MySQL Administrator

MySQL Administrator

Abstract

This is the MySQL Administrator Manual.

Document generated on: 2006-02-23 (revision: 1427)

Table of Contents

List of Figures

Chapter 1. Introduction to MySQL Administrator

MySQL Administrator is a program for performing administrative operations, such as configuring, monitoring and starting and stopping a MySQL server, managing users and connections, performing backups, and a number of other administrative tasks.

Most tasks can be performed using a command-line client such as `mysqladmin` [http://dev.mysql.com/doc/mysql/en/mysqladmin.html] or `mysql` [http://dev.mysql.com/doc/mysql/en/mysql.html], but MySQL Administrator is advantageous in the following respects:

- Its graphical user interface makes it more intuitive.
- It provides a better overview of the settings that are crucial for the performance, reliability, and security of your MySQL servers.
- It displays performance indicators graphically, thus making it easier to determine and tune server settings.

MySQL Administrator is designed to work with MySQL versions 4.0 and higher.

MySQL Administrator is to a large extent the result of feedback MySQL AB has received from many users over a period of several years. However, if you find it's lacking some feature important to you, or if you discover a bug, please use our MySQL Bug System [http://bugs.mysql.com] to request features or report problems.

Chapter 2. Installing MySQL Administrator

2.1. Introduction

MySQL Administrator runs on Windows, Linux, and Mac OSX. You can find MySQL Administrator for the operating system of your choice on the MySQL Downloads [http://dev.mysql.com/downloads/administrator/] page.

2.2. Installing Under Windows

MySQL Administrator runs on recent 32-bit Windows NT based operating systems, including Windows 2000, XP, and 2003. It doesn't run on Windows NT 4 and below.

MySQL Administrator is installed through the use of a Windows Installer (`.msi`) installation package, which can be used to install the MySQL Administrator on all Windows operating systems. The MSI package in contained within a ZIP archive named `mysql-administrator-`*`version`*`-win.msi`, where *`version`* indicates the MySQL Administrator version.

The Windows Installer engine was updated with the release of Windows XP; those using an older version can reference this Microsoft Knowledge Base article [http://support.microsoft.com/default.aspx?scid=kb;EN-US;292539] for information on upgrading to the latest version.

To install MySQL Administrator, right-click on the MSI file and select Install. The installation will begin automatically after the installer prompts you for your installation preferences. During installation, you can choose if you want the installer to place a shortcut in the `Start` menu, and/or an icon on the desktop.

If you are having problems running the installer, you can download a ZIP file without an installer as an alternative. That file is called `mysql-administrator-`*`version`*`-win-noinstall.zip`. Using a ZIP program, unpack it to a directory of your choice. You may want to create shortcuts to `MySQLAdministrator.exe` for your desktop or quick launch bar.

Unless you choose otherwise, MySQL Administrator is installed in `C:\`*`%PROGRAMFILES%`*`\MySQL\MySQL Administrator 1.0\MySQLAdministrator.exe`, where *`%PROGRAMFILES%`* is the default directory for programs on your machine. For example, this directory might be `C:\Program Files` or `C:\programme`.

2.3. Installing Under Linux

MySQL Administrator runs on Linux machines that have a graphical desktop installed. It is designed to run under the Gnome desktop with GTK2. It has been tested on Linux kernel versions 2.4 and 2.6, but it should also run on other versions, and even on a number of Unix-like operating systems.

To install MySQL Administrator, first download the installation tarball. The tarball file is called `mysql-administrator-`*`version`*`-linux.tar.gz`, where *`version`* indicates the MySQL Administrator version (for example 1.0.12).

To see all files in the tarball, run this command:

```
shell> tar -tzf  mysql-administrator-version-linux.tar.gz
```

To install MySQL Administrator, run this command:

```
shell> tar --directory=/opt -xzvf mysql-administrator-version-linux.tar.gz
```

This installs the application binary in `/opt/mysql-administrator/bin`. Change into that directory and run `mysql-administrator` to start the application.

Distribution-specific packages will be available at some point.

2.4. Installing Under Mac OSX

To install MySQL Administrator under Mac OS X, double-click the downloaded .dmg file and wait it to be opened and attached. Once a window containing the Administrator icon pops up, drag it to your Applications folder -- or any other location you prefer.

Once the copy is complete, you may eject the disk image.

Chapter 3. Starting MySQL Administrator

3.1. Introduction

The way you start MySQL Administrator depends on the operating system you are using:

- On Windows, start MySQL Administrator by double-clicking its desktop icon, or by selecting its item from the `Start` menu. Alternatively, you can open a DOS window and start it from the command line:

  ```
      C:\> C:\program
  directory\MySQL\MySQL Administrator 1.0\MySQLAdministrator.exe
  ```

  `program directory` is the default directory for programs on your machine, for example `C:\Program Files` or `C:\programme`. If your path contains spaces, you should enclose the command within quote characters. For example:

  ```
      C:\> "C:\Program Files\MySQL\MySQL Administrator 1.0\MySQLAdministrator.ex
  ```

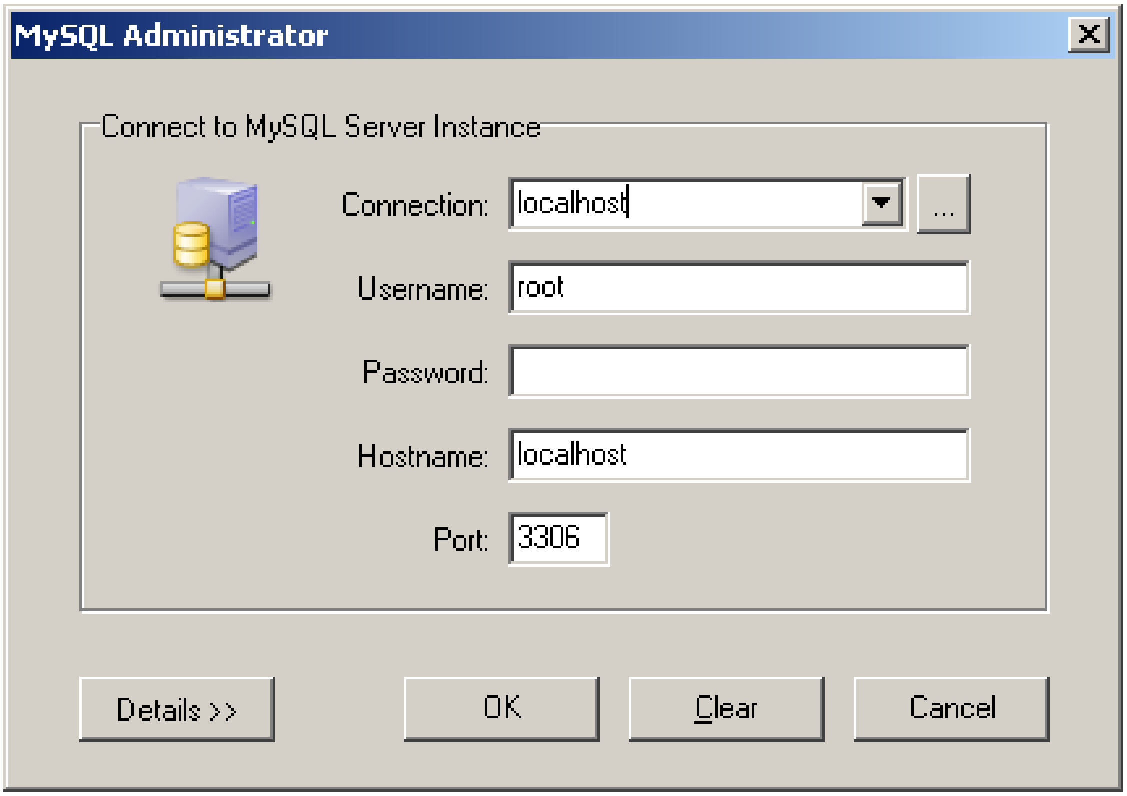
- On Linux desktops, start MySQL Administrator by changing directory into `/opt/mysql-administrator/bin`, and then running `mysql-administrator`.
- On Mac OSX, navigate to the MySQL Administrator installation directory and double-click on the MySQL Administrator icon.

3.2. Connection Dialog

Once MySQL Administrator has been started, it displays a connection dialog. You have to specify the MySQL server to which you would like to connect, the credentials needed for authorization on that server, and which machine that server runs on (and which port it listens to). You may also specify a number of other options, if required.

As an alternative to connecting to a running MySQL server, you can run MySQL Administrator in configure -service mode.

Figure 3.1. Connection dialog



Connection: localhost

Username: root

Password:

Hostname: localhost

Port: 3306

Note that MySQL Administrator will not work with MySQL servers prior to version 4.0. If you try to connect to an older server, an error message will appear, telling you that this is not possible.

If the server connection is successfully established, all of the values filled in the fields of the connection dialog will be saved for future connections (see the section that describes how MySQL Administrator stores connection information). The `Password` field, however, will always be empty: For security reasons, the password is not stored along with the other options, unless you explicitly specify otherwise in the Gen eral Options section of the Options dialog.

You can change any of the values in the connection profiles just by overwriting the existing values with new ones. Similarly, if you select another connection profile, you can change any of its values. When you click the OK button after changing a profile's values, the changes are stored permanently if a successful connection to the MySQL server is established. However, the preferred method for changing a profile is to click the ... button next to the Connection drop-down box. That brings up an `Options` dialog window that has a `Connections` section for modifying connection profiles. You can find the values you can set in the Connection dialog window in the description of the `Connections` section of the `Options` dialog (see Options dialog).

3.3. Configure-service Mode

Normally, you will use MySQL Administrator to connect to a running MySQL server. This is done by starting the program and specifying in the Connection dialog which server to connect to.

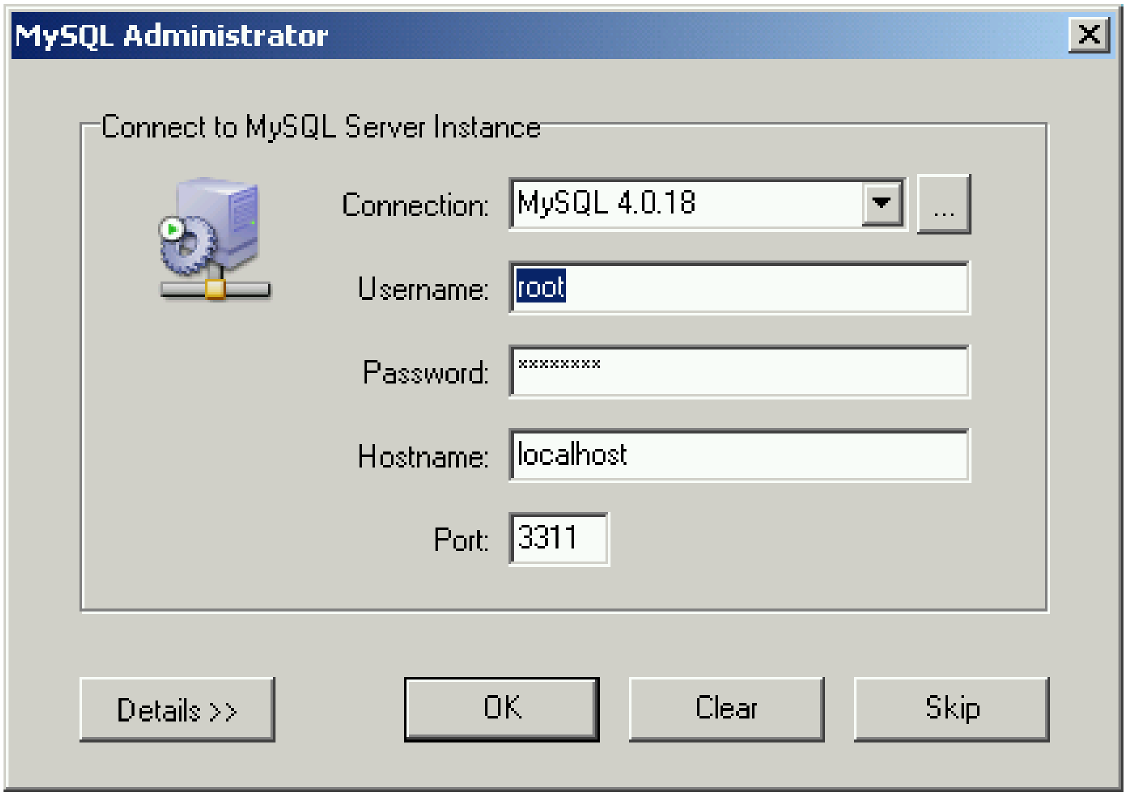
Alternatively, you can run MySQL Administrator in configure-service mode. This is useful in the following cases:

- When you want to start a MySQL server from within MySQL Administrator. Obviously, you cannot connect to a server that is not running, so you will have to use the configure-service mode to start that server. Note that, at the moment, you can only start servers on the same machine where MySQL Administrator is running (that is on the local host).

- When you have multiple servers [http://dev.mysql.com/doc/mysql/en/Multiple_servers.html] running on your machine. On Windows, MySQL Administrator only allows you to configure the Windows service that is listed first in the Windows services management when running in normal mode. This means that if you have more than one Windows service, you can only configure the first one of those, no matter which server you have connected to. If you want to configure a service other than the first service, you will have to run MySQL Administrator in configure-service mode.

To start MySQL Administrator in configure-service mode, hold down the **CTRL** key in the Connection dialog. The Cancel button will toggle to Skip when that key is pressed.

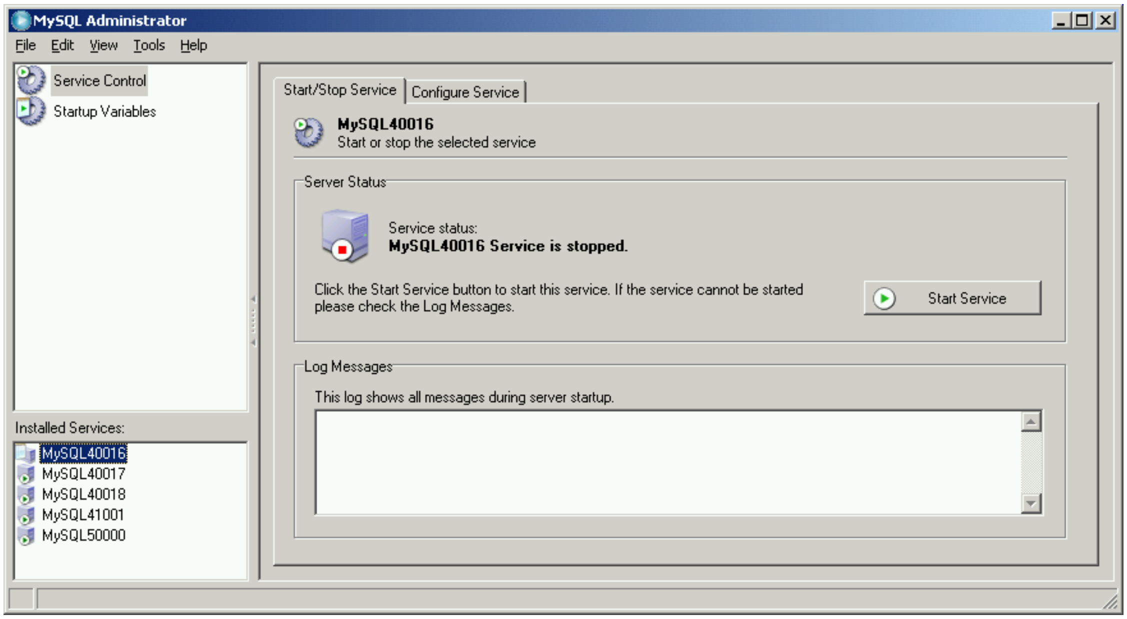
Figure 3.2. Switching to configure-service mode



Connection: MySQL 4.0.18
Username: root
Password: *********
Hostname: localhost
Port: 3311

Clicking on the Skip button will skip the normal mode, thus starting MySQL Administrator in configure-service mode.

Figure 3.3. Configure-service mode



MySQL40016
Start or stop the selected service

Service status:
MySQL40016 Service is stopped.

Click the Start Service button to start this service. If the service cannot be started please check the Log Messages.

This log shows all messages during server startup.

Under `Installed Services` in the sidebar, you can select the service that you wish to configure, or the server that you want to stop or start. In configure-service mode, as opposed to normal mode, there is only a subset of the sections in the sidebar avaiable:

- The Service Control section allows you to start and stop a MySQL server, and to configure the Windows service managing that server.
- The Startup Variables section allows you to set variables for server startup.

