

MySQL Query Browser

MySQL Query Browser

Abstract

This is the MySQL Query Browser Manual.

MySQL Query Browser has reached EOL. See the [EOL notice](#). Please upgrade to [MySQL Workbench](#).

Document generated on: 2010-02-04 (revision: 18947)

Copyright 2004-2008 MySQL AB, 2008-2010 Sun Microsystems, Inc.

This documentation is NOT distributed under a GPL license. Use of this documentation is subject to the following terms: You may create a printed copy of this documentation solely for your own personal use. Conversion to other formats is allowed as long as the actual content is not altered or edited in any way. You shall not publish or distribute this documentation in any form or on any media, except if you distribute the documentation in a manner similar to how Sun disseminates it (that is, electronically for download on a Web site with the software) or on a CD-ROM or similar medium, provided however that the documentation is disseminated together with the software on the same medium. Any other use, such as any dissemination of printed copies or use of this documentation, in whole or in part, in another publication, requires the prior written consent from an authorized representative of Sun Microsystems, Inc. Sun Microsystems, Inc. and MySQL AB reserve any and all rights to this documentation not expressly granted above.

For more information on the terms of this license, for details on how the MySQL documentation is built and produced, or if you are interested in doing a translation, please contact the <http://www.mysql.com/company/contact/>.

If you want help with using MySQL, please visit either the [MySQL Forums](#) or [MySQL Mailing Lists](#) where you can discuss your issues with other MySQL users.

For additional documentation on MySQL products, including translations of the documentation into other languages, and downloadable versions in variety of formats, including HTML, CHM, and PDF formats, see [MySQL Documentation Library](#).

Table of Contents

1. MySQL Enterprise	1
2. Introduction to the MySQL Query Browser	2
3. Installation	3
3.1. Introduction	3
3.2. Installing Under Windows	3
3.2.1. Using the Windows Installer	3
3.2.2. Installing From the ZIP File	3
3.3. Installing Under Linux	3
3.3.1. Installing the Generic Tarball	4
3.3.2. RPM Installation	4
3.4. Installing Under Mac OS X	5
4. Removing MySQL GUI Tools	6
4.1. Uninstall – Windows	6
4.2. Uninstall – Linux	6
4.3. Uninstall – Mac OS X	6
5. Upgrading MySQL GUI Tools	7
5.1. Upgrading – Windows	7
5.2. Upgrading – Linux	7
5.2.1. Tarball Upgrade	7
5.2.2. RPM Upgrade	7
5.3. Upgrading – Mac OS X	8
6. Running MySQL GUI Tools	9
6.1. Running Under Windows	9
6.2. Running Under Linux	9
6.3. Running On Mac OS X	9
7. The MySQL Query Browser Connection Dialog	11
7.1. Connection Dialog	11
8. A Tour of the MySQL Query Browser	14
8.1. The Main Query Window	14
8.2. The Query Toolbar	15
8.2.1. The Navigation Buttons	15
8.2.2. The Query Area	15
8.2.3. The Action Buttons	16
8.3. The Advanced Toolbar	16
8.4. The Result Area	17
8.5. The Object Browser	17
8.5.1. The Database Browser	17
8.5.2. The Bookmark Browser	19
8.5.3. The History Browser	19
8.6. The Information Browser	19
8.6.1. The Syntax Browser	19
8.6.2. The Function browser	19
8.6.3. The Parameter Browser	20
8.6.4. The Transaction Browser	20
8.7. The Script Editor	20
8.7.1. The Script Debugging Buttons	22
8.7.2. The Script Area	23
9. Using The MySQL Query Browser	24
9.1. Entering and Editing Queries Manually	24
9.2. Building Queries	24
9.3. Creating Views	25
9.4. Navigating Result Sets Within the Result Area	25
9.5. Exporting Result Sets	26
9.6. Handling BLOB and TEXT Columns	26
9.7. Editing Result Sets Within the Result Area	26
9.8. Comparing Result Sets	27
9.9. Managing Stored Procedures and Functions with MySQL Query Browser	27
9.10. Editing Queries From a Development Tool	28
9.11. Using the Regex Text Importer Tool	28

10. The MySQL Table Editor	29
10.1. Introduction	29
10.2. The Main Editor Window	29
10.3. The Columns And Indices Tab	30
10.3.1. The Column Editor	30
10.3.2. The Detail Area	31
10.4. The Table Options Tab	32
10.5. The Advanced Options Tab	32
10.5.1. The Various Section	33
10.5.2. The Row Options Section	33
10.5.3. The Storage Options Section	33
10.5.4. The Merge Table Options Section	33
10.5.5. The Table RAID Settings Section	33
10.6. Applying Your Changes	33
11. Options Dialog	35
11.1. Introduction	35
11.2. General Options	36
11.3. Connections	36
11.3.1. SSL Connections	38
11.4. Editors	39
11.5. The Browser Section	39
11.5.1. Display Options	40
11.5.2. Query Options	40
11.5.3. Various	41
A. Installing From Source	42
A.1. Introduction	42
A.2. Downloading The Source Code	42
A.3. Building from Source under Windows	42
A.3.1. Prerequisites	42
A.3.2. Compiling and Building	42
A.4. Building from Source under Linux	44
A.4.1. Prerequisites	44
A.5. Compiling and Building	45
A.5.1. Building Static Binaries	45
A.5.2. Building RPMs	46
B. Troubleshooting Application Errors	47
B.1. Troubleshooting Connection Errors	47
B.2. Troubleshooting Display Problems	47
C. How Connections Are Stored	48
D. XML Files Common to the MySQL GUI Applications	49
E. XML Files Used by MySQL Query Browser	50
F. Notes for Translators	51
G. Third-party Software Used by the MySQL GUI Tools	52
G.1. PCRE Library	52
G.2. PNG Support	52

List of Figures

7.1. Connection Dialog	11
8.1. The MySQL Query Browser Main Query Window	14
8.2. The Query Area Expanded to Ten Lines	15
8.3. The MySQL Query Browser Status Icons	16
8.4. The Advanced Toolbar	16
8.5. The Database Browser	17
8.6. The Script Editor	20
8.7. The File Open Dialog	21
9.1. The Table Tool	24
9.2. The BLOB Handling Icons	26
9.3. Stored Procedure in Database Browser	27
10.1. The MySQL Table Editor	29
10.2. The Column Editor	30
10.3. The Index Editor	31
10.4. The Confirm Table Edit Dialog	33
11.1. Options Dialog	35
11.2. Options Dialog: Connections	36
11.3. Options Dialog: Advanced Parameters	38
11.4. The Query Browser Options Dialog	40
B.1. Error Message Dialog	47

List of Tables

A.1. Required Components 44

A.2. Requirements for MySQL Query Browser 45

Chapter 1. MySQL Enterprise

A MySQL Enterprise subscription is the most comprehensive offering of MySQL database software, services and support; it ensures that your business achieves the highest levels of reliability, security, and uptime.

An Enterprise Subscription includes:

1. The MySQL Enterprise Server – the most reliable, secure, and up-to-date version of the world’s most popular open source database
2. The MySQL Enterprise Monitor – An automated virtual DBA assistant that monitors all your MySQL Servers around-the-clock, identifies exceptions to MySQL best practices, and provides expert advice on fixing any problems discovered
3. MySQL Production Support – Technical and consultative support when you need it, along with regularly scheduled service packs, hot-fixes, and more

For more information, visit <http://www.mysql.com/enterprise>.

Chapter 2. Introduction to the MySQL Query Browser

The MySQL Query Browser is a graphical tool provided by MySQL AB for creating, executing, and optimizing queries in a graphical environment. Where the <http://www.mysql.com/products/administrator/> is designed to administer a MySQL server, the MySQL Query Browser is designed to help you query and analyze data stored within your MySQL database.

While all queries executed in the MySQL Query Browser can also be run from the command-line using the `mysql` utility, the MySQL Query Browser allows for the querying and editing of data in a more intuitive, graphical manner.

MySQL Query Browser is designed to work with MySQL versions 4.0 and higher.

MySQL Query Browser is, to a large extent, the result of feedback 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](#) to request features or report problems.

For MySQL Query Browser, there's a dedicated forum available on <http://forums.mysql.com/list.php?108>.

Chapter 3. Installation

3.1. Introduction

MySQL GUI Tools run on Windows, Linux, and Mac OS X. You can find MySQL GUI Tools for the operating system of your choice on the [MySQL GUI Tools Downloads](#) page.

The MySQL Migration Toolkit requires the Java Runtime Environment (JRE). The minimum version supported is 5.0 update 8.

3.2. Installing Under Windows

MySQL GUI Tools run on recent 32-bit Windows NT based operating systems, including Windows 2000, XP, Vista, and 2003. They don't run on Windows NT 4 and below.

3.2.1. Using the Windows Installer

MySQL GUI Tools can be installed on all Windows operating systems using the Windows Installer ([.msi](#)) installation package. The MSI package is contained within a ZIP archive named `mysql-gui-tools-version-win32.msi`, where *version* indicates the MySQL GUI Tools version.

The Microsoft Windows Installer Engine was updated with the release of Windows XP; those using a previous version of Windows can reference [this Microsoft Knowledge Base article](#) for information on upgrading to the latest version of the Windows Installer Engine.

In addition, Microsoft has recently introduced the WiX (Windows Installer XML) toolkit. This is the first highly acknowledged Open Source project from Microsoft. We have switched to WiX because it is an Open Source project and it allows us to handle the complete Windows installation process in a flexible manner using scripts.

Improving the MySQL Installation Wizard depends on the support and feedback of users like you. If you find that the MySQL Installation Wizard is lacking some feature important to you, or if you discover a bug, please report it in our bugs database using the instructions given in [How to Report Bugs or Problems](#).

To install MySQL GUI Tools, right click on the MSI file and select [INSTALL](#).

Note

Installing MySQL GUI Tools on Windows Vista requires administrative privileges.

At the [Setup Type](#) window you may choose a [complete](#) or [custom](#) installation. If you do not wish to install all of the MySQL GUI Tools choose the custom option. Custom installation also gives you the option of installing support for languages other than English. MySQL GUI Tools supports, German, Greek, Japanese, Polish, and Brazilian Portuguese.

Unless you choose otherwise, MySQL GUI Tools are installed in `C:\%PROGRAMFILES%\MySQL\MySQL Tools for version\`, where `%PROGRAMFILES%` is the default directory for programs on your machine and *version* is the version number of MySQL GUI Tools. The `%PROGRAMFILES%` directory might be `C:\Program Files` or `C:\programme`.

Note

Installing MySQL GUI Tools using the Windows installer automatically creates entries in the [Start](#) menu.

MySQL Workbench is not included in the bundled MySQL GUI Tools download so must be installed separately. To do this, find the link to MySQL Workbench on <http://dev.mysql.com/downloads/>. After downloading install this file as described in [Section 3.2, "Installing Under Windows"](#).

3.2.2. Installing From the ZIP File

If you are having problems running the installer, as an alternative, you can download a ZIP file without an installer. That file is called `mysql-gui-tools-noinstall-version-win32.zip`. Using a ZIP program, unpack it to the directory of your choice. You may also want to create shortcuts to `MySQLAdministrator.exe`, `MySQLMigrationTool.exe`, and `MySQLQuery-Browser.exe` for your desktop or the quick launch bar.

3.3. Installing Under Linux

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

Note

Currently, the MySQL Migration Toolkit is not available for Linux.

3.3.1. Installing the Generic Tarball

The generic tar archive allows you to install MySQL GUI Tools on most Linux distributions. The tarball file is called `mysql-gui-tools-version.tar.gz`, where `version` indicates the MySQL GUI Tools version (for example, 5.0r3).

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

```
shell> tar -tzf mysql-gui-tools-version.tar.gz
```

To install MySQL GUI Tools, run this command:

```
shell> tar --directory=/opt -xzf mysql-gui-tools-version.tar.gz
```

This installs the various application binaries in the directory, `/opt/mysql-gui-tools-version`.

If you install MySQL GUI Tools to the `opt` directory, icons for use with desktop shortcuts or for creating menu items are found under the `opt/mysql-gui-tools-version/share/mysql-gui` directory.

If you install MySQL GUI Tools to a directory other than the `/opt` directory, you will need to update the installation directory. This is done by using the `--update-paths` option the first time that you run any one of the MySQL GUI Tools. For example, when running Query Browser for the first time, navigate to the installation directory and enter the following command:

```
shell> ./mysql-query-browser --update-paths
```

You need only run one MySQL GUI Tools application with the `--update-path` option. This updates the path for all MySQL GUI Tools.

If you do not use the default installation directory and create a Desktop shortcut using the `.desktop` files found under the `mysql-gui-tools-5.0` directory you must edit the properties of any shortcut you create. Enter the correct path for the application on your system.

3.3.2. RPM Installation

In addition to a generic tarball, some distribution-specific RPMs are available. Currently these include Red Hat Enterprise Linux (RHEL) 3 and 4, Fedora Core 5 (FC5), and SuSE Linux 10.x. For FC5 and SuSE the `gtkmm24` toolkit is a requirement for installing the RPM version of MySQL GUI Tools. You may also need to install the widget for displaying HTML pages. On FC5 you may install these RPMs from the command line in the following way:

```
shell> yum install gtkmm24 gtkhtml2
```

For installation on SuSE Linux:

```
shell> yast2 -i gtkmm24 gtkhtml2
```

Note

You may need root privileges to run the `yast2` command.

The Red Hat RPMs are self contained so no additional packages need to be installed.

The RPM downloads are made up of the individual GUI Tools components combined into a single TAR archive. Extract the individual RPMs in the following way:

```
shell> tar -zxf mysql-gui-tools-version.tar.gz
```

This will decompress the RPM files to the current directory.

