

MySQL GUI Tools Manual

MySQL GUI Tools Manual

Abstract

This manual describes all the MySQL GUI Tools.

Document generated on: 2009-06-02 (revision: 15165)

Copyright 2004-2008 MySQL AB, 2009 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 [Documentation Team](#).

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

Preface	x
1. MySQL Enterprise	1
2. Installation	2
2.1. Introduction	2
2.2. Installing Under Windows	2
2.2.1. Using the Windows Installer	2
2.2.2. Installing From the ZIP File	2
2.3. Installing Under Linux	2
2.3.1. Installing the Generic Tarball	3
2.3.2. RPM Installation	3
2.4. Installing Under Mac OS X	4
3. Removing MySQL GUI Tools	5
3.1. Uninstall – Windows	5
3.2. Uninstall – Linux	5
3.3. Uninstall – Mac OS X	5
4. Upgrading MySQL GUI Tools	6
4.1. Upgrading – Windows	6
4.2. Upgrading – Linux	6
4.2.1. Tarball Upgrade	6
4.2.2. RPM Upgrade	6
4.3. Upgrading – Mac OS X	6
5. Running MySQL GUI Tools	8
5.1. Running Under Windows	8
5.2. Running Under Linux	8
5.3. Running On Mac OS X	8
I. MySQL Administrator	9
6. Introduction to MySQL Administrator	12
7. Connection Dialog	13
7.1. Advanced Options	14
7.2. Configure-service Mode	14
8. Main Window	17
8.1. Introduction	17
8.2. Sidebar	17
8.3. Working Area	18
8.4. Menus	18
8.4.1. File	18
8.4.2. Edit	19
8.4.3. View	19
8.4.4. Tools	19
8.4.5. Help	19
9. Server Information	20
10. Service Control	21
10.1. Introduction	21
10.2. Start/Stop Service	21
10.3. Configure Service	21
11. Startup Variables	23
11.1. Introduction	23
12. Server Connections	24
12.1. Introduction	24
12.2. Threads	24
12.3. User Connections	24
13. User Administration	25
13.1. Introduction	25
13.2. User Accounts	25
13.3. User Information	26
13.4. Global Privileges	28
13.5. Schema Privileges	28
13.6. Table Column Privileges	28
13.7. Resources	29
14. Health	30
14.1. Introduction	30
14.2. Connection Health	30
14.3. Memory Health	30
14.4. Creating Custom Health Graphs	30

14.5. Status Variables	33
14.6. System Variables	33
15. Server Logs	34
15.1. Introduction	34
15.2. Error Log	35
15.3. Slow Log	35
15.4. General Log	35
16. Backup	36
16.1. Introduction	36
16.2. Backup Project	36
16.3. Advanced Options	36
16.3.1. Backup Execution Method	36
16.3.2. Output File Options	36
16.4. Schedule	37
17. Restore	38
17.1. Introduction	38
17.2. The General Tab	38
17.3. Restore Content	38
18. Replication Status	40
18.1. Introduction	40
18.2. Configuring Replication Servers	40
18.3. Using the Replication Status Section	40
19. Catalogs	42
19.1. Introduction	42
19.2. Schema Tables	42
19.2.1. Table Status	42
19.2.2. Row Status	43
19.3. Schema Indexes	43
II. MySQL Query Browser	44
20. Introduction to the MySQL Query Browser	46
21. A Tour of the MySQL Query Browser	47
21.1. The Main Query Window	47
21.2. The Query Toolbar	48
21.2.1. The Navigation Buttons	48
21.2.2. The Query Area	48
21.2.3. The Action Buttons	48
21.3. The Advanced Toolbar	49
21.4. The Result Area	49
21.5. The Object Browser	50
21.5.1. The Database Browser	50
21.5.2. The Bookmark Browser	51
21.5.3. The History Browser	51
21.6. The Information Browser	51
21.6.1. The Syntax Browser	52
21.6.2. The Function browser	52
21.6.3. The Parameter Browser	52
21.6.4. The Transaction Browser	52
21.7. The Script Editor	52
21.7.1. The Script Debugging Buttons	54
21.7.2. The Script Area	54
22. Using The MySQL Query Browser	56
22.1. Entering and Editing Queries Manually	56
22.2. Building Queries	56
22.3. Creating Views	57
22.4. Navigating Result Sets Within the Result Area	57
22.5. Exporting Result Sets	58
22.6. Handling BLOB and TEXT Columns	58
22.7. Editing Result Sets Within the Result Area	58
22.8. Comparing Result Sets	59
22.9. Managing Stored Procedures and Functions with MySQL Query Browser	59
22.10. Editing Queries From a Development Tool	60
22.11. Using the Regex Text Importer Tool	60
III. Common Administrator and Query Browser Components	61
23. Options Dialog	63
23.1. Introduction	63
23.2. General Options	63
23.3. Connections	64
23.3.1. SSL Connections	66
23.4. Editors	67
23.5. The Browser Section	67

23.5.1. Display Options	68
23.5.2. Query Options	68
23.5.3. Various	68
23.6. Administrator	69
23.6.1. User Administration	69
23.6.2. Health Graphs	69
23.6.3. Backup	69
IV. Common Administrator and Query Browser Components	70
24. The MySQL Table Editor	72
24.1. Introduction	72
24.2. The Main Editor Window	72
24.3. The Columns And Indices Tab	73
24.3.1. The Column Editor	73
24.3.2. The Detail Area	74
24.4. The Table Options Tab	75
24.5. The Advanced Options Tab	75
24.5.1. The Various Section	76
24.5.2. The Row Options Section	76
24.5.3. The Storage Options Section	76
24.5.4. The Merge Table Options Section	76
24.5.5. The Table RAID Settings Section	76
24.6. Applying Your Changes	76
V. MySQL Migration Toolkit	78
25. Introduction to the MySQL Migration Toolkit	80
26. An Overview of the Migration Process	81
27. The Migration Process In-Depth	83
27.1. Introduction	83
27.2. The Welcome Screen	83
27.3. The Configuration Type Screen	83
27.4. The Source Database Screen	84
27.4.1. Microsoft Access	84
27.4.2. Microsoft SQL Server	85
27.4.3. Oracle	86
27.4.4. MySQL	88
27.4.5. Saving Connection Information	89
27.5. The Target Database Screen	89
27.6. The Connect to Server Screen	90
27.7. The Source Schema Selection Screen	91
27.8. The Reverse Engineering Screen	92
27.9. The Object Type Selection Screen	93
27.9.1. Migrating a Sub-Set of Object Types	94
27.10. The Object Mapping Screen	95
27.10.1. GRT Object	96
27.10.2. Table Objects	96
27.11. The Meta Migration Screen	96
27.12. The Manual Editing Screen	97
27.13. The Object Creation Options Screen	99
27.14. The Creating Objects Screen	100
27.15. The Data Mapping Options Screen	101
27.16. The Bulk Data Transfer Screen	102
27.17. The Summary Screen	103
27.18. Saving the Current Application State	104
28. Scripted Migration	105
28.1. The Steps for Scripted Migration	105
28.2. Setting the Source and Target Connection	105
28.3. Reverse Engineering	106
28.4. Migration Methods	106
28.5. Map Objects and Migrate	106
28.6. The SQL Create Statements	106
28.7. Bulk Data Transfer	107
29. Extending The MySQL Migration Toolkit	108
29.1. Introduction	108
29.2. Architecture of the MySQL Migration Toolkit	108
29.3. The Modular Migration Process	108
29.4. Tools Required to Extend the MySQL Migration Toolkit	109
30. Preparing a Microsoft Access Database for Migration	110
31. The Generic Runtime Environment (GRT) Shell	112
31.1. Introduction	112
31.2. Exploring the GRT Shell	112
31.2.1. Menu Items	113

31.2.2. The Shell	114
31.2.3. The Globals Tree Panel	114
31.3. Using the GRT Shell	115
31.4. Invoking the GRT Shell From the Command Line	116
VI. Appendixes	118
A. Installing From Source	120
A.1. Introduction	120
A.2. Downloading The Source Code	120
A.3. Building from Source under Windows	120
A.3.1. Prerequisites	120
A.3.2. Compiling and Building	120
A.4. Building from Source under Linux	122
A.4.1. Prerequisites	122
A.5. Compiling and Building	123
A.5.1. Building Static Binaries	123
A.5.2. Building RPMs	123
B. Troubleshooting Application Errors	125
B.1. Troubleshooting Connection Errors	125
B.2. Troubleshooting Display Problems	125
C. How Connections Are Stored	126
D. XML Files Common to the MySQL GUI Applications	127
E. XML Files Used by MySQL Query Browser	128
F. XML Files Used by MySQL Administrator	129
G. The MySQL System Tray Monitor	131
H. Third-party Software Used by the MySQL GUI Tools	132
H.1. PCRE Library	132
H.2. PNG Support	132
I. Notes for Translators	133

List of Figures

7.1. Connection dialog	13
7.2. Switching to configure-service mode	14
7.3. Configure-service mode	15
8.1. MySQL Administrator main window	17
8.2. Marked area between sidebar and working area	18
13.1. User accounts	25
13.2. User Information tab	26
14.1. The Graph Settings dialog	31
15.1. Displaying entries from the middle of the General Log	34
17.1. The Restore Content tab	38
18.1. The Replication Status section	40
21.1. The MySQL Query Browser main query window	47
21.2. The query area expanded to ten lines	48
21.3. The MySQL Query Browser status icons	49
21.4. The advanced toolbar	49
21.5. The database browser	50
21.6. The script editor	52
21.7. The file open dialog	53
22.1. The table tool	56
22.2. The BLOB handling icons	58
22.3. Stored procedure in database browser	59
23.1. Options dialog	63
23.2. Options dialog: Connections	64
23.3. Options dialog: Advanced Parameters	66
23.4. The query browser options dialog	67
24.1. The MySQL Table Editor	72
24.2. The column editor	73
24.3. The index editor	74
24.4. The Confirm Table Edit dialog	76
26.1. The MySQL Migration Toolkit Migration Plan	81
27.1. The MySQL Migration Toolkit welcome screen	83
27.2. The Configuration Type screen	83
27.3. Source database – Microsoft Access	84
27.4. Source database – Microsoft SQL Server	85
27.5. Source database – Oracle	86
27.6. Oracle JDBC driver not attached	87
27.7. Source database – MySQL	88
27.8. Target Database – MySQL	89
27.9. The Connect to Servers screen	90
27.10. The Source Schema Selection screen	91
27.11. The Reverse Engineering screen	92
27.12. The Object Type Selection screen	93
27.13. The detail view of the Object Type Selection screen	94
27.14. The Add Ignore Pattern dialog	95
27.15. The Object Mapping screen	95
27.16. The Meta Migration screen	97
27.17. The Manual Editing screen	97
27.18. The Manual Editing screen – detailed view	98
27.19. The Object Creation Options screen	99
27.20. The Creating Objects screen	100
27.21. The Data Mapping Options screen	101
27.22. The Bulk Data Transfer screen	102
27.23. The Summary screen	103
30.1. The show section	110
30.2. The system objects	110
30.3. Granting access to the system objects	110
31.1. The GRT shell (Windows)	112
B.1. Error message dialog	125
G.1. The MySQL System Tray Monitor	131

List of Tables

A.1. Required Components	122
A.2. Requirements for MySQL Query Browser	122

Preface

The MySQL GUI Tools provide a suite of desktop applications for the administration of MySQL instances and for building and manipulating the data within MySQL databases.

This book provides a combined guide to the following tools:

- **MySQL Administrator** – provides administration tools and utilities to monitor and administer databases, users privileges and data within a MySQL instance.
- **MySQL Query Browser** – provides an interface for building and testing queries interactively.
- **MySQL Migration Toolkit** – used to migrate existing database information to MySQL (Windows only).

Note

Not all tools are available for all platforms.

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. Installation

2.1. 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.

2.2. 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.

2.2.1. 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 2.2, "Installing Under Windows"](#).

2.2.2. 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 `MySQLQueryBrowser.exe` for your desktop or the quick launch bar.

2.3. 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.

2.3.1. 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 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.

2.3.2. 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 4.2.2, “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.

2.4. 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 3. 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.

3.1. 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.

3.2. 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.

3.3. Uninstall – Mac OS X

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

Chapter 4. Upgrading MySQL GUI Tools

4.1. Upgrading – Windows

If you are upgrading using the installer file follow the instructions given in [Section 2.2, “Installing Under Windows”](#). There are 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 2.2, “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 2.2, “Installing Under Windows”](#).

4.2. Upgrading – Linux

4.2.1. 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 2.3.1, “Installing the Generic Tarball”](#).

4.2.2. 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 2.3.2, “RPM Installation”](#).

4.3. 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 2.4, “Installing Under Mac OS X”](#).

Chapter 5. Running MySQL GUI Tools

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

5.1. 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.

5.2. 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.

5.3. 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.

Part I. MySQL Administrator

Table of Contents

6. Introduction to MySQL Administrator	12
7. Connection Dialog	13
7.1. Advanced Options	14
7.2. Configure-service Mode	14
8. Main Window	17
8.1. Introduction	17
8.2. Sidebar	17
8.3. Working Area	18
8.4. Menus	18
8.4.1. File	18
8.4.2. Edit	19
8.4.3. View	19
8.4.4. Tools	19
8.4.5. Help	19
9. Server Information	20
10. Service Control	21
10.1. Introduction	21
10.2. Start/Stop Service	21
10.3. Configure Service	21
11. Startup Variables	23
11.1. Introduction	23
12. Server Connections	24
12.1. Introduction	24
12.2. Threads	24
12.3. User Connections	24
13. User Administration	25
13.1. Introduction	25
13.2. User Accounts	25
13.3. User Information	26
13.4. Global Privileges	28
13.5. Schema Privileges	28
13.6. Table Column Privileges	28
13.7. Resources	29
14. Health	30
14.1. Introduction	30
14.2. Connection Health	30
14.3. Memory Health	30
14.4. Creating Custom Health Graphs	30
14.5. Status Variables	33
14.6. System Variables	33
15. Server Logs	34
15.1. Introduction	34
15.2. Error Log	35
15.3. Slow Log	35
15.4. General Log	35
16. Backup	36
16.1. Introduction	36
16.2. Backup Project	36
16.3. Advanced Options	36
16.3.1. Backup Execution Method	36
16.3.2. Output File Options	36
16.4. Schedule	37
17. Restore	38
17.1. Introduction	38
17.2. The General Tab	38
17.3. Restore Content	38
18. Replication Status	40
18.1. Introduction	40
18.2. Configuring Replication Servers	40
18.3. Using the Replication Status Section	40
19. Catalogs	42
19.1. Introduction	42
19.2. Schema Tables	42
19.2.1. Table Status	42

19.2.2. Row Status	43
19.3. Schema Indexes	43

Chapter 6. Introduction to MySQL Administrator

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

Most tasks can be performed using a command-line client such as [mysqladmin](#), or [mysql](#), but MySQL Administrator has the following advantages:

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

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

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

Chapter 7. Connection Dialog

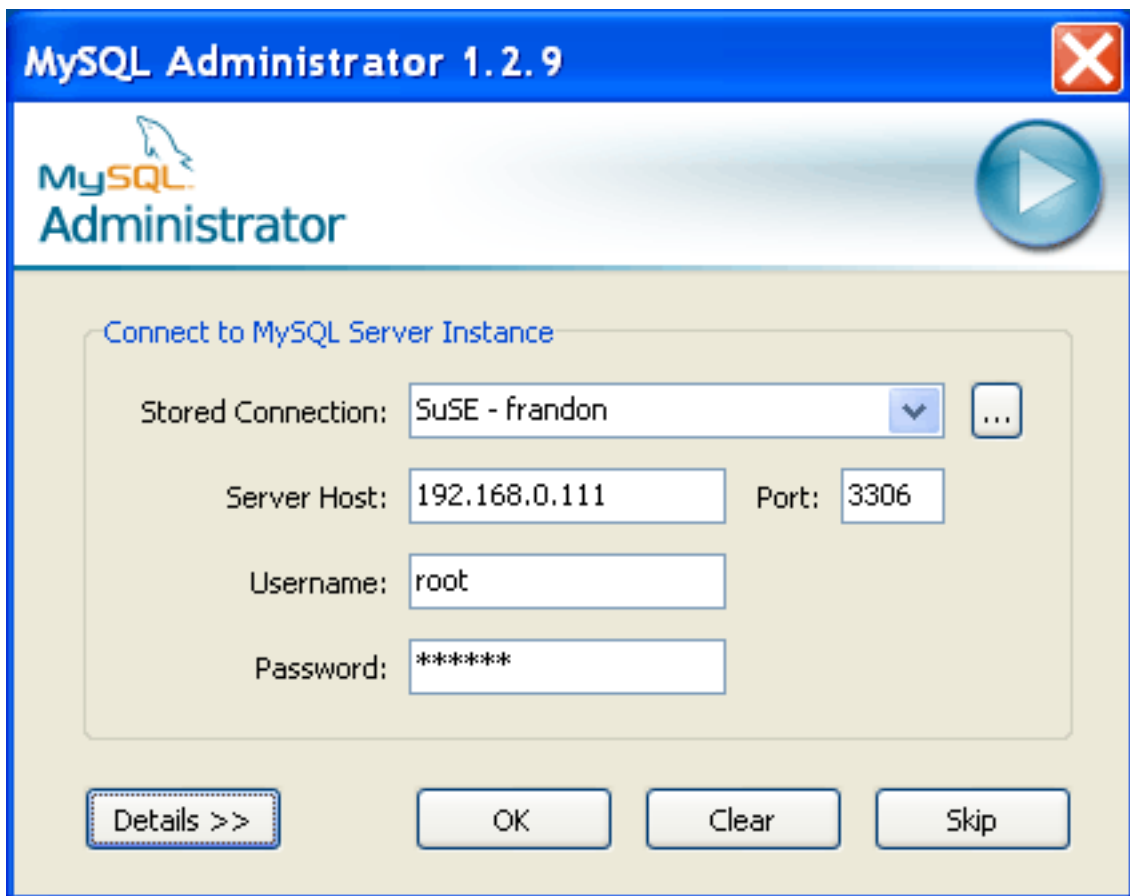
Once MySQL Administrator has been started, it displays a connection dialog box. Specify the MySQL server you would like to connect to, the necessary credentials, and both the host and port. You may also specify a number of additional options, if required.