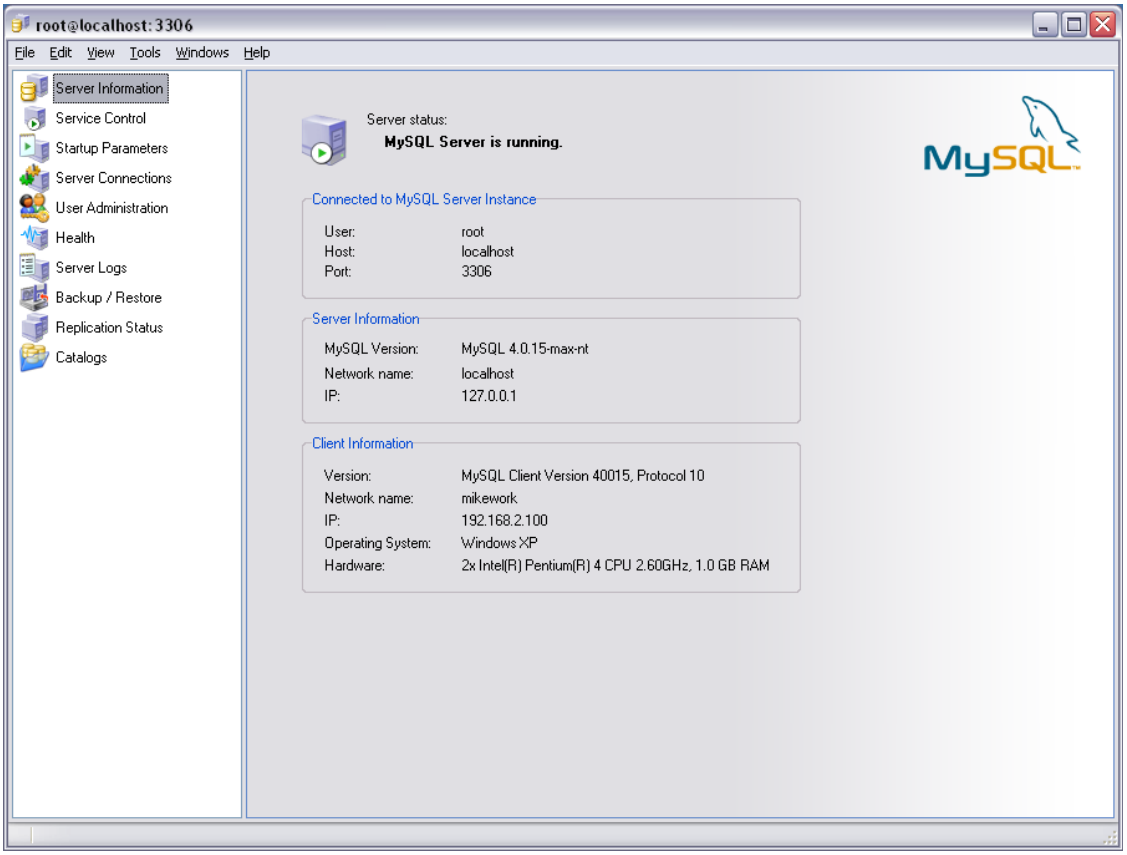
Both of those sections are described in detail below.

Chapter 4. Main Window

4.1. Introduction

After you have successfully connected to your MySQL server, the main window of MySQL Administrator appears (see figure). The main window contains a sidebar, a working area, and menus with submenus.

Figure 4.1. MySQL Administrator main window



Server status:
MySQL Server is running.

Connected to MySQL Server Instance

User: root
Host: localhost
Port: 3306

Server Information

MySQL Version: MySQL 4.0.15-max-nt
Network name: localhost
IP: 127.0.0.1

Client Information

Version: MySQL Client Version 40015, Protocol 10
Network name: mikework
IP: 192.168.2.100
Operating System: Windows XP
Hardware: 2x Intel(R) Pentium(R) 4 CPU 2.60GHz, 1.0 GB RAM

4.2. Sidebar

The sidebar consists of the following sections:

- Server Information: Information about the MySQL server you are connected to, about MySQL Administrator and the machine that it runs on, and about your connection.

- Service Control: Start and stop the MySQL server. On Windows, you can also configure the MySQL service.

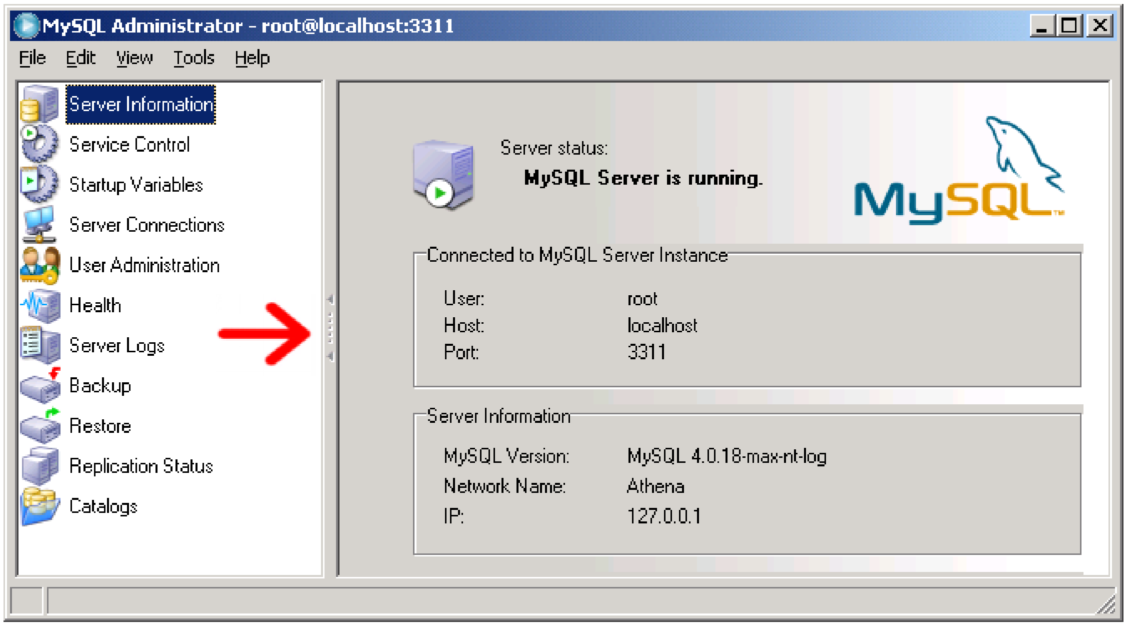
- Startup Variables: Configure the startup variables for the MySQL server.

- Server Connections: View or kill current connections to the MySQL server (also called threads).
- User Administration: Administer existing users, add new users, and delete existing users.
- Health: Graphical displays of a number of usage or hitrate values that affect server performance, and a hierarchical view of system and status variables.
- Server Logs: View log file entries.
- Backup: Plan and administer backup projects, select the databases to be backed up, and start the backup process.
- Restore: Restore databases from existing backups.
- Replication Status: View information about replication master and slave servers.
- Catalogs: View information about databases, tables, columns, indexes, and rows; optimize tables.

The following sections describe each of these in detail.

You can change the width of the sidebar in the main window by dragging the separator between sidebar and working area. To hide the sidebar completely, click on the marked area in the separator (located in the vertical center of it). To make the sidebar visible again, double-click the marked area (which then forms the left margin of the main window).

Figure 4.2. Marked area between sidebar and working area



Server status:
MySQL Server is running.

Connected to MySQL Server Instance

User: root
Host: localhost
Port: 3311

Server Information

MySQL Version: MySQL 4.0.18-max-nt-log
Network Name: Athena
IP: 127.0.0.1

4.3. Working Area

While the sidebar allows you to select the sections, the working area displays information, or fields, that pertain to the selected section. For some sections, the working area is split into two or more tabs.

4.4. Menus

4.4.1. File

4.4.1.1. New Instance Connection ...

Opens the Connection dialog, allowing you to open another connection to a MySQL server. You may open an arbitrary number of connections. MySQL Administrator opens a new main window for each successfully established connection. Connections may be to the same or different servers. Multiple connections may be opened using the same profile or different profiles.

4.4.1.2. Manage Connections ...

Opens the Options dialog, with the Con nections section preselected.

4.4.1.3. Save current Connection ...

Opens the Options dialog, with the Con nections section preselected. Furthermore, a new connection profile labeled `New Connection` is created, which is also preselected. The fields are filled with the values of the most recent connection. You can edit the values and save the settings in the new profile. You may also want to rename the profile from `New Connection` to something more appropriate. If you click the Close button without saving your changes with the Apply Changes button, you will be prompted as to whether you want to save or discard your changes.

4.4.1.4. Close

Closes the MySQL Administrator window and terminates the connection to the MySQL server that had been established in that window. If you have opened more than one instance (program window, or process) of MySQL Administrator, this command will not close the other windows.

4.4.2. Edit

The Cut, Copy, and Paste commands allow you to cut, copy, or paste selected text in the working area.

4.4.2.1. Options

Opens the Options dialog window.

4.4.3. View

The View menu selects sections from the sidebar. Selecting an item from this menu is the same as clicking the section with the same name from the sidebar.

4.4.4. Tools

The Tools menu starts other programs.

4.4.4.1. MySQL Command Line Client

This menu item is available only if MySQL Administrator finds the executable of the MySQL command-line client. At the moment, it will look up locations like `C:\mysql\bin`, `D:\mysql\bin`, and `C:\`*`program directory`*`\mysql\bin` to find `mysql.exe`. If it is found, you can start the command-line client with this command. MySQL Administrator will use the credentials provided in the

Connection dialog to start a server connection using the command-line client.

4.4.4.2. Windows Command Line

Opens a DOS window.

4.4.5. Help

4.4.5.1. Online Help

Opens a window displaying the MySQL Administrator section of the MySQL reference manual.

4.4.5.2. Report Bug to MySQL

Opens a connection to the MySQL bug reporting system in your web browser.

4.4.5.3. Launch MySQL Website

Opens a connection to the MySQL web site in your web browser.

4.4.5.4. About ...

Displays information about MySQL Administrator.

Chapter 5. Server Information

Provides basic information about connection, server, and client.

- Connected to MySQL Server Instance

  Information about the parameters used to connect to the MySQL server.

  - `User`: Username used for the current connection to the MySQL server.
  - `Host`: The hostname of the machine where the MySQL server runs. This is either identical to the name listed in `Network Name`, or it is `localhost`. The former indicates that you are connecting to a server that either runs on a remote machine, or that you have specified a hostname different from `localhost` when connecting. The latter indicates that the MySQL server runs on the same machine where MySQL Administrator runs.
  - `Port`: The port used for the connection to the MySQL server.

- Server Information

  Information about the MySQL server and the host where it is running.

  - `MySQL Version`: The version of the MySQL server.
  - `Network Name`: The hostname of the machine where the MySQL server runs.
  - `IP`: The IP number of the host where the MySQL server runs.

- Client Information

  Information about MySQL Administrator and the client host on which it is running.

  - `Version`: The version of MySQL Administrator, and the version number of the MySQL client/server protocol.
  - `Network name`: The name of the host where MySQL Administrator runs.
  - `IP`: The IP number of the host where MySQL Administrator runs.
  - `Operating System`: Information about the operating system on which MySQL Administrator runs.
  - `Hardware`: Information about the hardware on which MySQL Administrator runs.

Chapter 6. Service Control

6.1. Introduction

This section allows you to start and stop the MySQL server you have connected to. At the moment, this is only possible if the server is running on the same machine where MySQL Administrator runs. In future releases, MySQL servers will have an Instance Manager that allows you to also start and stop MySQL servers on remote machines. Also, you can start and stop the server only if a service has been configured. If the server runs as a stand-alone application, the Service Control section is not available.

The Configure Service tab is available only for MySQL servers that run under Windows.

6.2. Start/Stop Service

The `Start/Stop Service` tab displays the status of the MySQL service. This service starts and stops the MySQL server. Under `Service Status`, a text message indicates whether or not the service has been started. If the service is running, the button label is Stop Server button. Click it to stop the server. If the service is stopped, the button label is Start Server. Click it to start the service.

Under `Log Messages`, you will find messages that the MySQL server produces when it starts or stops. Messages produced by the server during startup look something like this:

```
    Trying to start the server ...
    040119 18:33:41  InnoDB: Started
    MySql: ready for connections.
    Version: '4.0.16-max-nt-log'  socket: ''  port: 3306
    Server was started.
```

When the server stops, it produces messages such as these:

```
    Trying to stop the server ...
    040119 18:31:45  MySql: Normal shutdown
    040119 18:31:46  MySql: Forcing close of thread 11  user:
'superuser'
    040119 18:31:46  MySql: Forcing close of thread 8  user: 'superuser'
    ...
    Server was stopped.
```

The details and format of the messages might vary depending on the version of the MySQL server you are using and its configuration.

6.3. Configure Service

This tab allows configuration of the Windows service that starts and stops the MySQL server, as well as setting some options for the server itself. This tab is specific to MySQL servers running under operating systems of the Windows NT family, including Windows NT, Windows 2000, and Windows XP; it is not available for servers running under a different operating system. Obviously, you need to have a Windows service installed before you can configure it. The Starting MySQL as a Windows Service [http://dev.mysql.com/doc/mysql/en/Windows_start_service.html] section describes how to set up a Windows service.

Note that if you have more than one MySQL server running as a Windows service, you should run MySQL Administrator in configure -service mode to administer these services. In normal mode, you can only administer the Windows service listed first in the Windows service manager, no matter which server you are connected to.

Note that the settings in the Server Features group take effect only after restarting the MySQL server.

The following settings are available:

- Service Settings
  - `Launch MySQL server automatically`: If checked, the Windows service starts the MySQL server automatically on operating system startup.
  - `Display Name`: The display name of the Windows service (not the service name!) as it will appear in the Windows services manager when displaying the properties of that service.
  - `Service Description`: The description of the Windows service as it will appear in the Windows services manager.
- Option File
  - `Config Filename`: You can enter the pathname of the server option file. MySQL will enter that information in the Windows registry, and the Windows service starting the MySQL server will use that file, rather than the default file (`C:\`*`windows installation directory`*`\my.ini`). *Note that this is available for MySQL servers as of version 4.0.3 only.* Details about the server option files can be found in the Creating an Option File [http://dev.mysql.com/doc/mysql/en/Windows_create_option_file.html] section.
  - `Section Name`: Setting a value other than `mysqld` is most useful when running multiple servers; see Running Multiple Servers on Windows [http://dev.mysql.com/doc/mysql/en/Multiple_Windows_servers.html].
- Server Features

  Note that all server features will only take effect after the MySQL server has been restarted.

  - `Support for InnoDB`: If checked, the MySQL server is started with support for InnoDB tables. If unchecked, it doesn't start the InnoDB storage engine; this saves memory, but InnoDB tables cannot be used. If this checkbox is unchecked and greyed out, it means that your MySQL server isn't compiled with support for InnoDB tables. (This is, for example, true for some versions that run under a commercial license.) You can find more information on InnoDB tables in the MySQL Storage Engines and Table Types [http://dev.mysql.com/doc/mysql/en/Storage_engines.html] section.

    Be careful when disabling this option! If any databases on the MySQL server contain InnoDB tables, disabling this option makes those tables inaccessible.

  - `Support for BDB`: If checked, the MySQL server is started with support for BDB tables. If unchecked, it doesn't start the BDB storage engine; this saves memory, but BDB tables cannot be used. If this checkbox is unchecked and greyed out, it means that your MySQL server isn't compiled with support for BDB tables. (This is, for example, true for non-max versions.) You can find more information on BDB tables in the MySQL Storage Engines and Table Types [http://dev.mysql.com/doc/mysql/en/Storage_engines.html] section.

    Be careful when disabling this option! If any databases on the MySQL server contain BDB tables, disabling this option makes those tables inaccessible.

  - `Named Pipes`: On operating systems of the Windows NT family, including Windows NT, Windows 2000, and Windows XP, you may use named pipes to connect to a MySQL server running on the local machine, rather than connecting via TCP/IP. If this option is checked, named pipe connections are allowed. Note that named pipes are slower than TCP/IP in many Windows configurations. See the Selecting a MySQL Server type [http://dev.mysql.com/doc/mysql/en/Windows_select_server.html] section for details.

- `Debug Information (slow)`: If checked, the MySQL server is started with an option that allows for easier debugging of problems. Using this option slows down the server significantly, so it is recommended only for debugging purposes. If the MySQL server has not been compiled with debugging support, this option is greyed out. See the `mysqld` Command-line Options [http://dev.mysql.com/doc/mysql/en/Server_options.html] section for details.
- `Path to Binary`: Contains the pathname of the MySQL server binary file as it is listed in the Windows service manager (for example `C:\mysql\bin\mysqld-max-nt`). Note that the the Windows service manager does not display the `.exe` extension.

Chapter 7. Startup Variables

7.1. Introduction

This section allows you to set variables that the MySQL server reads on startup. These variables are stored in the option file of the server. On Windows, the option file is called `my.ini`, and it is located in the Windows installation directory (for example, `C:\WinNT\my.ini`). Note that it can alternatively be stored in `C:\`, in which case it is named `my.cnf`. Also note that there can be more than one option file. On operating systems other than Windows, the option file is called `my.cnf`, regardless where it resides. On Windows, MySQL Administrator allows you to store the option file in any location you specify. You can find more information on the MySQL option files in the Using Option Files [http://dev.mysql.com/doc/mysql/en/Option_files.html] section.

Note that the server reads variables stored in the option files only at server startup. After changing an option file, you must restart the server for the changes to take effect. You can do this using the Stop Server and Start Server buttons in the Service Control section.

Chapter 8. Server Connections

8.1. Introduction

This section allows you to view and manage the current connections to the server (also called threads). If you have the PROCESS privilege, you can see all threads. If you have the SUPER privilege, you can also terminate threads other than your own. (Note that the SUPER privilege is available for MySQL servers as of version 4.0.3 only.) Otherwise, you can see your own threads only. You can find more information about threads in the SHOW PROCESSLIST [http://dev.mysql.com/doc/mysql/en/SHOW_PROCESSLIST.html] section. For more information on the PROCESS and the SUPER privilege (and other available privileges as well), see Privileges Provided by MySQL [http://dev.mysql.com/doc/mysql/en/Privileges_provided.html].

Note that you will see an additional thread for the user under which you are connected when you have clicked the Health section within your current MySQL Administrator session. The values displayed in that section are retrieved from the MySQL server via a separate connection.

8.2. Threads

This tab allows you to view all connections to the server (threads). Initially, those are sorted by their PID (MySQL process identification number), but you can change that order by clicking on the appropriate headings (User, Host, etc.). You can select a thread by clicking on it. You can select more than one thread using mouse or keyboard.

To reload the process list from the server, press the Refresh button.

If you have selected one or more threads, you can terminate them by clicking the Kill Thread button, or by right-clicking on that thread. Remember that you need the PROCESS privilege to see threads other than your own, and the SUPER privilege to kill threads other than your own. Furthermore, note that MySQL Administrator imposes a restriction that prevents you from terminating your own threads. On Windows, using the Kill Thread button on your own threads does not have an effect; on Linux, that button is disabled when selecting your own threads.

8.3. User Connections

This tab displays all connections to the server (threads), but in a different way than the Threads tab does: You will rather see users, not individual threads in the first place. The number of connections a user has currently established to the MySQL server is listed under Num.. Right-clicking on a username and selecting Show User Info will also display the Full Name and the Description, as specified in the User Information tab of the User Administration section. Initially, users are sorted by their MySQL Username, but you can change that order by clicking on the appropriate headings (Num., Full Name, or Description). Clicking on a username will display the individual threads of that user in the lower box of that tab, ordered by PID. Likewise, you can change that order by clicking on the appropriate headings.