Install all the RPM files by typing:

```
shell> rpm -ivh mysql-*.rpm
```

If you are upgrading to a newer version of MySQL GUI Tools see [Section 5.2.2, “RPM Upgrade”](#).

If you install the RPM files individually, you must install the `mysql-gui-tools-version.rpm` file first in order to satisfy dependencies.

If you wish, you may install only one of the MySQL GUI Tools. For example, to install MySQL Administrator only, do the following:

```
shell> rpm -ivh mysql-gui-toolsversion.rpm mysql-administratorversion.rpm
```

If possible, the RPM installation process creates shortcuts in the start menu of your window manager. For example, SuSE Linux with the KDE window manager adds shortcuts to the MySQL GUI Tools under the [DEVELOPMENT](#), [OTHER TOOLS](#) menu item. Likewise, with FC5, shortcuts are created under the [DEVELOPMENT](#) menu item.

Icons for use with desktop shortcuts or for creating menu items are found under the `/usr/share/mysql-gui` directory.

See the [MySQL GUI Tools Downloads](#) page for the most up-to-date listing of the various RPM packages available.

3.4. Installing Under Mac OS X

To install MySQL GUI Tools under Mac OS X, double-click the downloaded `.dmg` file and wait for it to be opened and attached. Once a window containing the MySQL GUI Tools 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.

The minimum supported version is Mac OS X 10.4.

Note

Currently, the MySQL Migration Toolkit is not available for Mac OS X.

Chapter 4. Removing MySQL GUI Tools

Under Linux and Mac OS X it is easy to remove all the MySQL GUI Tools or individual applications. Under Windows, removing individual applications is a bit more problematic.

4.1. Uninstall – Windows

To uninstall MySQL GUI Tools, open the [Control Panel](#) and Choose [Add or Remove Programs](#). Find the [MySQL Tools](#) entry and choose the REMOVE button. Choosing this option will remove all of the GUI Tools.

Note

Currently, there is no command-line option for removing MySQL GUI Tools.

After you have removed MySQL GUI Tools you may remove the [MySQL Tools for version](#) directory. Unless you chose otherwise on installation, you should find this directory below the `C:\%PROGRAMFILES%\MySQL\` directory.

You may remove individual applications manually. Go to the `C:\%PROGRAMFILES%\MySQL\` directory and delete the `exe` and `chm` files associated with the application you wish to remove. Also remove the appropriate icon from the [Start](#) menu.

Note

Removing an application manually will not remove all the files belonging to that application.

4.2. Uninstall – Linux

Unless you chose otherwise, you should find MySQL GUI Tools in the `/opt/mysql-gui-tools-version` directory.

If you installed MySQL GUI Tools using the RPM files you can remove all the tools by typing at the command line:

```
shell> rpm -e mysql-*.rpm
```

You may remove individual tools by using the `e` option with the name of the specific tool you wish to remove. For instance, to remove only the Administrator tool type:

```
shell> rpm -e mysql-administrator-version.rpm
```

Note

Remember, if you plan to keep any one of the MySQL GUI Tools you must also keep the [mysql-gui-tools-version.rpm](#) file.

After removal of all the MySQL GUI Tools, remove the `/opt/mysql-gui-tools-version` directory. if you have removed only selected tools, remove only the associated directories.

If you installed MySQL GUI Tools using the generic tarball, you can remove the tools by deleting the `/opt/mysql-gui-tools-version` directory. If you wish to remove an individual tool, find the directory associated with that tool and delete it.

4.3. Uninstall – Mac OS X

Find the directory where you installed MySQL GUI Tools, and remove applications by moving their icons to the [Trash](#).

Chapter 5. Upgrading MySQL GUI Tools

5.1. Upgrading – Windows

If you are upgrading using the installer file follow the instructions given in [Section 3.2, “Installing Under Windows”](#). There is no need to remove your current installation.

If you are not using the installer file remove the current MySQL GUI Tools directory, and extract and install the new version as described in [Section 3.2, “Installing Under Windows”](#).

MySQL Workbench is not included in the bundled MySQL GUI Tools download so must be installed separately. To do this, find the link to MySQL Workbench at <http://dev.mysql.com/downloads/>. After downloading, install this file as described in [Section 3.2, “Installing Under Windows”](#).

5.2. Upgrading – Linux

5.2.1. Tarball Upgrade

If you are upgrading using the generic tarball file, remove the current MySQL GUI Tools directory, and extract and install the new tarball as described in [Section 3.3.1, “Installing the Generic Tarball”](#).

5.2.2. RPM Upgrade

If you installed the MySQL GUI Tools using RPM files you can upgrade by navigating to the directory that contains the RPM files and typing at the command line:

```
shell> rpm -Uvh mysql-*.rpm
```

You may upgrade individual tools by using the `U` with the name of the specific tool you wish to upgrade. You will also need to upgrade the `mysql-gui-tools-version.rpm` file. For instance, to upgrade only the Administrator tool type:

```
shell> rpm -Uvh mysql-administrator-version.rpm mysql-gui-tools-version.rpm
```

Note

The `mysql-gui-tools-version.rpm` file is used by all MySQL GUI Tools. You must always upgrade this file.

When upgrading, packages cannot be installed separately because version conflicts will arise.

If your previous installation of MySQL GUI Tools included Workbench, (this tool has been removed from the GUI Tools package) you may encounter the following error when upgrading:

```
error: Failed dependencies:
mysql-gui-tools = old-version is needed by (installed)
mysql-workbench-version
```

To continue with the upgrade you must remove the Workbench RPM file. To determine the name of the Workbench RPM file issue the following command:

```
shell> rpm -qa | grep workbench
```

Remove Workbench by issuing the command:

```
shell> rpm -e mysql-workbench-version
```

You should now be able to upgrade MySQL GUI Tools as described above.

If you do not wish to remove Workbench, perform an RPM installation rather than an upgrade. For instructions on doing this see [Section 3.3.2, “RPM Installation”](#).

5.3. Upgrading – Mac OS X

Find the directory where you installed MySQL GUI Tools, and remove the applications by moving their icons to the [Trash](#).

Install the upgrade as described in [Section 3.4, “Installing Under Mac OS X”](#).

Chapter 6. Running MySQL GUI Tools

How you start any one of the MySQL GUI Tools depends on the operating system you are using.

6.1. Running Under Windows

Under Windows, the names of the executable files in the MySQL GUI Tools suite are:

- `MySQLAdministrator.exe`
- `MySQLQueryBrowser.exe`
- `MySQLMigrationTool.exe`

Start any one of the MySQL GUI Tools by double clicking its desktop icon, or by selecting it from the [Start](#) menu. Alternatively, you can open a DOS window and start it from the command line. For example, you could run the MySQL Administrator in the following way:

```
C:\> "C:\%PROGRAMFILES%\MySQL\MySQL Tools for version\MySQLAdministrator.exe"
```

`%PROGRAMFILES%` is the default directory for programs on your machine, for example `C:\Program Files` or `C:\programme`. If your path contains spaces, you must enclose the command within quotation marks as shown above.

6.2. Running Under Linux

Under Linux, the names of the executable files in the MySQL GUI Tools suite are:

- `mysql-administrator`
- `mysql-query-browser`

Note

There is no Linux version of MySQL Migration Toolkit.

If you installed MySQL GUI Tools to the `/opt` directory using the tar archive file, change into the `/opt/mysql-gui-tools-version` directory to run any one of the MySQL GUI Tools.

To run MySQL Administrator you would type:

```
shell> ./mysql-administrator
```

However, adding `/opt/mysql-gui-tools-version` to the `PATH` variable makes it much simpler to run MySQL GUI Tools — you need not worry about your present working directory.

When installed from RPM files, the MySQL GUI Tools are found in the `/usr/bin` directory. This directory is usually included in the `PATH` variable, so running any one of the MySQL GUI Tools simply requires typing the executable file name, regardless of your current directory. For example:

```
shell> mysql-administrator
```

For those distributions that create menu items, you may also, of course, start any one of the tools by choosing the menu item.

6.3. Running On Mac OS X

On Mac OS X, navigate to the MySQL GUI Tools installation directory and double-click on the application you wish to start.

Note

There is no Mac OS X version of MySQL Migration Toolkit.

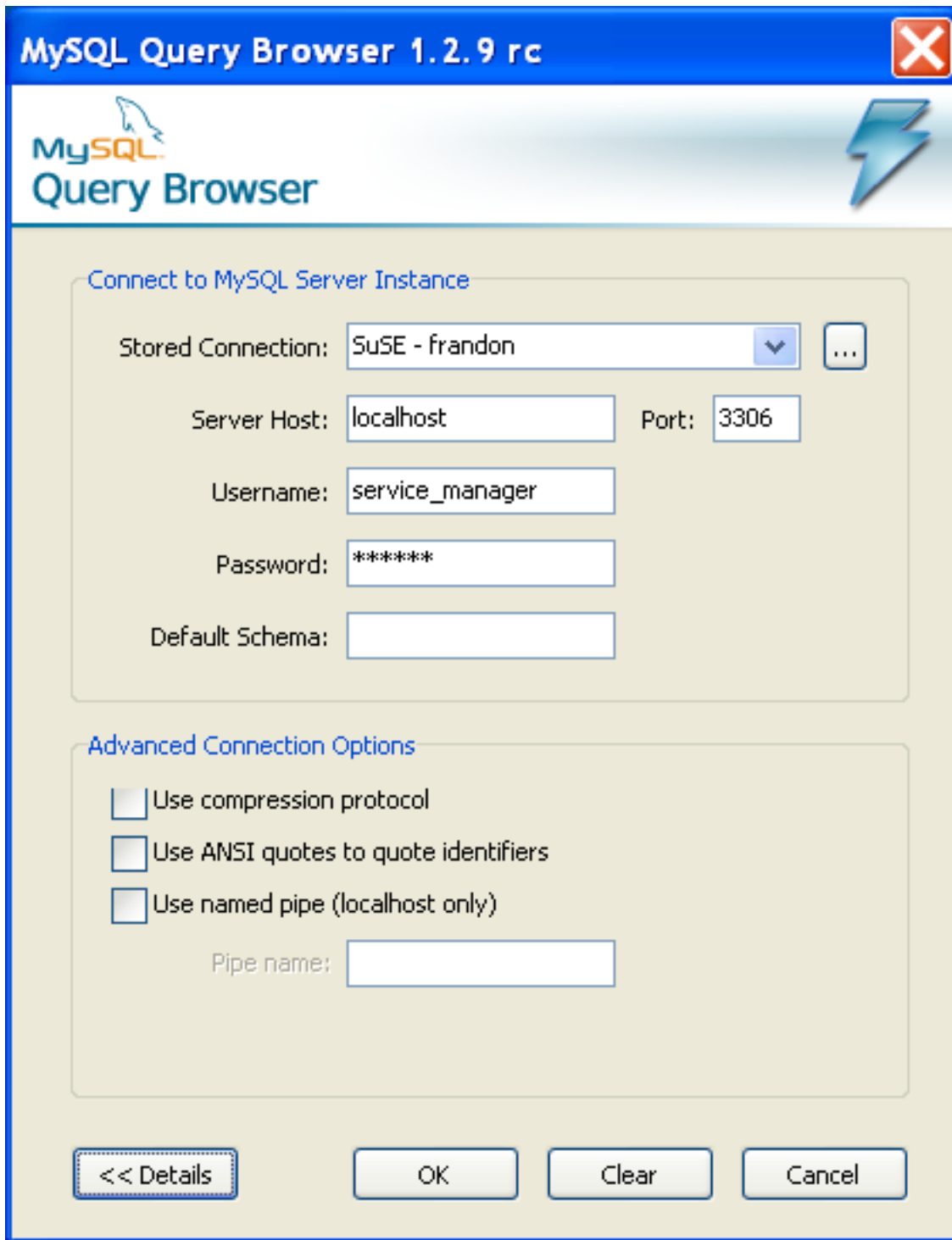
Chapter 7. The MySQL Query Browser Connection Dialog

7.1. Connection Dialog

Once MySQL Query Browser has been started, it displays a connection dialog window. You need to specify the MySQL server you are connecting to, the credentials needed for authorization on that server, which machine hosts that server (and which port it listens to), and the database (Schema) you will be querying. You may also specify a number of other options, if required.

You must choose a default database in order to issue queries. It is possible to choose a default database after connecting to the server, but setting the default from the connection dialog can save time on subsequent connections.

Figure 7.1. Connection Dialog



MySQL Query Browser 1.2.9 rc

MySQL Query Browser

Connect to MySQL Server Instance

Stored Connection: SuSE - frandon

Server Host: localhost Port: 3306

Username: service_manager

Password: *****

Default Schema:

Advanced Connection Options

☐ Use compression protocol

☐ Use ANSI quotes to quote identifiers

☐ Use named pipe (localhost only)

Pipe name:

<< Details OK Clear Cancel

If the server connection is successfully established, all the values are saved for future connections (see the section that describes [how MySQL Query Browser stores connection information](#)). The **Password** field, however, is always empty: For security reasons, the password is not stored along with the other options, unless you explicitly specify otherwise in the **General Options** section of the **Options dialog**.

The nature of the connection to the MySQL server varies depending upon the value specified in the **SERVER HOST**: text box.

Under Linux and Mac OS X the connection to the MySQL server will be a socket connection if “localhost” is used in the **SERVER**

OST: text box. Specifying `127.0.0.1`, the actual IP address that hosts the MySQL server, or the host name results in a TCP connection.

Under Windows, the connection is always a TCP connection unless you connect using named pipes. This is discussed in what follows.