For example, to connect to a newly installed server running on the same machine as the administrative (root) user you would use the host `localhost`, the user `root` and the password that you sent during installation (or blank if you haven't set one).

Note

As an alternative to connecting to a running MySQL server, you can run MySQL Administrator in Configure-service mode. For information on this topic see [Section 7.2, “Configure-service Mode”](#).

Figure 7.1. Connection dialog



Note that MySQL Administrator will not work with MySQL servers prior to version 4.0. If you try to connect to an older server, an error message will be displayed.

If the server connection is successfully established, the values in the fields of the connection dialog will be saved for future connections. For more information on this topic see [Appendix C, *How Connections Are Stored*](#). However, for security reasons the password is not stored along with the other options unless you explicitly specify otherwise. For instructions on storing the password see [Section 23.2, “General Options”](#).

You can change any of the values in the connection profiles by overwriting the existing values with new ones. Similarly, if you select another connection profile, you can change any of its values. If a successful connection to the MySQL server is established, clicking the OK button after changing a profile's values, permanently saves the changes. However, the preferred method for changing a profile is to click the ... button next to the CONNECTION drop-down box. That brings up an OPTIONS dialog window that has a CONNECTIONS section for modifying connection profiles. You can find the values you can set in the Connection dialog window in the description of the CONNECTIONS section of the Options dialog (see [Section 23.3, “Connections”](#)).

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 HOST**: 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 the following section.

7.1. Advanced Options

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 a 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.

7.2. Configure-service Mode

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

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

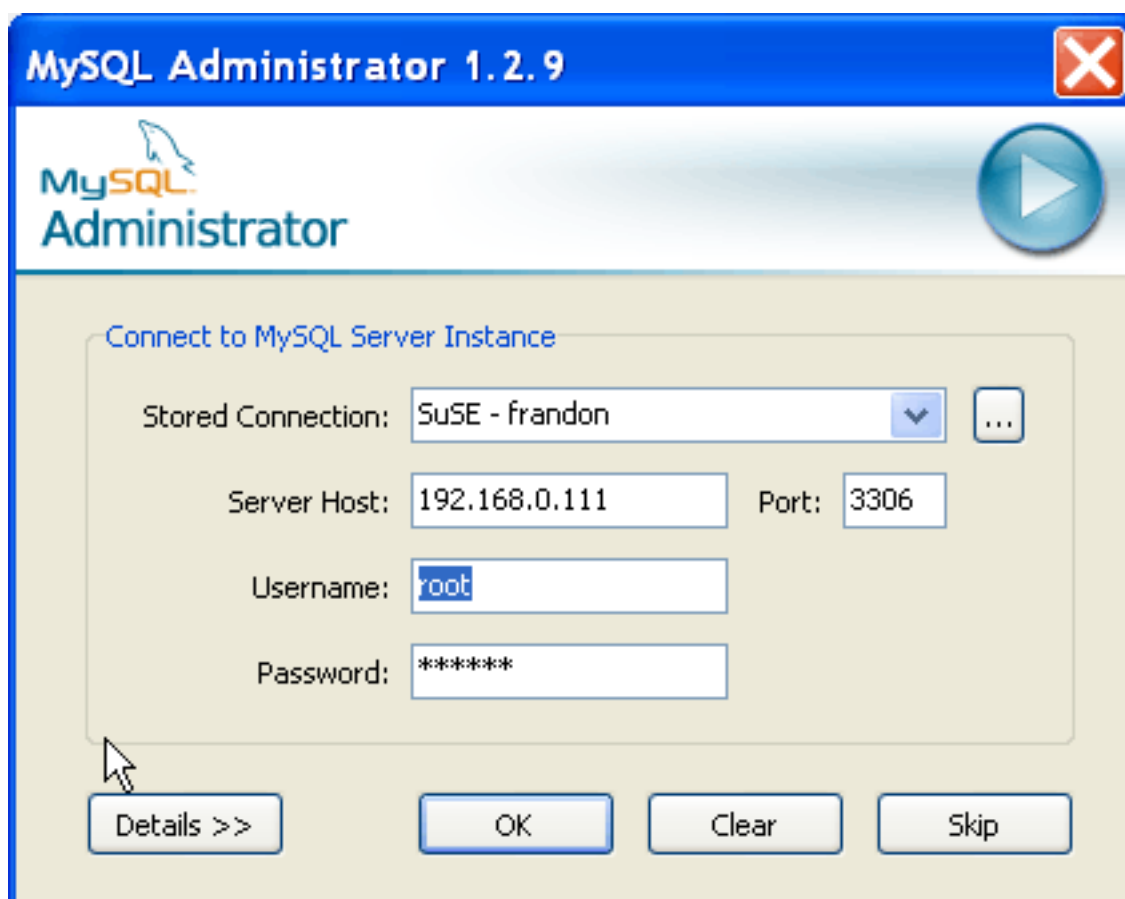
- When you want to start a MySQL server from within MySQL Administrator. Obviously, you cannot connect to a server that is not running, so you will have to use the configure-service mode to start that server. Note that you can only start servers on the same machine where MySQL Administrator is running (that is on the local host).
- When you have multiple servers running on your machine. When running in normal mode on Windows, MySQL Administrator only allows you to configure the Windows service that is listed first in the Microsoft Management Console window. This means that if you have more than one Windows service, you can only configure the first one of those, no matter which server you connect to. If you want to configure a service other than the first service, you will have to run MySQL Administrator in configure-service mode. For more information on this topic see [Running Multiple MySQL Servers on the Same Machine](#).

Note

The preceding applies only to MySQL Administrator when running on Windows.

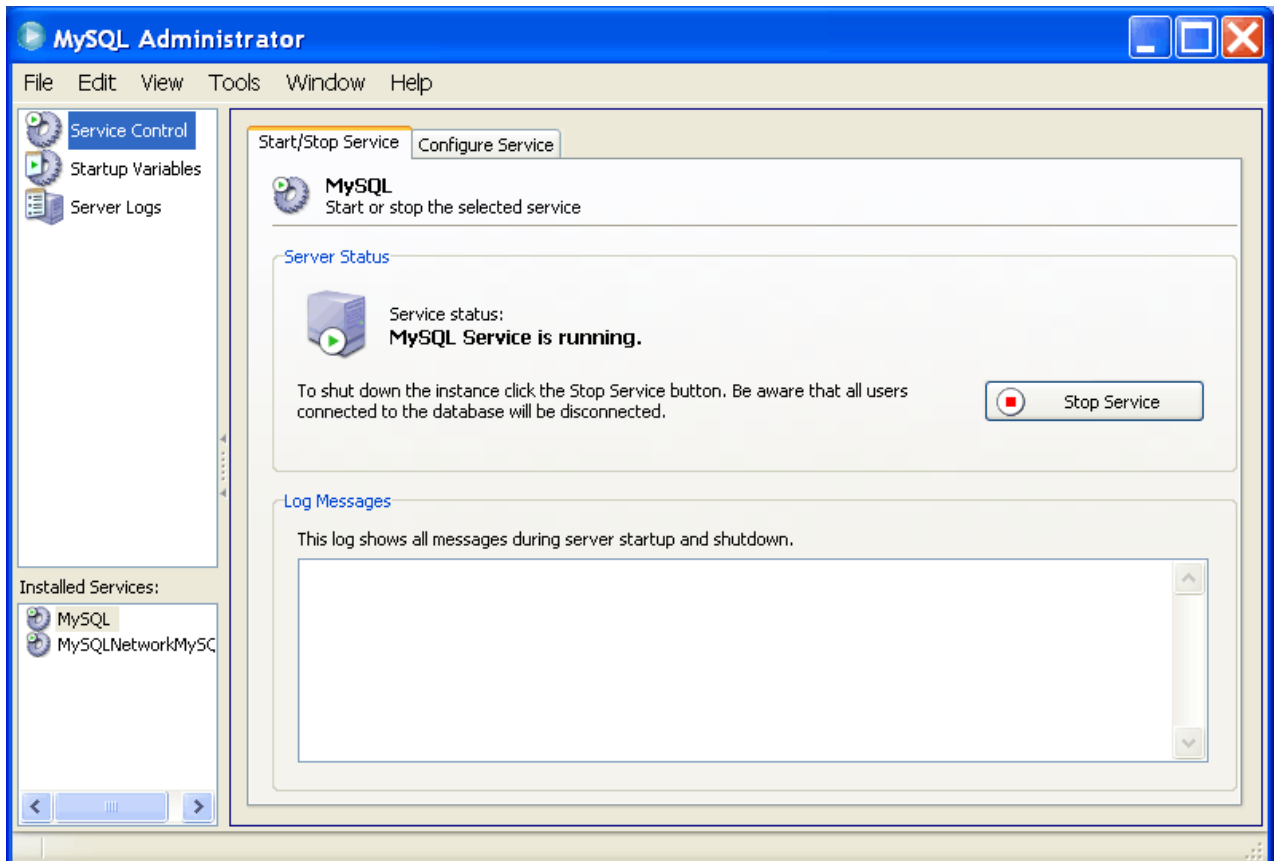
To start MySQL Administrator in configure-service mode, hold down the **CTRL** key in the [Connection](#) dialog. The CANCEL button will toggle to SKIP when the **CTRL** key is pressed.

Figure 7.2. Switching to configure-service mode



Clicking on the SKIP button will skip normal mode and start MySQL Administrator in configure-service mode. You should see something similar to the following.

Figure 7.3. Configure-service mode



Under **INSTALLED SERVICES** in the [sidebar](#), you can select the service that you wish to configure, or the server that you want to stop or start. Only servers that have been installed using the Windows installer will appear here.

Note

INSTALLED SERVICES is only displayed under Windows.

In configure-service mode, only the following sidebar options are available:

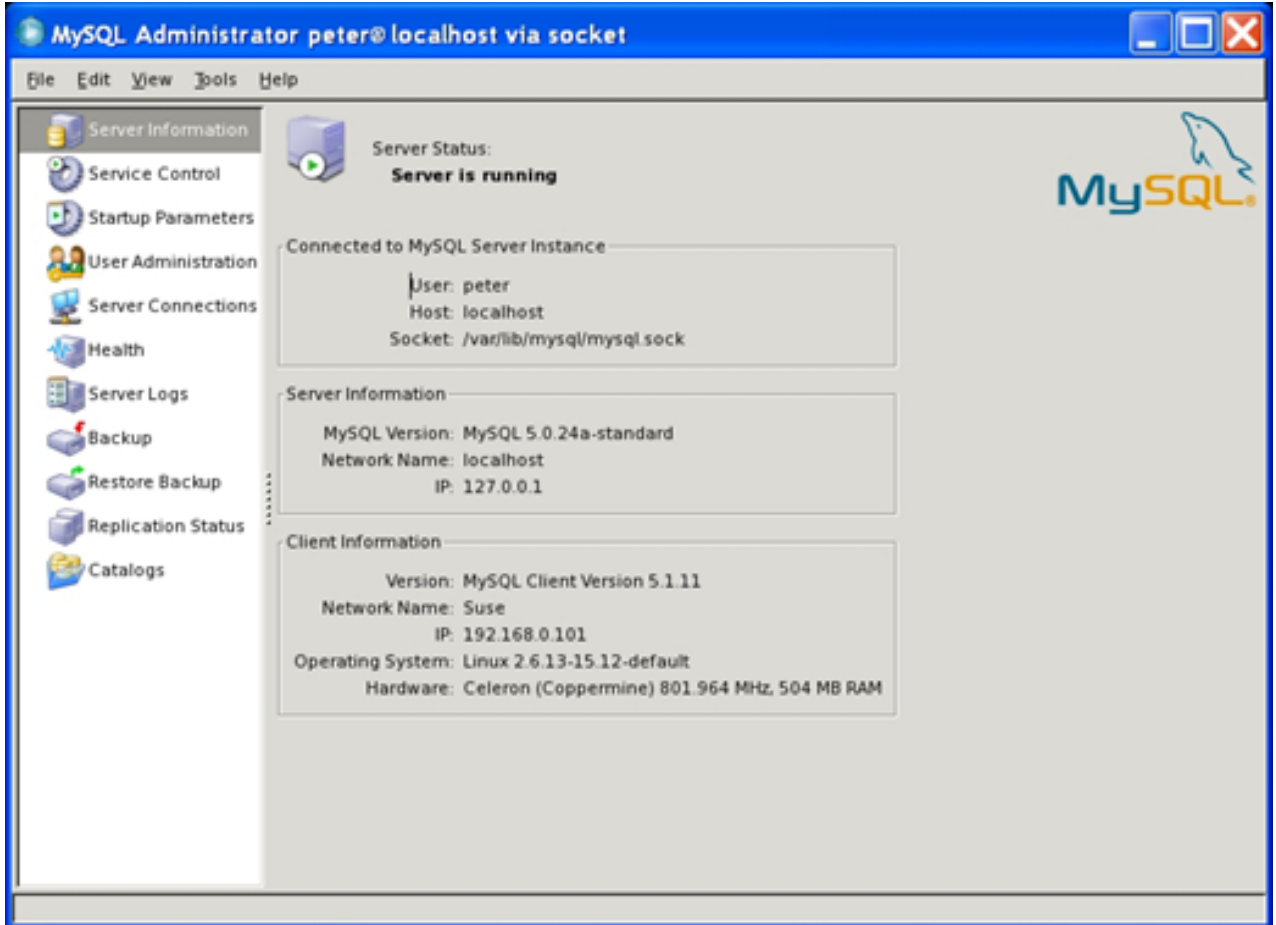
- The [Service Control](#) section allows you to start and stop a MySQL server, and to configure the Windows service managing that server. Servers are stopped using the `mysqladmin` utility. This program is part of the default MySQL installation but if you have removed it you will not be able to stop a server.
- The [Startup Variables](#) section allows you to set variables for server startup.
- The [Server Logs](#) section allows you to examine the server logs. Note: You do not need to connect in configure-service mode to examine the server logs. This option is available when you connect to a specific server instance.