Clicking on a user will enable the Kill User button, which allows you to terminate all threads of that user with a single click. That command is also available by right-clicking on the username. Clicking on an individual thread in the lower box will enable the Kill Thread button, which is used to terminate a connection. That command is available by right-clicking on a thread, too.

To reload the user list from the server, press the Refresh button.

Chapter 9. User Administration

9.1. Introduction

This section allows you to administer existing users, add new users, and delete existing users. For more information on the MySQL user account management and privilege system, see the MySQL User Account Management [http://dev.mysql.com/doc/mysql/en/User_Account_Management.html] section.

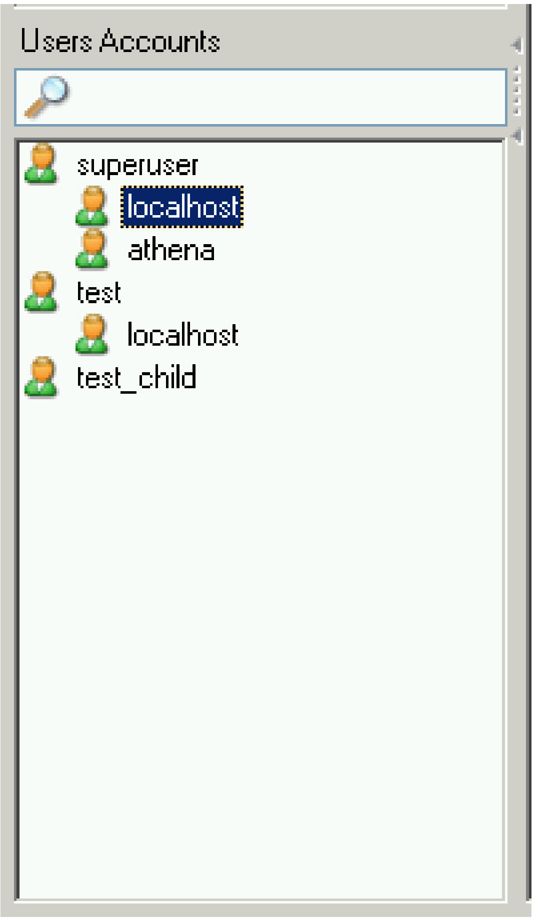
9.2. User Accounts

Existing users are listed in the lower left area of the sidebar. If there are many user accounts on your MySQL server, the field with the magnifying-glass icon may be handy for filtering the users you are looking for. Typing in `t` or `T`, for example, will set the filter to users whose names begin with a `t`. The filter works in a case-insensitive fashion.

If you are an experienced MySQL user, you will notice that users are not listed the typical MySQL way (*`user@host`*), but rather with their usernames only. If double-clicking on a username does not show subcategories for that user, this means that the user's privileges are not restricted to a specific host, or hosts. If there are subcategories, the user's privileges are restricted to a specific host, or set of hosts. You may assign various (and different) sets of privileges, depending on the host from which the user connects to the MySQL server. That concept is described in detail in the Access Control, Stage 1: Connection Verification [http://dev.mysql.com/doc/mysql/en/Connection_access.html] section.

Note that MySQL Administrator has a different concept of what a `user` is than MySQL has. In MySQL, a user is always identified by a `username/host` combination. This means that, for example, `'brian'@'%'` may be a user completely different from `'brian'@'localhost'`. The former might be `Brian Miller`, while the latter might be `Brian Schultz`. That distinction does not hold true for MySQL Administrator: User `brian` is always a particular user, no matter from which host he connects to the MySQL server. That said, `'brian'@'%'` may still have privileges different from `'brian'@'localhost'`.

Figure 9.1. User accounts



For example, you might have a user called `superuser`. If double-clicking on that username shows two subcategories, `localhost` and `athena`, this means that the user has a specific set of privileges if he/she connects from `localhost`, and a (probably different) set of privileges if he/she connects from `athena`.

Right-clicking on a username, or on one of the subcategories a user might have, provides access to the following commands:

- `Add a new User`: Selecting this command adds a new user with the default name `New User` to the user list. You will have to fill in at least a username in the `MySQL User` field of the User Information tab, which, after applying this change, will also rename that user in the user accounts list.
- `Add Host from which the User can connect`: Allows you to specify a host from which the user can connect. This command is unavailable if you highlight a subcategory.
- `Remove Host from which the User can connect`: Allows you to remove a host from which the user can connect. If there are no sub-categories, the user will be removed (you will be prompted to confirm the removal).
- `Clone User`: Makes a copy of the selected user, including all subcategories. The new user is called `New User` until you specify another name in the `MySQL User` field of the User Information tab. With that command, you can therefore create a user with identical privileges.
- `Delete User`: Allows you to delete the selected user. You will be prompted to confirm that you really want to delete that user.

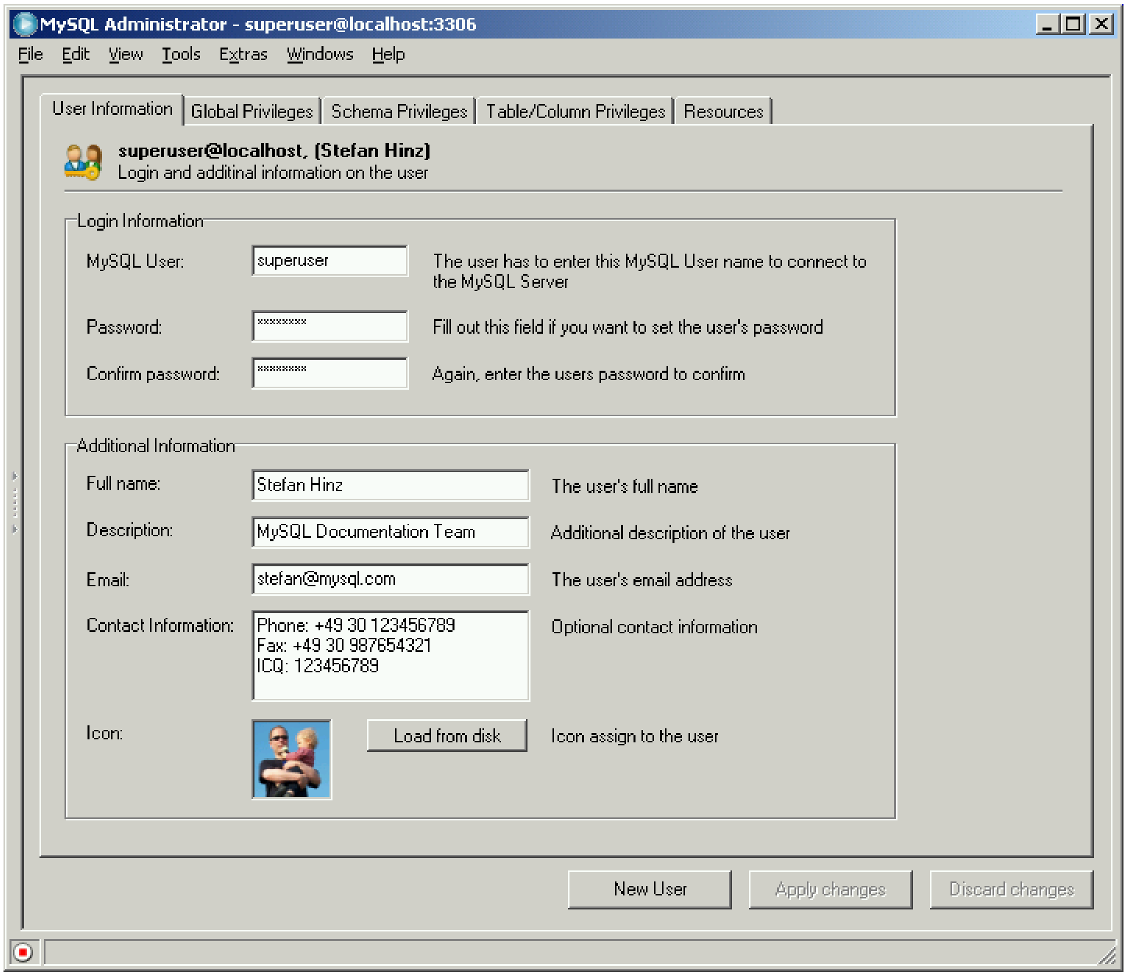
- `Refresh User List`: Rereads the user list from the MySQL server. This is helpful when other users (on other connections) are editing user accounts. Note that selecting that command will collapse all subcategories.

9.3. User Information

The fields in this tab are unavailable unless you either click on a username in the Us er Accounts area of the sidebar, or create a new user in that sidebar or using the New User button on this tab. In the former case, the fields are filled with the information stored for that user, in the latter case, all fields are empty.

Note that all fields will be stored only when you click the Apply Changes button. If you don't want to save your changes, click the Discard Changes button.

Figure 9.2. User Information tab



superuser@localhost, (Stefan Hinz)
Login and additinal information on the user

Login Information

MySQL User:	superuser	The user has to enter this MySQL User name to connect to the MySQL Server
Password:	*********	Fill out this field if you want to set the user's password
Confirm password:	*********	Again, enter the users password to confirm

Additional Information

Full name:	Stefan Hinz	The user's full name
Description:	MySQL Documentation Team	Additional description of the user
Email:	stefan@mysql.com	The user's email address
Contact Information:	Phone: +49 30 123456789 Fax: +49 30 987654321 ICQ: 123456789	Optional contact information
Icon:		Icon assign to the user

The values you enter in the fields of this tab are stored on the MySQL server. The fields in the Login Information group of this tab are stored in the `user` table of the `mysql` database, while the fields in the Additional Information group are stored in the `user-info` table of the `mysql` database. The latter table is created the first time you add a new user, or change an existing user. Its contents may look like this:

```
mysql> SELECT * FROM mysql.user-info \G
       *************************** 1. row ************
                      User: superuser
                 Full-name: Stefan Hinz
               Description: MySQL Documentation Team
                     Email: stefan@mysql.com
       Contact-information: Phone: +49 30 123456789
                            Fax: +49 30 987654321
                            ICQ: 123456789
                      Icon:
```

- Login Information
    - `MySQL User`: Username that is specified when connecting to the MySQL server. Note that this username has nothing to do with usernames you might use elsewhere on your operating system. It is recommendable not to use special characters or spaces for that username.
    - `Password`: Password that is specified when connecting to the MySQL server. Note that this password has nothing to do with passwords you might use elsewhere on your operating system. It is recommendable not to use special characters or spaces for that password. It is, however, recommendable to use non-trivial passwords. A very trivial password might be `abcde`, while a non-trivial password might be `lEtusMak3iThartdoGesz`. Note that it is not mandatory to specify a password, although it is highly recommendable to do so.
    - `Confirm Password`: Fill in the password once again to make sure you do not accidentally introduce a typo.
- Additional Information

  All information specified here is optional.

    - `Full Name`: As opposed to the username in `MySQL User`, the full name may contain any characters, including spaces.
    - `Description`: Additional description of user.
    - `Email`: Email address of user.
    - `Contact Information`: More contact information, like postal address, or messenger information.
    - `Icon`: Rather than using the default icon, you may select a different icon by pressing the Load from Disk button. You can either use one of the buttons that are shipped with MySQL Administrator , or use an icon of your own. In the latter case, be aware that the icon has to be exactly 48 by 48 pixels in size, and its file format has to be `PNG` (portable network graphics).

9.4. Global Privileges

This tab is available only if the `Show Global Privileges` checkbox in the Admin istrator section of the Options dialog has been checked.

Under Assigned Privileges, you find the global privileges assigned to the selected user. For more information about those privileges, see the Privileges Provided by MySQL [http://dev.mysql.com/doc/mysql/en/Privileges_provided.html] section. *Global* means that those privileges apply to the MySQL server in general (like the `Shutdown` privilege), or to all databases on the server (like the `SELECT` privilege).

Privileges not assigned to the user are listed under Available Privileges. You can remove privileges by selecting them in the left box, and then clicking the > button. Removed privileges will appear in the Available Privileges box. You can assign privileges by selecting them in the Available Privileges box, and then clicking the < button.

9.5. Schema Privileges

This tab allows you to grant privileges on a schema level. In MySQL, a schema is the same as a database. To grant privileges for a specific database, click on the database name in the left box. (You cannot select more than one database at a time.)

Privileges not assigned to the user are listed under Available Privileges. You can remove privileges by selecting them in the left box, and then clicking the > button. Removed privileges will appear in the Available Privileges box. You can assign privileges by selecting them in the Available Privileges box, and then clicking the < button.

9.6. Table Column Privileges

This tab is available only if the `Show Table/Column Privileges` checkbox in the Admin istrator section of the Options dialog has been checked. It allows you to grant privileges on a table level or on a column level.

To access the tables, double-click on the schema (database) name containing that table. You will not see and cannot grant privileges on a schema level in this tab; if you want to do that, use the Global Privileges tab instead. Also, you cannot select more than one table at a time.

If you want to grant privileges on a column level, double-click on the table name; this will give you access to its columns. You cannot select more than one column at a time.

Privileges not assigned to the user are listed under Available Privileges. You can remove privileges by selecting them in the left box, and then clicking the > button. Removed privileges will appear in the Available Privileges box. You can assign privileges by selecting them in the Available Privileges box, and then clicking the < button.

Chapter 10. Health

10.1. Introduction

The Health section of MySQL Administrator allows you to graphically monitor the real-time status of your MySQL server. A selection of pre-made status graphs are available, and you can easily create your own status graphs and organize them into custom pages and groups.

The default graphs provide graphical information about the MySQL server's current performance, load, and memory usage. Additional pages list all server variables, divided into pages for status variables and system variables. Note that clicking on the status pages will open an additional connection to the MySQL server, which you can see when viewing threads and users in the Server Connections section.

We'll begin by covering the different pre-made status graphs, and then describe how to create your own status graphs yourself. Finally we'll describe the pages in the Health section dedicated to the various status and system variables.

10.2. Connection Health

This tab provides information about connections to the MySQL server. The following three pre-made graphs are displayed:

- Connection Usage: Displays a percentage of the maximum number of simultaneous clients allowed (threads connected / max_connections). If this value gets close to 100%, it could indicate that you should increase the value of max_connections, which you can do in the Startup Variables section. The maximum number of connections since MySQL Administrator was loaded is also shown. For more information on server connections, see the SHOW VARIABLES [http://dev.mysql.com/doc/mysql/en/SHOW_VARIABLES.html] and Too many connections Error [http://dev.mysql.com/doc/mysql/en/Too_many_connections.html] sections of the reference manual.
- Traffic: This is a representation of the bytes_sent status variable. This does not represent total bytes sent since server startup, but instead represents the current number of bytes sent (the bytes_sent variable is a cumulative total). The initial maximum value of the graph is 102400, this maximum value will be automatically increased as needed.
- Number of SQL Queries: This is a representation os the com_select status variable. This does not represent total queries since server startup, but instead represents the current number of queries (the com_select variable is a cumulative total). The initial maximum value of the graph is 10, this maximum value will be automatically increased as needed.

10.3. Memory Health

This tab provides information about memory caches that affect the performance of the MySQL server. For each cache, there is a display that shows the current hitrate or usage of the cache, and a display that shows the hitrate or usage for the last few minutes.

- Query Cache Hitrate: When in use, the query cache of the MySQL server stores the text of a SELECT query together with the corresponding result that was sent to the client. If an identical query is later received, the server will retrieve the results from the query cache rather than parsing and executing the same query again. The hitrate indicates the ratio of queries that were cached and queries that were not cached. The higher the hitrate, the better the performance of the server regarding `SELECT` queries. See the MySQL Query Cache [http://dev.mysql.com/doc/mysql/en/Query_Cache.html] section for details.

- `Key Efficiency`: The horizontal bar graph indicates how much of the allocated key buffer is in use. If a fairly large percentage of the key buffer is in use you may wish to increase the `key_cache_size` setting in the Startup Variables section. The line graph is an indication of the number of `key_read_requests` that resulted in actual `key_reads`. The higher the hitrate the more effective your key buffer is. A low hitrate can indicate that you need to increate the size of your keycache.

