Table report display count or percentage information that can be formatted as bar charts, lists, or pie charts. As you drill down through subsequent reports, each report displays in the same format (bar chart, pie chart, or list), until you change the current report

format. The Job Activity over Time report displays time information and a list of completed jobs. It can only be displayed as a time chart.

Displaying Data as a Count

When you display data as a count, the data is an integer. For example, when calculating the tables hit, Tracker counts the number of times each table is accessed.

Displaying Data as a Percentage

When you display data as a percentage, the data displayed is a percentage of the total count for all the elements that have been accessed. If a similar element is not accessed, it is not included in the percentage calculation. For example, when calculating the percentage for table hits, Tracker counts all the hits for all tables. The percentage for a particular table is the number of hits for that table divided by the total number of hits multiplied by 100%.

Total Table Hits Report

The Total Table Hits report displays the number of times each table has been accessed (or hit) by queries submitted against the database within the defined date range. Alternatively, the report shows the percentage of hits per table relative to the total number of hits on all tables.

Note: Even though the hours do not display in the date range at the left side of this and other reports, the hours can be specified and are used to determine the date range for the information in the report.

From the Total Table Hits Report, you can drill down to get the following reports:

- Columns Hit in Table X
- Users That Hit Table X
- Tables Joined with Table X

Columns Hit in Table X Report

The Columns Hit in Table X report lists each column in a specific table and the number of times the column has been accessed by queries within the defined date range. Alternatively, the report shows the percentage of hits per column relative to all columns in the table.

Users That Hit Table X Report

The Users That Hit Table X report lists the ID of each user accessing a specific table and the number of times the table is accessed by that user within the

defined date range. Alternatively, the report shows the percentage for each user who accessed a table relative to all users who accessed the table. From this report, you can drill down to the columns hit in that table by a specific user.

Columns Hit by User Y Hitting Table X Report

The Columns Hit by User Y Hitting Table X report lists the columns accessed by a specific user for a specific table and the number of times those columns were accessed within the defined date range. Alternatively, the report shows the percentage each column was accessed by a specific user relative to the total accesses to all columns in that table by that user.

Tables Joined with Table X Report

The Tables Joined with Table X report lists the tables joined with the original table through a query and the number of times a join occurred within the defined date range. Alternatively, the report shows tables joined with the original table and the percentage the joins represent relative to the total number of accesses to the original table.

Rows Returned by Table Report

The Rows Returned by Table report lists the name of each table accessed and the number of rows returned for all queries submitted against the table within the defined date range. Alternatively, the report shows the number of rows returned from each table as a percentage of the total number of rows returned from all tables.

Job Activity Over Time Report

The Job Activity over Time report displays the following information:

- **Job Submit Time**
- **Job Start Time**
- **Job End Time**
- **Job ID** (job ID number)
- **User ID** (user who submitted the job)
- **Account ID**

The Job Activity over Time report displays all the jobs that were completed between the defined start and end dates. The graph at the top of the report summarizes the data for the completed jobs. The scroll list at the bottom of the report provides details on each job. The job submit time, job start time, and job end time are displayed graphically and in the detailed job list, which also provides the job ID, user ID, and account ID.

The job list shows jobs sorted by job elapsed time, so that you can easily identify the jobs that are taking a long time to complete. The graph displays lead time in yellow and run time in red.

For additional job detail information, such as execution cost, the number of result rows, result set destination, and the SQL statement, double-click on a specific job in the scroll list to open the Job Detail Information window.

Chapter 19. Using Tracker

This chapter describes how to start and stop Tracker and how to use Tracker to display job history data.

Starting Tracker

Complete the following steps to start the Tracker application:

1. From the **Start** menu, select **Programs > DB2 > DB2 Query Patroller > Tracker**.

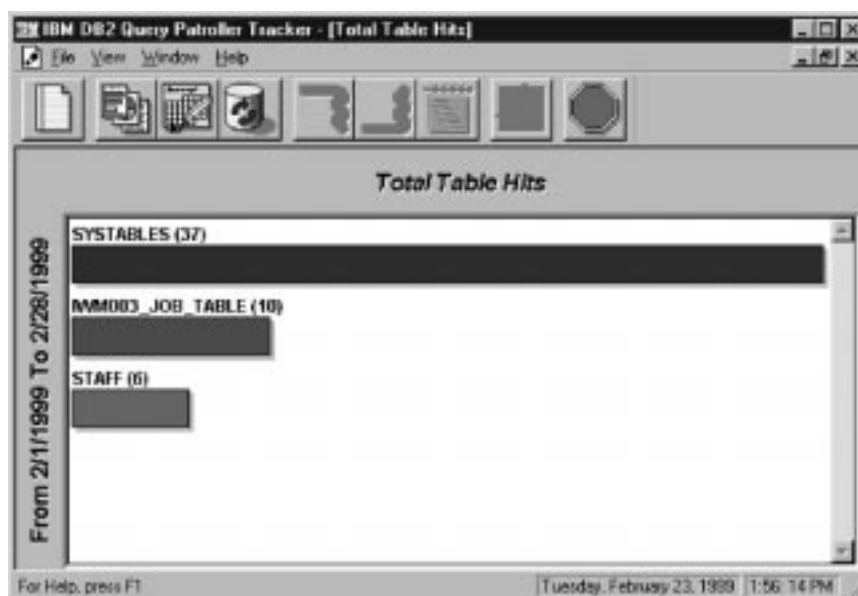
The Login to Tracker window opens.

2. Type your user ID, password, and database alias, and click **OK**. A list of reports display.

Note: The user ID and password are case-sensitive.

3. Select a report, and click **OK**.

The report displays using the default date range and the default report format as illustrated below. For information on changing the date range, see “Changing the Date and Time Range” on page 84. For information on changing the report format, see “Report Formats” on page 86.



Note: Even though the hours do not display in the date range at the left side of this and other reports, the hours can be specified and are used to determine the date range for the information in the report.

Displaying Job History Data

Tracker allows you to view job history data so that you can tune the database for heavily used tables and columns. For a description of each report available in Tracker see “Chapter 18. Tracker Reports” on page 79.

Tracker allows you to display reports in up to six child windows, a feature that allows you to compare data among reports. You can set the date and time range and you can also change the way the data displays for most reports. For example, the Total Table Hits report, its drill-down reports, and the Rows Returned by Table report allow you to view information in a list (as a count or as a percentage), in a bar chart (as a count or as a percentage), or in a pie chart (as a percentage only). The Job Activity over Time report allows you to view jobs completed in a time-chart format.

Opening a New Report

Complete the following steps to display a report:

1. Click **New Report** to display a list of available reports.
2. Select a report and click **OK** to display data for the defined date range. See “Chapter 18. Tracker Reports” on page 79 for more information about the reports.

Changing the Date and Time Range

A default date range exists for an entire Tracker session, which means that each new report opened during that session uses the default date range. While the date range can be changed for a particular window, the default date range cannot be changed within a session. Therefore, if you change the date range for a report, it neither resets the date range for the next report you open within the session nor does it necessarily reset the default date range for the next session. Changing the date range for the last report opened in a session is the only way to reset the default date range for the next session.

When you open a report, the data displays for the default date range. You can change the date range for the current window (which may change the data displayed in the report). You can set different date ranges for each window to view reports for different time periods.