Chapter 8. Main Window

8.1. Introduction

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

Figure 8.1. MySQL Administrator main window



8.2. Sidebar

The sidebar consists of the following options:

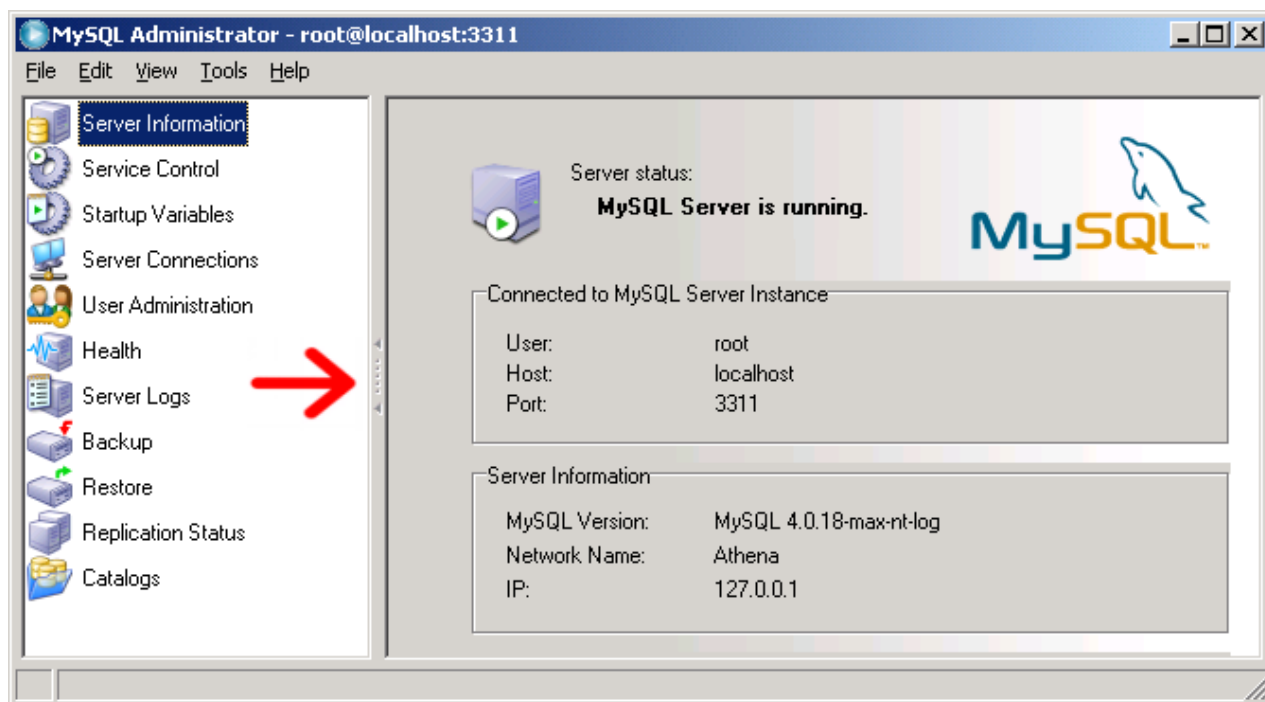
- **Server Information:** Information about the MySQL server you are connected to, about MySQL Administrator and the machine that it runs on, and about your connection.
- **Service Control:** Start and stop the MySQL server. If MySQL Administrator is running on Windows and connected to a MySQL server that is also running on Windows, you can configure the MySQL service.
- **Startup Variables:** Configure the startup variables for the MySQL server.
- **Server Connections:** View or kill current connections to the MySQL server (also called threads).
- **User Administration:** Administer existing users, add new users, and delete existing users.
- **Health:** Graphical displays of a number of usage or hit-rate values that affect server performance, and a hierarchical view of system and status variables.
- **Server Logs:** View log file entries.

- **Backup:** Plan and administer backup projects, select the databases to be backed up, and start the backup process.
- **Restore:** Restore databases from existing backups.
- **Replication Status:** View information about replication master and slave servers.
- **Catalogs:** View information about databases, tables, columns, indexes, and rows; optimize tables.

The following sections describe each of these in detail.

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

Figure 8.2. Marked area between sidebar and working area



8.3. Working Area

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

8.4. Menus

8.4.1. File

8.4.1.1. New Instance Connection ...

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

8.4.1.2. Reconnect

This option closes the current connection and opens the [Options dialog](#), with the [Connections section](#) preselected.

8.4.1.3. Copy Active Page as Text ...

This menu option copies the information displayed on the [Server Information](#) page to the clipboard. For more information about the [Server Information](#) page see [Chapter 9, Server Information](#).

Note

This menu option is only available on Windows and only copies the [Server Information](#) page.

8.4.1.4. Save current Connection ...

Opens the [Options dialog](#), with the [Connections section](#) preselected. Furthermore, a new connection profile labeled **NEW CONNECTION** is created, and it is also preselected. The fields are filled with the values of the most recent connection. You can edit the values and save the settings in the new profile. You may also want to rename the profile from **NEW CONNECTION** to something more appropriate. If you click the **CLOSE** button without saving your changes using the **APPLY CHANGES** button, you will be prompted to save or discard your changes.

8.4.1.5. Close

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

8.4.2. Edit

The [CUT](#), [COPY](#), and [PASTE](#) commands allow you to cut, copy, or paste selected text in the [working area](#).

8.4.2.1. Options

Opens the [Options dialog](#) window.

8.4.3. View

The [VIEW](#) menu selects sections from the [sidebar](#). Selecting an item from this menu is the same as clicking the section with the same name from the sidebar.

8.4.4. Tools

The [TOOLS](#) menu starts other programs.

8.4.4.1. MySQL Command Line Client

This menu item is available only if MySQL Administrator finds the executable file for the MySQL client. At the moment, it searches locations such as `C:\mysql\bin`, `D:\mysql\bin`, and `C:\program directory\mysql\bin` to find `mysql.exe`. If the executable file is found, you can start the command-line client with this command. MySQL Administrator will use the credentials provided in the [Connection dialog](#) to start a server connection using the command-line client.

8.4.4.2. Windows Command Line

Opens a DOS window.

8.4.5. Help

8.4.5.1. Help

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

8.4.5.2. Report Bug to MySQL

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

8.4.5.3. Launch MySQL Website

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

8.4.5.4. About ...

Displays information about MySQL Administrator.

Chapter 9. Server Information

Provides basic information about connection, server, and client.

- Connected to MySQL Server Instance

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

- **USERNAME:** Username used for the current connection to the MySQL server.
- **HOSTNAME:** The name of the machine hosting the MySQL server. This is either identical to the name listed in **NETWORK NAME**, or it is `localhost`. The former indicates that you are connecting to a server that either runs on a remote machine, or that you have specified a host name different from `localhost` when connecting. The latter indicates that the MySQL server runs on the same machine where MySQL Administrator runs.
- **PORT:** The port used for connecting to the MySQL server.

- Server Information

Information about the MySQL server and its host.

- **MYSQL VERSION:** The version of the MySQL server.
- **NETWORK NAME:** The name of the machine hosting the MySQL server.
- **IP:** The IP number of the MySQL server host.

- Client Information

Information about MySQL Administrator and the host client.

- **VERSION:** The version of MySQL Administrator, and the version number of the MySQL client/server protocol.
- **NETWORK NAME:** The name of the host where MySQL Administrator runs.
- **IP:** The IP number of the MySQL Administrator host.
- **OPERATING SYSTEM:** Information about the operating system on which MySQL Administrator runs.
- **HARDWARE:** Information about the hardware on which MySQL Administrator runs.

Chapter 10. Service Control

10.1. Introduction

This section shows how to start and stop the MySQL server you are connected to. Currently, this is only possible if the server is running on the same machine that hosts MySQL Administrator. You cannot start and stop MySQL servers on remote machines.

10.2. Start/Stop Service

The **START/STOP SERVICE** tab displays the status of the MySQL service. This service starts and stops the MySQL server. Under **SERVICE STATUS**, a text message indicates whether or not the service has been started. If the service is running, the button label is **STOP SERVICE**. Click it to stop the server. If the service is stopped, the button label is **START SERVICE**. Click it to start the service.

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

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

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

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

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

10.3. Configure Service

Use this tab to configure the Windows service that starts and stops the MySQL server and to set server options. This tab is specific to MySQL servers running under operating systems of the Windows NT family, including Windows 2000, and Windows XP; it is not available for servers running under a different operating system. Obviously, you need to have a Windows service installed before you can configure it. See [Starting MySQL as a Windows Service](#), for more information on how to set up a Windows service.

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

Note that the settings in the **SERVER FEATURES** group take effect only after restarting the MySQL server.

The following settings are available:

- Service Settings
 - **LAUNCH MYSQL SERVER AUTOMATICALLY:** If checked, the Windows service starts the MySQL server automatically on operating system startup.
 - **DISPLAY NAME:** The display name of the Windows service (not the service name!) as it will appear in the Windows services manager when displaying the properties of that service.
 - **SERVICE DESCRIPTION:** The description of the Windows service as it will appear in the Windows services manager.
- Option File
 - **CONFIG FILENAME:** You can enter the path name of the server option file. MySQL will enter that information in the Windows registry, and the Windows service starting the MySQL server will use that file, rather than the default file (`C:\windows\installation\directory\my.ini`). *Note that this is available for MySQL servers as of version 4.0.3 only.* See [Creating an Option File](#), for details about the server option files.
 - **SECTION NAME:** Setting a value other than `mysqld` is most useful when running multiple servers; see [Running Multiple](#)

[Servers on Windows.](#)

- Server Features

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

- **SUPPORT FOR INNODB:** If checked, the MySQL server is started with support for InnoDB tables. If unchecked, it doesn't start the InnoDB storage engine; this saves memory, but InnoDB tables cannot be used. If this checkbox is unchecked and grayed out, it means that your MySQL server isn't compiled with support for InnoDB tables. (This is, for example, true for some versions that run under a commercial license.) For more information on InnoDB tables, see [Storage Engines](#).

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

- **SUPPORT FOR BDB:** If checked, the MySQL server is started with support for BDB tables. If unchecked, it doesn't start the BDB storage engine; this saves memory, but BDB tables cannot be used. If this checkbox is unchecked and grayed out, it means that your MySQL server isn't compiled with support for BDB tables. (This is, for example, true for non-max versions.) For more information on BDB tables, see [Storage Engines](#).

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

- **NAMED PIPES:** On operating systems of the Windows NT family, including Windows 2000, and Windows XP, you may use named pipes to connect to a MySQL server running on the local machine, rather than connecting via TCP/IP. If this option is checked, named pipe connections are allowed. Note that named pipes are slower than TCP/IP in many Windows configurations. See [Selecting a MySQL Server Type](#).
- **DEBUG INFORMATION (SLOW):** If checked, the MySQL server is started with an option that allows for easier debugging of problems. Using this option slows down the server significantly, so it is recommended only for debugging purposes. If the MySQL server has not been compiled with debugging support, this option is grayed out. See [Server Command Options](#).
- **PATH TO BINARY:** Contains the path name of the MySQL server binary file as it is listed in the Windows service manager (for example `C:\mysql\bin\mysqld-max-nt`). Note that the Windows service manager does not display the `.exe` extension.

Chapter 11. Startup Variables

11.1. Introduction

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

Note that the server reads variables stored in the option files only at server startup. After changing an option file, you must restart the server for the changes to take effect. You can do this using the STOP SERVER and START SERVER buttons in the [Service Control](#) section.

Chapter 12. Server Connections

12.1. Introduction

This section allows you to view and manage the current connections to the server (also called threads). If you have the `PROCESS` privilege, you can see all threads. If you have the `SUPER` privilege, you can also terminate threads other than your own. (Note that the `SUPER` privilege is available for MySQL servers as of version 4.0.3 only.) Otherwise, you can see your own threads only. See [SHOW PROCESSLIST Syntax](#), for more information. See [Privileges Provided by MySQL](#), for more information on the `PROCESS` and the `SUPER` privilege (and other available privileges as well).

Note

When you click the [Server Connections](#) section You will see an additional thread for the user you logged in as because the values displayed are retrieved from the MySQL server via an additional separate connection.

12.2. Threads

This tab allows you to view all connections to the server (threads). By default, threads are sorted by their `PID` (MySQL process identification number), but you can change that order by clicking on the appropriate headings (`USER`, `HOST`, etc.). You can select one or more threads using the mouse or the keyboard.

To reload the process list from the server, press the `REFRESH` button.

If you have selected one or more threads, you can terminate them by clicking the `KILL THREAD` button, or by right-clicking on that thread. Remember that you need the `PROCESS` privilege to see threads other than your own, and the `SUPER` privilege to kill threads other than your own. Furthermore, note that MySQL Administrator imposes a restriction that prevents you from terminating your own threads. On Windows, using the `KILL THREAD` button on your own threads has no effect; on Linux, that button is disabled when selecting your own threads.

12.3. User Connections

This tab displays all connections to the server (threads), but in a different way than the [Threads](#) tab does: You will see users rather than individual threads. The number of connections for each user is listed under `NUM.` Right-clicking on a user name and selecting [SHOW USER INFOS](#) displays the `FULL NAME` and the `DESCRIPTION`, as specified in the `USER INFORMATION` tab of the [User Administration](#) section. Initially, users are sorted by their MySQL `Username`, but you can change the order by clicking on the headings (`NUM.`, `FULL NAME`, or `DESCRIPTION`). Clicking on a user name will display the individual threads of that user in the lower box of the tab, ordered by `PID`. Likewise, you can change the order by clicking on one of the headings.

Clicking on a user will enable the `KILL USER` button, allowing you to terminate all threads of that user with a single click. This command is also available by right clicking on the user name. Clicking on an individual thread in the lower box enables the `KILL THREAD` button, which is used to terminate a connection. This command is also available by right clicking on a thread.

To reload the user list from the server, press the `REFRESH` button.

Chapter 13. User Administration

13.1. Introduction

This section shows how to administer existing users, add new users, and delete existing users. For more information on the MySQL user account management and privilege system, see [MySQL User Account Management](#).

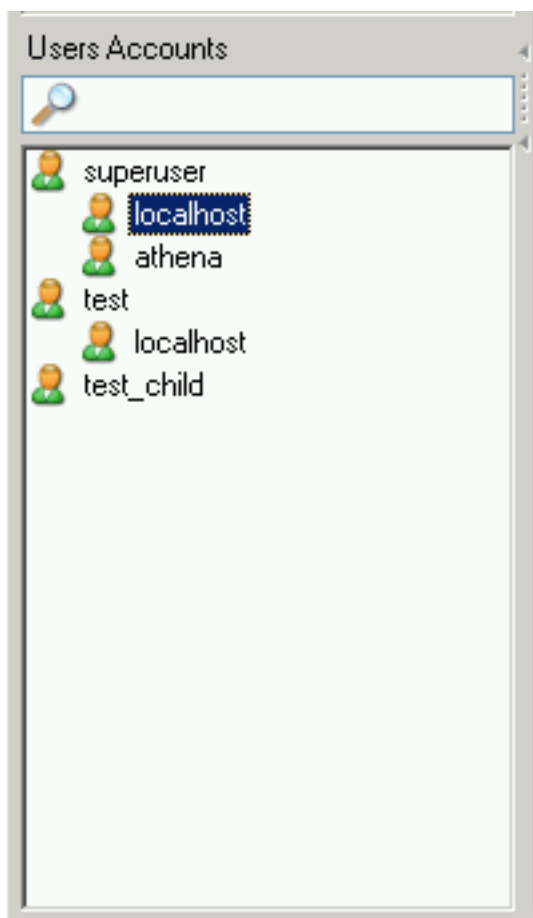
13.2. User Accounts

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

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

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

Figure 13.1. User accounts



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

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

- **ADD A NEW USER:** Selecting this command adds a new user with the default name `New User` to the user list. You will have to fill in at least a user name in the **MYSQL USER** field of the **User Information** tab, which, after applying this change, will also rename that user in the user accounts list.
- **ADD HOST FROM WHICH THE USER CAN CONNECT:** Allows you to specify a host from which the user can connect. This command is unavailable if you highlight a subcategory.
- **REMOVE HOST FROM WHICH THE USER CAN CONNECT:** Allows you to remove a host from which the user can connect. If there are no sub-categories, the user will be removed (you will be prompted to confirm the removal).
- **CLONE USER:** Makes a copy of the selected user, including all subcategories. The new user is called `New User` until you specify another name in the **MYSQL USER** field of the **User Information** tab. This command is useful for creating users with identical privileges.