10.4. Creating Custom Health Graphs

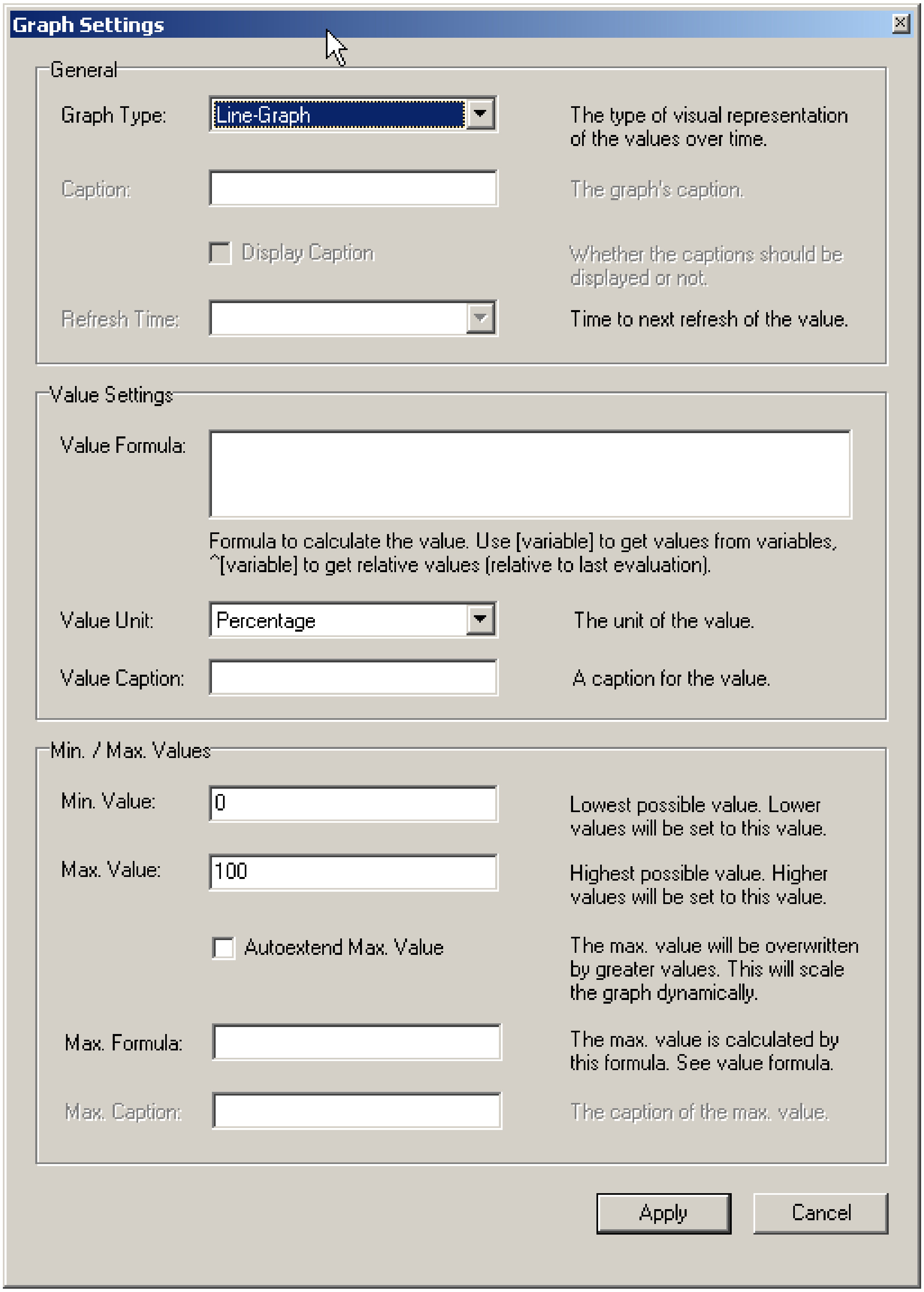
While the pre-made status graphs can provide you with informatin of the current state of your MySQL server, there may be additionally information that you wish to visually track. The MySQL Administrator `Health` section allows you to add additional tabs and graphs to track any information that is contained within the Status Variables.

The first step in creating custom health graphs is to create a new page in the `Health` section to store your graphs. To create a new page right-click anywhere in the working area and choose the Add a Page option. You will be prompted for a page name and description and a new page will be created.

Once you have created a new page, you will need to create a group. All graphs are located within groups. For example, the `Key buffer usage` and `Key buffer hitrate` graphs are both located within a group titled `Key Efficiency`. To create a group, right-click within your newly created page and choose the Add a Group option.

To create a custom graph, right-click within a group and choose the Add a graph option. The following dialog will appear:

Figure 10.1. The Graph Settings dialog



Graph Settings

General

Graph Type: Line-Graph — The type of visual representation of the values over time.

Caption: — The graph's caption.

□ Display Caption — Whether the captions should be displayed or not.

Refresh Time: — Time to next refresh of the value.

Value Settings

Value Formula:

Formula to calculate the value. Use [variable] to get values from variables, ^[variable] to get relative values (relative to last evaluation).

Value Unit: Percentage — The unit of the value.

Value Caption: — A caption for the value.

Min. / Max. Values

Min. Value: 0 — Lowest possible value. Lower values will be set to this value.

Max. Value: 100 — Highest possible value. Higher values will be set to this value.

□ Autoextend Max. Value — The max. value will be overwritten by greater values. This will scale the graph dynamically.

Max. Formula: — The max. value is calculated by this formula. See value formula.

Max. Caption: — The caption of the max. value.

The Line-Graph is appropriate for showing trends over time, while the Bar-Graph option will be more appropriate for showing percentage information. If you choose to create a bar-graph, you will need to choose a caption for your graph. Captions are not allowed for line-graphs.

The data on your graph is set using the `Value Formula`. You can create a formula using any of the variables available in the Status Variables and System Variables tabs. To use the value of a variable, wrap the variable name in square brackets (i.e. `[com_select]`), if you want the relative value of the variable prepend the square brackets with a `^` character.

For example, if we wanted to graph the percentage of temporary tables that were created on disk we could use the following formula:

```
[created_tmp_disk_tables] / [created_tmp_tables]
```

If we wanted to track the number of temporary tables created on a continious basis, we could use the following as our formula:

```
^[created_tmp_tables]
```

After creating your formula, select the `Value Unit` that best represents your data. You can choose from `Percentage`, `Count`, `Byte`, and `Seconds`. In our examples we would have used `Percentage` for the first example, and `Count` for the second. You can also assign a caption to the value.

Once your formula is assigned you should configure the `Max. Value` and `Min. Value` for the graph, so that your data is spread evenly across your graph. You can set arbitrary values based on your estimates of how large the values will grow, and check the Autoextend Max. Value option to allow MySQL Administrator to automatically increase the `Max Value` setting automatically to prevent your data from extending off of your graph.

You can also set the `Max. Value` option by way of a formula assigned in the Max Formular field. The same syntax applies in this field as applies in the Value Formula field. For example, if you were creating a bar graph tracking the number of temporary disk tables created, you could use `[created_tmp_disk_tables]` as the Value Formula and `[created_tmp_tables]` as the Max. Formular.

10.5. Status Variables

This tab provides information about the status variables of the MySQL server. In the left box of the working area, you find categories and subcategories. Double-click on a category name to expand or collapse its subcategories.

For each category, and each subcategory, the status variables are listed in the right box of the working area, together with their values, and with a short explanation. The status variables are described in detail in the `SHOW STATUS` [http://dev.mysql.com/doc/mysql/en/SHOW_STATUS.html] section of the reference manual.

Use the Refresh button to make MySQL Administrator retrieve the variables and their values once again.

Right-clicking in the right box allows you to copy variables to the clipboard.

10.6. System Variables

This tab provides information about the system variables of the MySQL server. In the left box of the working area, you can find categories and subcategories. Double-click on a category name to expand or collapse its subcategories.

For each category, and each subcategory, the system variables are listed in the right box of the working area, together with their values, and with a short explanation. The system variables are described in detail in the `SHOW VARIABLES` [http://dev.mysql.com/doc/mysql/en/SHOW_VARIABLES.html] section of the reference manual.

System variables that can be set at runtime are marked as editable, that is they are prefixed with an icon different from the variables that cannot be set at runtime. Double-clicking on editable variables will open a dialog where you can specify another value for the variable. The changes you make will take effect as soon as you click the OK button in that dialog. Click Abort if you want to keep the current value. Changes made in this manner will only persist until the server is restarted, after which the server will revert to it's startup settings. In MySQL Administrator, system variables are set on a global level only. Note that you need appropriate privileges to change variables globally. You can find more information on global variables in the `SET` Syntax [http://dev.mysql.com/doc/mysql/en/SET_OPTION.html] section.

Use the Refresh button to make MySQL Administrator retrieve the variables and their values once again.

Right-clicking in the right box allows you to copy variables to the clipboard.

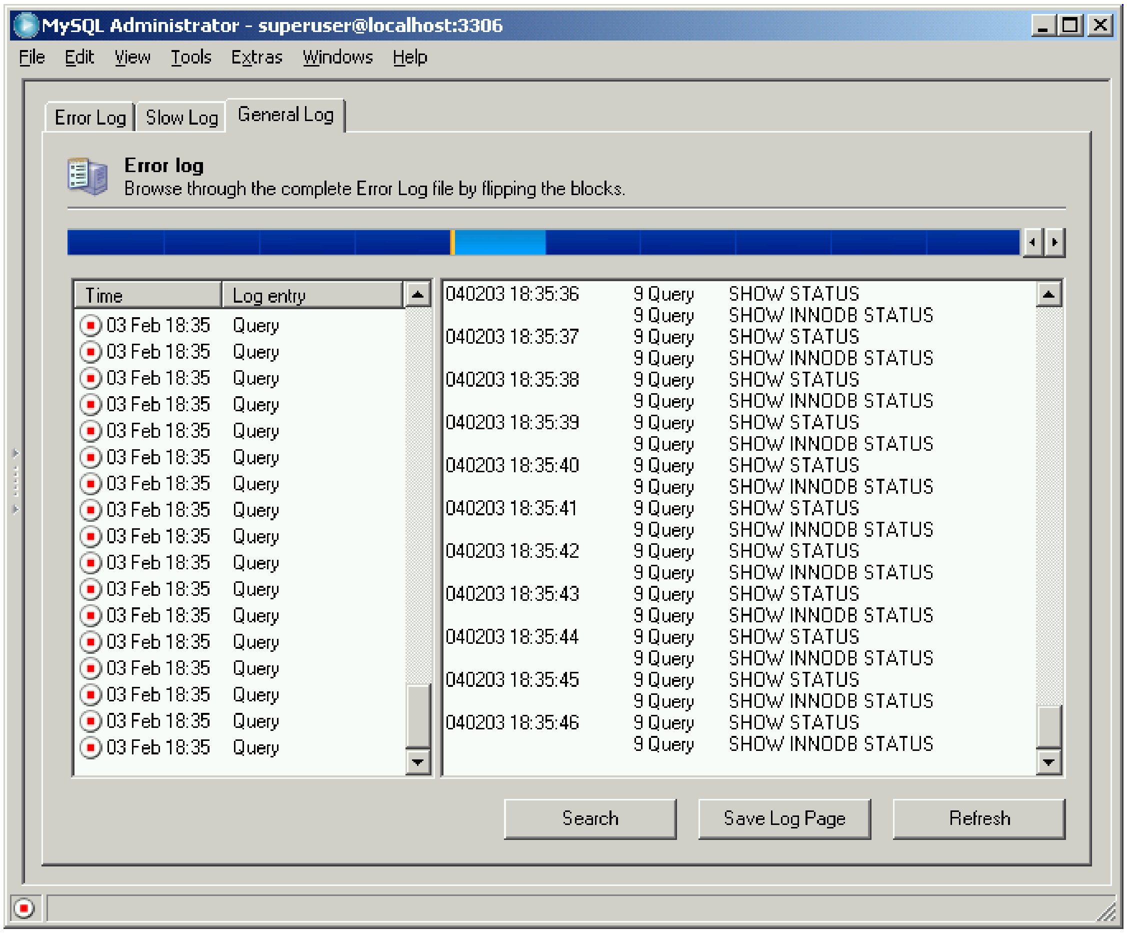
Chapter 11. Server Logs

11.1. Introduction

The `Server Logs` section provides easy access to the various log files generated by the MySQL server. You can access the server's error log, slow query log, and general log using this section.

This section has one tab for each of the log files, and each tab is split into two boxes. Above the boxes, there is a bar which allows you to navigate through larger log files by breaking the log file into individual pages. Clicking in the bar will display the individual pages. The light blue area represents the portion of the log file you are currently viewing.

Figure 11.1. Displaying entries from the middle of the General Log


MySQL Administrator - superuser@localhost:3306
File Edit View Tools Extras Windows Help
Error Log
Slow Log
General Log
Error log
Search
Save Log Page
Refresh

Browse through the complete Error Log file by flipping the blocks.

Time	Log entry
03 Feb 18:35	Query
03 Feb 18:35	Query
03 Feb 18:35	Query
03 Feb 18:35	Query
03 Feb 18:35	Query
03 Feb 18:35	Query
03 Feb 18:35	Query
03 Feb 18:35	Query
03 Feb 18:35	Query
03 Feb 18:35	Query
03 Feb 18:35	Query
03 Feb 18:35	Query
03 Feb 18:35	Query
03 Feb 18:35	Query
03 Feb 18:35	Query
03 Feb 18:35	Query
03 Feb 18:35	Query

```
040203 18:35:36      9 Query      SHOW STATUS
                     9 Query      SHOW INNODB STATUS
040203 18:35:37      9 Query      SHOW STATUS
                     9 Query      SHOW INNODB STATUS
040203 18:35:38      9 Query      SHOW STATUS
                     9 Query      SHOW INNODB STATUS
040203 18:35:39      9 Query      SHOW STATUS
                     9 Query      SHOW INNODB STATUS
040203 18:35:40      9 Query      SHOW STATUS
                     9 Query      SHOW INNODB STATUS
040203 18:35:41      9 Query      SHOW STATUS
                     9 Query      SHOW INNODB STATUS
040203 18:35:42      9 Query      SHOW STATUS
                     9 Query      SHOW INNODB STATUS
040203 18:35:43      9 Query      SHOW STATUS
                     9 Query      SHOW INNODB STATUS
040203 18:35:44      9 Query      SHOW STATUS
                     9 Query      SHOW INNODB STATUS
040203 18:35:45      9 Query      SHOW STATUS
                     9 Query      SHOW INNODB STATUS
040203 18:35:46      9 Query      SHOW STATUS
                     9 Query      SHOW INNODB STATUS
```

You can only view log files that are stored on the machine where MySQL Administrator is running, you cannot view log files on a remote machine.

Also note that some tabs may be disabled. If a tab is disabled it indicates that the log file in question has not been enabled on the MySQL server. You can enable the various log files in the Startup Variables section.

The left box (labeled `Events of page x / y`) displays the major events that have occured on the current page of the log file. The right box (labeled `Page Content`) displays the detailed lines of the log file. By clicking on an event in the left box, the first detailed entry of the event will be highlighted in the right box.

In addition to the log file tabs, the `Server Logs` section has four action buttons for managing log information:

- Open Other File will allow you to open a log file other than the currently active log file. This can be useful in situations where you rotate your log files as it allows you to open log files that have been rotated out of active use.
- Search opens a dialog where you can specify a search term to look for in the log file. If it's found, it will be marked as selected in the right box. This command is also available by right-clicking in the right box and choosing the Search ... option.
- Save Log Page opens a dialog that allows you to specify the path and file name where the current page of the log file should be stored. By default, MySQL Administrator suggests `ErrorLog.txt`, `SlowLog.txt`, or `GeneralLog.txt` as a filename. Note that executing this command does not affect the actual log file in any way; it just creates a copy of the *current* log file contents that you may store for documenting or analyzing. This command is also available by right-clicking in the right box and choosing the Save log Page ... option.
- Refresh will reload the log files.

11.2. Error Log

Displays the MySQL server's error log. The error log file contains information indicating when the MySQL server was started and stopped and also any critical errors found when running. For more information about that log file, see The Error Log [http://dev.mysql.com/doc/mysql/en/Error_log.html].

11.3. Slow Log

Displays the MySQL server's slow query log. If enabled (which you can do in the Startup Variables section), the MySQL server writes a log file containing all SQL commands that took more than a specific number of seconds to execute. (That number can be configured by the `long-query-time` variable; its default value is 10 seconds.) The time to get the initial table locks are not counted as execution time. For more information about that log file, see The Slow Query Log [http://dev.mysql.com/doc/mysql/en/Slow_query_log.html].

11.4. General Log

Displays the MySQL server's general query log. This log can be very useful when you suspect an error in a client and want to know exactly what the MySQL server thought the client sent to it. For more information about that log file, see The General Query Log [http://dev.mysql.com/doc/mysql/en/Query_log.html].

Chapter 12. Backup

12.1. Introduction

This section allows you to create backup projects, and to perform backups. This section will produce backup files that are plaintext and composed of multiple SQL statements that can be executed to restore your database, similar to the output generated by the `mysqldump` utility. For more information on `mysqldump`, see the `mysqldump` section [http://dev.mysql.com/doc/mysql/en/mysqldump.html] of the reference manual.

Existing backup projects are listed in the lower left area of the sidebar. If there are many backup projects, the field with the magnifying-glass icon can be used to filter the projects you are looking for. Typing in `t` or `T`, for example, will set the filter to projects whose names begin with a `t`. The filter works in a case-insensitive fashion.

You can update the list of backup projects by right-clicking on a project and choosing the Refresh Backup Projects option from the drop-down menu. You can delete a backup project by right-clicking on the project and choosing the Delete Backup Project(s) option from the drop-down menu.

If you are not working from a previously saved backup project, click the New Project button to create a new backup project. You cannot begin configuring a backup operation without either loading an existing project or creating a new project.

Once you have configured your backup project, you can click the Save Project button to save your project for future use. To start your backup, click the Execute Backup Now button. You will be prompted for a path and filename for the backup file, after which the backup operation will begin.

12.2. Backup Project

The `Backup Project` tab allows you to set the name of your backup project and also determine what data will be backed up.

You can name your project by filling out a name in the Project Name field. If you have enabled the `Add Date/Time to Backup Files` option in the `Administrator Options` dialogue, then the date and time of the backup will be appended to the project name automatically. See Section 17.5.3, “Backup” for more information on setting this option.

Specify a name that clearly denotes the project, such as `Full Backup` or `World Only`. The project name may contain arbitrary characters, including spaces.

To choose the databases and tables that will be backed up, select the database from the schemata list and click the > button. Once the database is in the `Backup Content` list, you can un-check specific tables that you do not wish to back up.

12.3. Advanced Options

The `Advanced Options` tab allows you to configure advanced options that affect the way the backup operation behaves. These include concurrency controls, backup type, and output formatting.

12.3.1. Backup Execution Method

The `Backup Execution Method` affects concurrency within the backup itself.

`Lock all tables` will cause all tables within the backup to be locked simultaneously to prevent writing. This will prevent users from writing to the tables during the backup operation and will ensure that the tables are in a consistent state. This option is recommended for MyISAM tables.

`Single transaction` will begin a transaction before reading from the tables. Because of the InnoDB versioning system this will allow all tables to be read in a consistent state. This method is recommended when backing up InnoDB tables.

`Normal Backup` will lock each table individually when performing the backup. While this can have less impact on the actions of users during the backup operation, these is a potential for tables to be backed up in an inconsistent state in relation to one another. This method is not recommended unless the risk on inconsistency in minimal.

12.3.2. Output File Options

The `Output File Options` control the formatting of the backup file. At present there is only one `Backup Type` available, the `SQL Files` format, which outputs a file similar to the output of the `mysqldump` utility.

`No CREATEs` will supress `CREATE DATABASE` statements. This can be useful if you need to restore the data to a different database than it was backed up from.