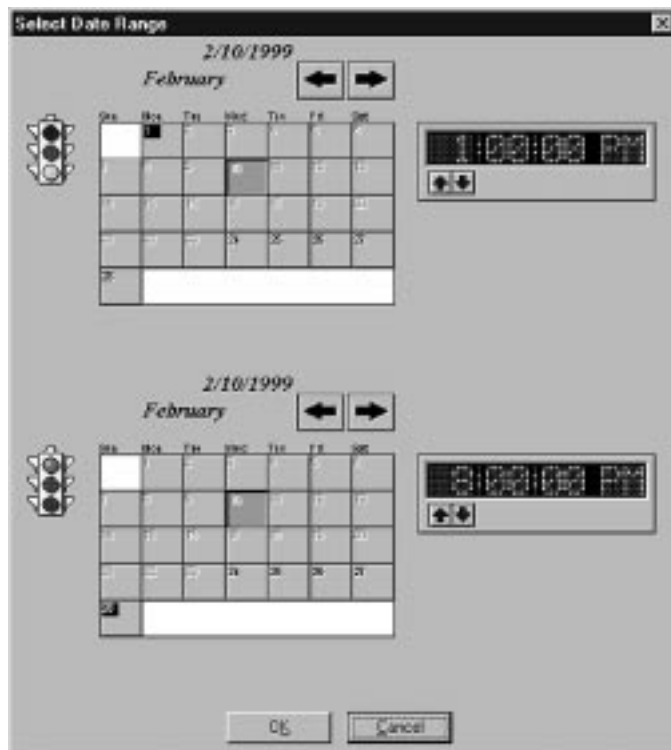
Note: Even though the hours do not display in the date range at the left side of each report, the hours can be specified and are used to determine the date range for the information in the report.

In addition, if you change the date range in a Job Activity over Time report, the scale of the displayed data may also change. For example, if the date range is less than 30 days, the increments displayed in the Job Activity over Time report are days, as illustrated below. See the procedure at the end of this section and also see "Zooming and Panning on a Time Chart" later in this chapter for information on how to change the date range for a Job Activity over Time report.

If the date range is greater than a year, the increments displayed in the Job Activity over Time report are years. If the date range is 24 hours or less, the increments displayed in the Job Activity over Time report are hours.

Complete the following steps to change the date range for a specific window:

1. Click the **Set Date Range** push button. The Set Date Range window opens, as illustrated below. This dialog box contains two calendars and two 12-hour clocks so that you can set the start and end dates and times.



2. Define the start date and time by completing the following steps:
 - Use the push buttons with the arrow icons to change the month of the Start calendar.
 - Click a day in the start calendar to select the start date.
 - Click the up or down push button to change the time (by hour) on the Start calendar.
3. Define the end date and time in the same way using the End calendar.
4. Accept the date range by clicking **OK**.

Refreshing Data

Complete the following steps to refresh the data for a report:

1. Select a window to make it active.
2. Click the **Refresh Data** push button to update the data in the window.

Note: For new data to be available to display, iwm_tracker (the back-end process for Tracker) must have been run to analyze any job accounting data that accrued since its last run.

Report Formats

The Total Table Hits report and the Rows Returned by Table report allow you to view data as bar charts (as a count or as a percentage), lists (as a count or as a percentage), or pie charts (as a percentage only).

Bar Charts

Bar charts display data as a count or as a percentage. The horizontal bars are arranged vertically, from largest to smallest. An example of a bar chart is illustrated. Alternatively, you can choose to view a bar chart with percentage information.

Lists

Lists display data as a count or as a percentage. The element with the highest value is at the top of the list. The same list can be viewed with count information.

Pie Charts

Pie charts show percentage data only. Each slice of the pie represents the percentage an element was accessed relative to the total accesses for all elements. Each wedge of the pie chart is also listed in a legend that lists the percentage value for each slice.

Changing the Report Format

Complete the following steps to change the default report format, which sets the format for the current report and subsequent reports, including the next time you run Tracker:

1. Select a window to make it active or open a new report.
2. Click the **Change Report Format** push button.

The **Change Report Format** window opens.

3. Select a format, and click **OK**.

Drilling Down for More Information

You can drill down to more detailed information if a drill-down report has been defined. See “Chapter 18. Tracker Reports” on page 79 for information on the reports available for drill down.

Complete the following steps to drill down to more specific data:

1. Select a window to make it active or open a new report.
2. Select an element from the report: a bar, a line in a list, or a bar in the pie chart legend. If a drill-down report is available, the **Drill Down** push button becomes active.
3. Click the **Drill Down** push button.

If only one drill-down report exists at the next level, the new report displays in the current window. Otherwise, if you can drill down to more than one type of report, a list displays as illustrated below:



4. Select the report that you want to view and click **Drill**.

The new report displays in the current window. The report uses the date range of the parent report and the current report format.

Drilling Up

If you have drilled down, you can drill back up through the same path. To drill back up through the reports, complete the following steps:

1. Select a window to make it active.
If a higher level of reports exists, the **Drill Up** push button becomes active.
2. Click the **Drill Up** push button to display the report from one level up.
The report displays in the current window using the current report format and the date range established for this window.

Viewing the Drill Path

You can view the drill path you followed for the active window. Complete the following steps to display the drill path:

1. Select a window to make it active.
If a drill path is available, the **Drill Path** push button becomes active.
2. Click the **Drill Path** push button to display the drill path for the active window.
The Drill-Down Path Display window opens. The currently displayed report is listed at the bottom of the drill path.
3. Click **OK** to close the Drill-Down Path Display window.

Zooming and Panning

Use the **Pan** push button or the **Zoom +/-** push button to toggle between pan and zoom modes. The active push button displays the mode to be selected, not the current mode.

Zooming and Panning on a Pie Chart

Complete the following steps to zoom and pan on a pie chart:

1. Select a window to make it active or to open a new report.
2. If the format is not a pie chart, change the format to a pie chart.
3. In zoom mode, place the cursor over the area that you want to zoom in on and click.
The pie chart displays enlarged, with the click point at the center of the display.
4. After you zoom in on an image, you can switch to pan mode by clicking the **Pan** push button.
5. In pan mode, place the cursor over the area that you want to be the center of the screen and click.
The area over which you placed the cursor becomes the center of the screen.
6. Click the **Zoom** push button to switch to zoom mode.
7. To zoom out, right-click.
The entire pie chart displays at the normal size

Zooming and Panning on a Time Chart

Complete the following steps to zoom and pan on a time chart:

1. Select a Job Activity over Time window to make it active or open a new Job Activity over Time report.
2. In zoom mode, place the cursor over the column dividers just below the scale legend on the report; the zoom cursor becomes visible.
3. Click to zoom in.
Zooming in increases the scale and decreases the number of days displayed in the report. For example, if the current scale displays data for 30 days, and you zoom in, the number of days displayed decreases to approximately 18 days.
4. After you zoom in on an image, you can switch to pan mode by clicking the **Pan** push button.
5. In pan mode, place the cursor over the time segment that you want to be the center of the screen and click.
The time segment over which you placed the cursor becomes the center of the screen and the date range is updated to reflect the change.
6. Click the **Zoom** push button to switch to zoom mode.
7. To zoom out, right-click.
Zooming out decreases the scale and increases the number of days displayed in the report.

Closing a Tracker Report Window

To close a report window, complete the following steps:

1. Select the window you want to close to make it active.
2. From the **File** menu, select **Close**.

Exiting Tracker

To exit Tracker, click the **Exit Tracker** push button. The push button is displayed below:



Part 4. Appendixes

Appendix A. Troubleshooting the DB2 Query Patroller Server

This chapter provides information to assist in troubleshooting the DB2 Query Patroller server and describes common problems that may be encountered.