To set advanced connection options, click the **DETAILS** button and check the desired options:

- **Use compression protocol** - Use compression when communicating with the MySQL server.
- **Use ANSI quotes to quote identifiers** - enclose all object names in double-quotes instead of back-ticks.
- **Windows Only:** **Use named pipe (localhost only)** is equivalent to starting the client from the command line specifying the `--protocol=pipe` option.
- **Unix Only:** **Connect using socket file** is equivalent to starting the client from the command line specifying the `--protocol=socket` option.

If you have difficulties connecting, please see the [Troubleshooting Connection Errors](#) section in the [Troubleshooting Application Errors](#) appendix.

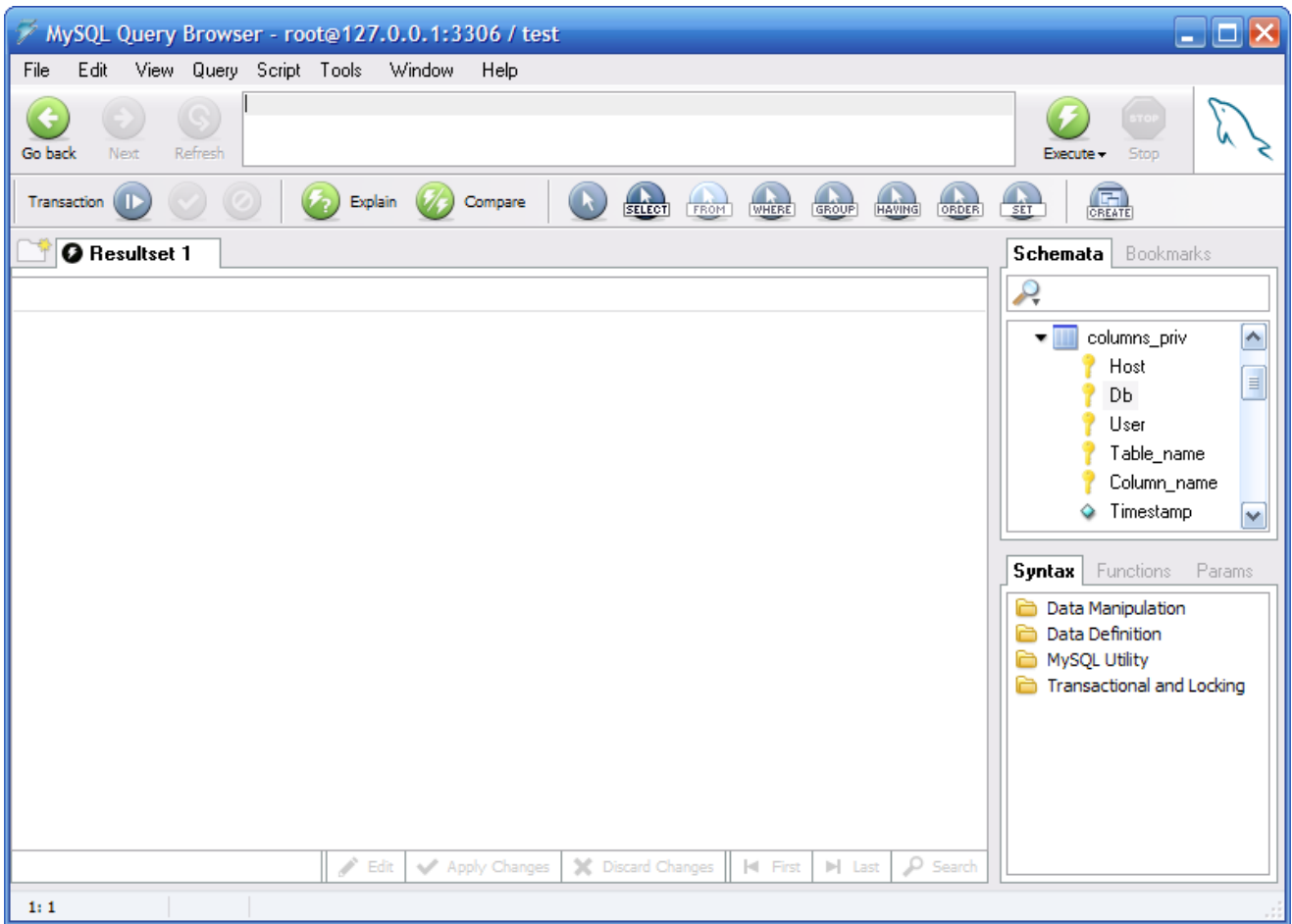
You can change any of the values in the connection profiles just by overwriting the existing values with new ones. Similarly, if you select another connection profile, you can change any of its values. When you click the OK button after changing a profile's values, the changes are stored permanently only if a successful connection to the MySQL server is established. You can also click the ... button next to the **CONNECTION** drop-down box. This brings up an **Options** dialog window that has a **Connections** section for modifying connection profiles. The values you can set are found in the Connection dialog window in the description of the **Connections** section of the [Options dialog](#).

Chapter 8. A Tour of the MySQL Query Browser

8.1. The Main Query Window

Once you successfully connect to a MySQL server you are presented with the main query window of the MySQL Query Browser. All of the application's functionality is available through this window.

Figure 8.1. The MySQL Query Browser Main Query Window



The main query window is divided up into several sections:

- *Query Toolbar:* The query toolbar is where you create and execute your queries. It is composed of three navigation buttons (GO BACK, NEXT, REFRESH), the query area, two action buttons (EXECUTE and STOP), and a status indicator.
- *Advanced Toolbar:* The advanced toolbar contains three sets of buttons: the Transaction Buttons (START, COMMIT, ROLLBACK), the query management buttons (EXPLAIN, COMPARE), and the query building buttons (SELECT, FROM, WHERE, and so on.)

The Advanced Toolbar is only visible when the [Show advanced toolbars](#) and [Show composer buttons toolbar](#) options in the Browser Options screen are checked. See [Section 11.5, “The Browser Section”](#).

- *Result Area:* All query results are displayed in the result area. You can have multiple tabs active at one time, allowing you to work on multiple queries. The result area can be split vertically and horizontally for performing comparisons, and queries in different parts of a split result area can be joined together for master-detail analysis.

- *Object Browser:* The object browser is part of the sidebar and allows you to manage your databases, bookmarks, and history. You can choose which database and tables to query, add commonly used queries to a collection of bookmarks, and browse through previously issued queries in order to use them again.

If there are many schemata on your MySQL server, the search 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.

You may also filter schemata using the wild card `?` to stand in for any single character and `*` for zero to any number of characters. For example the string `*boo?*` will find schemata with the names `books`, `coursebooks`, and `bookings`.

- *Information Browser:* The Information Browser is part of the sidebar and is used to look up built-in functions, and to manage queries that contain parameters. One example of this is when establishing a master-detail pair of queries. The data in the master query is available as parameters to the detail query.

Using the [VIEW](#) menu, most of these sections can be displayed and hidden by checking and unchecking the elements you wish to view.

Each of these topics are covered in more detail in the sections that follow.

8.2. The Query Toolbar

All queries, whether generated automatically, graphically, or manually, appear in the [Query Toolbar](#). The simplest way to use the MySQL Query Browser is to type a query into the query area and click the execute button. A statement terminator such as `;` or `\G` is not required.

8.2.1. The Navigation Buttons

To the left of the query area are the navigation buttons. The navigation buttons allow you to browse through your query history so that you can review and re-execute a previously executed query. The query history is available through the [object browser](#).

Clicking the GO BACK button loads the previous query in your history, whereas the GO NEXT button loads the following query. Only queries that execute without errors are added to your history.

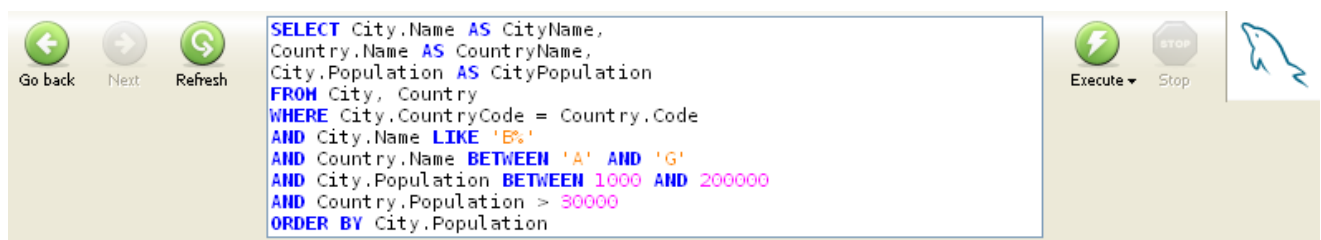
As you navigate with the NEXT and GO BACK buttons, the queries you navigate through are not executed unless you explicitly execute them by clicking the EXECUTE button. The REFRESH button re-executes the last executed query, which may or may not be the current query in the query area.

8.2.2. The Query Area

The query area is where the text of all queries and statements are displayed. By default, the query area is three lines high and automatically expands to a maximum of ten lines in height. For queries longer than ten lines, the query area is scrollable.

For additional space, you can press the **F11** key to maximize the query area. You can also do this by choosing [MAXIMIZE QUERY AREA](#) from the [VIEW](#) menu. When the query is maximized, line numbers are displayed and the query area can be resized by clicking and dragging the line that divides the query area from the result area. Under Windows, pressing the **F11** key again reverts to the normal view. Under Linux choose the [NORMAL VIEW](#) menu option from the [VIEW](#) menu.

Figure 8.2. The Query Area Expanded to Ten Lines



The following commands are available by right clicking the query area: [CUT](#) (**Ctrl+X**), [COPY](#) (**Ctrl+C**), [PASTE](#) (**Ctrl+V**), [CLEAR](#) (**Ctrl+N**), [OPEN QUERY ...](#) (**Ctrl+O**), and [SAVE QUERY AS ...](#) (**Ctrl+S**). The Open and Save commands load the contents of an SQL file

into the query area or save the contents of the query area into a text file.

8.2.3. The Action Buttons

To the right of the query area are the action buttons. The EXECUTE button executes any queries in the query area, whereas the STOP button ceases execution.

If you click the down arrow below the EXECUTE button, there are three potential execution options:

- **EXECUTE (Ctrl+Enter)**: This executes the query and displays the results in the currently active result area.
- **EXECUTE IN NEW TAB (Ctrl+Shift+Enter)**: This executes the query and displays the results in a newly created result area.
- **SPLIT TAB AND EXECUTE (Ctrl+Alt+Enter)**: This splits the result area horizontally and then displays the query results in the lower half of the result area.

To the right of the action buttons is the status indicator. The status indicator shows the standard MySQL logo when the MySQL Query Browser is idle, and displays an animated logo when the MySQL Query Browser is busy processing a query. Both icons can be seen in the following illustration:

Figure 8.3. The MySQL Query Browser Status Icons



8.3. The Advanced Toolbar

Below the query bar is the **Advanced Toolbar**. The **Advanced Toolbar** contains a set of buttons for transaction control, query management, and query building.

Note

The **Advanced Toolbar** is only available under Windows.

Figure 8.4. The Advanced Toolbar



The left panel of the advanced toolbar contains the transaction control buttons. From left to right, the buttons allow you to start, commit, and roll back a transaction. As when using the command-line client, you can only use transactions with table handlers that support them (InnoDB for example). For more information on transactions, see [The InnoDB Transaction Model and Locking](#).

The center panel provides buttons for query management. The EXPLAIN button can be used to get the EXPLAIN output for the current query from the MySQL server. When you use this feature, the results are shown below the result set window. To remove this window right click anywhere in the window and choose the [REMOVE RESULTSET](#) option.

The COMPARE button allows you to compare the results of two queries.

The next panel contains the query building buttons. You can use these buttons to build a query visually by clicking on the tables and columns that you wish to involve in your query, using specialized mouse pointers to indicate which part of the query the different fields and tables occupy.

The right panel contains the CREATE VIEW button. More information on the CREATE VIEW button can be found in the [Creating Views](#) section.

8.4. The Result Area

All query results are displayed in the result area. Within the result area you can use multiple tabs, and individual result areas can be split either vertically or horizontally. New tabs can be created either by choosing the [EXECUTE IN NEW TAB](#) option on the query bar, or by clicking the **NEW TAB** button at the top of the result area. In addition, you can also right click on the current result area and choose the [ADD NEW TABSHEET \(CTRL+T\)](#) option.

Individual tabs are closed by either right clicking within the result area and choosing the [REMOVE TAB](#) option, or by clicking the **X** icon on the tab you wish to close.

To split a result area, right click on it and choose either the [SPLIT TAB VERTICALLY](#) or the [SPLIT TAB HORIZONTALLY](#) option. After splitting the result area you can then choose one half of the result area and view result sets within it. You can remove sections of the result area by right clicking on the section you wish to remove and clicking [REMOVE RESULTSET](#).

When navigating a particularly large result set, you can press the **F12** key to maximize the result area or, alternately, you can choose the [MAXIMIZE TABSHEETS](#) option of the [VIEW](#) menu. Under Windows, pressing **F12** again will restore the result area to its original size. Under Linux, to revert to the normal view choose the [NORMAL VIEW](#) menu option under the [VIEW](#) menu.

The result sets of some queries do not display well in the **Results Area**; `SHOW CREATE TABLE tblname`, for example. For a better view of the results of such queries, right click the row you wish to view and choose the [View Field in Popup Editor](#) option from the pop-up menu.

The result area can be used to review and edit the results of a query, with editing permitted as long as the query is based on a single table and there is sufficient key information to uniquely identify rows. To edit the contents of the result area you must enable edit mode through the use of the **EDIT** button at the bottom of the result area. Any edits you make are not immediately applied, but instead you need to click the **APPLY CHANGES** button next to the **EDIT** button. Clicking the **DISCARD CHANGES** button throws away any changes you have made to the data.