`No EXTENDED INSERTS` will supress the default behavior of combining multiple inserts into a single statement (for example `INSERT INTO test VALUES ('1','Some data',2234),('2','Some More Data',23444)`). While the `EXTENDED INSERT` syntax is more efficient both in terms of storage and execution, it is not compatible with most RDBMSes.

`Complete INSERTs` will produce full `INSERT` statements that include a column list (i.e. `INSERT INTO tablename(columnA, columnB) VALUES(valueA, valueB)`). Without this option the column list will be omitted.

`Comment`

`Don't write full path`

`ANSI Quotes` will cause all table and database names to be quoted with ANSI style double quotes instead of back-ticks.

`Disable keys` will add an `ALTER TABLE ... DISABLE KEYS` statement to the backup file before the `INSERT` statements that populate the tables. This allows for faster recovery by loading all data before rebuilding the index information.

12.4. Schedule

Backup operations can be scheduled to be executed on a regular basis using the `Schedule` tab. Backup operations can be scheduled to execute on a daily, weekly, or monthly basis.

The enable scheduling for your backup project, check the Schedule this backup project checkbox. You will need to choose a target directory for the backup files to be placed and choose a name for the backup files. You can browse for a target directory by clicking the ... button.

If you have enabled the `Add Date/Time to Backup Files` option in the `Administrator Options` dialogue, then the date and time of the backup will be appended to the file name automatically. See Section 17.5.3, “Backup” for more information on setting this option.

Once you have decided on a file name and target directory, you can set the regularity of the backup operation. Regardless of you choice of daily, weekly, or monthly backups, you can set the time that the backup will occur by settiong the `Time` option. Set the backup time in a 24 hour format (i.e. 23:30).

When you select the `Execute Backup Weekly` option, a set of check boxes representing the days of the week will be displayed. Check the boxes corresponding to the days on you want to perform the backup.

When you select the `Execute backup monthly` option, a list of dates will appear, allowing you to choose which day of the month you want to perform the backup.

Click the Save Project button to save your preferences. After your project is saved it will be executed at the next scheduled interval. To run the backup operation immediatly click the Execute Backup Now button.

Chapter 13. Restore

13.1. Introduction

The Restore section helps with disaster recover by allowing you to restore backup files created with mysqldump or with the Backup section of MySQL Administrator. See Chapter 12, *Backup* for more information on creating backups with MySQL Administrator.

13.2. General

In this tab, you specify which backup file should be used for the restore procedure, along with information such as the backup file character set and format, and the target location for restored tables.

To set any options on this tab, you must first open a backup file by clicking the Open Backup File button at the bottom of the section. When all options are set, you can click the Start Restore button to begin the restore process.

The following options are available:

- File to restore: Specify path and filename of the backup file. You can browse for that file by clicking the Open Backup File button.
- Target Schema: You can choose the Original Schema option to have all tables restored into their original databases. You can also choose the New Schema... option to restore all tables into a new database. Finally you can choose from one of the existing databases in the drop-down list to force all tables to be restored to that database.
- Backup Type: At the moment, the only available option is SQL Files. Those are backup files containing SQL statements such as those produced by mysqldump.
- Force - Continue even if we get an sql error: This will cause a restore operation to continue executing even if errors are encountered during the restore operation.
- Create database(s) if they don't exist: If a database is referenced in the backup file that does not exist, the database will automatically be created if this option is checked.
- File Charset: Backup files created with MySQL Administrator are encoded in the utf8 character set. Backup files created with other tools such as winmysqladmin may be encoded with other character sets. Before a backup file can be successfully imported you will need to specify its character set.

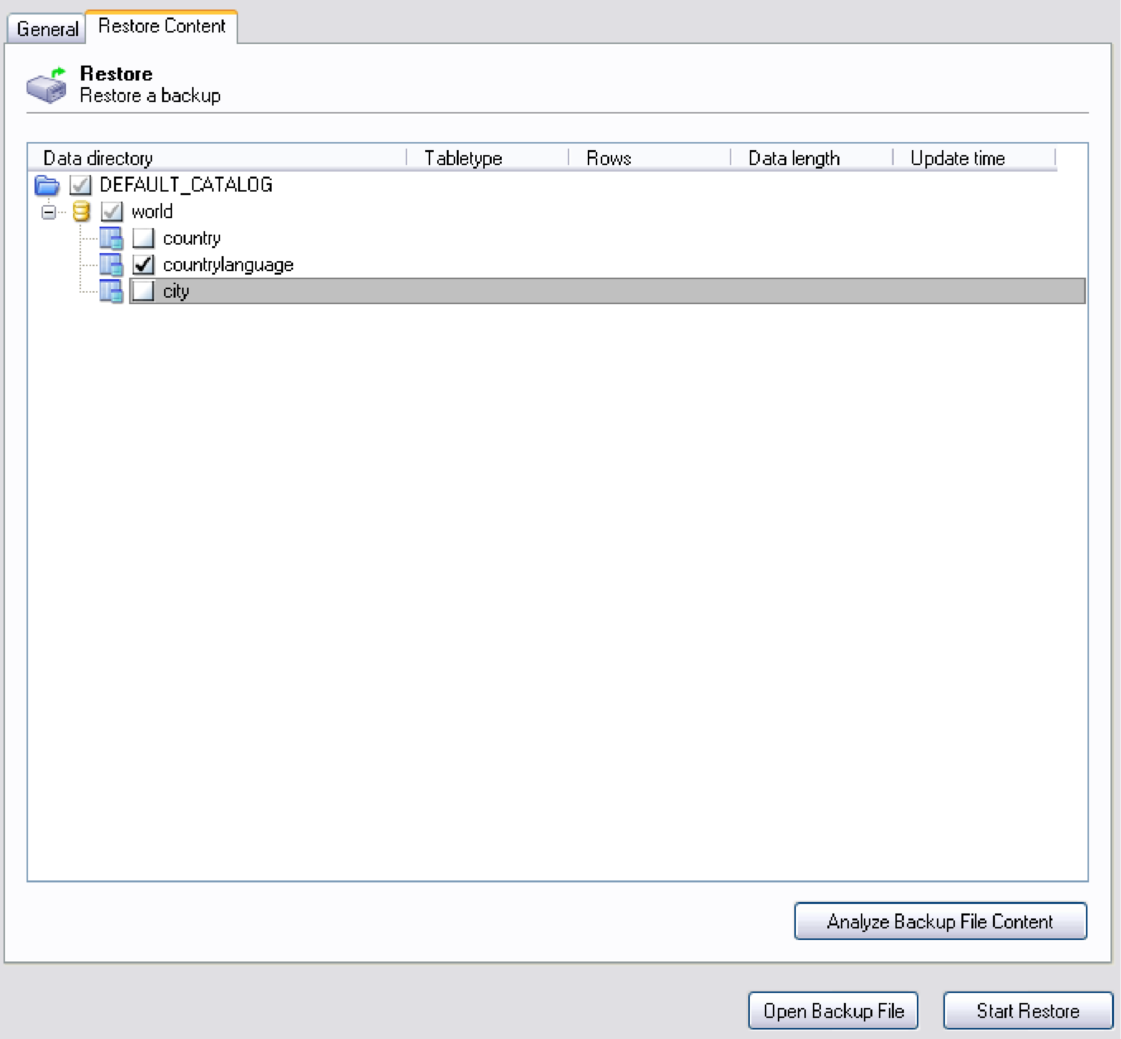
  If you do not know the character set of your backup file, click the Auto-Detect Character Set button to have MySQL Administrator attempt to determine it automatically.

13.3. Restore Content

If you do not wish to restore all the databases or tables listed within your backup file, you can use the Restore Content tab to limit which databases and tables are restored.

After clicking the Analyze Backup File Content button, a list of all databases and tables contained within the backup file will be displayed.

Figure 13.1. The Restore Content tab



- ☑ DEFAULT_CATALOG
  - ☑ world
    - □ country
    - ☑ countrylanguage
    - □ city

To prevent a database or table from being restored, un-check the box next to the object's name.

Chapter 14. Replication Status

14.1. Introduction

The `Replication Status` sections give you a visual overview of your MySQL replication master and slaves. You can tell which slave machines are connected to the master and which ones are offline through the server status icons beside each machine.

14.2. Configuring Replication Servers

To use the `Replication Status` section you must be connected to the server that acts as the master in your replication scheme. Your MySQL servers must be version 4 or higher to use this functionality.

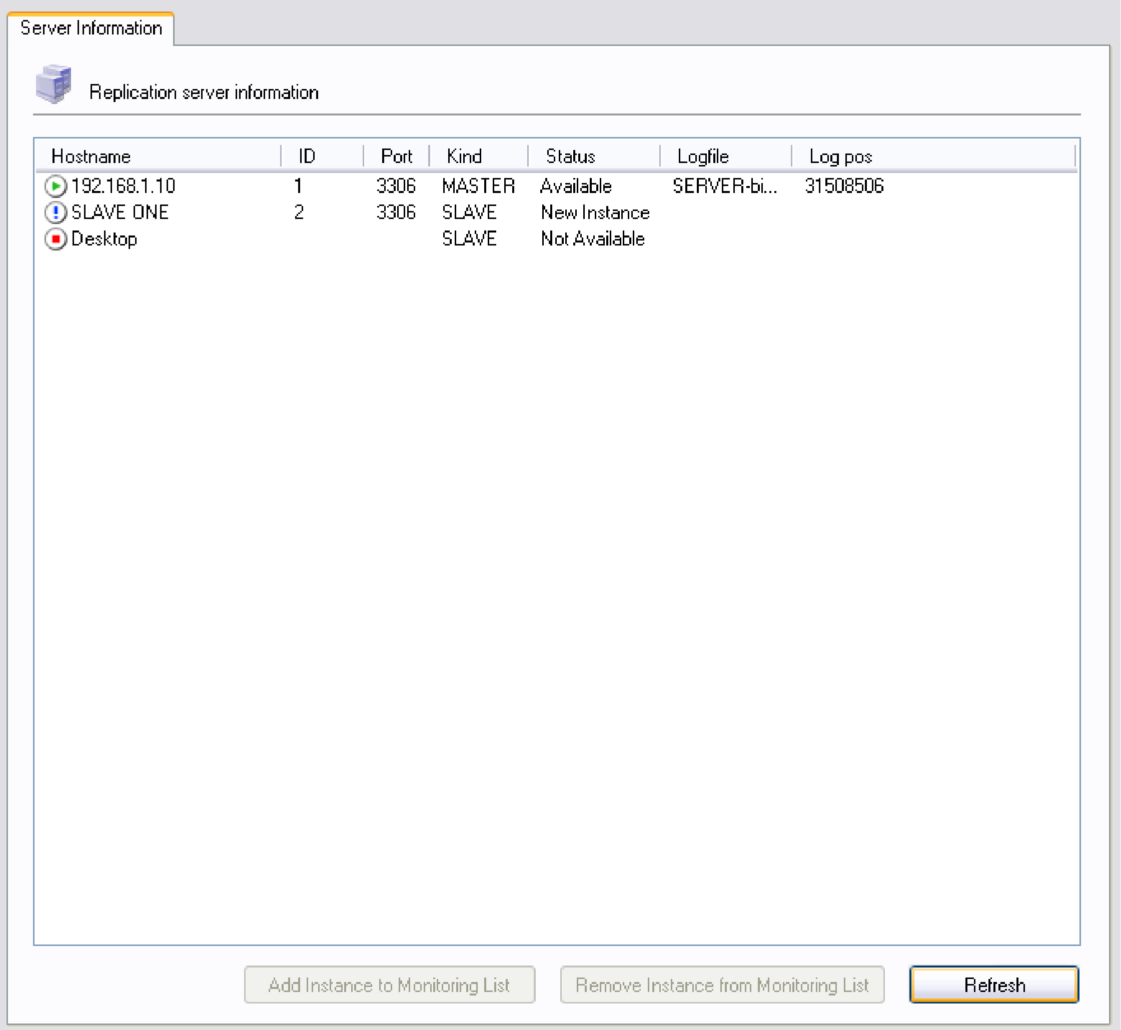
For a slave server to show in the `Replication Status` screen, it must register itself with the master server. To have a slave server register itself with the master, set the `report-host` option in your server configuration (my.cnf/my.ini) file. See http://dev.mysql.com/doc/mysql/en/Replication_Options.html for more information.

You can also set the value in the `Replication` tab of the `Startup Variables` section of MySQL Administrator.

14.3. Using the Replication Status Section

Once your servers are configured, they will appear in the `Replication Status` section, as seen in the following figure:

Figure 14.1. The Replication Status section


Server Information
Replication server information
Add Instance to Monitoring List
Remove Instance from Monitoring List
Refresh

Hostname	ID	Port	Kind	Status	Logfile	Log pos
▶ 192.168.1.10	1	3306	MASTER	Available	SERVER-bi...	31508506
! SLAVE ONE	2	3306	SLAVE	New Instance		
■ Desktop			SLAVE	Not Available		

All servers are listed in the main area of the `Replication Status` section, along with their server ID, Port, and current status.

New servers that are not currently being monitored will have an exclamation mark as their status symbol, and will be listed as a `New Instance` in the `Status` section of their entry.

Servers that are currently being monitored and are online will have a green status icon and will be listed as `Available` in the `Status` section of their entry.

Servers that are currently not in contact with the master server will have a red status icon and will be listed as `Not Available`.

There will be a slight delay between a slave machine going offline and having its status updated to allow for an adequate timeout in the communications between master and slave.

To start monitoring a new server that is not currently being monitored, click the Add Instance to Monitoring List button. To stop monitoring a server that is currently being monitored, click the Remove Instance from monitoring List button.

Chapter 15. Catalogs

15.1. Introduction

This section allows you to view catalogs, schemata, tables and their properties, and to perform a number of operations on tables. Existing catalogs and schemata are listed in the lower left area of the sidebar. If there are many schemata on your MySQL server, the field with the magnifying-glass icon may be handy for filtering the schemata you are looking for. Typing in `t` or `T`, for example, will set the filter to schemata whose names begin with a `t`. The filter works in a case-insensitive fashion.

Select the database that you wish to view. Right-click on any database and choose the Refresh Schemata List option to refresh the database list. Clicking on a database will cause the `Schema Tables` and `Schema Indices` tabs to be populated with the tables and indexes of the selected database.

To create a new database, right-click on an existing database and choose the Create New Schema option.

15.2. Schema Tables

The `Schema Tables` tab lists the different tables in the selected database, along with the rowcount, data length, and index length for those tables. For MyISAM tables, you also find the date and time when the table was last modified, under `Update Time`. You can update the list of tables by pressing the Refresh button.

The tables are ordered by table name initially, but you may change that sorting by clicking on the appropriate headings (`Type`, `Row Format`, etc.).

To drop a table, right-click on the table and choose the Drop Table option from the pop-up menu. To create a table you can either click the Create Table button at the bottom of the window, or right-click on a table and choose the Create Table option from the pop-up menu.

To edit a table's columns and indexes, right-click on the desired table and choose the Edit Table option from the pop-up menu. This will launch the MySQL Table Editor, which you can use to modify the table. See Chapter 16, *The MySQL Table Editor* .

To edit a table's actual data, right-click on the table and choose the Edit Table Data option. This will launch the MySQL Query Browser and load the table's data into the result area of the MySQL Query Browser.

You may select more than one table at a time in order to perform operations on multiple tables. For the selected tables, you can perform the following operations by either clicking the Maintenance button at the bottom of the table list or by right-clicking on one of the selected tables and choosing one of the following options from the Maintenance sub-menu of the pop-up menu:

- `Optimize Table`: This corresponds to the `OPTIMIZE TABLE` SQL command and should be used if you have deleted a large part of a table or if you have made many changes to a table with variable-length rows (tables that have `VARCHAR`, `BLOB`, or `TEXT` columns). Deleted records are maintained in a linked list and subsequent `INSERT` operations reuse old record positions. You can use this command to reclaim the unused space and to defragment the datafile. Note that table optimization works for MyISAM and BDB tables only. For more information see the OPTIMIZE TABLE section [http://dev.mysql.com/doc/mysql/en/OPTIMIZE_TABLE.html] of the reference manual.

- `Check Table`: This corresponds to the `CHECK TABLE` SQL command and is used to diagnose table problems. For more information see the CHECK TABLE section [http://dev.mysql.com/doc/mysql/en/CHECK_TABLE.html] of the reference manual.

- `Repair Table`: This corresponds to the `REPAIR TABLE` SQL command and should be used in case of table problems. Note that this command works for MyISAM tables only. For more information see the REPAIR TABLE section [http://dev.mysql.com/doc/mysql/en/REPAIR_TABLE.html] of the reference manual.

Clicking on a table name will display its properties in the details area when the details area is active. To display the details area, click the Details >> button.

15.2.1. Table Status

In this tab, you find detailed information about the selected table. This information could also be retrieved by issuing a `SHOW TABLE STATUS LIKE 'tbl'` SQL command. For more information see the SHOW TABLE STATUS section [http://dev.mysql.com/doc/mysql/en/SHOW_TABLE_STATUS.html] of the reference manual.

15.2.2. Row Status

In this tab, you find detailed information about the rows of the selected table. This information could also be retrieved by issuing a `SHOW TABLE STATUS LIKE 'tbl'` SQL command. For more information see the SHOW TABLE STATUS section [http://dev.mysql.com/doc/mysql/en/SHOW_TABLE_STATUS.html] of the reference manual.

15.3. Schema Indexes

The `Schema Indices` tab lists the indexes of the selected database. Besides the index names and the table the index belongs to, you can also see the index type, whether values are unique, and whether NULL values are allowed. You can also get this information by issuing a `SHOW INDEX` SQL command in a command-line client (see the `SHOW DATABASES` Syntax [http://dev.mysql.com/doc/mysql/en/SHOW_DATABASES.html] section of the reference manual). For more information about indexes, see the Column Indexes [http://dev.mysql.com/doc/mysql/en/Indexes.html] section of the reference manual.

To see the columns that form a given index, double-click on the index.