Distinguishing Between DB2 and DB2 Query Patroller Failures

DB2 Query Patroller is composed of components that run on the DB2 Query Patroller server, and components that run on the DB2 Query Patroller client. The client does not communicate directly with the server. DB2 Query Patroller uses DB2 tables to pass information between the client and the server. For this reason an administrator or operator needs to be able to distinguish between DB2 failures and DB2 Query Patroller failures.

Refer to the *Troubleshooting Guide* if a problem or system failure is suspected to have originated from DB2.

syserr.log

The DB2 Query Patroller server maintains a system error log named syserr.log in the path referenced by the environment variable IWMLOG. This file contains diagnostic information for the DB2 Query Patroller server and may explain the cause of the system failure.

Note: DB2 Query Patroller agents should be configured to log their diagnostic information to a unique location. This will assist in determining where the error originated.

DB2 Diagnostic Log

If the DB2 database manager configuration parameter DIAGLEVEL is set to 4, all SQLCAs with a negative SQLCODE will be dumped to the diagnostic log. This information is useful when trying to determine why the DB2 Query Patroller server failed when attempting to access DB2.

Processes

DB2 Query Patroller Server

On a properly functioning DB2 Query Patroller server, the following processes will be active:

- iwm_server
- iwm_nodemgr
- iwm_local (one or more)
- iwm_sched
- iwm_remote
- iwm_net

DB2 Query Patroller Agent

An DB2 Query Patroller agent will have the following processes active:

- iwm_net
- iwm_nodemgr

Note: Other DB2 Query Patroller processes are transitory and will be started and stopped as required.

Process Failures

If any of these processes have quit before completion, the DB2 Query Patroller system should be restarted. This process should be performed by systematically shutting down all DB2 Query Patroller agents followed by the DB2 Query Patroller server. The DB2 Query Patroller system can then be restarted by bringing up the server first, followed by the DB2 Query Patroller agents.

Under extreme circumstances, it may be necessary to generate a detailed DB2 Query Patroller server trace file to send to IBM support. In order to enable tracing, start the DB2 Query Patroller server or agent and implementing the **trace** option. More specifically, for the DB2 Query Patroller server specify **iwm start all trace**, and for an DB2 Query Patroller agent specify **iwm start agent trace**. The detailed trace will be written to the syserr.log file in the path specified by the DB2 profile variable IWMLOG. The trace results will continually be written to the log until DB2 Query Patroller is stopped and restarted without the tracing option.

Note: This file can grow quickly and should reside on a file system with a large amount of free disk space.

DB2 Query Patroller is also integrated with DB2 trace. The IBM service organization may request a DB2 trace in order to help identify and localize a problem. Refer to the *Troubleshooting Guide* for details on how to perform a DB2 trace.

Common Server Problems

1. The DB2 Query Patroller server fails to start:
 - Make sure DB2 is started.
 - Make sure that the DB2 Query Patroller TCP/IP port is not being used by another process.
 - Check environment variables to ensure that they are set correctly.
 - Check the syserr.log file for specific process errors.
2. The DB2 Query Patroller agent fails to start:
 - Ensure that the DB2 Query Patroller agent and DB2 Query Patroller server communications configuration specify the same TCP/IP port.
 - Check the setting of the DB2 profile variable IWM_Server to ensure that the correct TCP host name for the DB2 Query Patroller server is specified.
 - Ensure that the TCP/IP port is not in use on the DB2 Query Patroller Agent system.
3. A job is in the queue but will not run:
 - Check the DB2 Query Patroller error log to make sure there have been no DB2 Query Patroller process failures.
 - Check the DB2 Query Patroller system configurations to ensure that they are set correctly.
 - Restart the DB2 Query Patroller server.

Appendix B. Troubleshooting DB2 Query Patroller Clients

This chapter provides information to assist in troubleshooting the DB2 Query Patroller client and describes common problems that may be encountered.

The DB2 Query Patroller client may consist of the following tools: QueryEnabler, QueryMonitor, QueryAdministrator, and Tracker. Although there are different methods for troubleshooting each tool, because the tools are DB2 applications, methods used to troubleshoot DB2 clients as documented in the *Troubleshooting Guide* may often be applicable.

Troubleshooting QueryEnabler

The QueryEnabler is both an ODBC driver and an ODBC application. When trapping user queries, it acts as an ODBC application. When performing database access to DB2, it acts as an ODBC driver.

Whenever a problem occurs, you should check the following:

- the DB2 server is started
- the application can connect to the DB2 server directly using the DB2 cataloged names
- the QueryEnabler data source name is configured correctly to use the correct DB2 data source
- the application is using the correct QueryEnabler data source name

Note: You may also want to check the DB2 diagnostic log for DB2-related problems.

In order to help identify problems, IBM support representatives may employ the following tracing mechanisms collectively or separately:

- ODBC trace via the ODBC Data Source Administrator
- DB2 CLI trace
- DB2 trace
- DB2 QueryEnabler trace

The first three trace facilities are described in the *Troubleshooting Guide*. Refer to this document for detailed instructions.

DB2 QueryEnabler trace is enabled by editing the INI file DB2IWM.INI located in the Windows system directory and changing the value of LOG from FALSE to TRUE.

After this modification, the application accessing QueryEnabler should be restarted and the problem under investigation should be repeated to generate a diagnostic log called DB2IWM.LOG located in the root of the C partition, i.e. C:\DB2IWM.LOG.

This file will contain all ODBC function calls accessed in QueryEnabler by the application. QueryEnabler will continue to log information to this file until the LOG parameter in the DB2IWM.INI file is reset to FALSE. Ensure that there is sufficient free disk space on the C partition to support this file.

Common QueryEnabler Problems

1. The application is unable to connect to DB2:
 - Verify that the application is able to connect to DB2 using the DB2 data source names. If this fails, refer to the *Troubleshooting Guide* for information on how to resolve a connectivity problem.
 - Verify that the QueryEnabler data source is configured to use the correct DB2 data source.
 - Ensure that the user account and user passwords are correct.
2. The application is unable to access the QueryEnabler query submission dialogs:
 - Ensure the application is using the correct QueryEnabler data source name.
3. The QueryEnabler dialogs appear unexpectedly:
 - QueryEnabler will trap all DB2 queries through the ODBC interface except for queries accessing the DB2 system catalogs. Some applications will query DB2 in support of the application. These queries supplement the main user query. QueryEnabler cannot distinguish between the main user query and the supplemental query. In this event, select the **Submit and Wait** button to execute these queries. This allows time for the result set to be constructed and unwanted QueryEnabler dialogs do not appear.
4. The query application is unable to submit a query to the DB2 Query Patroller server using QueryEnabler:
 - Ensure the DB2 Query Patroller server stored procedures have been installed on the DB2 server.
 - Ensure the user has been granted the proper DB2 access to the stored procedures.
5. The application has submitted a query to the DB2 Query Patroller server but the query is not being executed:

- Ensure that the DB2 Query Patroller server is running correctly. It may be necessary to restart the DB2 Query Patroller server.
- Use QueryMonitor to inspect the details of the Job. The job may be held because the query exceeded user thresholds. An DB2 Query Patroller administrator or operator must be contacted to release this job.
- The job may be in the queue state because system resources have exceeded configured limitations. Under these conditions, the query will execute once the system resources fall within acceptable limits as configured by the DB2 Query Patroller administrator.