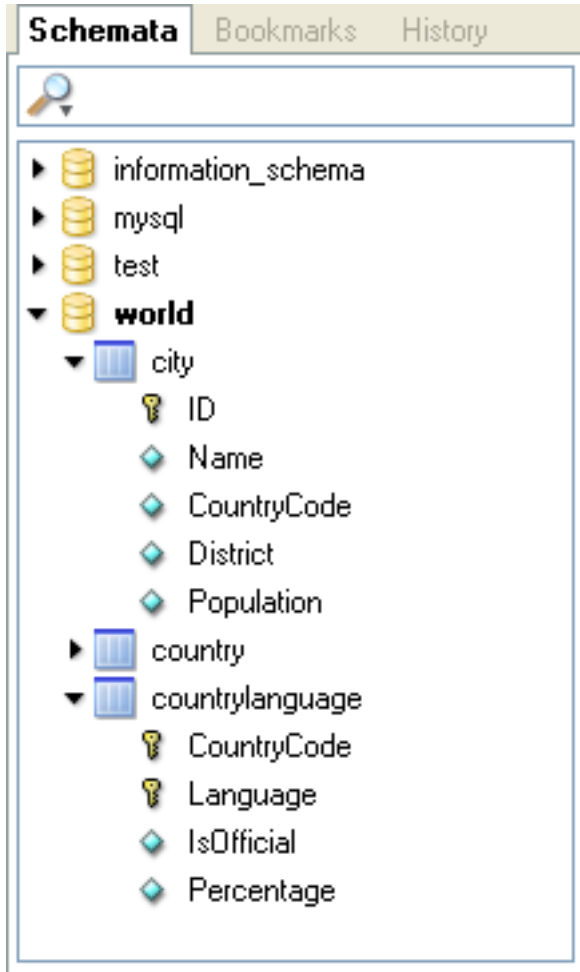
8.5. The Object Browser

Use the object browser to browse your server's databases, bookmarks, and query history.

8.5.1. The Database Browser

The database (schemata) browser is the primary screen of the object browser. You can use the database browser not only to select tables and fields to query, you can also edit tables, create new tables and databases, and drop tables and databases. The database browser can be used to set the default database; this is required before you can issue queries against tables.

Figure 8.5. The Database Browser



The current default database will always be highlighted with bold text. To change the default database, right click on the desired database and choose the [MAKE DEFAULT SCHEMA](#) option or simply double click on the desired database.

You can filter the databases shown by using the search bar at the top of the database browser. As you fill in the search bar, databases that do not match your search are hidden from view. You can clear the search bar by clicking the X button on the right side of the search bar. All databases will once again be displayed. You can specify what objects are filtered by clicking on the magnifying glass icon at the left side of the search bar and choosing the appropriate option. Options include [Schemata](#), [Schema Assets](#), [Columns/Indices](#) and [Custom Selection](#).

To view a database's tables, stored procedures, and views, click the black arrow on the left of the database name. You can view a table's columns by clicking on the black arrow to the left of the table name. Columns that form part of a primary key have a small key icon to the left of their name. Otherwise they have a blue diamond icon.

When you click the black arrow to the left of a view, the columns that form the view are shown. When you click the black arrow to the left of a stored procedure or stored function, the arguments of the stored procedure or function are shown.

To create a new database, right click within the database browser and choose the [CREATE NEW SCHEMA](#) option. To create a new table, right click the database you wish to add a table to and choose the [CREATE NEW TABLE](#) option. To create a new view, right click within the database you wish to add a view to and choose the [CREATE NEW VIEW](#) option. To create a new stored procedure or stored function, right click within the database you wish to add to and choose the [CREATE NEW PROCEDURE / FUNCTION](#) option. You can drop objects by right clicking on the object you wish to drop and choosing the appropriate option ([DROP SCHEMA](#), [DROP TABLE](#), etc.).

You can edit an object by right clicking on it and choosing the appropriate edit option ([EDIT TABLE](#), [EDIT VIEW](#), etc.). When you choose the [EDIT TABLE](#) option, the MySQL Table Editor will be displayed with the selected table. For more information on editing tables, see [The MySQL Table Editor](#). When the [EDIT VIEW](#), [EDIT FUNCTION](#), or [EDIT PROCEDURE](#) options are chosen, the [CREATE VIEW](#), [CREATE PROCEDURE](#), or [CREATE FUNCTION](#) statement will be shown in a new tab using the Script Editor. It is possible to edit all server functions/procedures at once by choosing the [EDIT ALL STORED PROCEDURES / FUNCTIONS](#) option from the [SCRIPT](#) menu.

If you need the CREATE statement for any object, right click on the object and choose the [COPY SQL TO CLIPBOARD](#) option. The appropriate [CREATE](#) statement will be copied to the clipboard and can be used to recreate the selected object.

Note

Only the object itself is created with the [COPY SQL TO CLIPBOARD](#) command, the statements required to populate the object are not created.

8.5.2. The Bookmark Browser

You can place your more commonly used queries in bookmarks so that you can quickly retrieve them and re-use them later. To add a query to your bookmarks, highlight and drag it from the query area into the bookmark browser.

Your bookmarks can be organized into folders and subfolders to help with management of your queries. To add a new subfolder right click on an existing folder and choose the [CREATE BOOKMARK FOLDER](#) option. You can remove bookmarks and folders by right clicking on them and choosing the [DELETE ITEMS](#) option. After you confirm that you wish to delete the item it will be removed from your bookmark list.

Warning

If you delete a folder all items and subfolders within the folder are also deleted.

The Bookmark Browser is one of a number of XML files use for internal purposes by the Query Browser. For a listing of these files see [Appendix E, XML Files Used by MySQL Query Browser](#).

8.5.3. The History Browser

With the history browser you can browse through all the queries you have previously issued. To expand a given day's queries, double click on the day. To load a history item into the query area, double click on it or drag it to the query area.

You can remove history items by right clicking on them and choosing the [DELETE SELECTED HISTORY ENTRIES](#) option from the drop-down menu. You can also use the [CLEAR HISTORY](#) option to erase all history entries.

You can create bookmarks from history items by right clicking on a selected history item and choosing the [ADD HISTORY ITEM AS BOOKMARK](#) menu option.

8.6. The Information Browser

The Information Browser provides access to all information that is not directly related to actual data within your database. The parameter browser provides different local, global, and dynamic parameters that can help build your queries, whereas the syntax browser provides a convenient reference to MySQL query syntax. The Function browser is a quick reference to the various functions built into MySQL. The Transaction Browser lists all queries that make up a single transaction and serves as a history for a single transaction.

8.6.1. The Syntax Browser

The syntax browser provides a quick reference tool for determining proper syntax to use when constructing queries. By double clicking on the various types of syntax (SELECT, REPLACE, and so forth) you can bring up reference information using the in-line help system.

Reference information is displayed in a new tab within the result area.

To expand a given syntax category, double click on the category header.

8.6.2. The Function browser

The function browser provides a quick reference tool for usage of the built-in functions of the MySQL server. By double clicking on the various functions you can bring up reference information using the in-line help system.

To expand a given function category, double click on the category header.

Reference information is displayed in a new tab within the result area.

8.6.3. The Parameter Browser

The parameter browser contains all the local, global, and dynamic parameters that can be included in your query. Local parameters affect the query in the current query window only. Global parameters affect all queries. Dynamic parameters are generated automatically from existing queries.

Parameter categories can be displayed and hidden by clicking on the **GLOBAL PARAMS**, **LOCAL PARAMS** and **DYNAMIC PARAMS** headings.

To set the value of a parameter, click on the value and press the **F2** key, or double click the value.

To add a new parameter, right click within the parameter browser and select the ADD PARAMETER option.

To delete a parameter, right click on the parameter and choose the DELETE PARAMETER option.

To change a Local parameter into a Global parameter, right click on the parameter and choose the MOVE PARAMETER TO GLOBAL PARAMETERS option.

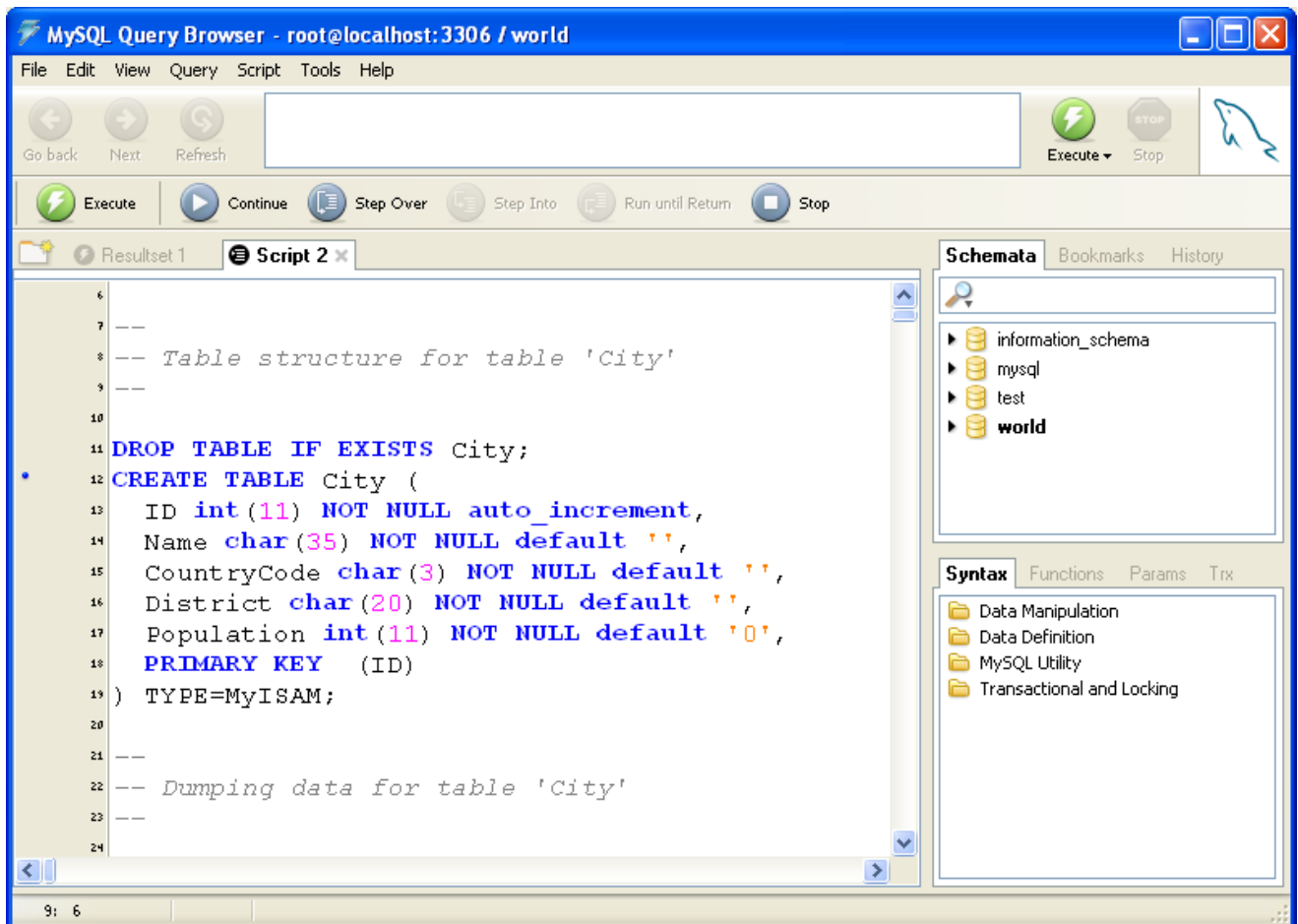
8.6.4. The Transaction Browser

The Transaction Browser provides a list of all statements that have been executed in the current transaction. The Transaction browser will be displayed when you start a transaction using the transaction control buttons.

8.7. The Script Editor

The **Script Editor** provides an alternate interface for use when dealing with large SQL scripts that involve multiple SQL statements.

Figure 8.6. The Script Editor



The Script Editor displays within an individual tab in the [Result Area](#). When a Script Editor tab is active, the [Advanced Toolbar](#) buttons are replaced with script debugging buttons.