Note

Cloning is only available on Windows.

- **DELETE USER:** Allows you to delete the selected user. You will be prompted to confirm that you really want to delete that user.
- **REFRESH USER LIST:** Rereads the user list from the MySQL server. This is helpful when other users (on other connections) are editing user accounts. Note that selecting that command will collapse all subcategories.

13.3. User Information

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

Note that all fields will be stored only when you click the **APPLY CHANGES** button. If you don't want to save your changes, click the **DISCARD CHANGES** button.

Figure 13.2. User Information tab

MySQL Administrator - superuser@localhost:3306

File Edit View Tools Extras Windows Help

User Information Global Privileges Schema Privileges Table/Column Privileges Resources

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

Login Information

MySQL User: The user has to enter this MySQL User name to connect to the MySQL Server

Password: Fill out this field if you want to set the user's password

Confirm password: Again, enter the users password to confirm

Additional Information

Full name: The user's full name

Description: Additional description of the user

Email: The user's email address

Contact Information: Optional contact information

Icon: Icon assign to the user

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

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

- **Login Information**
 - **MYSQL USER:** User name that is specified when connecting to the MySQL server. Note that this user name has nothing to do with user names you might use elsewhere on your operating system. Using special characters or spaces for a user name is not recommended.
 - **PASSWORD:** Password that is specified when connecting to the MySQL server. Note that this password has nothing to do with passwords you might use elsewhere on your operating system. Again, don't use special characters or spaces for the password. It is, however, recommended that you use non-trivial passwords. A very trivial password might be `abcde`, while a non-trivial password might be `lEtusMak3iThartdoGesZ`. Note that it is not mandatory to specify a password, although it is strongly recommended that you do so.

- **CONFIRM PASSWORD:** Fill in the password once again to make sure you do not accidentally introduce a typo.
- Additional Information

All information specified here is optional.

- **FULL NAME:** As opposed to the user name in **MYSQL USER**, the full name may contain any characters, including spaces.
- **DESCRIPTION:** Additional description of user.
- **EMAIL:** Email address of user.
- **CONTACT INFORMATION:** More contact information, like postal address, or messenger information.
- **ICON:** Rather than using the default icon, you may select a different icon by pressing the **LOAD FROM DISK** button. The icon you use must be exactly 48 by 48 pixels in size, and its file format must be **PNG** (portable network graphics).

13.4. Global Privileges

This tab is available only if the **SHOW GLOBAL PRIVILEGES** checkbox in the [Administrator section of the Options dialog](#) has been checked.

Under **ASSIGNED PRIVILEGES**, you find the global privileges assigned to the selected user. For more information about those privileges, see [Privileges Provided by MySQL](#). *Global* means that those privileges apply to the MySQL server in general (like the [Shutdown](#) privilege), or to all databases on the server (like the [SELECT](#) privilege).

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

13.5. Schema Privileges

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

You may use wild cards to grant privileges on a number of schemata simultaneously. Use “%” to replace multiple characters and “_” to replace a single character. To use this feature, right click any database name in the schemata list, choose the [ADD SCHEMA WITH WILDCARDS](#) option from the pop-up menu and enter the pattern you wish to match. Entering `m%` for example, will add the entry, `m%` to the schemata list. Highlight this entry and any privileges granted will be granted for all schemata starting with the letter “m”.

Note

This pop-up menu option is only available under Windows.

If wild card characters appear in schemata names, you may escape them using the “\” character.

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

The **SCHEMA PRIVILEGES** tab only displays privileges that have been granted for a specific schema. No schema privileges will show in the **ASSIGNED PRIVILEGES** column for users with global privileges. For example, the **ASSIGNED PRIVILEGES** column will be blank for a `root` user with global privileges on all schemata. Likewise, the **ASSIGNED PRIVILEGES** column will be blank for a user with a global [SELECT](#) privilege, unless [SELECT](#) has also specifically been granted for the currently selected schema.

13.6. Table Column Privileges

This tab is available only if the **SHOW SCHEMA OBJECT PRIVILEGES** checkbox in the [Administrator section of the Options dialog](#) has been checked. It allows you to grant privileges on a table level or on a column level.

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

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

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

13.7. Resources

This tab allows you to limit a user's resources. You may set limits for the following items:

- **MAX_QUESTIONS** – The number of questions allowed per hour
- **MAX_UPDATE** – The number of updates allowed per hour
- **MAX_CONNECTIONS** – The number of connections allowed per hour
- **MAX_USER_CONNECTIONS** – The number of simultaneous connections allowed

Note

This option is only available under Windows

The default value for each of these settings is 0, indicating no restrictions whatsoever.

Chapter 14. Health

14.1. Introduction

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

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

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

14.2. Connection Health

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

- **CONNECTION USAGE:** Displays a percentage of the maximum number of simultaneous clients allowed ([threads connected](#) / [max_connections](#)). If this value gets close to 100%, it could indicate that you should increase the value of [max_connections](#), which you can do in the [Startup Variables](#) section. The maximum number of connections since MySQL Administrator was loaded is also shown. For more information on server connections, see [SHOW VARIABLES Syntax](#), and [Too many connections](#).
- **TRAFFIC:** This is a representation of the [bytes_sent](#) status variable. This does not represent total bytes sent since server startup, but instead represents the current number of bytes sent (the [bytes_sent](#) variable is a cumulative total). The initial maximum value of the graph is 102400, this maximum value will be automatically increased as needed.
- **NUMBER OF SQL QUERIES:** This is a representation of the [com_select](#) status variable. This does not represent total queries since server startup, but instead represents the current number of queries (the [com_select](#) variable is a cumulative total). The initial maximum value of the graph is 10, this maximum value will be automatically increased as needed.

14.3. Memory Health

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

- **QUERY CACHE HITRATE:** When in use, the query cache of the MySQL server stores the text of a [SELECT](#) query together with the corresponding result that was sent to the client. If an identical query is later received, the server will retrieve the results from the query cache rather than parsing and executing the same query again. The hit rate indicates the ratio of queries that were cached and queries that were not cached. The higher the hit rate, the better the performance of the server regarding [SELECT](#) queries. See [The MySQL Query Cache](#).
- **KEY EFFICIENCY:** The horizontal bar graph indicates how much of the allocated key buffer is in use. If a fairly large percentage of the key buffer is in use you may wish to increase the [key_cache_size](#) setting in the [Startup Variables](#) section. The line graph is an indication of the number of [key_read_requests](#) that resulted in actual [key_reads](#). The higher the hit rate the more effective your key buffer is. A low hit rate can indicate that you need to increase the size of your keycache.

14.4. Creating Custom Health Graphs

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

The first step in creating custom health graphs is to create a new page in the [Health](#) section to store your graphs. To create a new page right click anywhere in the working area and choose the [ADD A PAGE](#) option. You will be prompted for a page name and description and a new page will be created.

Once you have created a new page, you will need to create a group. All graphs are organized into groups. For example, the [Key buffer usage](#) and [Key buffer hit rate](#) graphs are both members of the group titled [Key Efficiency](#). To create a group, right click within your newly created page and choose the [ADD A GROUP](#) option.

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

Figure 14.1. The Graph Settings dialog

Graph Settings ✕

General

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

Caption: The graph's caption.

Display Caption Whether the captions should be displayed or not.

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

Value Settings

Value Formula:

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

Value Unit: Percentage ▼ The unit of the value.

Value Caption: A caption for the value.

Min. / Max. Values

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

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

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

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

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

Apply
Cancel

The **LINE-GRAPH** is appropriate for showing trends over time, while the **BAR-GRAPH** option will be more appropriate for showing

percentage information. If you choose to create a bar-graph, you will need to choose a caption for your graph. Captions are not allowed for line-graphs.

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

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

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

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

```
^[created_tmp_tables]
```

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

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

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

14.5. Status Variables

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

For each category, and each subcategory, the status variables are listed in the right box of the working area, together with their values, and with a short explanation. See **SHOW STATUS Syntax**, for more detail on server status variables.

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

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

14.6. System Variables

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

For each category, and each subcategory, the system variables are listed in the right box of the working area, together with their values and a short explanation. See **SHOW VARIABLES Syntax**, for more information on system variables.

System variables that can be set at runtime are marked as editable, that is they are prefixed with an icon different from the variables that cannot be set at runtime. Double clicking on editable variables will open a dialog box allowing you to specify another value for the variable. The changes you make will take effect as soon as you confirm the changes by clicking the **OK** button. Click **ABORT** if you want to keep the current value. Changes made in this manner will only persist until the server is restarted, after which the server will revert to its startup settings. In MySQL Administrator, system variables are set on a global level only. Note that you need appropriate privileges to change variables globally. See **SET Syntax**.

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

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

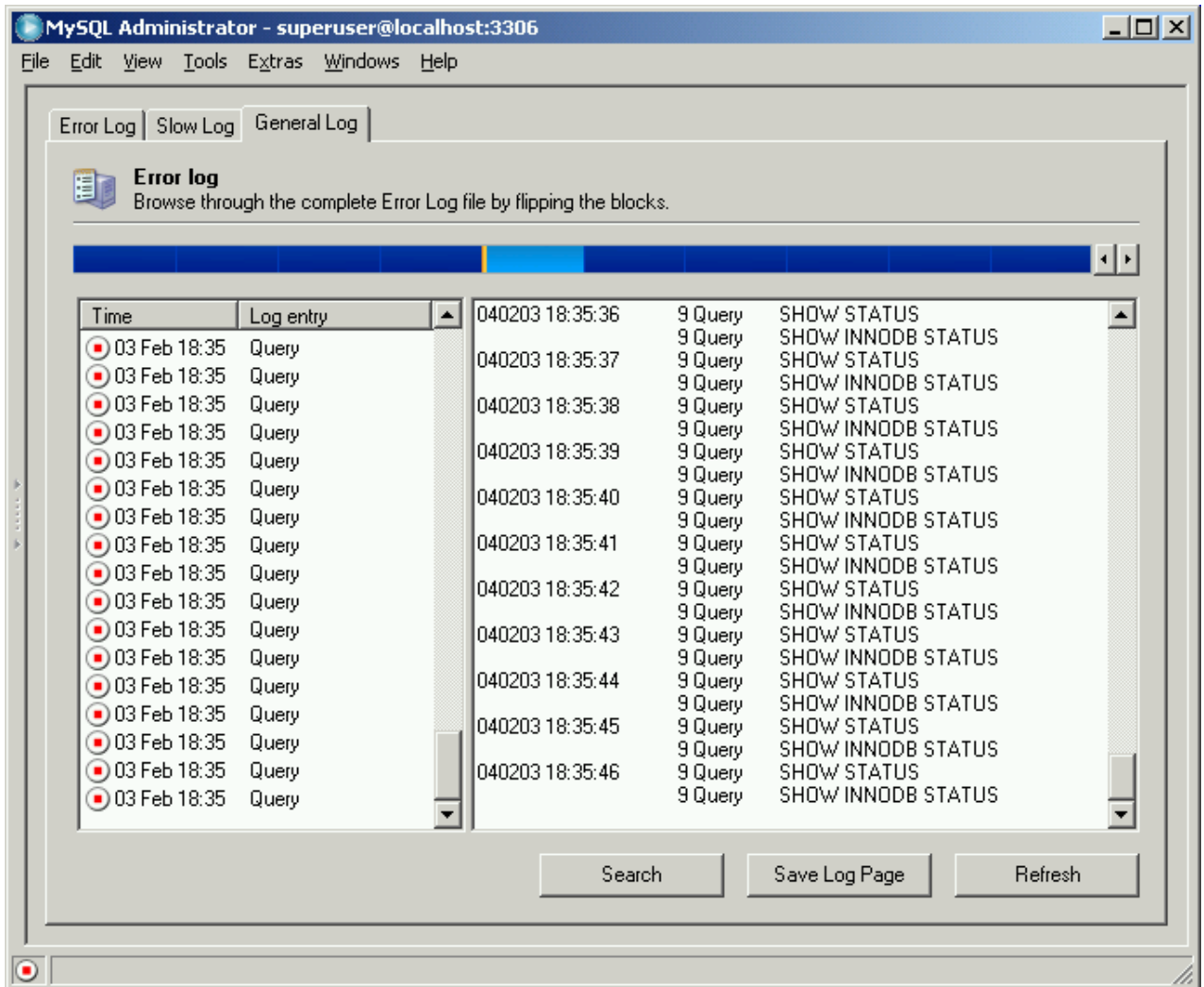
Chapter 15. Server Logs

15.1. Introduction

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

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

Figure 15.1. Displaying entries from the middle of the General Log



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

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

The left box (labeled **EVENTS OF PAGE x / y**) displays the major events that have occurred on the current page of the log file. The right box (labeled **PAGE CONTENT**) displays the detailed lines of the log file. By clicking on an event in the left box, the first detailed entry of the event will be highlighted in the right box.

In addition to the log file tabs, the [Server Logs](#) section has four action buttons for managing log information:

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

15.2. Error Log

Displays the MySQL server's error log. The error log file contains information indicating when the MySQL server was started and stopped and also any critical errors found when running. See [The Error Log](#), for more information about that log file.

15.3. Slow Log

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

15.4. General Log

Displays the MySQL server's general query log. This log can be very useful when you suspect an error in a client and want to know exactly what the MySQL server thought the client sent to it. For more information about that log file, see [The General Query Log](#).

Chapter 16. Backup

16.1. Introduction

This section shows you how to create backup projects and to perform backups. This section will produce backup files that are plain text and composed of multiple SQL statements that can be executed to restore your database, similar to the output generated by the `mysqldump` utility. For more information on `mysqldump`, see `mysqldump`, in the reference manual.

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

You can update the list of backup projects by right clicking on a project and choosing the [REFRESH BACKUP PROJECTS](#) option from the drop-down menu. You can delete a backup project by right clicking on the project and choosing the [DELETE BACKUP PROJECT\(S\)](#) option from the drop-down menu.

If you are not working from a previously saved backup project, click the [NEW PROJECT](#) button to create a new backup project. You cannot begin configuring a backup operation without either loading an existing project or creating a new project.

Once you have configured your backup project, you can click the [SAVE PROJECT](#) button to save your project for future use. To start your backup, click the [EXECUTE BACKUP NOW](#) button. You will be prompted for a path and file name for the backup file, after which the backup operation will begin.

16.2. Backup Project

The [BACKUP PROJECT](#) tab allows you to set the name of your backup project and also determine what data will be backed up.

You can name your project by filling out a name in the [PROJECT NAME](#) field. If you have enabled the [ADD DATE/TIME TO BACKUP FILES](#) option in the [Administrator Options](#) dialog then the date and time of the backup will be appended to the project name automatically. See [Section 23.6.3, “Backup”](#) for more information on setting this option.

Specify a name that clearly identifies the project, such as [Full Backup](#) or [World Only](#). The project name may contain arbitrary characters, including spaces.

To choose the databases and tables that will be backed up, select the database from the schemata list and click the `>` button. Once the database is in the [Backup Content](#) list, you can uncheck specific tables that you do not wish to back up.

16.3. Advanced Options

The [ADVANCED OPTIONS](#) tab allows you to configure advanced options that affect the way the backup operation behaves. These include concurrency controls, backup type, and output formatting.

16.3.1. Backup Execution Method

The [Backup Execution Method](#) affects concurrency within the backup itself.

[INNO DB ONLINE BACKUP](#) will begin a transaction before reading from the tables. Because of the InnoDB versioning system this will allow all tables to be read in a consistent state. This method is recommended when backing up InnoDB tables.