Troubleshooting QueryMonitor and QueryAdministrator

QueryMonitor and QueryAdministrator are Java applications that access DB2 using JDBC. Diagnostic information can be obtained from four sources:

- the DB2 diagnostic log
- ODBC traces
- CLI traces
- DB2 traces

Typically those trace tools are launched using a Java Virtual Machine (JVM) that does not have a console (JREW.EXE). These trace tools may also be run in a JVM that has a console (JRE.EXE), in which case DB2 Query Patroller will print information to the console. You can use that information to help identify the cause of a problem.

Common QueryMonitor and QueryAdministrator Problems

1. The tool is unable to connect to DB2:
 - Ensure that the DB2 data source name is used in the connect dialog.
 - Ensure that you have supplied the correct user name and password.
2. The application stops responding:
 - Close and restart the application. If the problem persists, attempt to run the application from a JVM with a console to obtain additional diagnostic information.

Troubleshooting Tracker

Tracker is an ODBC application that accesses the DB2 Query Patroller schema in DB2 in order to provide reports that display database usage history for queries that have been managed by DB2 Query Patroller.

Diagnostic information for Tracker can be obtained from four sources:

- the DB2 diagnostic log

- ODBC traces
- CLI traces
- DB2 traces

Common Tracker Problems

1. Tracker is unable to access DB2:
 - Ensure that the correct DB2 data source name, user name, and password have been specified.
2. Tracker does not display any data:
 - Ensure that the DB2 Query Patroller server has been configured to capture this information.
 - From QueryAdministrator, check that the Accounting option has been enabled.
 - Ensure that the date range specified in the Tracker application is set correctly to a time period when queries were in progress.
3. The data displayed by Tracker is not up-to-date:
 - Ensure the Tracker server on the DB2 Query Patroller server is scheduled to execute periodically. The Tracker server component may be executed interactively to generate up-to-date statistics.

Appendix C. DB2 Query Patroller Messages

This chapter provides information about the log file and the messages that display in DB2 Query Patroller.

Messages

DB29501E **DB2 has detected an inconsistent environment. Please investigate the following : "<error>"**

Explanation: An error in the operating environment has been detected.

User Response: This error may arise due to an inconsistent DB2 or operating system environment. Attempt to correct the problem identified in the error message and re-issue the command.

DB29502E **An incompatible Java Runtime Environment has been detected. Please verify that the required level of the Java Runtime Environment has been installed. The Java application may be launched using the Java runtime installed with DB2.**

Explanation: The installed Java runtime is not supported by the DB2 Java application.

User Response: The installed DB2 Java runtime will be used to launch the DB2 Java application. Other Java applications using the installed Java runtime should be terminated prior to executing the DB2 Java application to ensure correct behaviour.

DB29526E **You cannot drop this table; it does not exist. Please select a job that has results, and try again, or select a recurring job that is scheduled and try again.**

Explanation: A request was made to drop a result table for a job which did not have an associated result table.

User Response: Select a job that has a results or select a recurring job that is scheduled and try again.

DB29543E **Resource DLL rqsres.dll not found.**

Explanation: Resource DLL rqsres.dll could not be loaded successfully.

User Response: During initialization of DB2 Query Patroller Recurring Query Scheduler the resource DLL rqsres.dll could not be loaded as it was either damaged or deleted. Please reinstall DB2 Query Patroller QueryEnabler.

DB29544E **Resource DLL qeres.dll not found.**

Explanation: Resource DLL qeres.dll could not be loaded successfully.

User Response: During initialization of DB2 Query Patroller QueryEnabler the resource DLL qeres.dll could not be loaded as it was either damaged or deleted. Please reinstall DB2 Query Patroller QueryEnabler.

DB29545E **NAME INVALID: Query Name can only be 20 characters or less in length. Please change the name accordingly and retry.**

Explanation: A name for a query was specified which was longer than 20 characters.

User Response: Specify a query name that has at most 20 characters.

DB29546E **NAME INVALID: Query Name can only contain alphanumeric characters and spaces...(\"a..z\", \"n\"A..Z\", \"0..9\"). Please change the name accordingly and retry.**

Explanation: A name for a query was specified which contained non-alphanumeric characters.

User Response: Specify a query name composed of only alphanumeric characters.

DB29608E **Script Load Failed.**

Explanation: Script files required by DB2 Query Patroller Tracker could not be found.

User Response: Please reinstall DB2 Query Patroller Tracker.

DB29610E **Could not load the ODBC Driver Manager.**

Explanation: The ODBC driver dll ODBC32.DLL could not be loaded.

User Response: The ODBC driver dll ODBC32.DLL could not be loaded as it was damaged or destroyed. Please reinstall ODBC.

DB29611E **Three Login Attempts Failed! Exiting Tracker!**

Explanation: DB2 Query Patroller Tracker requires to connect to the ODBC Data Source specified on the login dialog. The userid, password combination specified on the login dialog was not authorized to connect to the specified database.

User Response: Please relaunch DB2 Query Patroller Tracker and provide a valid userid password combination for the ODBC Data source specified on the login dialog.

DB29613E **Resource DLL tkres.dll not found.**

Explanation: DB2 Query Patroller Tracker requires the resource dll tkres.dll for successful operation. The dll could not be loaded.

User Response: During initialization of DB2 Query Patroller Tracker the resource DLL tkres.dll could not be loaded as it was either damaged or deleted. Please reinstall DB2 Query Patroller Tracker.

DB29700E **The application programming interface "<program>" did not complete successfully for object "<name>" with reason code "<name>". Try again or contact your local support.**

Explanation: An application programming interface call did not complete.

User Response: Try the operation again or contact your system administrator.

DB29701E **Class "<class name>" not found. Please check you classpath.**

Explanation: The "<class_name>" cannot be located in the specified classpath.

User Response: Verify the classpath is correct. If you are using QueryAdmin.bat or QueryMonitor.bat please check the classpath setting in the batch file.

DB29702E **User "<user_id>" does not have administrative authority to use QueryAdministrator.**

Explanation: User "<user_id>" does not have Administrative authority defined in the **DB2 Query Patroller** User Profile table.

User Response: Have a **DB2 Query Patroller** ensure that "<user_id>" has the authorization necessary to perform the operation.

DB29703E **User “<user_id>” is not defined in User Profile.**

Explanation: User “<user_id>” is not defined in the **DB2 Query Patroller** user profile table.

User Response: Have a **DB2 Query Patroller** ensure that “<user_id>” is defined in the User Profile table.

DB29704E **User iwm cannot be removed.**

Explanation: User iwm is the mandatory userid needed for **DB2 Query Patroller** so it cannot be removed.

User Response:

DB29720E **User “<user_id>” does not have the necessary authority to use QueryMonitor.**

Explanation: User “<user_id>” is defined in the **DB2 Query Patroller** User Profile table but has no authority.

User Response: Have a **DB2 Query Patroller** ensure that “<user_id>” has the authorization necessary to perform the operation.

DB29721E **Error encountered when submitting a new job. Reason code : “<reason_code>”.**

Explanation: An error was encountered when submitting a new query. Check the reason code returned.

Reason codes map to a valid SQL message. Check the help for the corresponding SQL message.

User Response: Correct the problem and retry this operation.

DB29722E **Error encountered during job sequence number generation. Reason code : “<reason_code>”. Job will not be submitted.**

Explanation: An error was encountered during job sequence number generation. Check the reason code returned.

Reason codes map to a valid SQL message. Check the help for the corresponding SQL message.

User Response: Correct the problem and retry this operation.

DB29723E **Error encountered during purging of a result set for job “<job_id>”. Reason code : “<reason_code>”.**

Explanation: An error was encountered when purging the specified result set. Check the reason code returned.

Reason codes map to a valid SQL message. Check the help for the corresponding SQL message.