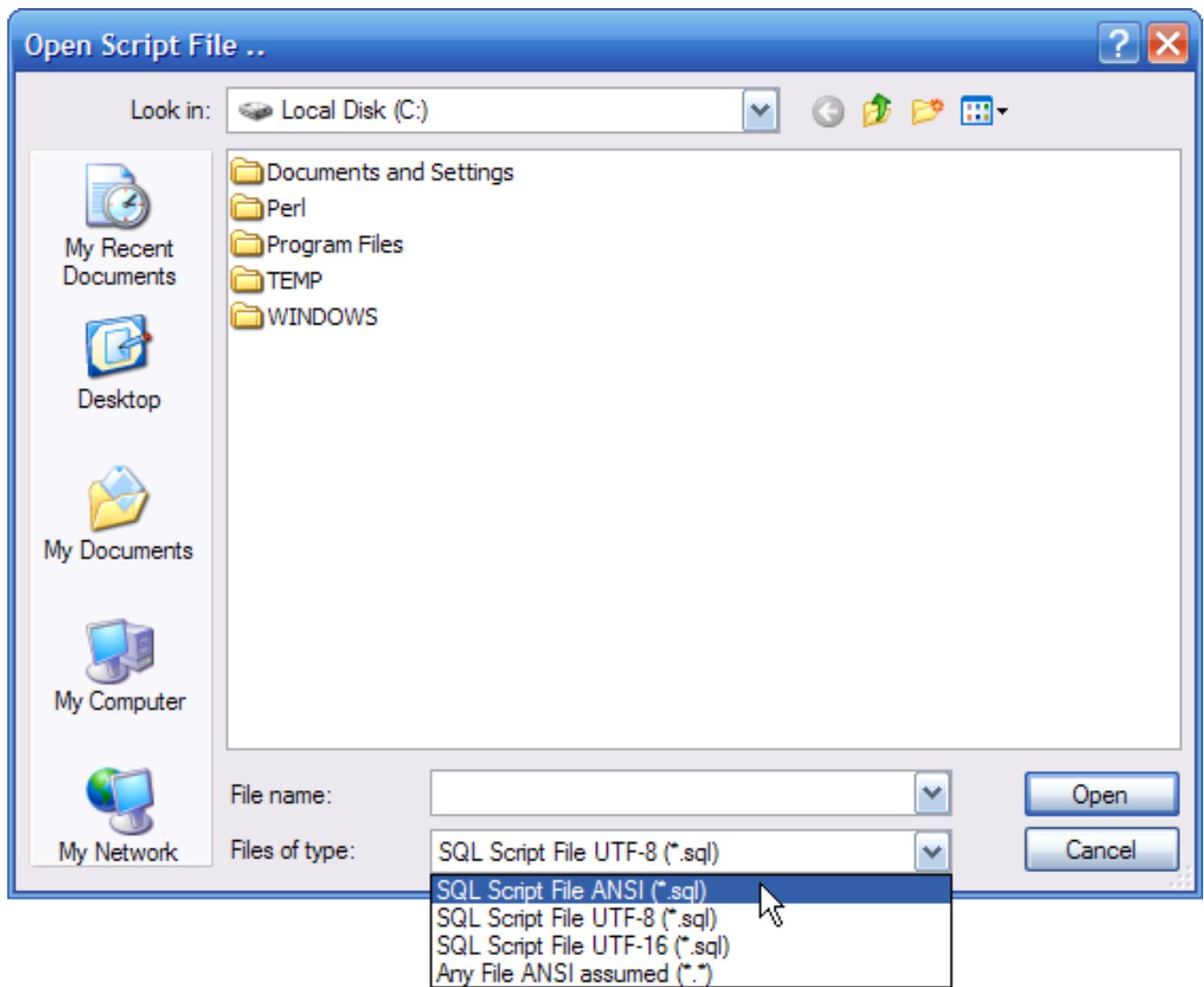
To open a script in the Script Editor, choose the [OPEN SCRIPT ...](#) option from the [FILE](#) menu. You can also press the **Ctrl + O** keys to open an SQL script.

To create a new Script Editor window without loading a script, choose the [NEW SCRIPT TAB](#) option from the [FILE](#) menu.

If you're experiencing problems with character sets (for example, Spanish accented characters, Russian, or Chinese characters aren't displayed properly), here's some advice:

Avoid ANSI encoding in your script files. Rather than using ANSI, save scripts in *UTF-8* encoding. Take care when loading script files generated by other tools. Many tools save files in ANSI encoding, which in turn means you have to open them as ANSI in Query Browser (see the **FILES OF TYPE** combo box in the file open dialog):

Figure 8.7. The File Open Dialog



There are significant limitations to ANSI formatted files; the current system locale must be the same as the machine where the file originated. For example, you cannot store a script in ANSI on a French version of Windows and expect it to load correctly on a Spanish version of Windows. Because of this and many other limitations we strongly recommend always using Unicode for scripts. It doesn't matter whether you use UTF-8 or UTF-16, since both are Unicode transformation formats and can be read properly by any of the MySQL GUI tools. Using UTF, you could even write Spanish text with Chinese comments on a German version of Windows.

The MySQL Query Browser forum provides many contributions by users of non-Latin1 characters sets; see <http://forums.mysql.com/list.php?108>.

8.7.1. The Script Debugging Buttons

The following buttons are available when using the Script Editor:

- **Execute:** Execute the script from the beginning and do not stop execution until the end of the script is reached.
- **Continue:** Execute the script from the beginning (or current position) and stop for errors or break points.
- **Step:** Execute the next statement, functions will be executed but not stepped into.
- **Pause:** Stops script execution and highlights on the next statement to be executed.

- **Stop**: Stop execution of the script.

8.7.2. The Script Area

The current script being edited in the Script Editor is displayed in the **Script Area**. The Script Area features line numbering and syntax highlighting to make it easier to work with scripts.

Script lines that are marked with a blue dot to the left of the line are eligible for use as breakpoints. To set a line as a breakpoint, click the blue dot. You can also set a breakpoint by positioning the cursor in the line you wish to break at and clicking the TOGGLE BREAK-PPOINT option in the SCRIPT menu. To remove all breakpoints, choose the CLEAR ALL BREAKPOINTS option of the SCRIPT menu.

To execute a selection of statements, highlight the statements you wish to execute and choose the EXECUTE SELECTION option of the SCRIPT menu.

Chapter 9. Using The MySQL Query Browser

In this chapter we further describe the tools available in the MySQL Query Browser by using practical examples.

9.1. Entering and Editing Queries Manually

The query browser is most commonly used for executing queries and analyzing their results. The most direct way to create a query is to type it directly into the query area. As you type in your query, the SQL syntax portions of the query (SELECT, FROM, WHERE, and so on) are highlighted in blue.

As you enter your query, the query area will expand from an initial three lines in height to a maximum ten lines in height. For additional space, you can press the **F11** key to maximize the query area. You can also choose the MAXIMIZE QUERY AREA of the VIEW menu to maximize the query area. When the query is maximized, line numbers are displayed for the query, and the query area can be resized by clicking and dragging the line that divides the query area from the result area. To restore the query area, press the **F11** key again.

Once you have entered a query, click on the EXECUTE button and your query results are displayed in the result area. You can also press **Ctrl+ Enter** to execute the query. If there is an error in your query, an error area appears at the bottom of the result area and displays the relevant error message and error number.

In addition to loading the query results into the current active result area, you can also create a new result area for the results of your query or split your current result area and load results into the new section.

To execute the query and load the results into a new result area click the down-arrow below the EXECUTE button and choose the EXECUTE IN NEW TAB option or press **Ctrl+Shift+Enter**.

To split the active result area and display the query results click the down-arrow below the EXECUTE button and choose the SPLIT TAB AND EXECUTE option or press **Ctrl+Alt+Enter**.

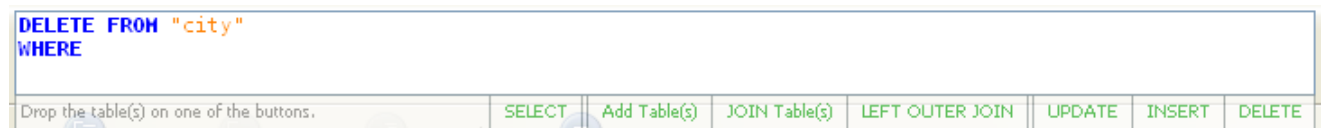
You must set a default database before you can query specific tables (though you can still perform queries that are not database-specific). You can set the default database at the connection screen, or by right-clicking on a database in the database browser and choosing MAKE DEFAULT SCHEMA, or by choosing the CHANGE DEFAULT SCHEMA option from the FILE menu.

9.2. Building Queries

One feature of the MySQL Query Browser is the ability to build queries. Use the database browser to select the columns and tables you wish to query and have the query created automatically based on your choices.

The first step to building a query is to choose a table to query. Click and drag a table to the query area to start a query. For example, by dragging the City table to the query area, SELECT * FROM City C is added to the query area. You can also double-click a table to start a new SELECT query.

Figure 9.1. The Table Tool



When you select a table from the database browser and drag it over the query area, a table tool with some query composition actions is displayed. Drop the table you're dragging on the desired action and the query is modified accordingly. The following actions are possible:

- **SELECT** replaces the current statement with a **SELECT** query containing the dragged table.
- **Add Table** adds the dragged table to the list of tables in the current **SELECT** query
- **JOIN Table**: If a **SELECT** query is already in the query box, with a table in it, the new dragged table is added and the appropriate **WHERE** clauses to perform a **JOIN** will be added
- **LEFT OUTER JOIN** is the same as the previous, but does a **LEFT OUTER JOIN** instead of a **JOIN**

- **UPDATE** replaces the current statement with an **UPDATE** statement containing the dragged table
- **INSERT** replaces the current statement with an **INSERT** statement containing the dragged table
- **DELETE** replaces the current statement with a **DELETE** statement containing the dragged table

Tables are joined based on identical column names for MyISAM tables and foreign key information for InnoDB tables.

Once a table is selected, you can choose specific columns to query; click the **SELECT** button from the query building buttons on the button bar. Your mouse pointer is changed to a **Select** pointer, which you can use to choose columns from the database browser. For example, by clicking on the Id, Name, and Country fields of the **world** sample database, the query **SELECT C.Id, C.Name, C.Country FROM City C** is built in the query area.

Once you have chosen the columns you wish to query, you can use the other query building buttons to complete your query with **WHERE**, **GROUP BY**, and **ORDER BY** clauses. When a new section of the query is added with the query building buttons, the cursor in the query area is placed in position for editing; if you click a field with the **WHERE** pointer, the cursor is in position for you to type in the details of the **WHERE** clause.

You can change between the different query building pointers by clicking on the query building buttons in the button bar, or by using a combination of **Ctrl+Alt** and the first letter of the pointer you wish to use (**Ctrl+Alt+S** for **SELECT**, **Ctrl+Alt+W** for **WHERE**, and so forth.)

If the query building buttons are not visible, they can be displayed through the Browser Options. See [Section 11.5.1, “Display Options”](#) for more information.

9.3. Creating Views

The simplest way to create a view using the MySQL Query Browser is to use the **CREATE VIEW** button. Execute a query that represents the view you wish to create. Once the query is executed, click the **CREATE VIEW** button, provide a name for the view, and the view is created.

Note

Creating views is not yet supported on all Operating systems.

You can also create a view by right-clicking on the database you wish to add a view to in the database browser and choosing the **CREATE NEW VIEW** option. After you enter a name for the view, a view template will be displayed in the script editor.

9.4. Navigating Result Sets Within the Result Area

Once you have successfully executed a query you can then view and manipulate the result set within the result area.

You can navigate the result area using the arrow keys, tab key, and PageUp/PageDown keys. The Home and End keys can be used to move to the first and last column within a given row. Your current position within the result set is shown in the bottom-left corner of the application window. The **FIRST** and **LAST** buttons at the bottom of the result area can be used to move to the first and last rows of the result set.

NULL data will be indicated with a special **NULL** icon to differentiate NULL data from empty strings. BLOB fields will be empty and will have a special **BLOB** icon.

When navigating a particularly large result set, you can press the **F12** key to maximize the result area. You can also choose the **MAXIMIZE TABSHEETS** option of the **VIEW** menu to maximize the result area. Pressing **F12** again will restore the result area to its original size.

To search for a particular value within the result set, click the **SEARCH** button. To see additional search options, click the **DETAILS >>** button. The following options are available:

- **Case Sensitive**: The search is performed in a case-sensitive manner. By default, searches are not case-sensitive.
- **Whole Words Only**: The search does not allow partial matches. By default, partial matches are allowed (for example, **Edm** will match **Edmonton**).
- **Search From Top**: The search begins with the first row in the result set.

- **Search From Cursor**: The search begins from the currently selected row.
- **Search all Text / Columns**: The search will involve all columns of the result set.
- **Search Only in Selected Text / Column**: The search will only involve the currently selected column(s) of the result set.
- **Search Up**: The search will scan the result set, starting at the designated start location and moving upwards.
- **Search Down**: The search will scan the result set, starting at the designated start location and moving down.

To perform a search and replace operation on a result set, click the **Replace** tab in the search dialog window. The options for performing a replace operation are the same as for a search. To replace a single instance of a string, click the **REPLACE** button. To replace all instances of a string, click the **REPLACE ALL** button.

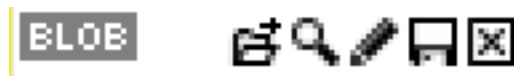
9.5. Exporting Result Sets

You can export any result set from MySQL Query Browser by right-clicking within the result set and choosing an option from the **EXPORT RESULTSET** sub-menu. You can choose to export the result set in **CSV**, **XML**, **HTML**, Microsoft Excel **XLS** or **PLIST** formats.

9.6. Handling BLOB and TEXT Columns

The MySQL Query Browser provides functionality for dealing with **BLOB** and **TEXT** columns through a series of special icons.

Figure 9.2. The BLOB Handling Icons



These icons appear in any **BLOB** or **TEXT** columns in your result set. From left to right the following icons are available:

- **Open File**: This icon looks like a file folder and is used to open a file for loading into the field.
- **View**: This icon looks like a magnifying glass and is used to open the field viewer to view the contents of the field. The field viewer can be used to view **TEXT** fields and **BLOB** fields that contain images.
- **Edit**: This icon looks like a pencil and opens the field viewer in edit mode, allowing you to make changes to the data and apply the changes to the result set.
- **Save**: This icon looks like a floppy disk and is used to save the contents of a **TEXT** or **BLOB** field into a file.
- **Clear**: This icon looks like an X within a black box and is used to clear the contents of a **TEXT** or **BLOB** field.

Only the **View** and **Save** icons are visible if you have not enabled edit mode. See [Section 9.7, “Editing Result Sets Within the Result Area”](#) for information on editing result sets.

The functionality represented by the icons is also available by right-clicking on the field. The blob handling icons can be displayed and hidden through the display options. See [Section 11.5.1, “Display Options”](#) for more information.