[LOCK ALL TABLES](#) will cause all tables within the backup to be locked simultaneously to prevent writing. This will prevent users from writing to the tables during the backup operation and will ensure that the tables are in a consistent state. This option is recommended for MyISAM tables.

[NORMAL BACKUP](#) will lock each table individually when performing the backup. While this can have less impact on the actions of users during the backup operation, there is a potential for tables to be backed up in an inconsistent state in relation to one another. This method is not recommended unless the risk of inconsistency is minimal.

[ONLINE WITH BINLOG POS](#) is the same as [InnoDB Online Backup](#) but also logs the current position within the binary log.

[COMPLETE BACKUP](#) overrides the table selection made in the [BACKUP PROJECT](#) tab and backs up all tables in the selected schema. This option can be useful when scheduling the backup of a schema that changes regularly.

16.3.2. Output File Options

The [OUTPUT FILE OPTIONS](#) section controls the formatting of the backup file. At present there is only one [Backup Type](#) available, the [SQL Files](#) format, which outputs a file similar to the output of the `mysqldump` utility.

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

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

ADD DROP STATEMENTS adds `DROP TABLE IF EXISTS` statements to the backup file.

Note

If your backup project includes views then you *must* choose this option. Restoration of a view is a two step process that involves creating a dummy table with the same name as the view. This table must be dropped before the view can be recreated.

COMPLETE INSERTS will produce full `INSERT` statements that include a column list (i.e. `INSERT INTO table-name(columnA, columnB) VALUES(valueA, valueB)`). Without this option the column list will be omitted.

COMMENT writes additional backup information to the backup file in the form of SQL comments.

FULLY QUALIFIED IDENTIFIERS causes MySQL Administrator to write statements in the format, `schema_name.table_name`, instead of writing only the object name. If you enable fully qualified identifiers you will not be able to restore your backup to a different schema. Selecting this option will also display a warning dialog requesting acknowledgement of this.

Note

Under Linux this check box is labeled **DON'T WRITE FULL PATH**. In this case you need to check this box if you later want to restore to another database schema.

COMPATIBILITY MODE creates backup files that are compatible with older versions of MySQL Administrator.

ANSI QUOTES will cause all table and database names to be quoted with ANSI style double quotes instead of backticks.

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

16.4. Schedule

The **SCHEDULE** tab. is used to execute backup operations on a daily, weekly, or monthly basis.

In order to run a scheduled backup, you must be using a stored connection. Without one you will see the following warning:

```
A scheduled backup requires a stored connection. You are currently
connected without using such a stored connection. Please use the connection
manager to create one and use it then for the login.
```

For instructions on creating a stored connection, see [Chapter 7, Connection Dialog](#).

To enable scheduling for your backup project, check the **SCHEDULE THIS BACKUP PROJECT** checkbox. You will need to choose a target directory for the backup files and also a name for them. You can browse for a target directory by clicking the ... button.

If you have enabled the **ADD DATE/TIME TO BACKUP FILES** option in the `Administrator Options` dialog, then the date and time of the backup will be appended to the file name automatically. See [Section 23.6.3, "Backup"](#) for more information on setting this option.

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

When you select the **EXECUTE BACKUP WEEKLY** option, a set of check boxes representing the days of the week will be displayed. Check the boxes corresponding to the days when you want to perform backups.

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

Click the **SAVE PROJECT** button to save your preferences. After your project is saved it will be executed at the next scheduled interval. To run the backup operation immediately click the **EXECUTE BACKUP NOW** button.

Chapter 17. Restore

17.1. Introduction

The **RESTORE** section helps with disaster recover by helping you restore backup files created using MySQL Administrator. See [Chapter 16, Backup](#) for more information on creating backups with MySQL Administrator.

The backup files created by MySQL Administrator are similar but not identical to those created by `mysqldump`. A consequence of the differences is that MySQL Administrator cannot read dump files created by `mysqldump`.

17.2. The General Tab

In this tab, you specify which backup file should be restored, the character set and format, and the target location for restored tables.

To set any options on this tab, you must first open a backup file by clicking the **OPEN BACKUP FILE** button at the bottom of the window. When all options are set, you can click the **START RESTORE** button to begin the restore process.

The following options are available:

- **FILE TO RESTORE:** Specify the path and file name of the backup file. You can browse for that file by clicking the **OPEN BACKUP FILE** button.
- **TARGET SCHEMA:** You can choose the **ORIGINAL SCHEMA** option to have all tables restored into their original databases. You can also choose the **NEW SCHEMA ...** option to restore all tables into a new database. Finally you can choose from one of the existing databases in the drop-down list to force all tables to be restored to that database.
- **BACKUP TYPE:** At the moment, the only available option is **SQL FILES**. Those are backup files containing SQL statements such as those produced by `mysqldump`.
- **IGNORE ERRORS:** Selecting this option allows a user to continue with the restore operation even if errors are encountered. However, results are not likely to be 100% correct.
- **CREATE DATABASE(S) IF THEY DON'T EXIST:** If a non-existent database is referenced it will automatically be created if this option is checked.
- **FILE CHARSET:** Backup files created with MySQL Administrator are encoded in the `utf8` character set. Backup files created with other tools such as `winmysqladmin` may be encoded with other character sets. Before a backup file can be successfully imported you will need to specify its character set.

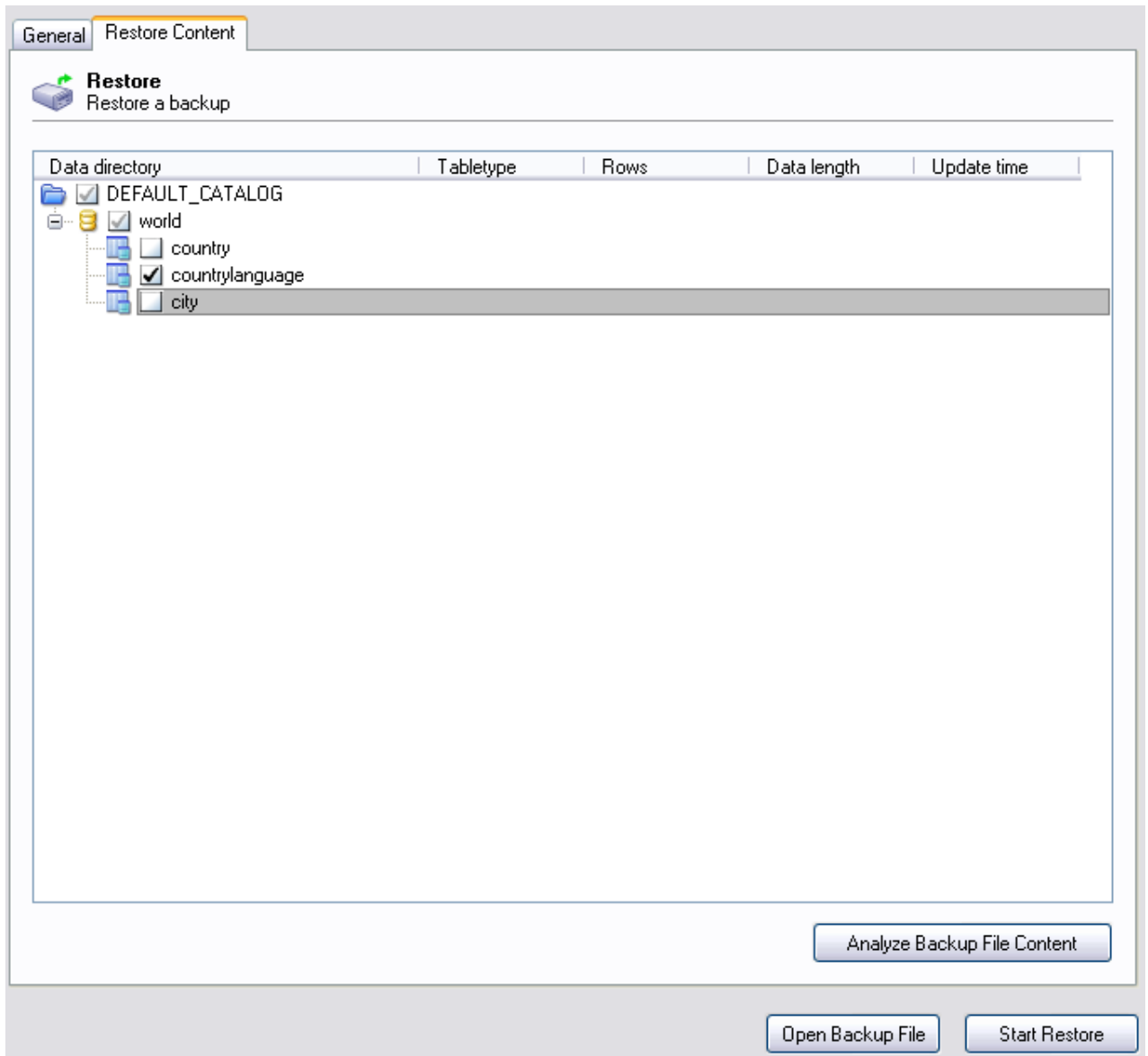
If you do not know the character set of your backup file, click the **AUTO-DETECT CHARACTER SET** button to have MySQL Administrator attempt to determine it automatically.

17.3. Restore Content

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

After clicking the **ANALYZE BACKUP FILE CONTENT** button, a list of all databases and tables contained within the backup file will be displayed.

Figure 17.1. The Restore Content tab



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

Chapter 18. Replication Status

18.1. Introduction

The **REPLICATION STATUS** sections give you a visual overview of your MySQL replication master and slaves. The server status icons beside each machine tells you which slave machines are connected to the master and which ones are offline.

18.2. Configuring Replication Servers

To use the **REPLICATION STATUS** section you must be connected to the server that acts as the master in your replication scheme. Your MySQL servers must be version 4 or higher to use this feature.

For a slave server to show in the [Replication Status](#) screen, it must register itself with the master server. To have a slave server register itself with the master, set the `report-host` option in your server configuration (my.cnf/my.ini) file. See [Replication and Binary Logging Options and Variables](#).

You can also set the value in the **REPLICATION** tab of the **STARTUP VARIABLES** section of MySQL Administrator.

18.3. Using the Replication Status Section

Once your servers are configured, they will appear in the [Replication Status](#) section, as seen in the following figure:

Figure 18.1. The Replication Status section

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

All servers are listed under the **HOSTNAME** column, along with their server **ID**, **Port**, and current status.

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

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

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

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

To start monitoring a new server that is not currently being monitored, click the **ADD INSTANCE TO MONITORING LIST** button. To stop monitoring a server that is currently being monitored, click the **REMOVE INSTANCE FROM MONITORING LIST** button.

Chapter 19. Catalogs

19.1. Introduction

This section deals with viewing catalogs, schemata, tables and their properties, creating databases, and performing a number of operations on tables. Existing catalogs and schemata are listed in the lower left area of the [sidebar](#).

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

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`.

Select the database that you wish to view. Right click on any database and choose the [REFRESH SCHEMATA LIST](#) option to refresh the database list. Clicking on a database will cause the [SCHEMA TABLES](#) and [SCHEMA INDICES](#) tabs to be populated with the tables and indexes of the selected database.

To create a new database, right click on an existing database and choose the [CREATE NEW SCHEMA](#) option.

19.2. Schema Tables

The [SCHEMA TABLES](#) tab lists the different tables in the selected database, along with the rowcount, data length, and index length for those tables. For MyISAM tables, you also find the date and time when the table was last modified, under [UPDATE TIME](#). You can update the list of tables by pressing the [REFRESH](#) button.

The tables are ordered by table name initially, but you may change that sorting by clicking on the appropriate headings ([TYPE](#), [ROW FORMAT](#), etc.).

To drop a table, right click on the table and choose the [DROP TABLE](#) option from the pop-up menu. To create a table you can either click the [CREATE TABLE](#) button at the bottom of the window, or right-click on a table and choose the [CREATE TABLE](#) option from the pop-up menu.

To edit a table's columns and indexes, right-click on the desired table and choose the [EDIT TABLE](#) option from the pop-up menu. This will launch the MySQL Table Editor, which you can use to modify the table. For more information, see [Chapter 24, The MySQL Table Editor](#).

To edit a table's actual data, right click on the table and choose the [EDIT TABLE DATA](#) option. This will launch the MySQL Query Browser and load the table's data into the result area of the MySQL Query Browser.

To perform operations on multiple tables select more than one table. For the selected tables, you can perform operations by either clicking the [MAINTENANCE](#) button at the bottom of the table list or by right clicking on one of the selected tables and choosing from the [MAINTENANCE](#) sub-menu of the pop-up menu:

The available options are:

- [OPTIMIZE TABLE](#): This corresponds to the [OPTIMIZE TABLE](#) SQL command and should be used if you have deleted a large part of a table or if you have made many changes to a table with variable-length rows (tables that have [VARCHAR](#), [BLOB](#), or [TEXT](#) columns). Deleted records are maintained in a linked list and subsequent [INSERT](#) operations reuse old record positions. You can use this command to reclaim the unused space and to defragment the datafile. Note that table optimization works for MyISAM and BDB tables only. For more information, see [OPTIMIZE TABLE Syntax](#).
- [CHECK TABLE](#): This corresponds to the [CHECK TABLE](#) SQL command and is used to diagnose table problems. For more information, see [CHECK TABLE Syntax](#).
- [REPAIR TABLE](#): This corresponds to the [REPAIR TABLE](#) SQL command and should be used in case of table problems. Note that this command works for MyISAM tables only. For more information, see [REPAIR TABLE Syntax](#).

Clicking on a table name will display its properties in the details area when the details area is active. To display the details area, click the [DETAILS >>](#) button.

19.2.1. Table Status

In this tab, you find detailed information about the selected table. This information could also be retrieved by issuing a [SHOW TABLE STATUS LIKE 'tbl'](#) SQL command. For more information, see [SHOW TABLE STATUS Syntax](#).

19.2.2. Row Status

In this tab, you find detailed information about the rows of the selected table. This information can also be retrieved by issuing a `SHOW TABLE STATUS LIKE 'tbl'` SQL command. For more information, see [SHOW TABLE STATUS Syntax](#).

19.3. Schema Indexes

The **SCHEMA INDICES** tab lists the indexes of the selected database. Besides the index names and the table the index belongs to, you can also see the index type, whether values are unique, and whether NULL values are allowed. You can also get this information by issuing a `SHOW INDEX` SQL command in a command-line client (see [SHOW DATABASES Syntax](#)). For more information about indexes, see [Column Indexes](#).

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

Part II. MySQL Query Browser

Table of Contents

20. Introduction to the MySQL Query Browser	46
21. A Tour of the MySQL Query Browser	47
21.1. The Main Query Window	47
21.2. The Query Toolbar	48
21.2.1. The Navigation Buttons	48
21.2.2. The Query Area	48
21.2.3. The Action Buttons	48
21.3. The Advanced Toolbar	49
21.4. The Result Area	49
21.5. The Object Browser	50
21.5.1. The Database Browser	50
21.5.2. The Bookmark Browser	51
21.5.3. The History Browser	51
21.6. The Information Browser	51
21.6.1. The Syntax Browser	52
21.6.2. The Function browser	52
21.6.3. The Parameter Browser	52
21.6.4. The Transaction Browser	52
21.7. The Script Editor	52
21.7.1. The Script Debugging Buttons	54
21.7.2. The Script Area	54
22. Using The MySQL Query Browser	56
22.1. Entering and Editing Queries Manually	56
22.2. Building Queries	56
22.3. Creating Views	57
22.4. Navigating Result Sets Within the Result Area	57
22.5. Exporting Result Sets	58
22.6. Handling BLOB and TEXT Columns	58
22.7. Editing Result Sets Within the Result Area	58
22.8. Comparing Result Sets	59
22.9. Managing Stored Procedures and Functions with MySQL Query Browser	59
22.10. Editing Queries From a Development Tool	60
22.11. Using the Regex Text Importer Tool	60

Chapter 20. 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 [MySQL 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 MySQL AB has received from many users over a period of several years. However, if you find it's lacking some feature important to you, or if you discover a bug, please use our [MySQL Bug System](#) to request features or report problems.