User Response: Correct the problem and retry this operation. The table may have been dropped manually outside of the system environment.

DB29724E **Error encountered when attempting to change status for job “<job_id>”. Reason code : “<reason_code>”.**

Explanation: An error was encountered when changing the status for the specified job. Check the reason code returned.

Reason codes map to a valid SQL message. Check the help for the corresponding SQL message.

User Response: Correct the problem and retry this operation.

DB29725E **Error encountered during resubmit of job “<job_id>”. with SQL statement “<SQL_stmt>”. Reason code : “<reason_code>”.**

Explanation: An error was encountered when resubmitting the specified job. Check the reason code returned.

Reason codes map to a valid SQL message. Check the help for the corresponding SQL message.

User Response: Correct the problem and retry this operation.

DB29726E **Request completed successfully.**

Explanation: The request was successful.

User Response:

DB29801E **Invalid command line parameters.**

Explanation: The parameter for command line specified is missing or incorrect.

User Response: Correct the problem and retry this operation.

DB29802E **Completed with Error.**

Explanation: The program terminated due to an error. Please see the associated error messages for the reason.

User Response: Correct the problem and retry this operation.

DB29803E **Environment variable “<environment_variable>” is not set.**

Explanation: The required “<environment_variable>” is not set.

User Response: Set the “<environment_variable>” and retry the operation.

DB29804E **A memory allocation error has occurred.**

Explanation: During processing, there was not enough memory to continue processing.

User Response: Ensure that your system has sufficient real and virtual memory.

DB29805E **User “<user_id>” is undefined.**

Explanation: “<user_id>” is not defined on the operating system.

User Response: Create an operating system user account or use a correct user id.

DB29806E **Cannot change to effective user id to “<user_id>”. Error : “<os_error>”.**

Explanation:

User Response:

DB29807E **An I/O error (reason = “<code>”) occurred while reading the file “<file>”.**

Explanation: The read operation against “<file>” failed.

User Response: Determine if the I/O error can be corrected and resubmit the command. /p>

DB29808E **An I/O error (reason = “<code>”) occurred while writing the file “<file>”.**

Explanation: The write operation against “<file>” failed.

User Response: Determine if the I/O error can be corrected and resubmit the command.

DB29809E **An I/O error (reason = “<code>”) occurred during a read operation.**

Explanation: Incomplete data was read during an I/O operation.

User Response: Determine if the I/O error can

be corrected and resubmit the command.

DB29810E **An I/O error (reason = “<code>”) occurred during a write operation.**

Explanation: An I/O error occurred during a write operation. The data may be incomplete.

User Response: Determine if the I/O error can be corrected and resubmit the command.

DB29811E **File “<file_name>” could not be opened (reason = “<code>”).**

Explanation: Error encountered while trying to open file “<file_name>”.

User Response: Ensure that the “<file_name>” is correct and exists in the file system, and that the file permissions are correct.

DB29812E **Error occurred while terminating process “<process_id>”.**

Explanation: Error encountered while trying to kill process “<process_id>”.

User Response: Ensure the terminating process has sufficient authority and the process exists.

DB29813E **There is not enough memory available to process the command.**

Explanation: Not enough random access memory (RAM) is available to process the command.

User Response: Check if the system has sufficient paging space. Free up some memory by stopping unused applications.

DB29814E **The request failed because an operating system process, thread, or swap space limit was reached.**

Explanation: An operating system process, thread, or swap space limit was reached.

User Response: Increase the limit which was reached (or have your system administrator increase it).

DB29815E **Execution of “<program_name>” failed (Reason = “<program_name>”).**

Explanation: Error encountered while executing “<program_name>”.

User Response: Determine if the problem can be fixed and retry the operation.

DB29820E **Unknown job type “<type>” for job “<job_id>”.**

Explanation: The **iwm_nodemgr** process encountered a job with an unrecognized type.

User Response: Resubmit the query. If problem persists contact an IBM Service representative.

DB29821E **Another node manager is running.**

Explanation: Only one **iwm_nodemgr** process can run on each node.

User Response:

DB29822E **Node manager could not recover jobs.**

Explanation: The **iwm_nodemgr** failed to recover jobs that were running when it last failed.

User Response:

DB29823W **sysinfo statistics are not being collected.**

Explanation: CPU utilization cannot be collected for this node because **sysinfo** facility is disabled.

User Response:

DB29824E **Job “<job_id>” not found.**

Explanation: The job “<job_id>” was not found in the job table.

User Response:

DB29825I **Completion message for job “<job_id>” saved to file “<file_name>”.**

Explanation: The agent process has lost contact with the Server component, and was unable to notify it of job completion. When the Node Manager process successfully restarts on this node, the Server component receives notification of the job’s completion.

User Response:

DB29826E **Error encountered when executing job “<job_id>” with SQL “<SQL_statement>”. Error “<sqlcode>”.**

Explanation: The “<SQL_statement>” failed for the job.

User Response: Correct the “<SQL_statement>” and resubmit the job.

DB29827I **Too many columns selected.**

Explanation: The job’s SQL statement selected too many columns.

User Response: Correct the SQL statement and resubmit the job.

DB29828E **Cannot create result table “<result_table>”.**

Explanation: The “<result_table>” could not be created.

User Response: Make sure system has sufficient free disk space and the table does not already exist.

DB29829E **SQLDA type “<type>” is not supported for column “<column_name>”.**

Explanation: DB2 Query Patroller does not recognize the database column type.

User Response:

DB29830E **Job “<job_id>” cannot be updated.**

Explanation: The request to update this job cannot be performed because the request change is not valid for the current state of the job.

User Response:

DB29831W **Job “<job_id>” (process id “<process_id>”) is no longer running.**

Explanation: The job terminated without notifying the Server component.

User Response:

DB29832E **Node “<node_name>” not found.**

Explanation: The RequestHandler Node Manager process is not active on the requested node.

User Response:

DB29835W **Node “<node_name>” has not responded for 5 minutes, attempting recovery.**

Explanation: The **iwm_nodemgr** process has not updated the node information for five minutes. The **iwm_sched** process assumes that the node has crashed and will recover jobs scheduled for the node.

User Response:

DB29836E **Scheduler could not recover jobs for node “<node_name>”.**

Explanation: An error occurred during node recovery.

User Response:

DB29837W **No active nodes available to assign jobs.**

Explanation: There are no nodes with an active node manager process.

User Response:

DB29838I “<number>” result tables over
“<number>” days old have been
purged.

Explanation: Purging of result tables has been activated through the **DB2 Query Patroller** system parameters. This message indicates how many result tables have been purged.

User Response: Check SQL statement for correctness.

DB29839I “<number>” jobs over
“<number>” days old have been
purged.

Explanation: Purging of **DB2 Query Patroller** jobs has been activated through the **DB2 Query Patroller** system parameters. This message indicates how many jobs have been purged.

User Response:

DB29840I PREPARE SQL statement
“<SQL_statement>” failed for
user “<user_id>” with SQLCODE
“<sqlcode>”

Explanation: The SQL statement could not be PREPARED.

User Response:

DB29841E “<SQL_statement>” failed. Exit
code “<code>”.

Explanation:

User Response:

DB29842E Error “<sqlcode>” encountered
during execution of the following
SQL statement
“<SQL_statement>”.

Explanation: The SQL statement failed.

User Response: Correct the SQL statement and resubmit request.

DB29843E Failed to connect to database
“<database_alias>” with userid
“<user_id>”. Error code :
“<sqlcode>”.

Explanation: The SQL CONNECT failed.

User Response:

DB29844E User “<user_id>” is not defined in
the User Profile table.