9.7. Editing Result Sets Within the Result Area

When a query is based on a single table and there is sufficient key information, the result set can be edited from within the result area. To edit a result set click the **EDIT** button at the bottom of the window. If the **EDIT** button is not active, your result set is not editable. (Under Linux this button is labeled **START EDITING**.)

Note

Queries derived from a single table lacking a primary key or from more than one table are *not* editable; they are read-only.

Once in edit mode you can insert, update, and delete rows from the result set. Navigate the fields using the **Tab** and arrow keys, and press **Enter** to edit the content of a field. You can also double-click a field to make it editable. When editing a field, use the tab key to move to the next editable field. All edited fields are highlighted in blue for easy identification.

To add rows to the result set, scroll to the blank row at the bottom of the result area and fill in the fields. All new rows are highlighted in green.

To delete a row right-click on the row and choose the DELETE ROW option. All deleted rows are highlighted in red.

Changes made to the result set are not applied immediately, but instead are cached until the **APPLY CHANGES** button is pressed. You can abort your edits with the **DISCARD CHANGES** button. Exiting edit mode without choosing to apply or discard your changes prompts you to apply or discard your work.

9.8. Comparing Result Sets

You can compare result sets graphically with the MySQL Query Browser, allowing you to easily determine where rows have been inserted, updated, or deleted.

To compare two result sets, execute the first of the queries you wish to compare. Once the result set has loaded, right-click on the result set and choose the SPLIT TAB HORIZONTALLY option. Load your second query into the new section of the result area and click the **COMPARE** button to compare the two result sets.

When you have activated the compare mode, both result sets will scroll in unison, both vertically and horizontally. Rows are matched for comparison, with blank rows added when one set has a row that the other set lacks.

If one result set has a row that the other result set does not have, that row is highlighted in green. The other result set has a blank row inserted that is highlighted in red. If both result sets have a matching row, but individual fields are different, those fields are highlighted in blue.

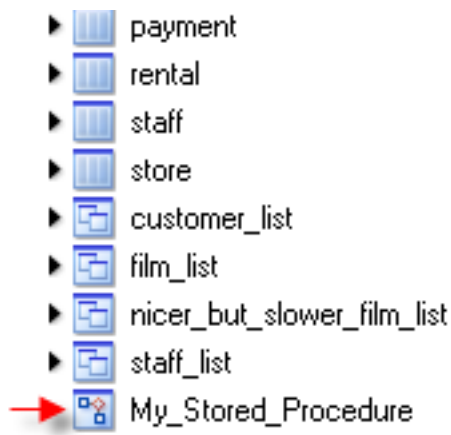
In order to successfully compare two result sets, you need two queries with matching column names and column order. For the MySQL Query Browser to match rows, primary keys must be defined in the tables used.

9.9. Managing Stored Procedures and Functions with MySQL Query Browser

When used with MySQL version 5 and higher, the MySQL Query Browser supports creating, editing, and viewing stored procedures and functions.

Stored procedures and stored functions are displayed in the database browser with a special icon to distinguish them from regular tables and views. Click the arrow to the left of a stored procedure or stored function to display the parameter list for that procedure or function.

Figure 9.3. Stored Procedure in Database Browser



To edit a stored procedure or stored function, right-click on it in the database browser and choose the EDIT PROCEDURE or EDIT FUNC-

[TION](#) option. This opens a new script editor tab with the selected procedure/function displayed. Once you have finished editing, click the [EXECUTE](#) button above the script area or click the [EXECUTE](#) option of the [SCRIPT](#) menu to update the procedure/function.

To create a new stored procedure or function, choose the [CREATE STORED PROCEDURE / FUNCTION](#) option from the [SCRIPT](#) menu. Enter the desired procedure/function name and click either the [CREATE PROCEDURE](#) or [CREATE FUNCTION](#) button. A template similar to the following will be created:

```
DELIMITER \\  
  
DROP PROCEDURE IF EXISTS `test`.`new_proc`\  
CREATE PROCEDURE `test`.`new_proc` ()  
BEGIN  
  
END\  
  
DELIMITER ;
```

After typing the procedure/function, click the [EXECUTE](#) button above the script area or click the [EXECUTE](#) option of the [SCRIPT](#) menu to create it.

To edit all stored procedures/functions at once, choose the [EDIT ALL STORED PROCEDURES / FUNCTIONS](#) option of the [SCRIPT](#) menu. A new script editing tab will be created, containing all the stored procedures and stored functions for the current default database.

To remove an existing stored procedure or stored function, right-click on it in the database browser and choose the [DROP PROCEDURE](#) or [DROP FUNCTION](#) option.

9.10. Editing Queries From a Development Tool

In order to help programmers optimize and troubleshoot their queries more efficiently, the MySQL Query Browser can copy queries from application code using your favorite integrated development environment (IDE).

Note

This functionality is only available for the Windows version of MySQL Query Browser.

The following PHP code will be used as an example:

```
$SQL = "SELECT Id, Name, Country FROM City" .  
      "WHERE Name LIKE $cityname";
```

To copy the listing into the MySQL Query Browser, copy the block of code (including the assignment portion), right click within the query area of the MySQL Query Browser, and choose the [PASTE CLIPBOARD CONTENT AS PHP CODE](#) option. The nonquery portions of the highlighted area will be stripped and the query will be pasted into the query area.

The dynamic elements of the query are converted into local parameters, visible in the [parameter browser](#):

```
SELECT Id, Name, Country FROM City  
WHERE Name LIKE :cityname
```

To set the value of a local parameter, highlight the value in the parameter browser and press **F2**. You can also double-click on the value to edit it. The value you assign will be used when the query is executed.

After editing the query, right-click within the query area and choose the [COPY QUERY AS PHP CODE](#) option. The surrounding PHP code will be re-inserted along with the modified query. This functionality allows you to edit queries quickly while programming.

9.11. Using the Regex Text Importer Tool

Caution

This feature is currently experimental, and so should be used with caution. Further information will be available in the near future.

Chapter 10. The MySQL Table Editor

10.1. Introduction

The MySQL Table Editor is a component of the MySQL Query Browser and the 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.

The appearance of the MySQL Table Editor varies slightly depending upon the operating system (OS) and whether the MySQL Table Editor is docked or not. For example, under Linux the Columns and Indexes are separate tabs but under Windows they are combined in one. However, the same basic functionality is available regardless of docking or the particular OS.

Note

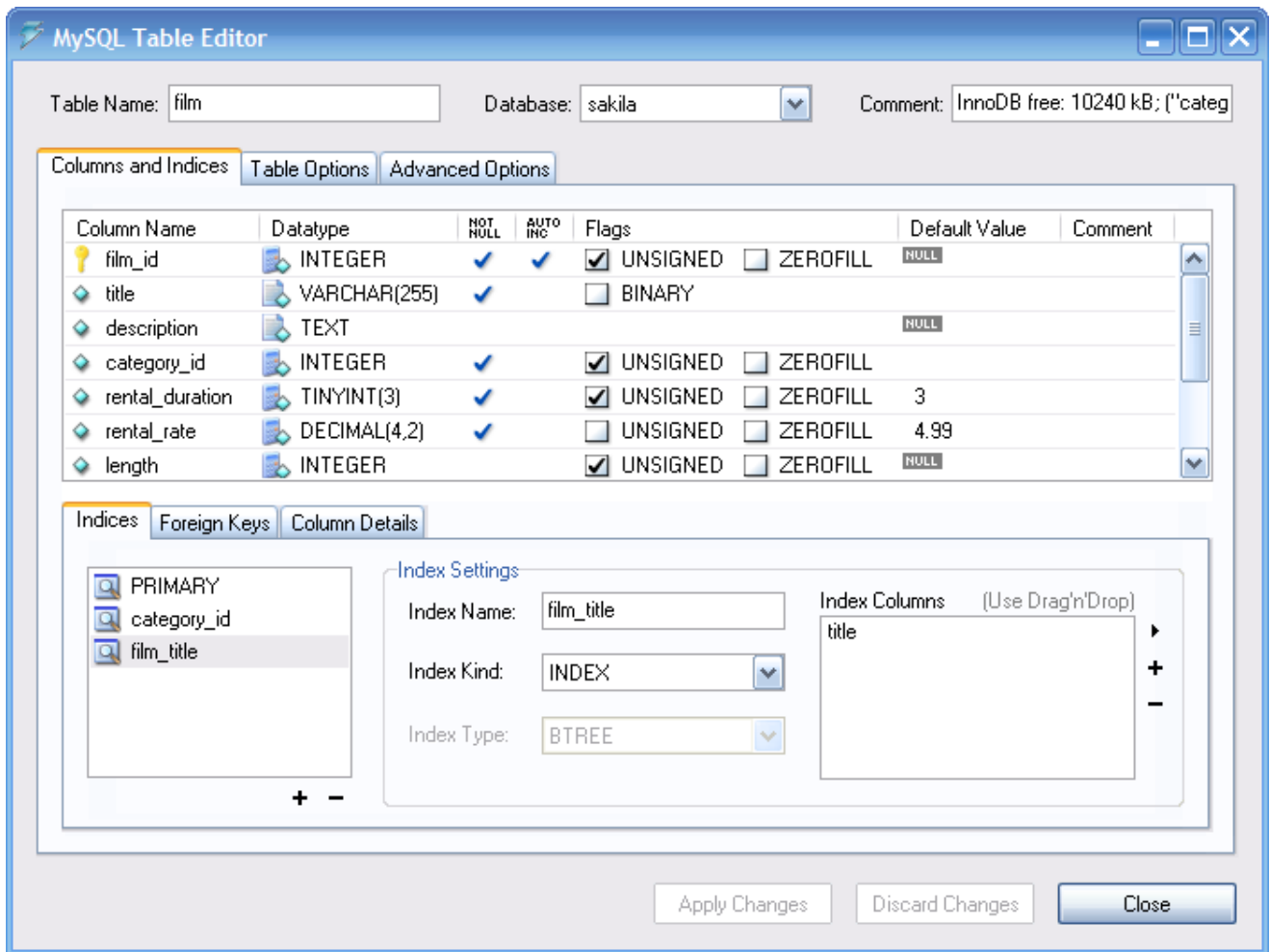
Docking is not supported on all operating systems.

10.2. The Main Editor Window

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

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

Figure 10.1. The MySQL Table Editor



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 configure options 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.

10.3. 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.

10.3.1. 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 10.2. 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 add a column simply double-click a column in an empty row and enter an appropriate value.

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 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.

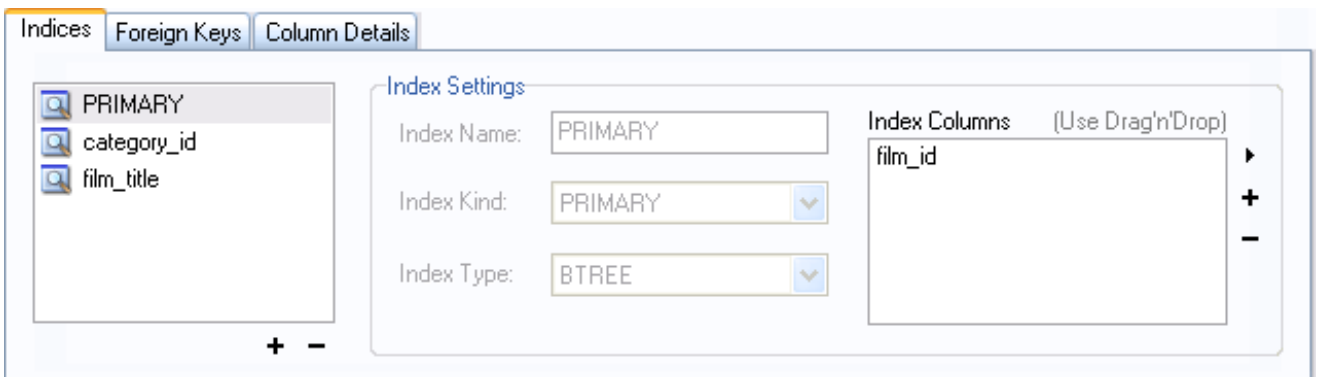
10.3.2. The Detail Area

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.

10.3.2.1. 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 10.3. The Index Editor



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. Under Windows, you can change the automatically generated index name by using the **Options** dialog window. For more information, see [Section 11.4, “Editors”](#).

To drop an index, select the index and click the – (minus) button.

Use the **INDEX NAME** and **INDEX KIND** dialog boxes 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.

10.3.2.2. The Foreign Keys Tab

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

To add a foreign key, click the + (plus) button below the foreign key list. A dialog box prompts for a foreign key name and the new foreign key is created. Under Windows, you can change the automatically generated foreign key name by using the **Options** dialog window. For more information, see [Section 11.4, “Editors”](#).

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 dialog boxes provided in the **FOREIGN KEY SETTINGS** section of the tab.

The options are:

- NO ACTION
- RESTRICT
- CASCADE
- SET NULL

To establish a foreign key relationship, choose a table from the **REF. TABLE** drop-down list. (Under Linux this list box is called **REFER. TABLE**.) The fields that can be related to create a foreign key are listed in the frame below the drop-down list. This frame contains two columns; one labeled **Column** and the other **Reference Column**. To select a column as a foreign key, double click the area below **COLUMN** and select a column from the drop-down list. Alternately, drag a column from the column editor to the **COLUMN** section. To add a related column from the referenced table, double click the area below **REFERENCE COLUMN** and select the related column from the drop-down list.

10.3.2.3. 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.

Depending upon circumstances one or the other of these views may be more convenient. However, changing the data type in the tabular view is especially easy because a drop-down list shows all available data types.