Chapter 16. The MySQL Table Editor

16.1. Introduction

The MySQL Table Editor is a component of both the MySQL Query Browser and MySQL Administrator, enabling the visual creation and modification of tables.

The MySQL Table Editor can be accessed from the MySQL Query Browser by right-clicking on a table within the database browser and choosing the Edit Table option, or by right-clicking on a database within the database browser and choosing the Create New Table option.

The MySQL Table Editor can be accessed from MySQL Administrator through the Catalogs screen. Once you have selected a database, right-click on a table and choose the Edit Table option from the drop-down menu. You can also select a table and click the Edit Table button to access the MySQL Table Editor.

16.2. The Main Editor Window

The MySQL Table Editor consists of a work space divided into three tabs, some general information prompts, and three action buttons.

Regardless of the active tab, you can always edit the table name and the table comment.

Figure 16.1. The MySQL Table Editor

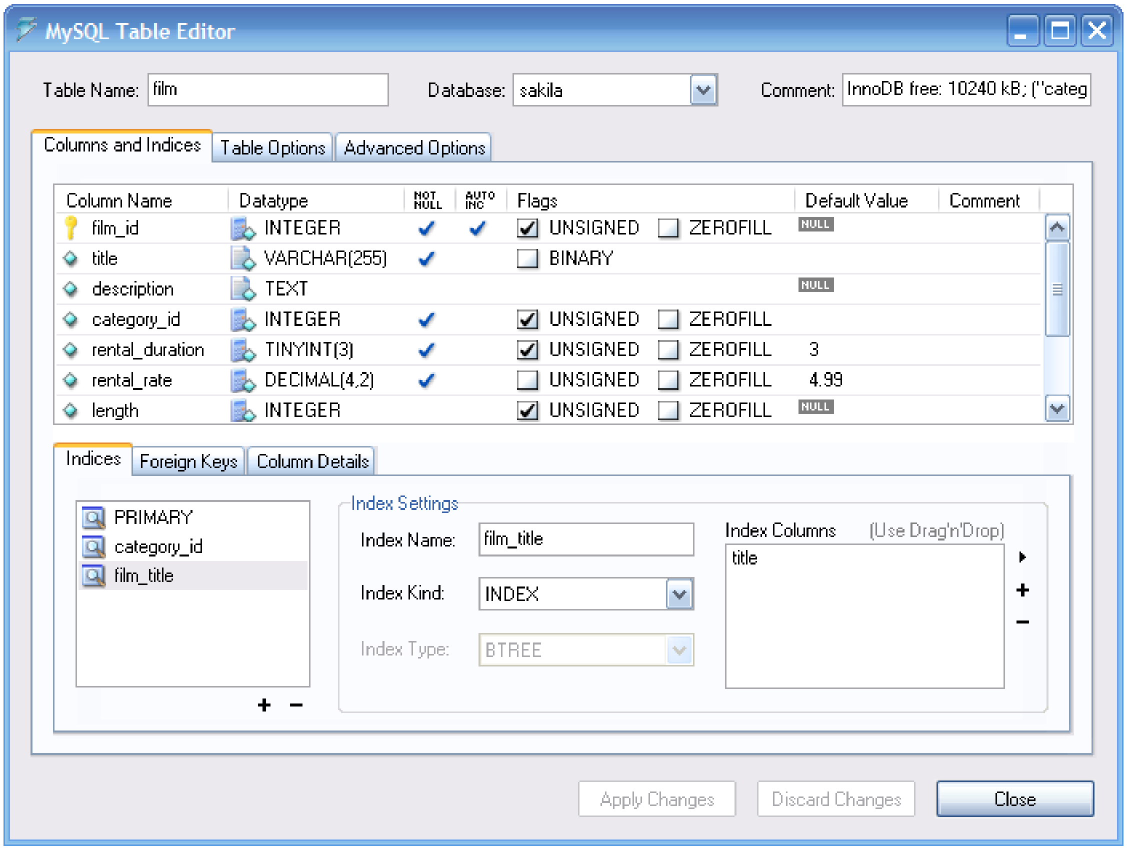


Table Name: film
Database: sakila
Comment: InnoDB free: 10240 kB; ("categ

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
film_id	INTEGER	✓	✓	☑ UNSIGNED ☐ ZEROFILL	NULL	
title	VARCHAR(255)	✓		☐ BINARY		
description	TEXT				NULL	
category_id	INTEGER	✓		☑ UNSIGNED ☐ ZEROFILL		
rental_duration	TINYINT(3)	✓		☑ UNSIGNED ☐ ZEROFILL	3	
rental_rate	DECIMAL(4,2)	✓		☐ UNSIGNED ☐ ZEROFILL	4.99	
length	INTEGER			☑ UNSIGNED ☐ ZEROFILL	NULL	

PRIMARY
category_id
film_title

Index Name: film_title
Index Kind: INDEX
Index Type: BTREE
Index Columns (Use Drag'n'Drop): title

The tabbed area is divided into three sections:

- *Columns and Indices:* Use the Columns and Indices tab to create and modify the table's column and index information. You can also create FOREIGN KEY relationships using this tab.
- *Table Options:*Use the Table Options tab to choose the storage engine and default character set used by the table.
- *Advanced Options:* Use the Advanced Options tab to configureoptions such as the per-table storage directory, MERGE and RAID table options, and table/row length options.

Each of these areas are discussed in further detail in the following sections.

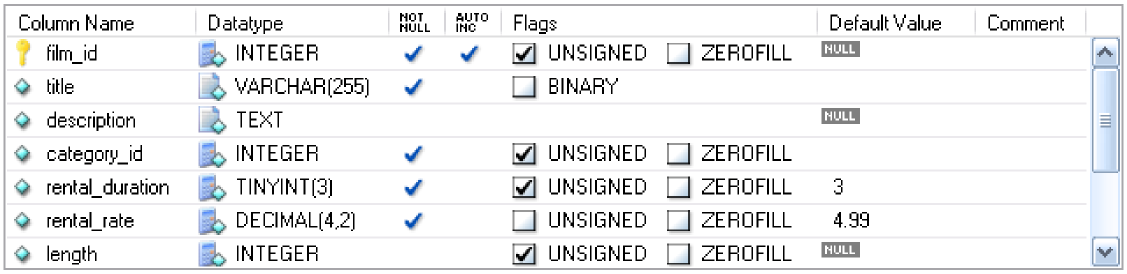
16.3. The Columns And Indices Tab

The `Columns and Indices` tab can be used to display and edit all column and index information for your table. Using this tab, you can add, drop, and alter columns and indexes.

16.3.1. The Column Editor

You can use the column editor to change the name, data type, default value, and other properties of your table's columns.

Figure 16.2. The column editor



Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
film_id	INTEGER	✓	✓	☑ UNSIGNED ☐ ZEROFILL	NULL	
title	VARCHAR(255)	✓		☐ BINARY		
description	TEXT				NULL	
category_id	INTEGER	✓		☑ UNSIGNED ☐ ZEROFILL		
rental_duration	TINYINT(3)	✓		☑ UNSIGNED ☐ ZEROFILL	3	
rental_rate	DECIMAL(4,2)	✓		☐ UNSIGNED ☐ ZEROFILL	4.99	
length	INTEGER			☑ UNSIGNED ☐ ZEROFILL	NULL	

To change the name, data type, default value, or comment of a column, double-click on the value you wish to change. The value becomes editable and you can the complete your changes by pressing the **Enter** key.

To modify the flags on a column (`UNSIGNED`, `BINARY`, `ASCII`, and so on) check and uncheck the boxes corresponding to the flag you wish to change.

To the left of the column name is an icon that indicates whether the column is a member of the primary key. If the icon is a small key, that column belongs to the primary key, otherwise the icon is a blue diamond. To add or remove a column from the primary key, you can click on the icon.

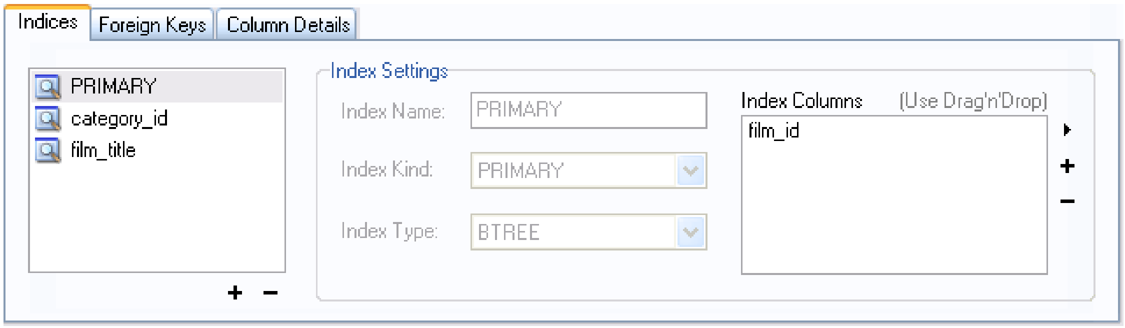
16.3.2. The Detail Area

The detail area of the `Columns and Indices` tab is divided into three tabs used for modifying the index, foreign key, and miscellaneous information regarding your columns.

16.3.2.1. The Indices Tab

The `Indices` tab holds all index information for your table. You can add, drop, and modify indexes using the indices tab, in combination with the column editor.

Figure 16.3. The index editor



PRIMARY
category_id
film_title

Index Name: PRIMARY

Index Kind: PRIMARY

Index Type: BTREE

Index Columns: film_id

To add an index, click the + (plus) button below the index list. The MySQL Query Browser prompts for an index name and the new index is created. To drop an index, select the index and click the - (minus) button.

Use the Index Name and Index Kind dialogs to modify the name and type (`UNIQUE`, `FULLTEXT`, and so on) of the index.

To add columns to an index, either click and drag the column to the Index Columns box or select the column you wish to add and click the + (plus) button to the right of the Index Columns box. You can remove a column from the index by selecting the column and clicking the - (minus) button to drop the column from the index.

To change an index to only refer to a column prefix (such as with the CHAR and VARCHAR string data types), select the index column you wish to prefix in the Index Columns box and then click the arrow icon to the right of the Index Columns box. Select the Set Index Column Length option from the drop-down menu that appears.

16.3.2.2. The Foreign Keys Tab

The `Foreign Keys` tab is divided into two sections, one with a list of foreign keys and one with various dialogs for foreign key settings.

To add a foreign key, click the + (plus) button below the foreign key list. The MySQL Query Browser prompts for a foreign key name and the new foreign key is created.

To drop a foreign key, select the foreign key and click the - (minus) button below the foreign key list.

You can modify the name of the foreign key, its `ON DELETE`, and its `ON UPDATE` actions using the dialogs provided in the Foreign Key Settings section of the tab.

To define a foreign key relationship, choose a table from the Ref. Table drop-down list. The columns that can be referenced as foreign keys are listed in the area below, and you can drag a column from the column editor to the Column section to the left of the column you wish to reference.

16.3.2.3. The Column Details Tab

The `Column Details` tab provides an interface for setting the parameters of a column without using the table interface of the column editor.

All settings that are available in the Column Editor are also available in the Column Details tab, and in addition you can also configure the column character set and column default collation from the Column Details tab.

16.4. The Table Options Tab

The Table Options tab allows you to change the storage engine and default character set of your table. The potential storage engines are listed, along with a brief summary of each storage engine's features and strengths.

To change the storage engine for your table, click on the radio button next to the desired storage engine.

To change the default character set or collation of your table, choose a new option from the drop-down list of available character sets.

16.5. The Advanced Options Tab

The Advanced Options tab is used to configure table options that would be considered outside the standard set of options that most users designate when creating and modifying tables.

The Advanced Options tab is divided into several sub-sections, each of which is described in the upcoming sections of this manual.

Descriptions of most options set using the Advanced Options tab can be found in the CREATE TABLE section [http://dev.mysql.com/doc/refman/5.0/en/create-table.html] of the MySQL Reference Manual [http://dev.mysql.com/doc/refman/5.0/en/].

16.5.1. The Various Section

The Various section of the Advanced Options tab contains options for you to set the PACK KEYS behavior, the table password, the initial AUTO_INCREMENT value, and the delayed key update behavior.

The AUTO_INCREMENT and delayed key update behaviors apply only to MyISAM tables.

16.5.2. The Row Options Section

The Row Options section can be used to configure options such as the row format, checksum use, and the row size parameters needed for large tables.

To set the row format, choose the desired row format from the drop-down list. See http://dev.mysql.com/doc/mysql/en/MyISAM_table_formats.html for more information on the different row formats that are available. This option only applies to MyISAM tables.

When you expect a table to be particularly large, use the Avg Row Length, Min Rows, and Max Rows options to enable the MySQL server to better accommodate your data. See http://dev.mysql.com/doc/mysql/en/CREATE_TABLE.html for more information on how to use these options.

16.5.3. The Storage Options Section

The Storage Options section is used to configure a custom path to the table storage and data files. This option can help improve data integrity and server performance by locating different tables on different hard-drives.

This option is only available for MyISAM tables and is not available for servers running under the Windows operating system.

16.5.4. The Merge Table Options Section

The `Merge Table` Options section is used to configure MERGE tables in MyISAM. To create a MERGE table, select MERGE as your storage engine in the Table Options Tab and then specify the tables you wish to MERGE in the Union Tables dialog.

You can also specify the action the server should take when users attempt to perform INSERT statements on the merge table. See http://dev.mysql.com/doc/mysql/en/MERGE.html for more information on MERGE tables.

16.5.5. The Table RAID Settings Section

The `Table RAID Settings` section allows you to configure RAID support for MyISAM tables. RAID allows MyISAM table data files to grow larger than the 2GB/4GB size limit imposed by some operating systems.

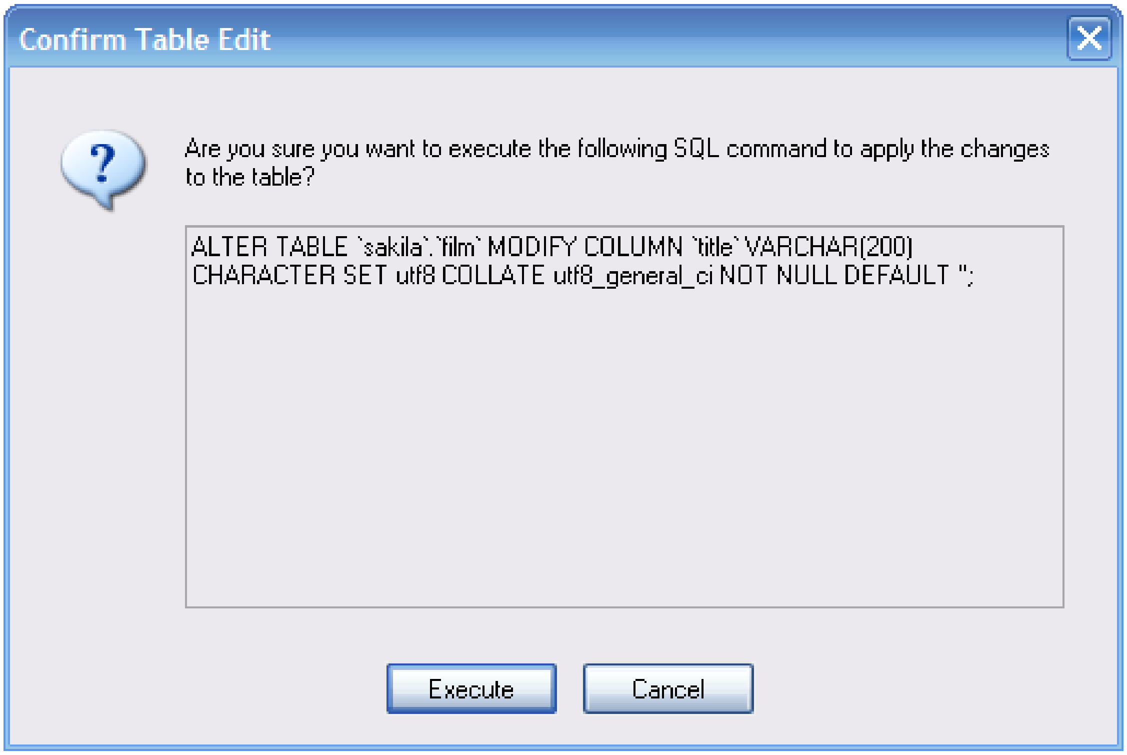
For more information on using RAID support with MyISAM, see http://dev.mysql.com/doc/mysql/en/CREATE_TABLE.html

16.6. Applying Your Changes

The changes you make with the MySQL Table Editor are not immediately applied but are instead queued to be applied in batches after you have made all your edits.

To apply the changes you have made, click the Apply Changes button. The `Confirm Table Edit` dialog will appear.

Figure 16.4. The Confirm Table Edit dialog



Are you sure you want to execute the following SQL command to apply the changes to the table?

```
ALTER TABLE `sakila`.`film` MODIFY COLUMN `title` VARCHAR(200)
CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL DEFAULT '';
```

You can click the Execute button to confirm the changes and have them applied, or click the Cancel button to cancel the changes (the table editor window is redisplayed with your changes intact). You can also click the Discard Changes button in the main MySQL Table Editor window to discard all changes you have made.

You can also copy the proposed changes to the clipboard for further editing by highlighting the ALTER TABLE or CREATE TABLE statement, right-clicking and choosing Copy from the drop-down menu.