Explanation: There is no row in the User Profile table for the user.

User Response: Correct the userid or have an Administrator add a user to the User Profile table.

DB29845E Error opening socket. Error :
“<error_code>”.

Explanation: The socket specified cannot be open.

User Response: Determine if the problem can be corrected by examining the “<error_code>”.

DB29846E Error closing socket. Error :
“<error_code>”.

Explanation: The socket specified cannot be closed.

User Response: Determine if the problem can be corrected by examining the “<error_code>”.

DB29847E Error occurred while sending or
receiving data. Error :
“<error_code>”.

Explanation: Error happened while sending / receiving data.

User Response: Determine if the problem can be corrected by examining the “<error_code>”.

DB29848E **Error encountered on TCP/IP protocol support. TCP/IP function “<function>”. Socket was “<socket>”. Error : “<error_code>”.**

Explanation: The TCP/IP function failed.

User Response: Determine if the problem can be corrected by examining the “<error_code>”.

DB29849E **The specified listener port “<listener_port>” is not valid.**

Explanation: The specified listener port is not valid.

User Response: Verify the **DB2 Query Patroller** configuration parameters are correctly set.

DB29850E **The specified connect port “<connector_port>” is not valid.**

Explanation: The specified connect port is not valid.

User Response: Verify the **DB2 Query Patroller** configuration parameters are correctly set.

DB29851E **Invalid internet address “<internet_addr>”.**

Explanation: The specified internet address is not valid.

User Response: Verify the **DB2 Query Patroller** configuration parameters are correctly set.

DB29852E **TCP/IP could not find host name “<host_name>” in the local Hosts file.**

Explanation: The “<host_name>” specified cannot not be resolved.

User Response: Verify the **DB2 Query Patroller** configuration parameters are correctly set.

DB29853E **Service “<service_name>” is not defined in the TCP/IP services file.**

Explanation: The “<service_name>” specified cannot not be resolved in the TCP/IP service file.

User Response: Verify the **DB2 Query Patroller** configuration parameters are correctly set.

DB29854E **Invalid network identifier “<network_string>”.**

Explanation: The network identifier must be in the form **host-id:port_id** where **host-id** represents either a resolvable host name or the dotted internet address and **port-id** represents either a resolvable service name or a port number.

User Response: Verify the **DB2 Query Patroller** configuration parameters are correctly set.

DB29855E **DB2 Query Patroller server is not available.**

Explanation: The Server component cannot be contacted.

User Response: Verify the **DB2 Query Patroller** configuration parameters are correctly set.

DB29856E **A severe internal processing error has occurred. Error “<error_code>”.**

Explanation: An internal **DB2 Query Patroller** failure occurred.

User Response: Please contact an IBM Service representative.

DB29857E **“<name>” value is missing.**

Explanation: An internal **DB2 Query Patroller** failure occurred.

User Response: Please contact an IBM Service representative.

DB29858E "<name>" value is invalid.

Explanation: An internal **DB2 Query Patroller** failure occurred.

User Response: Please contact an IBM Service representative.

DB29859E Destination "<name>" is no longer valid.

Explanation: An destination name is no longer active.

User Response:

DB29860E Expected response message not received.

Explanation: An internal **DB2 Query Patroller** failure occurred.

User Response: Please contact an IBM Service representative.

DB29861E Connection to **iwm_net** lost.

Explanation: The **iwm_net** program is no longer active.

User Response: Restart the Server component or agent on this node.

DB29862E Not connected.

Explanation: An internal **DB2 Query Patroller** failure occurred.

User Response: Please contact an IBM Service representative.

DB29863E Message not understood.

Explanation: An internal **DB2 Query Patroller** failure occurred.

User Response: Please contact an IBM Service representative.

DB29864E **iwm_local** database access is not available.

Explanation: The **iwm_local** program failed or was not started successfully.

User Response: Restart the DB2 Query Patroller and all agents.

DB29865E **iwm_remote** database access is not available.

Explanation: The **iwm_remote** program failed or was not started successfully.

User Response: Restart the Server component.

DB29866E **iwm_local** database access failed.

Explanation: The **iwm_local** program failed or was not started successfully.

User Response: Please contact an IBM Service representative.

DB29867E **iwm_remote** database access failed.

Explanation: The **iwm_remote** program failed or was not started successfully.

User Response: Please contact an IBM Service representative.

DB29868E Invalid ticket.

Explanation: An internal **DB2 Query Patroller** failure occurred.

User Response: Please contact an IBM Service representative.

DB29869E Logon has expired.

Explanation: Your logon to **DB2 Query Patroller** has expired.

User Response: Log on again.

DB29870E Not authorized.

Explanation: You are not authorized to perform the requested function.

User Response: Contact your System Administrator to obtain the necessary authorization.

DB29988E The IWM_SERVER environment variable is not set!

Explanation: Environment variable IWM_SERVER was not set.

User Response: Set the environment variable IWM_SERVER to [host]:[port] where [host] refers to IP address of the machine where DB2 Query Patroller Server is installed and [port] refers to the TCP port that the DB2 Query Patroller Server can use for communication.

DB29990E The IWM_RUNTIME environment variable is not set!

Explanation: Environment variable IWM_RUNTIME was not set.

User Response: Set the environment variable IWM_RUNTIME to the fully qualified path where DB2 Query Patroller is installed.

DB29991E DB2 Query Patroller Server does not seem to be installed correctly!

Explanation: DB2 Query Patroller Server was not installed correctly!

User Response: Please reinstall DB2 Query Patroller Server.

Log File

Messages written to the log file display in the following format:

<date-time> <node> <program> <pid> <message-id> <source-line> <text>

where:

- **<date-time>** represents the date and time that the message was recorded in *YYYYMMDD HHMMSS* format.
- **<node>** represents the host name of the node that recorded the message.
- **<program>** is the name of the program that recorded the message.
- **<pid>** represents the UNIX process ID that recorded the message.
- **<message-id>** represents the message identifier in the format *iwm-<number>-<severity>*. The severity code is **V** (verbose - messages displayed for information only), **I** (informational messages), **W** (warning messages), or **E** (error messages).
- **<text>** represents the text of the message. It may include a message from the database software, identified in message descriptions below as *<dbms-error>*. Refer to your database documentation for descriptions of these messages. The text may also include a message from the operating system.

The log server file name is `syserr.log`. By default, DB2 Query Patroller creates this file in the `$IWM_RUNTIME/log` directory. However, if the user sets the `$IWMLOG` environment variable, DB2 Query Patroller creates the file in that directory.

Appendix D. Notices

Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM's licensed program may be used. Any functionally equivalent product, program or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter 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 the

IBM Director of Licensing
IBM Corporation, North Castle Drive
Armonk, NY 10504-1785
U.S.A.

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 which has been exchanged, should contact:

IBM Canada Limited
Office of the Lab Director
1150 Eglinton Ave. East
North York, Ontario
M3C 1H7
CANADA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

This publication 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.

Trademarks

The following terms are trademarks or registered trademarks of the IBM Corporation in the United States and/or other countries:

ACF/VTAM	MVS/ESA
ADSTAR	MVS/XA
AISPO	OS/400
AIX	OS/390
AIXwindows	OS/2
AnyNet	PowerPC
APPN	QMF
AS/400	RACF
CICS	RISC System/6000
C Set++	SP
C/370	SQL/DS
DATABASE 2	SQL/400
DataHub	S/370
DataJoiner	System/370
DataPropagator	System/390
DataRefresher	SystemView
DB2	VisualAge
DB2 Connect	VM/ESA
DB2 Universal Database	VSE/ESA
Distributed Relational Database Architecture	VTAM
DRDA	WIN-OS/2
Extended Services	
FFST	
First Failure Support Technology	
IBM	
IMS	
LAN Distance	

Trademarks of Other Companies

The following terms are trademarks or registered trademarks of the companies listed:

C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both.