10.4. 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 in a drop-down list box. Select a particular engine to display a brief summary of that engine's features and capabilities.

Under Windows, engines that are not available are shown in red. Under Mac OS X, unavailable engines are grayed out and under Linux there is no differentiation. If your server has built-in support for the **Falcon** engine, that engine will show up in the list of available engines. Otherwise, the **Falcon** engine does not show at all.

To change the storage engine for your table, select the desired storage engine from the list box.

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

10.5. 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.

For descriptions of most options set using the Advanced Options tab, see [CREATE TABLE Syntax](#).

10.5.1. 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.

10.5.2. 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 [MyISAM Table Storage Formats](#), 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 [CREATE TABLE Syntax](#), for more information on how to use these options.

10.5.3. 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.

10.5.4. 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 [The MERGE Storage Engine](#), for more information on MERGE tables.

10.5.5. 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.

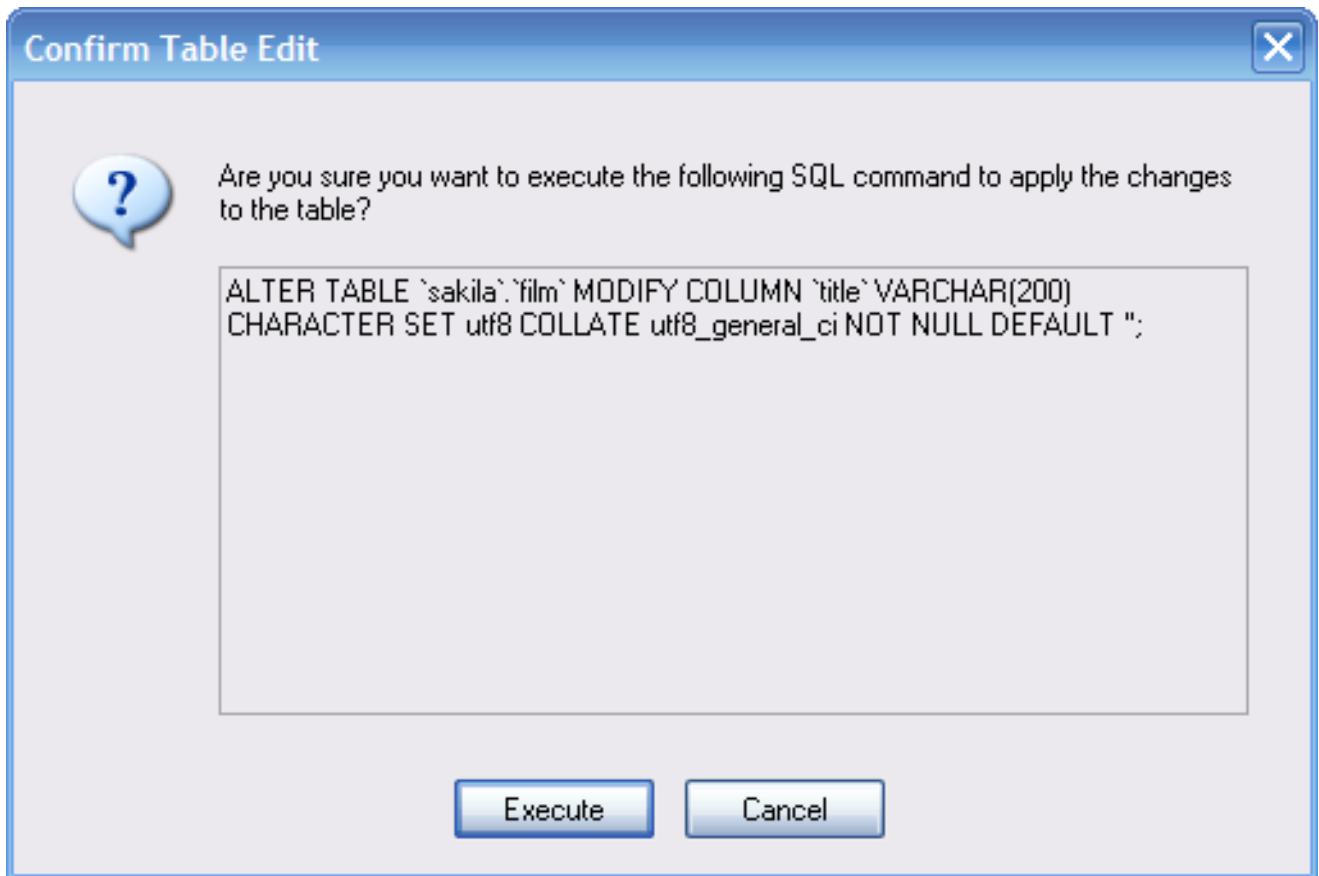
For more information on using RAID support with MyISAM, see [CREATE TABLE Syntax](#).

10.6. 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 10.4. The Confirm Table Edit Dialog



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.

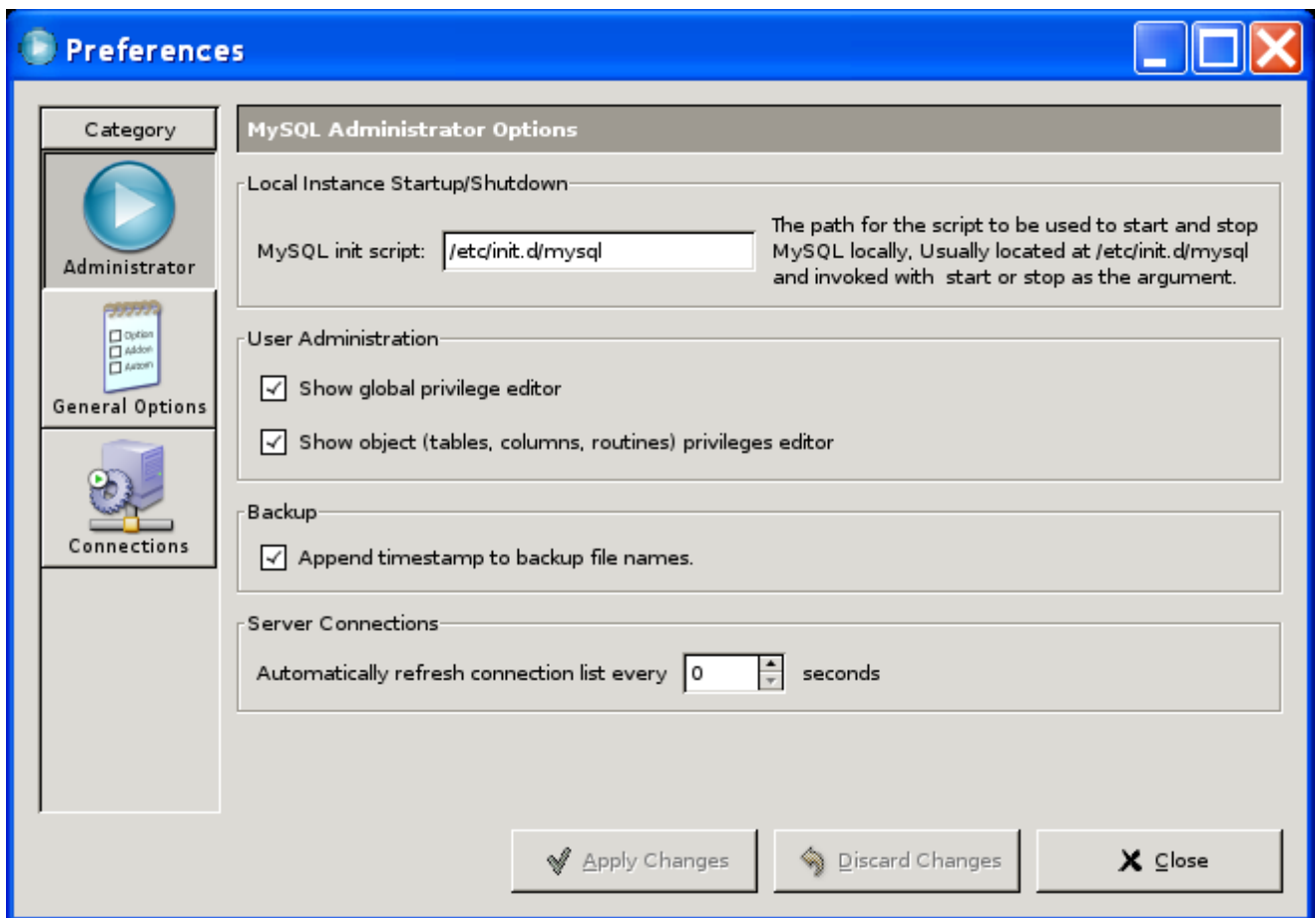
Chapter 11. Options Dialog

11.1. 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 11.1. Options Dialog



In the sidebar of the dialog, you can select the section you wish to configure. These options are **Administrator**, **General Options**, **Connections**, and, on Windows only, **Editors**.

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.

11.2. 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 are stored 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 application 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.

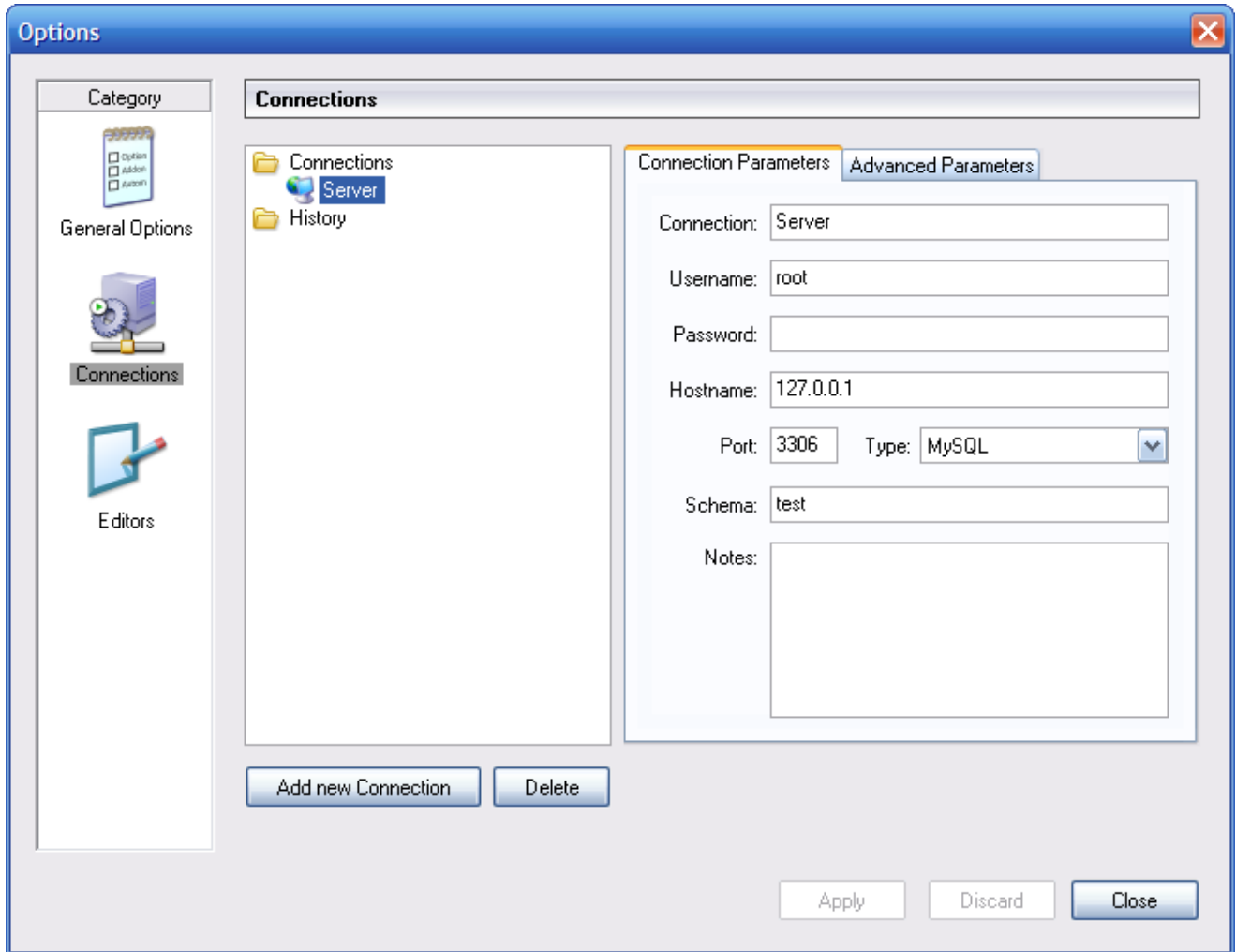
You may change the font and font size of any of the application fonts by clicking the CHOOSE ... button to the right of the font.

11.3. 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 11.2. Options Dialog: Connections



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 user name 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 reference 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.

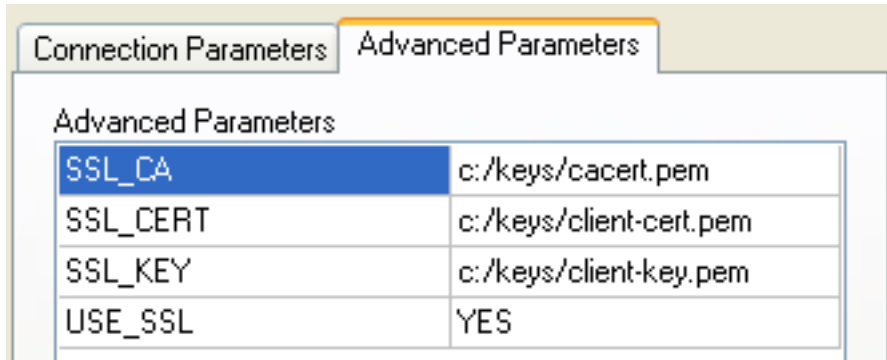
- **Allow interactive_timeout seconds of inactivity before disconnect:** Normally, the connection is closed by the MySQL server after a certain period of inactivity on the client side. This period can be set with the `interactive_timeout` variable. If checked, the server will not close the connection 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 you 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.)