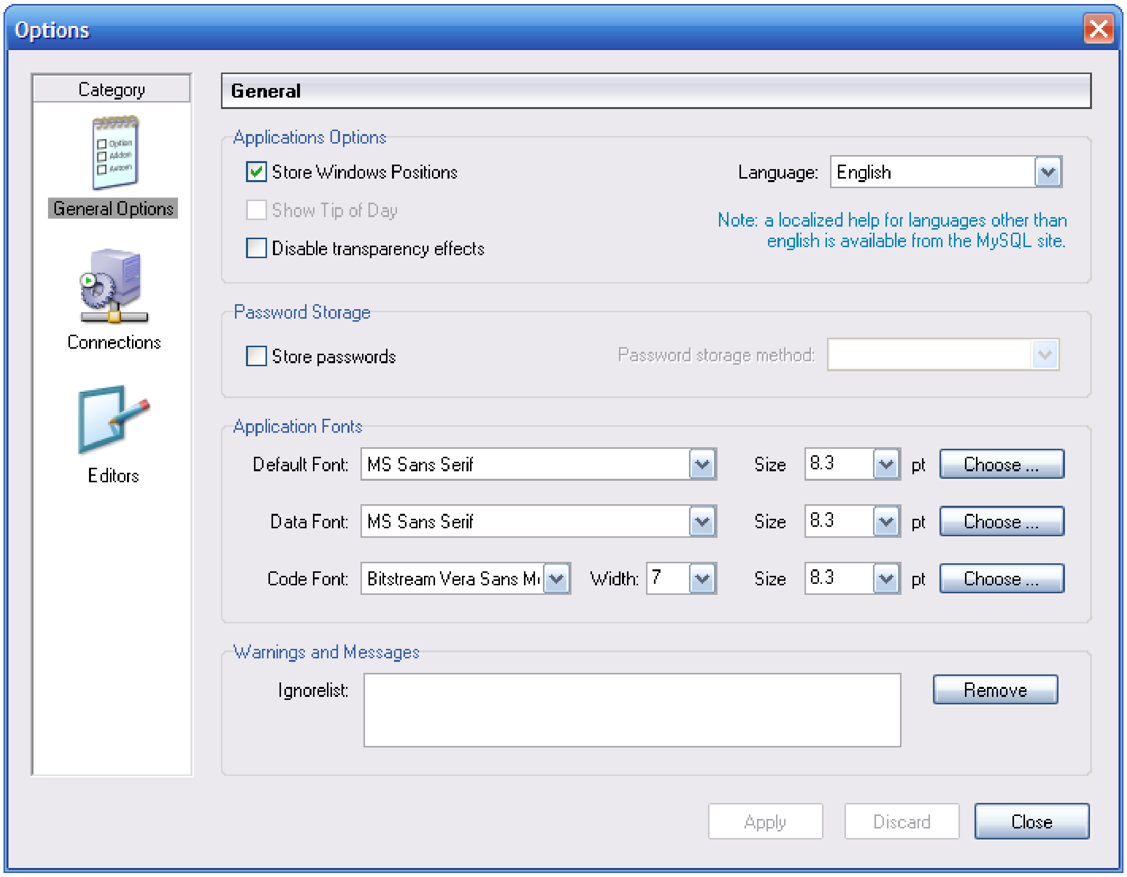
Chapter 17. Options Dialog

17.1. Introduction

The `Options` dialog allows you to configure connection profiles, general program settings, and more. You can open the `Options` dialog window using one of the following methods:

- In the connection dialog window, click the ... button.
- In the main application window select Options ... from the Tools menu.

Figure 17.1. Options dialog



Options

General

Applications Options

☑ Store Windows Positions

□ Show Tip of Day

□ Disable transparency effects

Language: English

Note: a localized help for languages other than english is available from the MySQL site.

Password Storage

□ Store passwords

Password storage method:

Application Fonts

Default Font: MS Sans Serif — Size 8.3 pt

Data Font: MS Sans Serif — Size 8.3 pt

Code Font: Bitstream Vera Sans Mı — Width: 7 — Size 8.3 pt

Warnings and Messages

Ignorelist:

In the sidebar of the dialog, you can select the section you wish to configure. There are three action buttons in the lower right corner of the window:

- Apply: Applies and saves changes.
- Discard: Discards any changes you have made.
- Close: Closes the `Options` dialog window. If you have not applied or discarded your changes, you

will be prompted to do so.

17.2. General Options

The General Options section allows you to specify a number of settings that are valid for all graphical MySQL applications.

- Store Window Positions: When an application is started the next time, its latest window position will be reused.
- Show Tip of Day: If checked, a pop-up window with the tip of the day appears at program startup.
- Store Passwords: If checked, passwords are stored in the user's connection profile. You can specify the password storage method:

  Plaintext: Passwords arestored without encryption; this can be insecure. Obscured: Passwords are encrypted using a weak algorithm. This encryption method is operating system independent. OS Specific: Use the default encryption method provided by your operating system. The default option is Plaintext.
- Language: Select the interface language. The default is English.
- Default Font: The font used for all aplication text.
- Data Font: The font used for all query and table data displayed.
- Code Font: The font used for all queries entered by the user..
- Ignorelist: Whenever you check the Do Not Show This Message Again option on error and message prompts, they are added to this list. If you would like a particular message to be shown again, click the Remove button after selecting the message from the list.

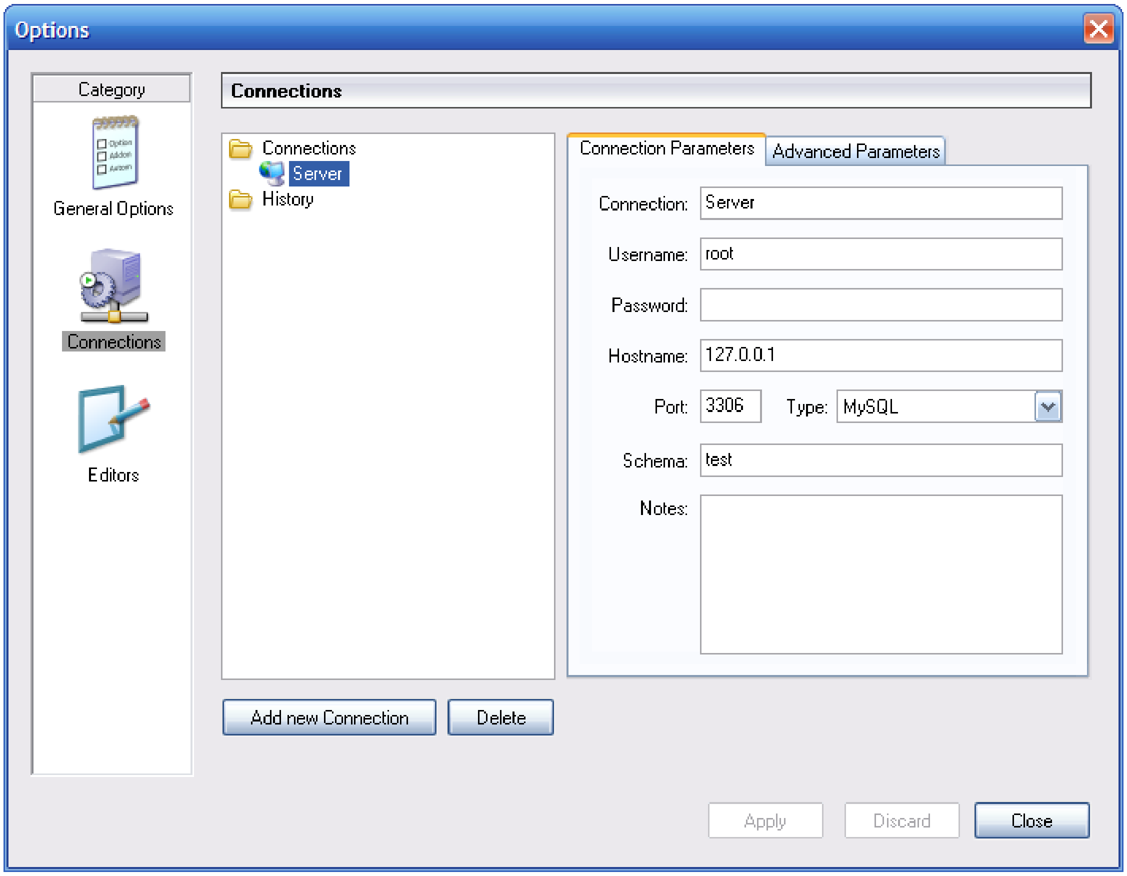
You may change the font and font size of any of the application fonts by clicking the Choose ... button to the right of the font.

17.3. Connections

The Connections section allows you to create, edit, and delete connection profiles. The center box displays a list of currently available profiles, together with a history of connections that were made without being stored in a profile. You can collapse or expand both the Connections and History trees by double-clicking them.

Connections are automatically added to the History tree whenever you establish a connection to a MySQL server without using one of the profiles stored under the Connections tree. They do not appear in the drop-down box of the Connection dialog, but you can use any of them by manually typing their name into the Connection box of the Connection dialog.

Figure 17.2. Options dialog: Connections



Connection: Server

Username: root

Password:

Hostname: 127.0.0.1

Port: 3306 Type: MySQL

Schema: test

Notes:

To edit an existing connection profile, click on its name and change the values that appear in the `Connection Parameters` and `Advanced Parameters` tabs, then click on the Apply Changes button to save your changes.

When you select a connection profile from either the `Connections` or `History` trees, the `Connection Parameters` tab displays the following fields:

- `Connection`: The connection profile label. This is the name by which you refer to the profile and that appears in the `Connection` drop-down box of the Connection dialog. It may contain any characters, including spaces. Choose distinctive names so that you can easily tell which profiles they refer to. The names can help you distinguish connections to different MySQL servers, or connections as different MySQL users to a given server.
- `Username`: The username used to connect to the MySQL server.
- `Password`: The password used to connect to the MySQL server. Note that passwords are not stored in the connection profile, unless you specify otherwise in the General Options section.
- `Hostname`: The name of the host machine where the MySQL server runs, or its IP address.
- `Port`: The TCP/IP port that the MySQL server listens to on the host machine.
- `Type`: Specifies the protocol used to connect to the database server. The default protocol is `MySQL` (which uses the native MySQL protocol).
- `Schema`: The default database for a connection when using the MySQL Query Browser.

- `Notes`: You can use this field to enter comments or additional information describing the connection profile.

> **Note**
>
> The `Advanced Parameters` tab is not available on all platforms. Advanced parameters can still be configured in the Connection dialog. Use the Details ... button to display the `Advanced Connection Options`.

When you select a connection profile from either the `Connections` or `History` list, the `Advanced Parameters` tab displays the following checkboxes:

- `Use compressed protocol`: If checked, the communication between the application and the MySQL server will be compressed, which may increase transfer rates. This corresponds to starting a MySQL command-line tool with the `--compress` option.

- `Return number of found rows, not number of affected rows`: By default, MySQL returns the number of rows changed by the last `UPDATE`, deleted by the last `DELETE` or inserted by the last `INSERT` statement. When this option is checked, the server returns the number of rows matched by the `WHERE` statement for `UPDATE` statements.

- `Ignore spaces after function names, make them reserved words`: Normally, any refereence to a function name in an SQL statement must be followed immediately by an-opening parenthesis. If this option is checked, spaces may appear between the function name and the parenthesis, like this:

  ```
  COUNT (*)
  ```

  Enabling this option has the effect that function names become reserved words. This option corresponds to starting a MySQL command-line tool with the `--ignore-spaces` option.

- `Allow interactive_timeout seconds of inactivity before disconnect`: Normally, the connection is closed by the MySQL server after a certain period of inactivity on the client side. This period can be set with the `interactive_timeout` variable. If checked, the server will not close the connection unless the period of inactivity exceeds the value set by `interactive_timeout`. This corresponds to starting a MySQL command-line tool with the `--connect-timeout=`*`seconds`* option.

- `Enable LOAD DATA LOCAL handling`: By default, the `LOCAL` option of the `LOAD DATA` statement is disabled for security reasons. Enabling this option will allow to load data from the local machine (the machine where the client GUI application is running). This option corresponds to starting a MySQL command-line tool with the `--local-infile=1` option. (Note that this option is ineffective unless the MySQL server allows `LOCAL` handling.)

17.4. Editors

The `Editors` section is used to configure options specific to the different editors available within the MySQL GUI Suite. At the time of writing this is limited to the MySQL Table Editor.

- `Show SQL command before applying changes`: Toggles whether the MySQL Table Editor will show you the `CREATE TABLE` or `ALTER TABLE` statement it is about to execute for confirmation when you click Apply Changes.

- `All columns Not Null per default`: Determines whether the MySQL Table Editor will designate columns as being `NOT NULL` by default when creating new columns.

- `All integer columns unsigned per default`: Sets whether integer columns are declared `UNSIGNED` by default when creating new columns.

- `Default storage engine`: Sets the storage engine to be assigned to newly created tables. this value is independent of the default storage engine of the MySQL server.

- `PK Naming`: Dictates the format that should be used to name `PRIMARY KEY` columns that are automatically generated. The *`%tablename%`* portion will be replaced with the name of the appropriate table.

- `Index Naming`: Sets the name automatically generated for new indexes. The *`%nr%`* string will be replaced with an automatically incrementing number.

- `FK Naming`: Configures the name used when creating new foreign keys.

- `PK Datatype`: The datatype used when creating a `PRIMARY KEY` column.

- `Def. data type` The default datatype assigned to all new columns that are not part of a `PRIMARY KEY`.

17.5. Administrator

The `Administrator` section allows you to specify options that affect the behavior of the MySQL Administrator. The `Administrator` section is divided into three sub-sections that will be discussed further in the sections that follow.

17.5.1. User Administration

These options affect the interface of the User Administration section of MySQL Administrator.

- `Show Global Privileges`: If checked, the Global Privileges tab will be visible in the User Administration section. In that tab, you can grant users global privileges (as opposed to privileges on a database, table, or column level only). This may affect security. See The MySQL Access Privilege System [http://dev.mysql.com/doc/mysql/en/Privilege_system.html] for details. If unchecked, the Global Privileges tab will not be visible in the User Administration section.

- `Show Table/Column Privileges`: In most situations, it is sufficient to grant either global or database privileges for users. You can hide the Table/Column Privileges tab in the User Administration section by enabling this checkbox.

17.5.2. Health Graphs

These options affect the appearance of the Health Graphs section of MySQL Administrator.

- `Use Peak Level Indicator`: This will display a thin yellow line on the health graphs indicating the highest value recorded by the graph.

- `Reset peak Level after a number of ticks`: If this box is not checked, the `Peak Level Indicator` will be permanently positioned at the highest value ever recorded on the health graph. With this box checked, the `Peak Value Indicator` will be reset after a period of time indicated by the `Time after the Peak level resets` option. Use this option if you would like to know the peak level over an interval rather than the peak level overall.

- `Time after the Peak level resets`: When the `Reset peak Level after a number of ticks` option is set, this determines the amount of time (in ticks) that will pass before the `Peak Level Indicator` will be reset.

17.5.3. Backup

The backup options configure how MySQL Administrator performs backup operations.

- `Add Date/Time to Backup Files`: This determines whether the date and time of the backup operation are added to the name of the backup file. See the backup section for more information.

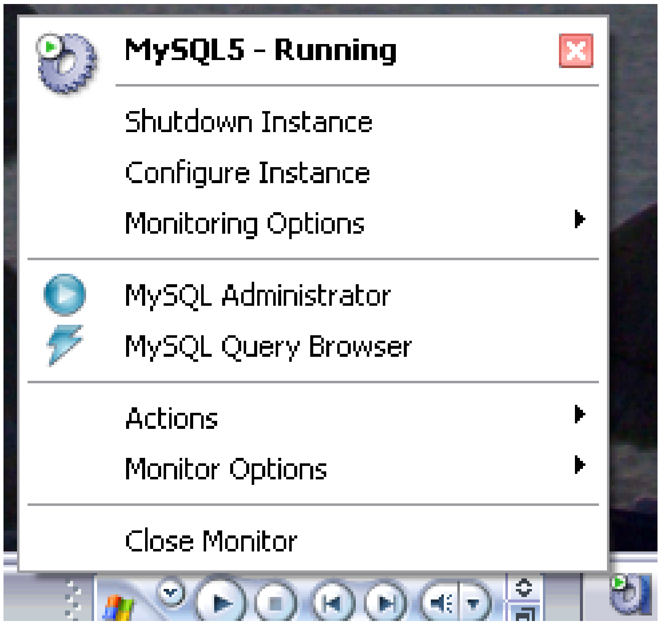
Appendix A. The MySQL System Tray Monitor

The MySQL System Tray Monitor is a tool that allows you to monitor the status of your local mysql server through an indicator that resides in the system tray. The MySQL System Tray Monitor also gives quick access to the various MySQL GUI tools through its context menu.

To start the MySQL System Tray Monitor, you can either execute the `MySQLSystemTrayMonitor.exe` application that resides in your MySQL Administrator installation directory, or choose the MySQL System Tray Monitor option of the Tools menu.

Once the MySQL System Tray Monitor is running, it resides in the system tray and provides visual status information for your MySQL server. A green icon is displayed at the top left corner of the MySQL System Tray Monitorif the MySQL server is running, and a red icon is displayed if the service is stopped. In addition, the CPU load may optionally be displayed along the right side of the MySQL System Tray Monitor.

Figure A.1. The MySQL System Tray Monitor



If you right-click on the MySQL System Tray Monitor, a pop-up menu will appear, listing all installed MySQL services and their individual statuses. For each service, the following options are available:

- `Shutdown Instance:` this will begin the shutdown operation on the selected running service.
- `Start Instance:` this will start a currently stopped server instance.
- `Configure Instance:` this will launch the MySQL Administrator in configure service mode for the currently selected instance. See Configure Service Mode for more details.
- `Monitoring Options:` currently the `Monitoring Options` submenu has one available op-

tion: `Ignore Instance State`. `Ignore Instance State` will cause the MySQL System Tray Monitor to stop monitoring the selected service.

In addition to the service controls, you can also start the MySQL Query Browser and MySQL Administrator from the pop-up menu.

The `Manage MySQL Instances` option of the `Actions` menu can be used to start MySQL Administrator in configure service mode so you can manage your server configurations. See Configure Service Mode for more details.

The `Monitor Options` menu has two options, which affect the general behavior of the MySQL System Tray Monitor. `Display CPU Load` determines whether the CPU usage is displayed on the right of the MySQL System Tray Monitor. `Launch Monitor after Login` sets the MySQL System Tray Monitorto launch when you log into Windows.

Appendix B. Installing From Source

B.1. Introduction

Normally, you will want to install the MySQL GUI tools from binaries. We put a lot of effort into making sure that our binaries are built with the best possible options. If, for whatever reason, you wish to compile the tools yourself, follow these instructions.