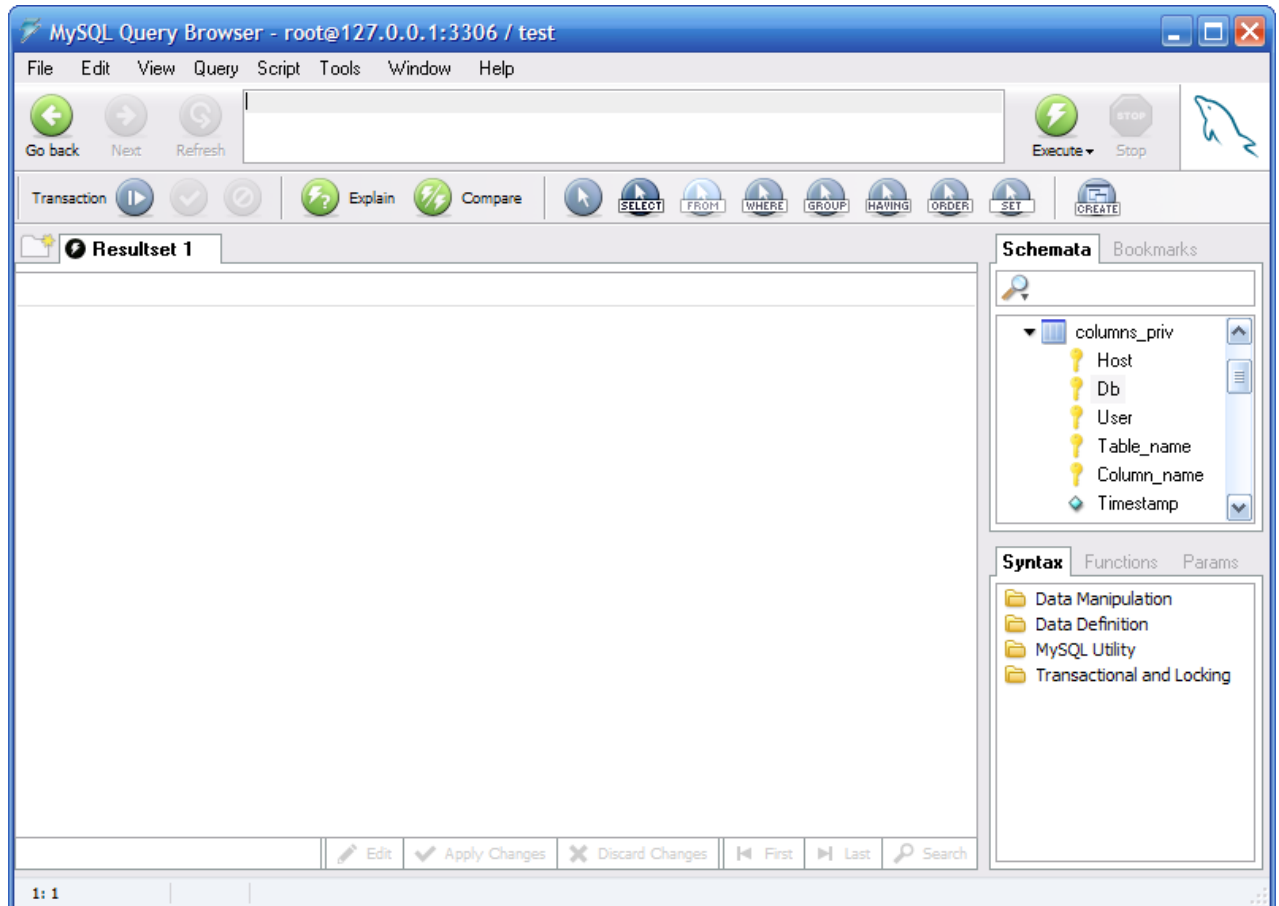
For MySQL Query Browser, there's a dedicated forum available on <http://forums.mysql.com/list.php?108>.

Chapter 21. A Tour of the MySQL Query Browser

21.1. 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 21.1. 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 23.5, “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.

21.2. 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.

21.2.1. 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.

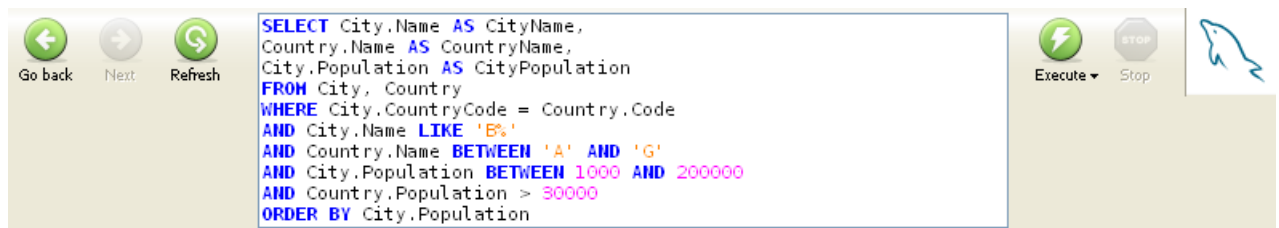
As you navigate with the **NEXT** and **GO BACK** buttons, the queries you navigate through are not executed unless you explicitly execute them by clicking the **EXECUTE** button. The **REFRESH** button re-executes the last executed query, which may or may not be the current query in the query area.

21.2.2. 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 21.2. 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.

21.2.3. 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 21.3. The MySQL Query Browser status icons



21.3. 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 21.4. 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.

21.4. 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.

21.5. The Object Browser

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

21.5.1. 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 21.5. 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.

21.5.2. 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](#).

21.5.3. 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.

21.6. 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.

21.6.1. 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.

21.6.2. 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.

21.6.3. 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.

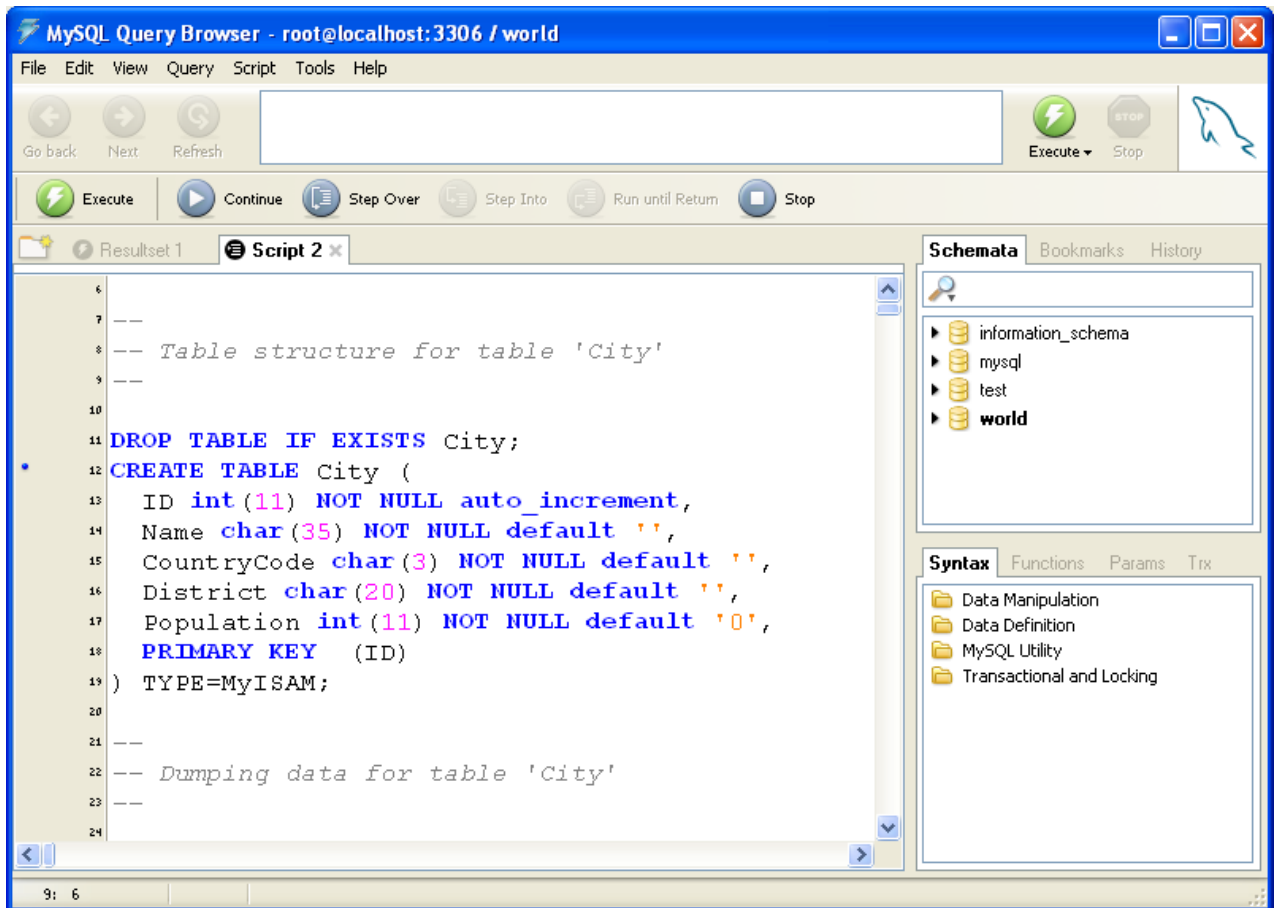
21.6.4. The Transaction Browser

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.

21.7. The Script Editor

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

Figure 21.6. 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.

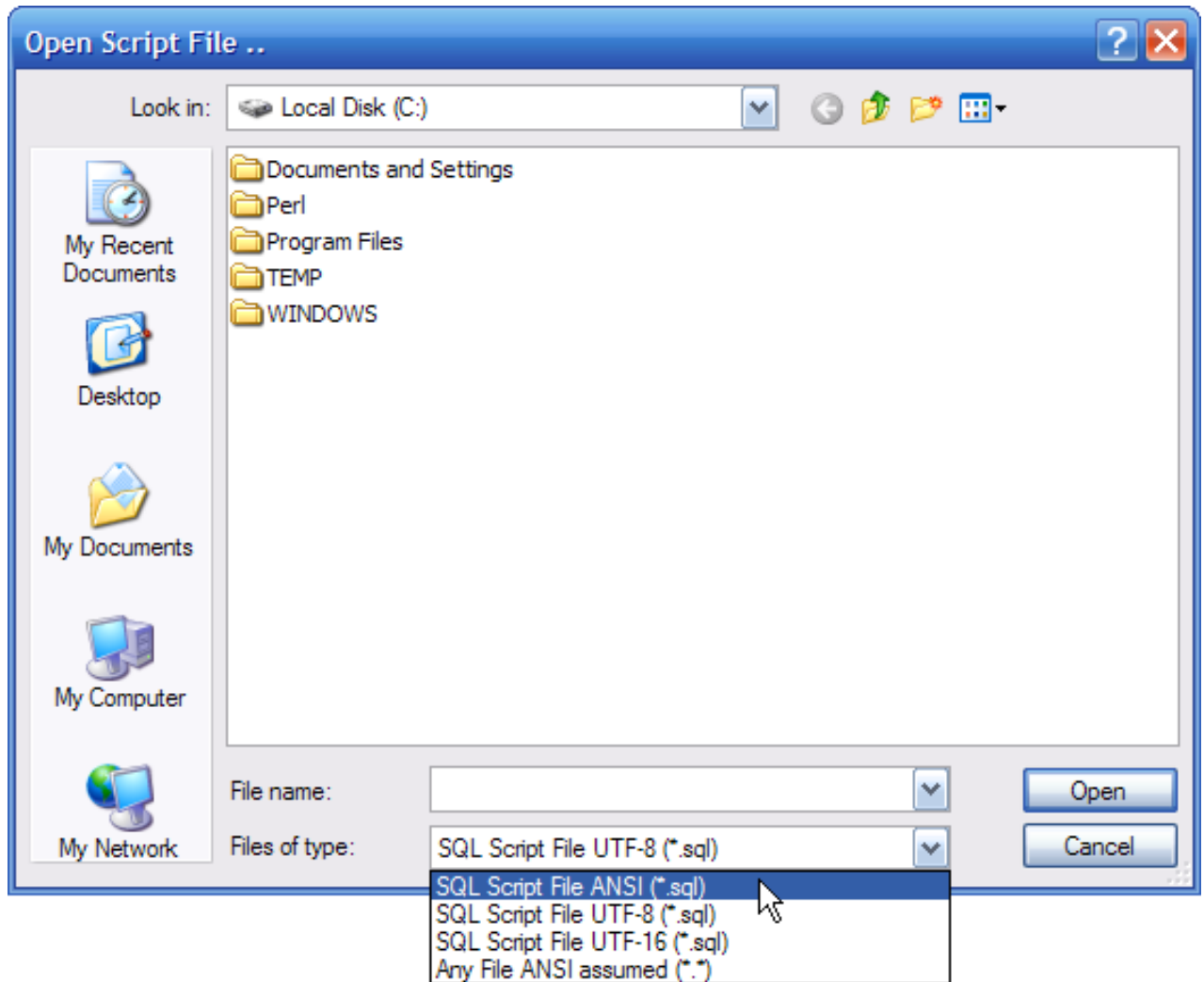
To open a script in the Script Editor, choose the [OPEN SCRIPT...](#) option from the [FILE](#) menu. You can also press the **Ctrl + O** keys to open an SQL script.

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 21.7. 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>.

21.7.1. 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.

21.7.2. 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 BREAKPOINT](#) 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 22. Using The MySQL Query Browser

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

22.1. 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**.

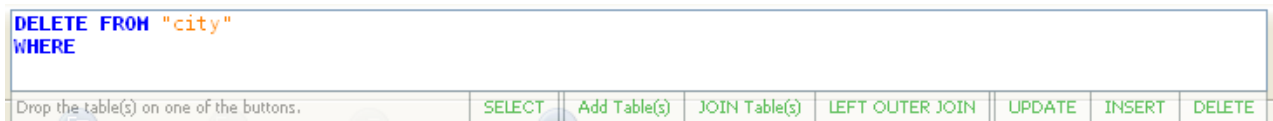
You must set a default database before you can query specific tables (though you can still perform queries that are not database-specific). You can set the default database at the connection screen, or by right-clicking on a database in the database browser and choosing MAKE DEFAULT SCHEMA, or by choosing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

22.2. 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 22.1. 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 23.5.1, “Display Options”](#) for more information.

22.3. 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.

22.4. 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 (i.e. `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.

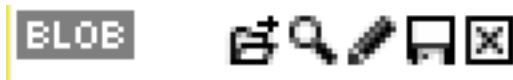
22.5. Exporting Result Sets

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.

22.6. 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 22.2. 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 22.7, “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 23.5.1, “Display Options”](#) for more information.

22.7. 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.

22.8. 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.

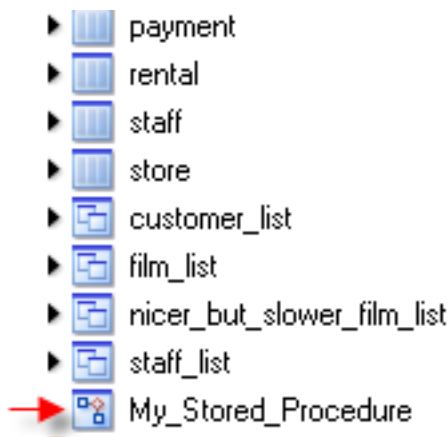
In order to successfully compare two result sets, you need two queries with matching column names and column order. For the MySQL Query Browser to match rows, primary keys must be defined in the tables used.

22.9. 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 22.3. 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 FUNCTION** 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.

22.10. 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 non-query 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.

22.11. 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.