HP-UX is a trademark of Hewlett-Packard.

Java, HotJava, Solaris, Solstice, and Sun are trademarks of Sun Microsystems, Inc.

Microsoft, Windows, Windows NT, Visual Basic, and the Windows logo are trademarks or registered trademarks of Microsoft Corporation in the United States, other countries, or both.

PC Direct is a trademark of Ziff Communications Company in the United States, other countries, or both and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium, and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark in the United States, other countries or both and is licensed exclusively through X/Open Company Limited.

Other company, product, or service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

Index

A

- adding users 15
- administration 11
 - data sources 55
 - nodes 63
 - result destinations 67
 - system parameters 71
 - user profiles 59
- agent 6, 10
- agent processes 94

B

- bar charts 86

C

- canceling jobs 18
- changing the date and time range 84
- changing the report format 87
- client configuration 47
- client configuration options 48
- clients, troubleshooting 97
- columns
 - displaying 52
 - resizing 53
- Columns Hit by User Y Hitting Table X report 81
- Columns Hit in Table X report 80
- command-line interface 6
- command-line interface environment 21
- commands
 - EXCLUDE 37
 - INCLUDE 37
 - INTERVAL 36
 - iwm 17, 18
 - iwm_cmd 21, 25
 - iwm_submit 21, 22
 - LOGFILE 36
 - MAILTO 37
- common parameters 27
- common QueryEnabler problems 98
- common server problems 95
- common Tracker problems 100
- components 4
 - agent 6, 10
 - command-line interface 6
 - cost analyzer 5, 15
 - executor 6, 10, 44

- components 5 (*continued*)
 - job scheduler 6, 9
 - log monitor 5, 12, 19, 35
 - node manager 6, 10
 - notifier 5, 10, 39, 44
 - QueryAdministrator 6
 - QueryEnabler 7
 - QueryMonitor 7
 - request server 5
 - server 4
- configuration
 - client 47
 - data warehouse 46
 - estimating and monitoring 45
 - monitoring 46
 - QueryEnabler 47
- configuration file, log monitor 35
- configuring DB2 Query Patroller server 45
- configuring the DB2 Query Patroller client 47
- connections, socket 8
- control command 25
- control program 18
- controlling
 - job execution 17
 - job flow 17
 - jobs, using iwm_cmd 29
 - nodes, using iwm_cmd 30
 - system parameters, using iwm_cmd 33
- controlling data sources 31
- conventions, syntax 3
- cost analysis 5, 9, 14
 - exit 43
- cost analysis required 14
- cost analyzer 5, 15
- CPU utilization 10

D

- data source administration 55
- data source parameters 13
 - data source status 13, 32
 - job cost limit 13, 32
 - job count limit 14, 32
 - node CPU limit 14, 32
 - node disk limit 14, 32
- data source status 13, 32

- data sources
 - controlling using iwm_command 31
 - editing 55
 - listing 55
 - listing using iwm_command 31
- Data Sources table 8, 13
- data warehouse configuration 46
- date range 84
- DB2 diagnostic log 93
- DB2 failures 93
- DB2 Query Patroller
 - components 4
 - starting 17
 - stopping 17
 - system architecture 4
 - tables 7
- DB2 Query Patroller failures 93
- diagnostic log 93
- disk space monitoring 11, 32
- displaying columns 52
- displaying data as a count 80
- displaying data as a percentage 80
- displaying job history data 83, 84
- drill path 88
- drilling down 87
- drilling up 87

E

- e-mail 5, 10, 12, 15, 35, 37
- e-mail messages 44
- environment variables 11
 - IWM_DISKMON 11
 - IWM_EXIT_AN 11, 43
 - IWM_INTERVAL 12
 - IWM_LOGMON 12, 35
 - IWM_MAIL 12
 - IWM_NET 12, 21
 - IWM_NOEXPLAIN 12
 - IWM_RES_TBSPC 12
 - IWM_RUNTIME 12, 21, 36, 111
 - IWM_server 13
 - IWMLOG 12, 36, 111
 - PATH 21
- error messages 101
- estimating system configuration 45
- EXCLUDE 37
- executor 6, 10, 44
- exiting Tracker 89

F

- failures
 - DB2 93
 - DB2 Query Patroller 93
 - process 94
- filters
 - NUM 38
 - PGM 38
 - SEV 37
- formats
 - bar charts 86
 - changing 87
 - lists 86
 - pie chart 86
- formats, report 86

I

- INCLUDE 37
- installing QueryAdministrator 51
- INTERVAL 36
- iwm 17, 18
- iwm account 11
- iwm_cmd 21, 25
 - controlling data sources 31
 - controlling jobs 29
 - controlling nodes 30
 - controlling system parameters 33
 - listing data sources 31
 - listing system parameters 33
 - monitoring jobs 27
 - monitoring nodes 30
- IWM_DISKMON 11
- IWM_EXIT_AN 11, 43
- IWM_INTERVAL 12
- IWM_LOGMON 12, 35
- IWM_MAIL 12
- IWM_NET 12, 21
- iwm_net 94
- iwm_nodemgr 94
- IWM_NOEXPLAIN 12
- IWM_RES_TBLSPC 12
- IWM_RUNTIME 12, 21, 36, 111
- IWM_SERVER 13
- iwm_submit 21, 22
- IWMLOG 12, 36, 111

J

- job
 - accounting 39
 - execution 10
 - scheduling 9
 - submission 9
- job accounting
 - status 13, 33, 39
 - table 5

- Job Accounting
 - table 8, 10, 13, 33
- job accounting
 - table 39
- Job Activity over Time report 81
- job cost limit 13, 14, 32
- Job Cost Limit 33
- job count limit 13, 14
- Job Count Limit 32
- job count limit 33
- job execution
 - controlling 17
- job flow 9
 - controlling 17
 - cost analysis 9
 - job execution 10
 - job notification 10
 - job scheduling 9
 - job submission 9
- job history data 83, 84
- job purge days 13, 33
- job queue parameters 14
 - maximum cost 14
 - maximum jobs 14
- Job Queues table 8, 14
- job scheduler 5, 9
- Job table 7
- jobs
 - canceling 18
 - controlling 25, 29
 - controlling using iwm_cmd 29
 - monitoring 25, 27
 - monitoring using iwm_cmd 27
 - releasing 18

L

- listing
 - data sources, using iwm_cmd 31
 - system parameters, using iwm_cmd 33
- Lists 86
- log file 5, 35, 101
- log file monitoring 35
- log monitor 5, 12, 19, 35
 - configuration file 35
 - running 35
- LOGFILE 36

M

- mail 5, 10, 12, 15, 35, 37, 44
- MAILTO 37
- maintaining cost statistics 15
- MaxElapsedTime 48
- MaxResultRows 48

- monitoring
 - CPU utilization 10, 32
 - disk space 32
 - jobs, using iwm command 27
 - jobs, using the control command 25
 - log file 35
 - nodes, using iwm_cmd 30
- monitoring system
 - configuration 45, 46

N

- navigating 76
- node administration 63
- node CPU limit 14, 32
- node disk limit 14, 32
- Node Information table 7
- node manager 6, 10
- nodes
 - controlling, using iwm_cmd 30
 - listing 63
 - monitoring, using iwm_cmd 30
- notification 10
- notifier 5, 10, 39, 44
- NUM filter 38