B.2. Downloading The Source Code

You can get the source code of the MySQL GUI tools from our public BitKeeper trees. To be able to access trees, you have to have BitKeeper installed. BitKeeper is freely available from Bitmover [http://www.bitmover.com/]. To run under Windows, BitKeeper requires Cygwin. If, during the installation process, BitKeeper finds that either Cygwin is not installed, or that the Cygwin version is too old, it will suggest to download and install Cygwin (or a newer version). You may, however, install Cygwin before installing BitKeeper. You can get Cygwin from Cygwin.com [http://www.cygwin.com/].

Our public BitKeeper trees that contain the MySQL GUI tools source code can be retrieved by the following BitKeeper commands:

```
bk clone bk://mysql.bkbits.net/mysql-administrator mysql-administrator
bk clone bk://mysql.bkbits.net/mysql-gui-common mysql-gui-common
bk clone bk://mysql.bkbits.net/mysql-query-browser mysql-query-browser
```

You will need the `mysql-gui-common` tree regardless of which GUI tools you intend to build. The `mysql-gui-common` tree contains source and graphics files used for all MySQL GUI tools.

You can also download the source for the various GUI tools from the downloads page [http://dev.mysql.com/downloads].

B.3. Building from Source under Windows

Installing from source under Windows requires multiple commercial compilers and because of this it is preferable to install from binaries. If you meet the prerequisites needed you can build the MySQL GUI tools from source under Windows.

B.3.1. Prerequisites

To compile the MySQL GUI tools from source under Windows, you need Borland Delphi 7, Microsoft Visual Studio.NET 2003, and some libraries which are available on the internet as open source software. Make sure the programs `DCC32.exe` and `DEVENV.exe /?` can be launched from the command line.

You will need the following libraries to build from source:

- glib-2.0
- libxml-2
- mysql client libraries (4.0 or newer, 5.0 prefered)
- pcre-1.4

- MS Visual C runtime libraries 7.1
- Java include files 1.4.2_04
- Lua 5.0.2

B.3.2. Compiling and Building

Make sure you have cloned the `mysql-administrator`, `mysql-gui-common`, and `mysql-query-browser` trees, and that their local directories are located in the same directory, for example in `Personal Files\MySQLGUI\`.

The source files and libraries should be placed into the following tree structure:

```
Working Directory
|
|- mysql-administrator
|- mysql-gui-common
|- mysql-query-browser
|- mysql-gui-win-res
   |
   |-include
   | |
   | |-glib-2.0
   | |-java
   | | |
   | | |-win32
   | |
   | |-libxml
   | |-lua
   | |-mysql
   | |-pcre
   | |-windows
   |
   |-lib
     |
     |-glib-2.0
     |-java
     |-libxml
     |-lua
     |-mysql
     |-pcre
     |-windows
```

B.3.2.1. Building from Command Line

To build from the command line open a windows command line prompt and change into the main repository of the tool (i.e. for MySQL Administrator this is `work\mysql-administrator`). There are three batch-files to start the build:

- build_debug.cmd
- build_release.cmd
- build_commercial.cmd

Execute one of those to generate a runtime image of the application. It will be generated in a `bin\windows` directory.

Another method is to doubleclick on one of the `build_`*`buildtype`*`.cmd` files, where *`buildtype`* is one of `debug`, `release`, and `commercial`. The `commercial` build script is for users with commercial licenses, and makes use of the commercial version of the MySQL client library.

B.3.2.2. Building from the IDE

Before you can open any projects, you have to install these components. Otherwise some of the forms will not open and you will receive error messages.

To install the components, goto the `mysql-gui-common\source\windows\` directory. Copy the following directories to your Delphi source directory (`C:\Program Files\Borland\Delphi7\Source\`).

- png
- SynEdit
- TNT
- UniCodeEditor
- VirtualTreeview

Then doubleclick each `*.dpk` file outlined below. It will open Delphi. Press compile and install to install the components. Repeat this for each `*.dpk` file.

- SynEdit\Packages\SynEdit_D7.dpk
- TNT\Packages\TntUnicodeVcl_D70.dpk
- VirtualTreeview\Packages\VirtualTreesD7D.dpk
- UniCodeEditor\Packages\UniCodeEditorD7.dpk

These components have to be installed via the Install option of the Component menu. Click on the Browse button and select the the files below. Press OK after each file.

- png\PNGImage.pas

After installing the components you can open the projects as outlined below.

To build the application from the Delphi 7 IDE simply doubleclick the `edit_xxxx.cmd`. This will launch Delphi 7 and open the file. To generate the complete runtime image select Build All Projects option from the Project menu.

Like for building from command line there are three different files.

- edit_debug.cmd
- edit_release.cmd

- edit_commercial.cmd

The runtime image will be generated in a `bin\windows` directory.

Opening the Delphi project not using the `edit_*.cmd` script will lead to incorrect project settings and problems while building the projects.

B.4. Building from Source under Linux

B.4.1. Prerequisites

To be able to build the MySQL GUI tools from source, you need the following software:

- g++ development suite
- gtk-2.x
- gtkmm-2.0 or gtkmm-2.2
- libglade-2.x
- libxml-2.6.x
- mysql client libraries (4.0 or newer, 5.0 prefered)
- pcre-1.4
- autoconf-2.54, automake-1.7.x

In addition, you will need gtkhtml-3.0.x to build the MySQL Query Browser.

B.4.2. Compiling and Building

To build any of the tools, you need to have the bk tree for `mysql-gui-common` and the tree for the desired app (i.e.: `mysql-administrator`, `mysql-query-browser`) in the same top-level directory.

First build `mysql-gui-common`:

1. cd mysql-gui-common
2. sh ./autogen.sh
3. ./configure --prefix=/opt/mysql-*`administrator`*
4. make
5. make install

The `--prefix` option can have whatever value you want, normally one would specify `/usr/local` (the default) or `/usr`, but for making precompiled binaries, something in `/opt` might be more appropriate.

After you have successfully built `mysql-gui-common`, you can build the individual tools (in this example we will build mysql-administrator):

1. cd mysql-administrator
2. sh ./autogen.sh
3. ./configure --prefix=/opt/mysql-*administrator*
4. make
5. make install

If everything went well, things should be properly built and installed.

B.4.2.1. Building Static Binaries

Building static binaries is non-trivial under Linux, due to the enormous amount of library dependencies that come from GNOME and associated libraries. Additionally:

- Many libraries do not come with their static (lib*.a) versions shipped, so we need to build them ourselves.
- Some libraries can't be linked statically, such as glibc, because of libnss*.so.
- Some libraries depend on data files and modules from the original package, esp. in case of gtk/ gnome. Problems don't just appear when the user don't have these data files installed; things may not work if they use distributions that put data files in different paths.

The current solution is to build a partially static binary, with the most common libraries left dynamically linked (`glibc` and `gtk`). `gtkmm`, `gtksourceview`, `gtkhtml` and dependencies (like gnome) are being linked statically. The ideal solution would be to build custom binaries for each major/chosen distribution, but we don't have time/resources for that at the moment.

To build these binaries:

1. Look at the `source/linux/static_make` script for all the files that should be removed/ moved out of `/usr/lib`.
2. Run the script.

That's needed to force the linker to look for the static version of each library and also to explicitly list the dependency libs that were otherwise linked to the dynamic libs.

B.4.2.2. Building RPMs

For building RPMs, a spec file is already supplied and will be made automatically after `./configure` is ran. The spec file expects a source tarball with the following structure:

```
mysql-administrator/ mysql-administrator/mysql-gui-common/* mysql-administrator/my
```

The contents of each subdir is the entire bk tree for each tool (Windows specific files and bk metadata are optional, of course). You must make a tar.gz file and put it in `/usr/src/redhat/SOURCES` (or whatever is your rpm SOURCES directory).

Then, execute: `rpmbuild -ba mysql-administrator.spec`

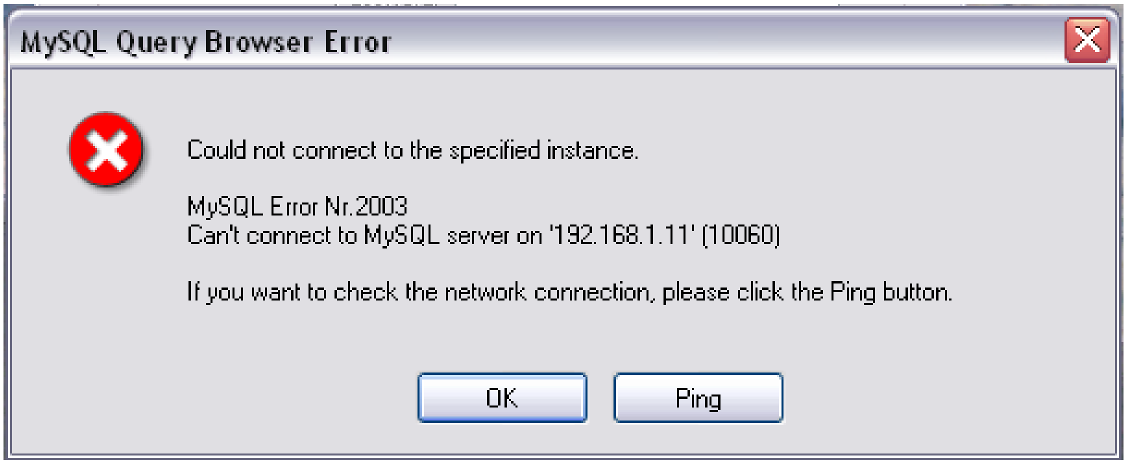
That should build the srpm and rpm files for the tool.

Appendix C. Troubleshooting Application Errors

C.1. Troubleshooting Connection Errors

If an error occurs when you select a connection profile and attempt to establish a connection to the MySQL server named in the profile, a window containing error information will be displayed (see figure).

Figure C.1. Error message dialog



MySQL Query Browser Error

Could not connect to the specified instance.

MySQL Error Nr.2003
Can't connect to MySQL server on '192.168.1.11' (10060)

If you want to check the network connection, please click the Ping button.

Normally, this means either that the credentials specified in the profile are wrong (wrong username, wrong password), or that you do not have privileges to connect to the MySQL server from the client machine. You can find more information about the error in the Causes of `Access denied` Errors [http://dev.mysql.com/doc/mysql/en/Access_denied.html] section of the reference manual.

However, another cause of connection failure is that there might be a network problem that prevents you from reaching the machine where the MySQL server is running. For this reason, the error dialog provides a PING button. Clicking this button will send an ICMP PING request to the MySQL server host machine. If the host machine is available over the network, you will see something like this:

```
Reply from 127.0.0.1: Time=0ms TTL=128
Reply from 127.0.0.1: Time=1ms TTL=128
```

This would indicate that the machine is reachable, and that the cause of connection failure is not a network problem.

C.2. Troubleshooting Display Problems

If, under Windows, the controls in the windows look somehow displaced, select Start, Control Panel, Display. Select the Settings tab and click the Advanced... button. Change your DPI settings to `Normal (96 DPI)`.

If, under Linux, messages and captions appear very big, your display resolution is probably set too high. Use `gnome-font-properties` to change your default font to a smaller value or go to `Details` and set your resolution to `72` or some other more appropriate value.

Appendix D. How Connections Are Stored

On Windows, connection profiles are stored in `C:\Documents and Settings\`*`UserName`*`\Application Data\MySQL` directory, where *`UserName`* is the name of the current Windows user. On Linux, the files are stored in the `~/.mysqlgui` directory. The settings are stored in a file called `mysqlx_user_connections.xml`. The contents of that file look like this:

```
<?xml version="1.0"?>
<user_connections>
 <last_connection>1</last_connection>
 <password_storage_type>2</password_storage_type>
 <user_connection>
  <connection_name></connection_name>
  <username>root</username>
  <hostname>localhost</hostname>
  <port>3306</port>
  <schema></schema>
  <advanced_options/>
  <storage_path></storage_path>
  <notes></notes>
  <connection_type>0</connection_type>
  <storage_type>2</storage_type>
  <password></password>
 </user_connection>
 ...
```

You can edit the file manually, but take care not to invalidate the XML. When applying changes by editing and saving the file, those changes will show up the next time you open the Connections section of the Options Dialog. You do not need to restart your application for the changes to take effect.

As a database administrator, you may also edit the file according to your preferences, and then copy it to any other machine running the MySQL GUI tools. This makes it easy to have identical connection profiles on all machines, without having to set up those profiles individually. All XML configuration files for the MySQL GUI Tools are cross-platform compatible and can be transferred between Windows and Linux machines.

Appendix E. XML Files Common to the MySQL GUI Applications

On Windows, XML files common to all MySQL GUI applications are stored in the `C:\Documents and Settings\`*`UserName`*`\Application Data\MySQL` directory, where *`UserName`* is the name of the current Windows user. On Linux, the files are stored in the `~/.mysqlgui` directory.

- `mysqlx_common_options.xml`: Stores options selected in the Options dialog.
- `mysqlx_user_connections.xml`: This file is described in How Connections are Stored.

Appendix F. XML Files Used by MySQL Administrator

Besides the documentation you are just reading (which is written in DocBook XML), MySQL Administrator uses a number of XML files for internal purposes.

- `mysqlx-options.xml`: Stores options selected in the Options dialog.
- `mysqlx-status-variables.xml`: Stores the status variables known to MySQL Administrator, so that they can be assigned descriptions in various languages (see below for the XML file containing those descriptions). Those descriptions will show up in the Status Variables tab of the Health section. You can select the language of the descriptions in the General Options section of the Options dialog. That file also determines in which categories or subcategories a variable will appear, and the order in which categories and subcategories appear. It does not, however, determine in which order variables appear within a category or subcategory, as those are sorted lexically by MySQL Administrator.
- `mysqlx-system-variables.xml`: Stores the system variables known to MySQL Administrator, so that they can be assigned descriptions in various languages (see below for the XML file containing those descriptions). Those descriptions will show up in the System Variables tab of the Health section. You can select the language of the descriptions in the General Options section of the Options dialog. That file also determines in which categories or subcategories a variable will appear, and the order in which categories and subcategories appear. It does not, however, determine in which order variables appear within a category or subcategory, as those are sorted lexically by MySQL Administrator.
- `mysqlx-translations-administrator.xml` : Stores descriptions specific to MySQL Administrator, like descriptions for status and server variables, or category names, or labels used in dialogs. Here is an excerpt from that file, which contains English and German texts:

```
     <textgroup id="MySQLVariables">
      <text id="status-var-group-General">
       <trans xml:lang="en">General</trans>
       <trans xml:lang="de">Allgemein</trans>
      </text>
      <text id="status-var-Uptime">
       <trans xml:lang="en">How many seconds the server has been
up.</trans>
      </text>
      ...
      <textgroup id="Form-InstanceForm">
       <text id="FileMI">
         <trans xml:lang="en">&File</trans>
         <trans xml:lang="de">&Datei</trans>
       </text>
       <text id="NewConnectionMI">
         <trans xml:lang="en">&New Connection
...</trans>
         <trans xml:lang="de">&Neue Verbindung
...</trans>
       </text>
       <text id="SaveConnectionMI">
         <trans xml:lang="en">&Save Connection
...</trans>
         <trans xml:lang="de">Verbindung &speichern
...</trans>
       </text>
```

```
 <text id="CloseMI">
   <trans xml:lang="en">Clos&e</trans>
   <trans xml:lang="de">Be&enden</trans>
 </text>
</textgroup>
...
```

From that excerpt, you can learn a number of things:

  - A description (called *text*) does not necessarily have to be translated (see `status-var-Uptime`). If it isn't, English will be used instead.
  - Adding another translation is as simple as adding another `<trans xml:lang="`*`language`*`">` tag.

  Therefore, translating the software is a painless task; however, please have a look at the Notes for Translators below!

- `mysqlx-translations-general.xml`: Stores terms and descriptions that appear not only in MySQL Administrator, but also in other graphical MySQL programs. Its format, however, is identical with the format of the `mysqlx-translations-administrator.xml` file. Here is an excerpt from that file:

```
<textgroup id="ResourceStrings">
 <text id="OKBtn">
   <trans xml:lang="en">OK</trans>
 </text>
 <text id="CancelBtn">
   <trans xml:lang="en">Cancel</trans>
   <trans xml:lang="de">Abbruch</trans>
 </text>
 ...
```

- `mysqlx-user-connections.xml`: This file is described in How MySQL Administrator Stores Connections.

Appendix G. Notes for Translators

You may find that the MySQL GUI tools and/or their documentation are not available in your preferred language. If you would like to translate the software, or its documentation, or participate in that translation, *please contact the MySQL documentation team before starting your translation!*

The MySQL documentation team's address is `<docs@mysql.com>`. In your mail, please state what you would like to translate (the tools, their documentation, or even the whole MySQL reference manual if you dare), and give some information regarding your background:

- What is your name?
- In which country and city are you located?
- How long have you used MySQL?
- Have you done other translations?
- Will you work alone, or with a group of translators?
- What's your timetable regarding the translation?
- What is your motivation for translating?

We will get back to you as soon as possible.

The format used for both software and documentation is *DocBook XML*. From that base format, all other available formats (HTML, CHM, PDF, etc.) are being generated. It would be a pity if you started translating, say, the HTML version of this documentation, because we will not be able to use it in that format. Also, you might find that someone else has already done (or is in the process of doing) that translation. So, once again, please contact the MySQL documentation team first!

Appendix H. Third-party Software Used by the MySQL GUI Tools

The MySQL GUI tools incorporate PCRE and PNG support through the use of third-party libraries.

H.1. PCRE Library

Regular expression support is provided by the PCRE library package, which is open source software, written by Philip Hazel, and copyright by the University of Cambridge, England. The source for the PCRE library can be found at: ftp:/ /ftp.csx.cam.ac.uk/pub/software/programming/pcre/ [ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/].

H.2. PNG Support

PNG support for the Windows version is provided by the TPNGImage component which is open source software, written and copyright by Gustavo Daud. The source for the TPNGImage can be found at: http://pngdelphi.sourceforge.net [http://pngdelphi.sourceforge.net].