11.3.1. SSL Connections

To set up an SSL connection under Windows, open the **ADVANCED PARAMETERS** tab of the **Options Dialog** window. Select the connection you wish to configure, or create a new connection. Use the **Advanced Parameters** text boxes to send options to the MySQL client library in much the same way that you would pass parameters to the command-line client.

The parameters are essentially the same as those you would use for the command-line client, with underscores replacing “-” and **USE_SSL** replacing `--ssl`. See the following example.

Figure 11.3. Options Dialog: Advanced Parameters

**Note**

The file separator used for path parameters is “/” and not “\”.

The parameters used with an SSL connection are found in the [SSL Command Options](#) section of the reference manual. For instructions on setting up SSL certificates, see [Setting Up SSL Certificates for MySQL](#). Links to the different versions of the reference manual are found at <http://dev.mysql.com/doc/refman>.

There is currently no provision for creating an SSL connection when running MySQL GUI Tools on Mac OS X.

11.4. 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](#).

Note

Currently, configuring editors is only available on the Windows platform.

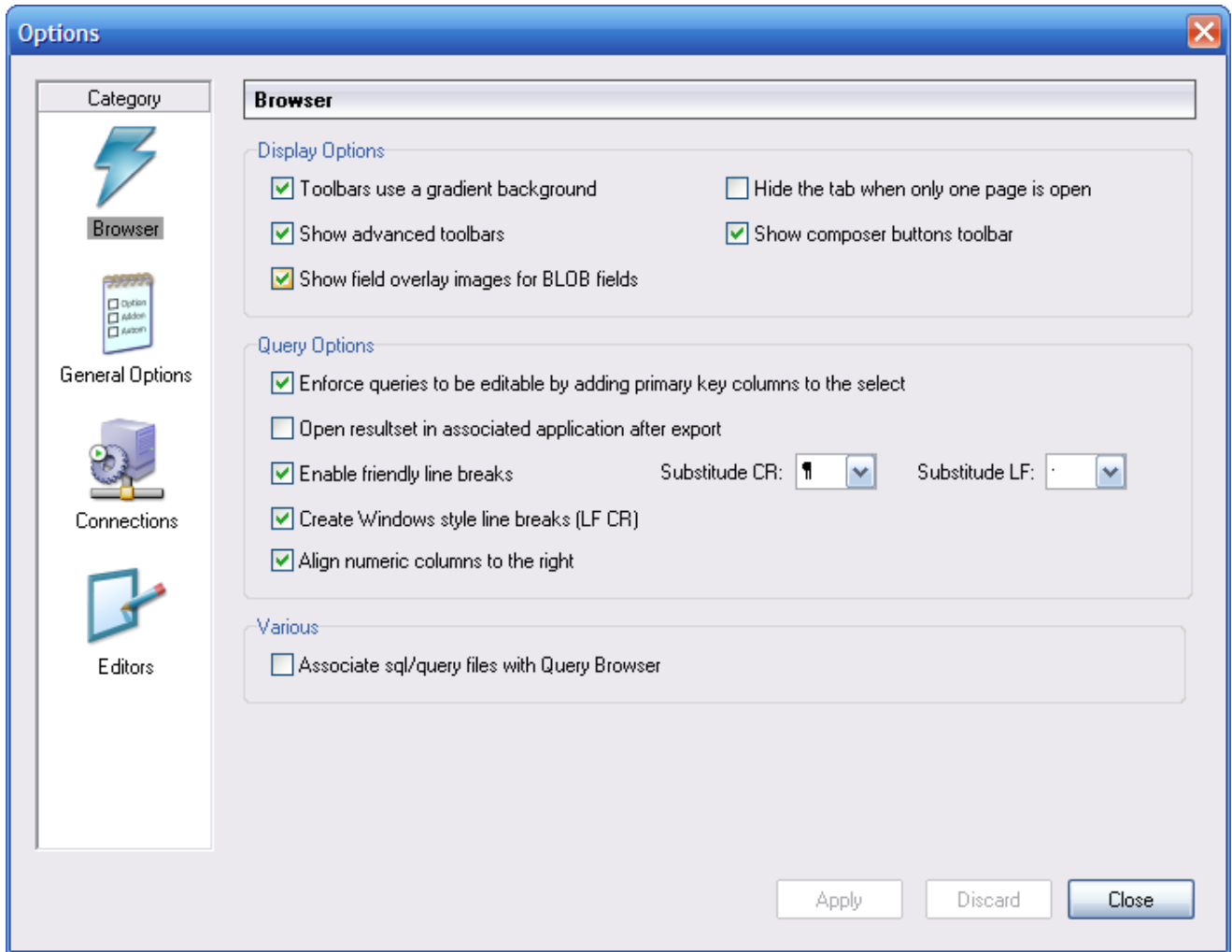
- [Show SQL command before applying changes](#): Toggles whether the MySQL Table Editor will show you the [CREATE TABLE](#) or [ALTER TABLE](#) statement it is about to execute for confirmation when you click APPLY CHANGES.
- [All columns Not Null per default](#): Determines whether the MySQL Table Editor 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](#).

11.5. The Browser Section

The [Browser](#) section allows you to specify options that relate to the MySQL Query Browser. The [Browser](#) section is divided into

three areas: [Display Options](#), [Query Options](#), and [Various](#).

Figure 11.4. The Query Browser Options Dialog



11.5.1. Display Options

The display options affect the appearance of the MySQL Query Browser.

- [Show advanced toolbars](#): Display or hide the [The Advanced Toolbar](#).
- [Hide the tab when only one page is open](#): Determines whether or not to display a tab icon at the top of the [Result Area](#) if there is only one result area.
- [Toolbars use gradient background](#): Enables and disables use of gradient effects in the background of the [Query Toolbar](#).
- [Show field overlay images for long VARCHAR, TEXT, and BLOB fields](#): Toggles on and off the BLOB management icons.
- [Show composer buttons toolbar](#): Toggles on and off the query composition buttons on the Advanced Toolbar.

11.5.2. Query Options

[Enforce queries to be editable by adding primary key columns to the select](#): Adds **PRIMARY KEY** columns to a query that does not explicitly **SELECT** them to ensure that the results of a query can be edited. The **PRIMARY KEY** columns will not be displayed in this case, but will still be stored.

[Open resultset in associated application after export](#): Causes the associated application to be opened after choosing the **EXPORT RESULTSET** option from the right click menu of the [Result Area](#).

11.5.3. Various

[Associate sql/query files with Query Browser](#): Configures Windows to associate all .sql files with the MySQL Query Browser. As a result, all such files will be opened with the MySQL Query Browser by default.

Appendix A. Installing From Source

A.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.

A.2. Downloading The Source Code

The source code for the MySQL GUI tools is available from our public Subversion trees at <http://dev.mysql.com/tech-resources/sources.html>.

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.

To be able to access these trees, you must have Subversion installed. Subversion is freely available from <http://subversion.tigris.org/>.

A.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.

A.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 preferred)
- pcre-1.4
- MS Visual C runtime libraries 7.1
- Java include files 1.4.2_04
- Lua 5.0.2

A.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
```

A.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 (for example, 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.

A.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 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.*

A.4. Building from Source under Linux

A.4.1. Prerequisites

The easiest way to install required packages is by using an automatic dependency resolving package manager, such as **apt-get** or **yum**. You can install these dependencies by hand, but it will be a very tedious process since some of them require other packages that require other packages and so on.

Common requirements are listed below. For some distributions the package name is also mentioned.

Table A.1. Required Components

Software	Version	Ubuntu 6	SuSE 10.x
gcc/g++	4	g++	gcc-c++
autoconf, automake libtool			
x11-libs		libx11-dev	xorg-x11-devel
glibc		libc6-dev	xorg-x11-devel
libstdc++		libstdc++6-4.0-dev	libstdc++devel
glib	2.6+	libglib2.0-dev	glib2-devel
gtk+	2.6+	libgtk2.0-dev	gtk2-devel
libglade2	2.5+	libglade2-dev	libglade2-devel
libsigc++	2.0.x	libsigc++-2.0-dev	libsigc++2-devel
glibmm-2.4	2.4	libglibmm-2.4-dev	glibmm24-devel
gtkmm-2.4	2.4	libgtkmm-2.4-dev	gtkmm24-devel
pcre	4.x or newer	libpcre3-dev	pcre-devel
pkgconfig	0.15+	pkg-config	pkgconfig
libxml2	2.6.x	libxml2-dev	libxml2-devel
MySQL 5.0 or 5.1 client		libmysqlcli-	

Software	Version	Ubuntu 6	SuSE 10.x
libs and headers		ent15-dev	

Additional requirements for building MySQL Query Browser are:

Table A.2. Requirements for MySQL Query Browser

Software	Version	Ubuntu 6	SuSE 10.x
libgtkhtml	2.11 or 3.x	libgtkhtml2-dev	
libgnomeprint	2.2	libgnomeprint2.2-dev	

A.5. Compiling and Building

To build any of the tools, you need to have the source tree for [mysql-gui-common](#) and the tree for the desired app (for example: [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.

A.5.1. Building Static Binaries

Building static binaries is nontrivial under Linux, due to the enormous number 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, especially in the case of gtk/gnome. Problems don't only

appear when the user is missing these data files; things may not work because different distributions put 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 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.

A.5.2. 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/mysql-administrator/*
```

The contents of each subdirectory is the entire source tree for each tool (Windows specific files and source 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`

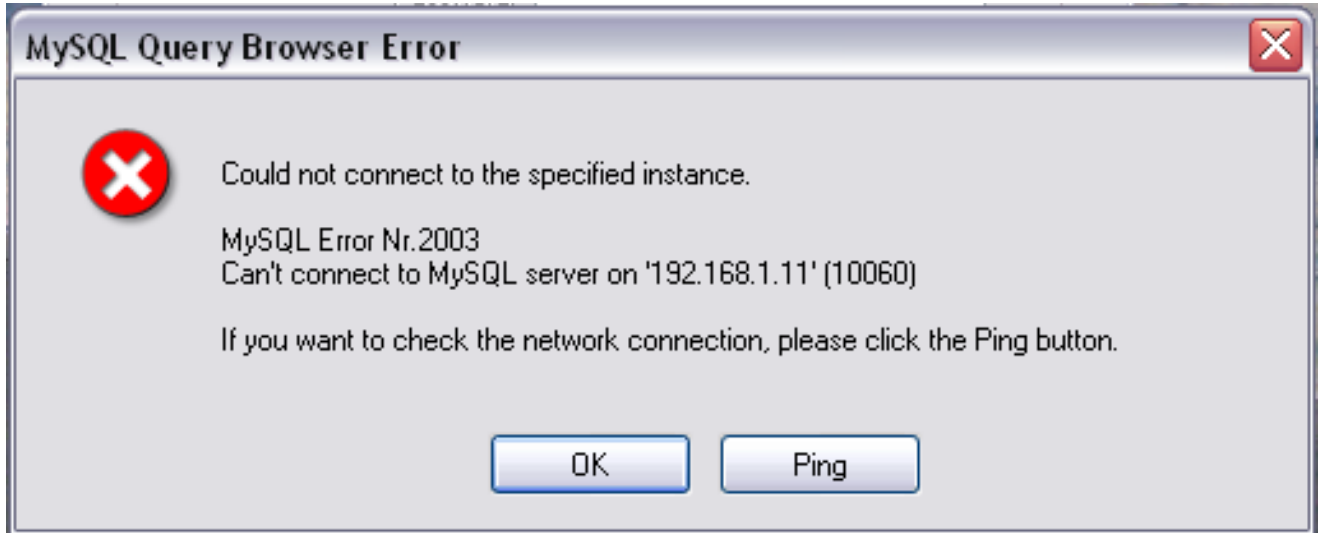
That should build the srpm and rpm files for the tool.

Appendix B. Troubleshooting Application Errors

B.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 B.1. Error Message Dialog



Normally, this means either that the credentials specified in the profile are wrong (wrong user name, wrong password), or that you do not have privileges to connect to the MySQL server from the client machine. See [Causes of Access-Denied Errors](#), for more information.

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.

B.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 C. 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>
  ...
</user_connections>
```

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.

Database administrators may want to edit the file and then copy it to another machine running the MySQL GUI tools. This makes it easy to have identical connection profiles on a number of 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 D. 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 E. XML Files Used by MySQL Query Browser

Besides the [common files](#), MySQL Query Browser uses a number of XML files for internal purposes.

- `mysqlqb_functions.xml`: Stores a list of functions and operators for use in the [Function Browser](#).
- `mysqlx_dbm_data_types.xml`: Stores a list of the available MySQL data types for use with auto-completion and with the [MySQL Table Editor](#)
- `mysqlx_dbm_charsets.xml`: Stores the available MySQL character sets.
- `mysqlqb_history.xml` : Stores the queries that make up the [History Browser](#)
- `mysqlqb_bookmark.xml`: Stores the queries that form the [Bookmark Browser](#) on Windows and Linux
- `~/Application Support/MySQL/QueryBrowser/bookmarks.xml`: Stores the queries that form the [Bookmark Browser](#) on Mac OS X

Appendix F. 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 can be contacted using <http://www.mysql.com/company/contact/>. In your message, 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?
- Where do you live?
- 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 G. 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.

G.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/>.

G.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>.