O

- opening a new report 84
- operation 17
- options, client configuration 48

P

- panning on a pie chart 88
- panning on a time chart 89
- parameters
 - common for iwm_cmd 27
 - cost analysis required 14
 - data source 13
 - data source, for iwm_cmd 31
 - data source status 13, 32
 - DB2 Query Patroller 11
 - iwm_submit 22
 - job accounting status 13
 - job control, for iwm_cmd 29
 - job cost limit 13, 32, 33
 - job count limit 13, 32, 33
 - job monitoring for iwm_cmd 28
 - job purge days 13, 33
 - job queue 14
 - maximum cost 14
 - maximum jobs 14
 - mode disk limit 32
 - node control and monitoring, for iwm_cmd 30
 - node CPU limit 14, 32
 - node disk limit 14

- parameters *(continued)*
 - priorities 27
 - results purge days 33
 - system 13
 - system, for iwm_cmd 33
 - user profile 14
- PATH 21
- pausing DB2 Query Patroller 17
- PGM filter 38
- pie chart, zooming and panning 88
- pie charts 86
- priorities 14
- problems, common server 95
- problems, QueryAdministrator 99
- problems, QueryEnabler 98
- process failures 94
- processes, agent 94
- processes, server 94
- profiling queries 45
- purging 33
 - jobs 10, 33
 - result tables 10
- purging, result tables 33

Q

- queries
 - profiling 45
- QueryAdministrator 6
 - common problems 99
 - installing 51
 - starting 51
 - stopping 52
- QueryEnabler 7
 - configuring 47
 - troubleshooting 97
- QueryEnabler, problems 98
- QueryMonitor 7
 - common problems 99
- QueryMonitor, troubleshooting 99

R

- refreshing report data 86
- releasing 18
- releasing jobs 18
- report data 86
- report formats 86
 - bar charts 86
 - changing 87
 - lists 86
 - pie chart 86
- reports 79
 - changing the date and time range 84
 - closing a report window 89
 - Columns Hit by User Y Hitting Table X 81

- reports 80 *(continued)*
 - Columns Hit in Table X 84
 - date range 84
 - displaying 79
 - displaying data as a count 80
 - displaying data as a percentage 80
 - drilling down for more information 87
 - drilling up 87
 - formats 86
 - Job Activity Over Time 79
 - Job Activity over Time 81
 - opening a new report 84
 - refreshing data 86
 - Rows Returned by Table 79, 81
 - Tables Joined with Table X 80, 81
 - time range 84
 - Total Table Hits 79
 - Total Table Hits report 80
 - Users That Hit Table X 80
 - viewing the drill path 88
 - zooming and panning on a pie chart 88
 - zooming and panning on a time chart 89
- request server 5
- resizing columns' 53
- result destination administration 67
- result destination variables 44
- result destinations 8, 43
 - creating 67
 - listing 67
 - pipelined 44
 - removing 68
- Result Destinations table 8, 43
- result tables 8, 10
 - purging 33
- result tables, purging 10
- results purge days 13, 33
- Rows Returned by Table report 81
- running the log monitor 35

S

- server 4
 - common problems 95
 - troubleshooting 93
- server, configuring 45
- server processes 94
- SEV filter 37
- socket connections 8
- starting
 - DB2 Query Patroller 17
 - log monitor 35
 - Tracker 83

- starting QueryAdministrator 51
- stopping
 - DB2 Query Patroller 17
 - log monitor 35
- stopping QueryAdministrator 52
- submit command 22
- submitting 22
- syntax conventions 3
- syserr.log 93
- system administration 71
- system monitoring 18
- system parameters 13, 33
 - controlling using iwm_cmd 33
 - editing 71
 - job accounting status 13, 33, 39
 - job cost limit 13, 33
 - job count limit 13, 33
 - job purge days 13, 33
 - listing 71
 - listing using iwm_cmd 33
 - results purge days 13, 33
- System Parameters table 8, 13

T

- tables
 - Data Sources 8, 13
 - DB2 Query Patroller 7
 - Job 7
 - Job Accounting 5, 8, 10, 33, 39
 - Job Accounting Status 13
 - Job Queues 8, 14
 - Node Information 7
 - Result Destinations 8, 43
 - Results 8
 - Results Purge Days 13
 - System Parameters 8, 13
 - User Profile 5, 7, 14, 15
- Tables Joined with Table report 80
- Tables Joined with Table X
 - report 81
- time chart, zooming and panning 89
- time range 84
- tool bar 76
- Total Table Hits report 80
- Tracker
 - closing a report window 89
 - common problems 100
 - exiting 89
 - navigating 76
 - reports 79
 - starting 83
 - tool bar 76
 - troubleshooting 99
- troubleshooting
 - DB2 Query Patroller clients 97

- troubleshooting (*continued*)
 - DB2 Query Patroller server 97
 - QueryEnabler 97
 - QueryMonitor 99
 - Tracker 99

U

- user administration 59
- user exits 43
- user profile parameters 14
 - cost analysis required 14
 - job cost limit 14
 - job count limit 14
 - priorities 14
- User Profile table 5, 7, 14
- user profiles
 - changing node status 63
 - creating 59
 - editing 61
 - listing 59
 - removing 61
- users, adding 15
- Users That Hit Table X report 80
- using the command-line interface 21

V

- variables, result destination 44
- viewing the drill path 88

Z

- zooming on a pie chart 88
- zooming on a time chart 89

Contacting IBM

This section lists ways you can get more information from IBM.

If you have a technical problem, please take the time to review and carry out the actions suggested by the *Troubleshooting Guide* before contacting DB2 Customer Support. Depending on the nature of your problem or concern, this guide will suggest information you can gather to help us to serve you better.

For information or to order any of the DB2 Universal Database products contact an IBM representative at a local branch office or contact any authorized IBM software remarketer.

Telephone

If you live in the U.S.A., call one of the following numbers:

- 1-800-237-5511 to learn about available service options.
- 1-800-IBM-CALL (1-800-426-2255) or 1-800-3IBM-OS2 (1-800-342-6672) to order products or get general information.
- 1-800-879-2755 to order publications.

For information on how to contact IBM outside of the United States, see Appendix A of the IBM Software Support Handbook. You can access this document by accessing the following page:

<http://www.ibm.com/support/>

then performing a search using the keyword “handbook”.

Note that in some countries, IBM-authorized dealers should contact their dealer support structure instead of the IBM Support Center.

World Wide Web

<http://www.software.ibm.com/data/>

<http://www.software.ibm.com/data/db2/library/>

The DB2 World Wide Web pages provide current DB2 information about news, product descriptions, education schedules, and more. The DB2 Product and Service Technical Library provides access to frequently asked questions, fixes, books, and up-to-date DB2 technical information. (Note that this information may be in English only.)

Anonymous FTP Sites

<ftp.software.ibm.com>

Log on as anonymous. In the directory /ps/products/db2, you can find demos, fixes, information, and tools concerning DB2 and many related products.

Internet Newsgroups

comp.databases.ibm-db2, bit.listserv.db2-l

These newsgroups are available for users to discuss their experiences with DB2 products.

CompuServe

GO IBMDB2 to access the IBM DB2 Family forums

All DB2 products are supported through these forums.

To find out about the IBM Professional Certification Program for DB2 Universal Database, go to http://www.software.ibm.com/data/db2/db2tech/db2cert.html
--



Part Number: CT6N0NA



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC09-2859-00



CT6N0NA