Part III. Common Administrator and Query Browser Components

Table of Contents

23. Options Dialog	63
23.1. Introduction	63
23.2. General Options	63
23.3. Connections	64
23.3.1. SSL Connections	66
23.4. Editors	67
23.5. The Browser Section	67
23.5.1. Display Options	68
23.5.2. Query Options	68
23.5.3. Various	68
23.6. Administrator	69
23.6.1. User Administration	69
23.6.2. Health Graphs	69
23.6.3. Backup	69

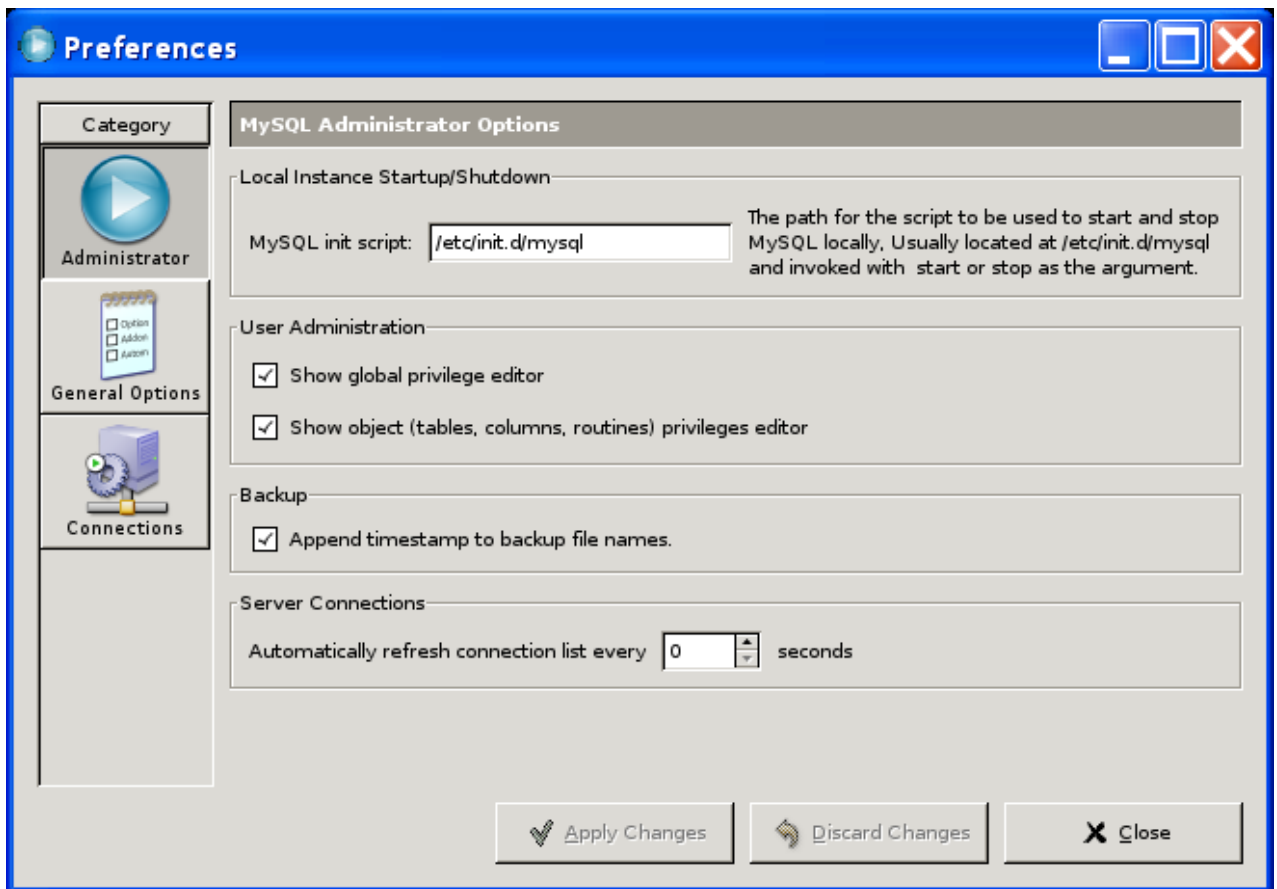
Chapter 23. Options Dialog

23.1. 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 23.1. 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.

23.2. 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.

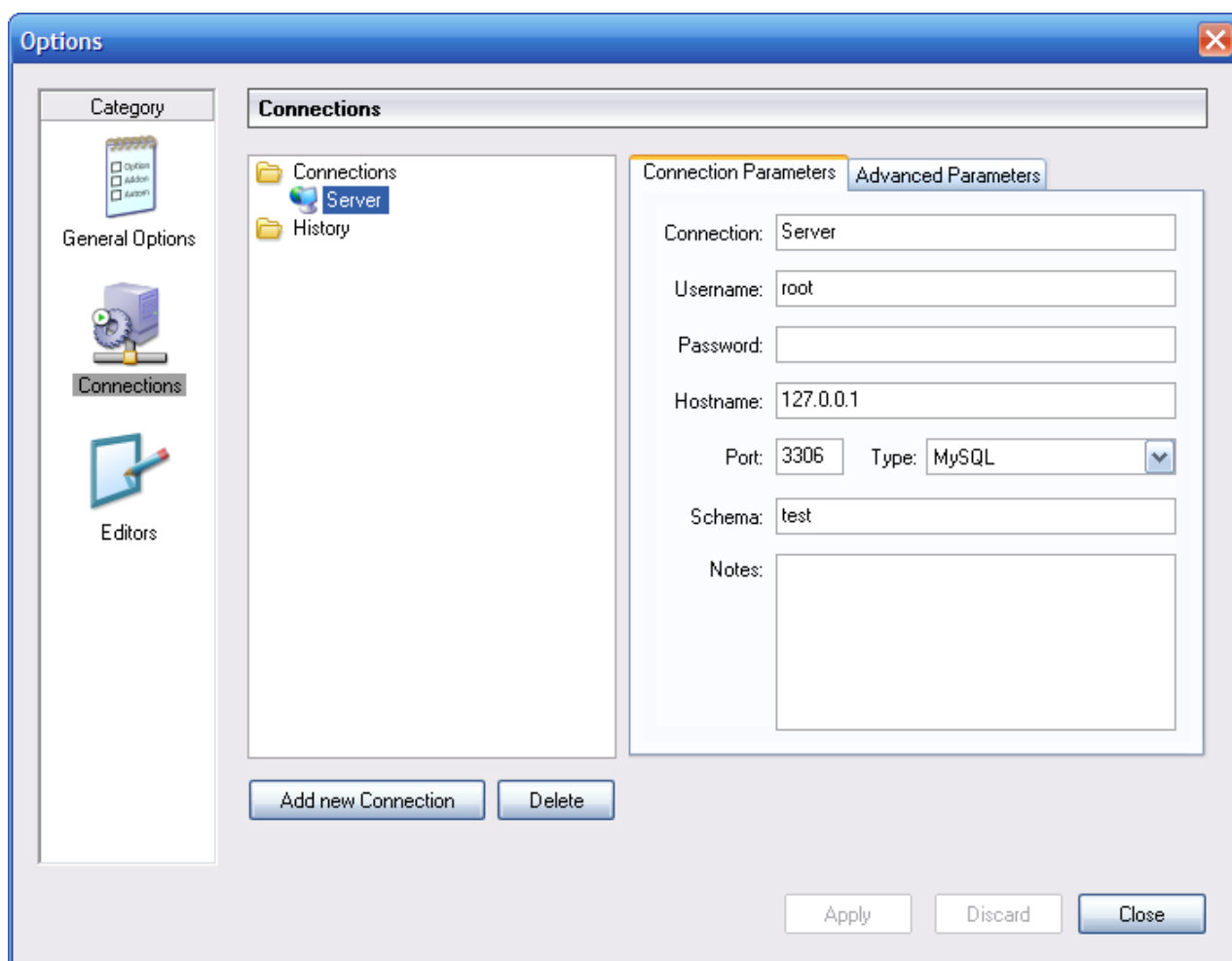
You may change the font and font size of any of the application fonts by clicking the CHOOSE ... button to the right of the font.

23.3. 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 23.2. 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.

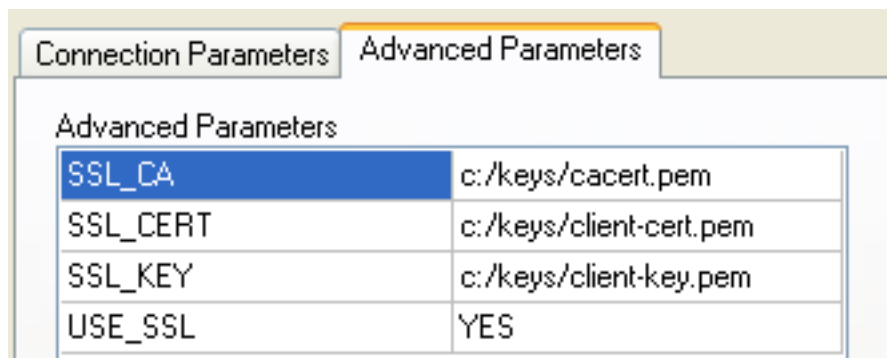
- [Allow interactive_timeout seconds of inactivity before disconnect](#): Normally, the connection is closed by the MySQL server after a certain period of inactivity on the client side. This period can be set with the `interactive_timeout` variable. If checked, the server will not close the connection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

23.3.1. 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 23.3. 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.

23.4. 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

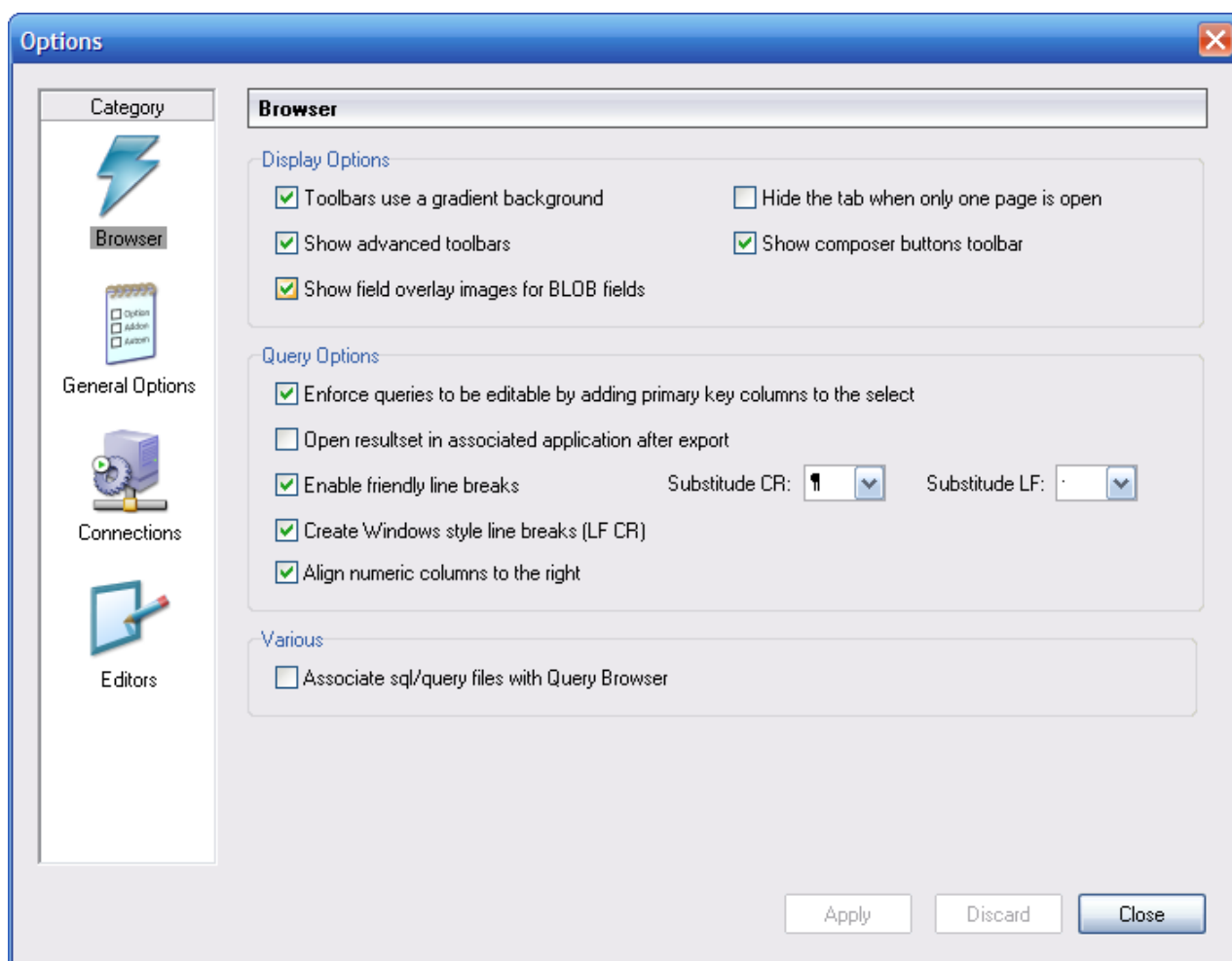
Currently, configuring editors is only available on the Windows platform.

- [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`.

23.5. 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 23.4. The query browser options dialog



23.5.1. 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.

23.5.2. 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](#).

23.5.3. 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.

23.6. Administrator

The [Administrator](#) section allows you to specify options that affect the behavior of the MySQL Administrator. The [Administrator](#) section is divided into four sub-sections that will be discussed further in the sections that follow.

23.6.1. User Administration

These options affect the interface of the [User Administration](#) section of MySQL Administrator.

- **SHOW GLOBAL PRIVILEGES**: If checked, the [Global Privileges](#) tab will be visible in the [User Administration](#) section. In that tab, you can grant users global privileges (as opposed to privileges on a database, table, or column level only). This may affect security. See [The MySQL Access Privilege System](#). If unchecked, the [Global Privileges](#) tab will not be visible in the [User Administration](#) section.
- **SHOW TABLE/COLUMN PRIVILEGES**: In most situations, it is sufficient to grant either global or database privileges for users. You can hide the [Table/Column Privileges](#) tab in the [User Administration](#) section by enabling this checkbox.

23.6.2. Health Graphs

These options affect the appearance of the [Health Graphs](#) section of MySQL Administrator.

- **USE PEAK LEVEL INDICATOR**: This will display a thin yellow line on the health graphs indicating the highest value recorded by the graph.
- **RESET PEAK LEVEL AFTER A NUMBER OF TICKS**: If this box is not checked, the [Peak Level Indicator](#) will be permanently positioned at the highest value ever recorded on the health graph. With this box checked, the [Peak Value Indicator](#) will be reset after a period of time indicated by the **TIME AFTER THE PEAK LEVEL RESETS** option. Use this option if you would like to know the peak level over an interval rather than the peak level overall.
- **TICKS AFTER THE PEAK LEVEL RESETS**: When the **RESET PEAK LEVEL AFTER A NUMBER OF TICKS** option is set, this determines the amount of time (in ticks) that will pass before the [Peak Level Indicator](#) will be reset.

23.6.3. Backup

The backup options configure how MySQL Administrator performs backup operations.

- **ADD DATE/TIME TO BACKUP FILES**: This determines whether the date and time of the backup operation are added to the name of the backup file. See [the backup section](#) for more information.

Part IV. Common Administrator and Query Browser Components

Table of Contents

24. The MySQL Table Editor	72
24.1. Introduction	72
24.2. The Main Editor Window	72
24.3. The Columns And Indices Tab	73
24.3.1. The Column Editor	73
24.3.2. The Detail Area	74
24.4. The Table Options Tab	75
24.5. The Advanced Options Tab	75
24.5.1. The Various Section	76
24.5.2. The Row Options Section	76
24.5.3. The Storage Options Section	76
24.5.4. The Merge Table Options Section	76
24.5.5. The Table RAID Settings Section	76
24.6. Applying Your Changes	76

Chapter 24. The MySQL Table Editor

24.1. 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

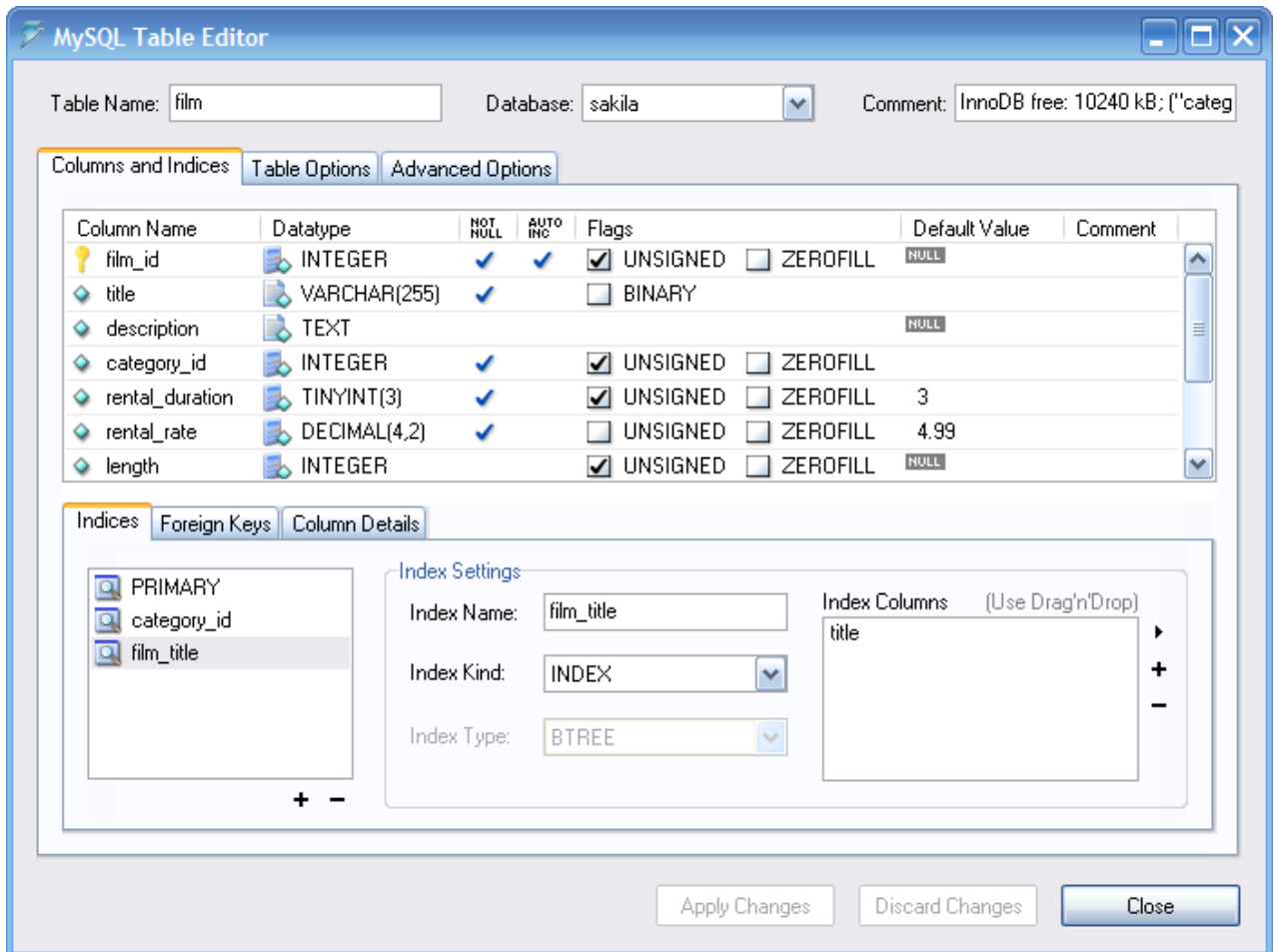
Docking is not supported on all operating systems.

24.2. 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 24.1. 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.








24.3. 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.

24.3.1. 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 24.2. The column editor

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
 film_id	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL	
 title	VARCHAR(255)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY		
 description	TEXT				NULL	
 category_id	INTEGER	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL		
 rental_duration	TINYINT(3)	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	3	
 rental_rate	DECIMAL(4,2)	<input checked="" type="checkbox"/>		<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	4.99	
 length	INTEGER			<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> 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.

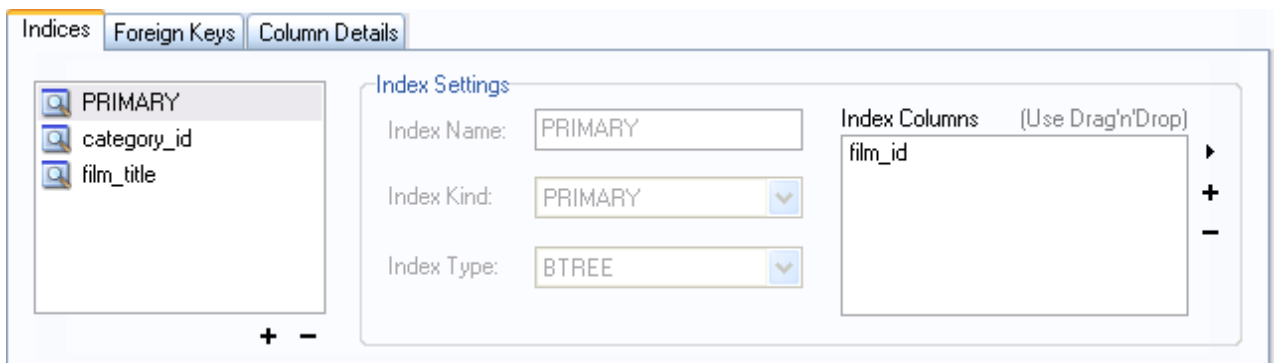
24.3.2. The Detail Area

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.

24.3.2.1. 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 24.3. 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 23.4, “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.

24.3.2.2. 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 23.4, “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.

24.3.2.3. 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.

24.4. 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.

24.5. 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](#).

24.5.1. 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.

24.5.2. 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.

24.5.3. 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.

24.5.4. 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.

24.5.5. 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.

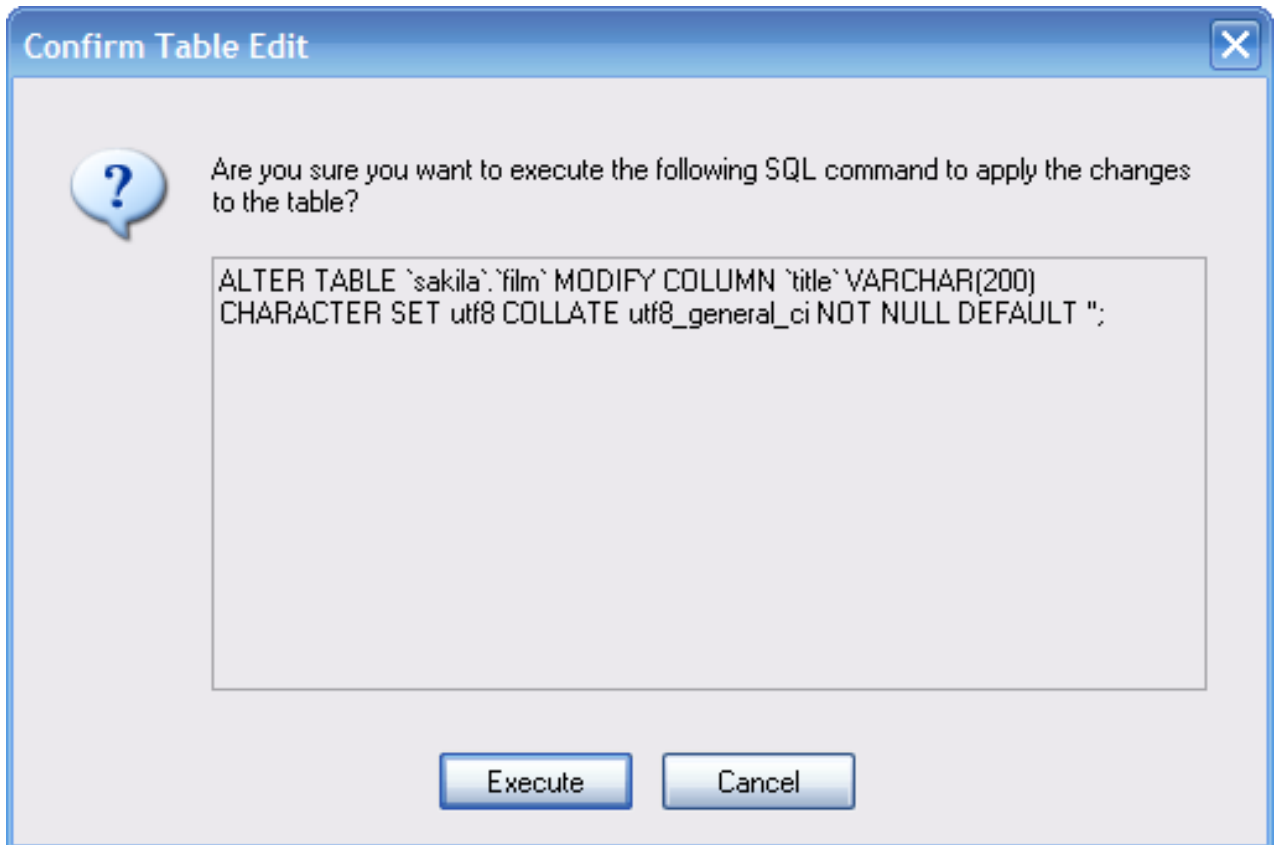
For more information on using RAID support with MyISAM, see [CREATE TABLE Syntax](#).

24.6. 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 24.4. 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.

Part V. MySQL Migration Toolkit

Table of Contents

25. Introduction to the MySQL Migration Toolkit	80
26. An Overview of the Migration Process	81
27. The Migration Process In-Depth	83
27.1. Introduction	83
27.2. The Welcome Screen	83
27.3. The Configuration Type Screen	83
27.4. The Source Database Screen	84
27.4.1. Microsoft Access	84
27.4.2. Microsoft SQL Server	85
27.4.3. Oracle	86
27.4.4. MySQL	88
27.4.5. Saving Connection Information	89
27.5. The Target Database Screen	89
27.6. The Connect to Server Screen	90
27.7. The Source Schema Selection Screen	91
27.8. The Reverse Engineering Screen	92
27.9. The Object Type Selection Screen	93
27.9.1. Migrating a Sub-Set of Object Types	94
27.10. The Object Mapping Screen	95
27.10.1. GRT Object	96
27.10.2. Table Objects	96
27.11. The Meta Migration Screen	96
27.12. The Manual Editing Screen	97
27.13. The Object Creation Options Screen	99
27.14. The Creating Objects Screen	100
27.15. The Data Mapping Options Screen	101
27.16. The Bulk Data Transfer Screen	102
27.17. The Summary Screen	103
27.18. Saving the Current Application State	104
28. Scripted Migration	105
28.1. The Steps for Scripted Migration	105
28.2. Setting the Source and Target Connection	105
28.3. Reverse Engineering	106
28.4. Migration Methods	106
28.5. Map Objects and Migrate	106
28.6. The SQL Create Statements	106
28.7. Bulk Data Transfer	107
29. Extending The MySQL Migration Toolkit	108
29.1. Introduction	108
29.2. Architecture of the MySQL Migration Toolkit	108
29.3. The Modular Migration Process	108
29.4. Tools Required to Extend the MySQL Migration Toolkit	109
30. Preparing a Microsoft Access Database for Migration	110
31. The Generic Runtime Environment (GRT) Shell	112
31.1. Introduction	112
31.2. Exploring the GRT Shell	112
31.2.1. Menu Items	113
31.2.2. The Shell	114
31.2.3. The Globals Tree Panel	114
31.3. Using the GRT Shell	115
31.4. Invoking the GRT Shell From the Command Line	116

Chapter 25. Introduction to the MySQL Migration Toolkit

The MySQL Migration Toolkit is a graphical tool provided by MySQL AB for migrating schema and data from various relational database systems to MySQL.

MySQL Migration Toolkit is designed to work with MySQL versions 5.0 and higher.

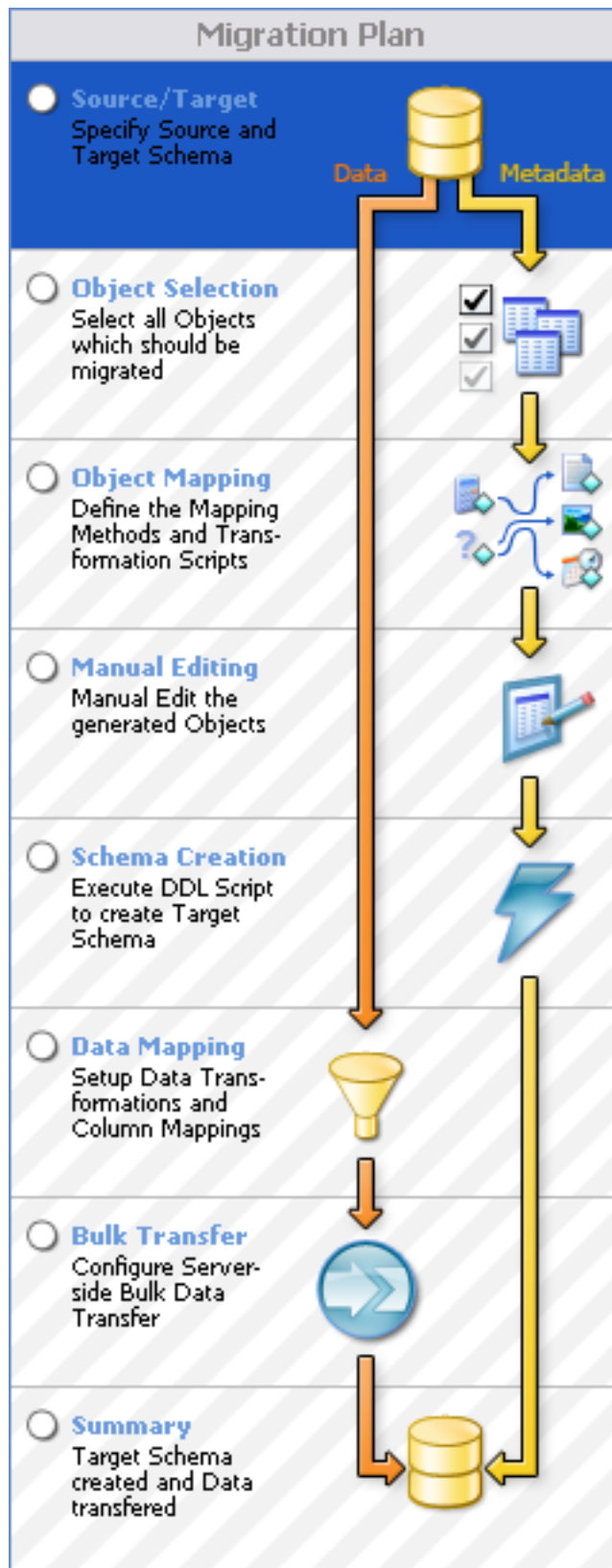
If you find that MySQL Migration Toolkit lacks some feature important to you, or if you discover a bug, please use our [MySQL Bug System](#) to request features or report problems.

At the moment, MySQL Migration Toolkit is only available for Windows.

Chapter 26. An Overview of the Migration Process

Migrating data from an external RDBMS to MySQL is an eight step process :

Figure 26.1. The MySQL Migration Toolkit Migration Plan



- **Source/Target Selection:** In the first step you specify the connection parameters for the source and target database servers.
- **Object Selection:** In the second step you select the objects (tables, views, stored procedures) that will be migrated.
- **Object Mapping:** In the third step you choose the method used for mapping and transforming the objects.
- **Manual Editing:** In the fourth step you can manually edit the new objects to ensure a proper transformation.
- **Schema Creation:** In the fifth step the MySQL Migration Toolkit creates the transformed object on the target MySQL server.
- **Data Mapping:** In the sixth step you specify any changes that need to be made to the data as it is migrated.
- **Bulk Transfer:** In the seventh step the MySQL Migration Toolkit transfers the data from the source server to the target server.
- **Summary:** In the eighth and final step the MySQL Migration Toolkit creates a summary report of the migration process for you to review.

Each of these sections will be covered in more detail in the coming chapters.

Chapter 27. The Migration Process In-Depth

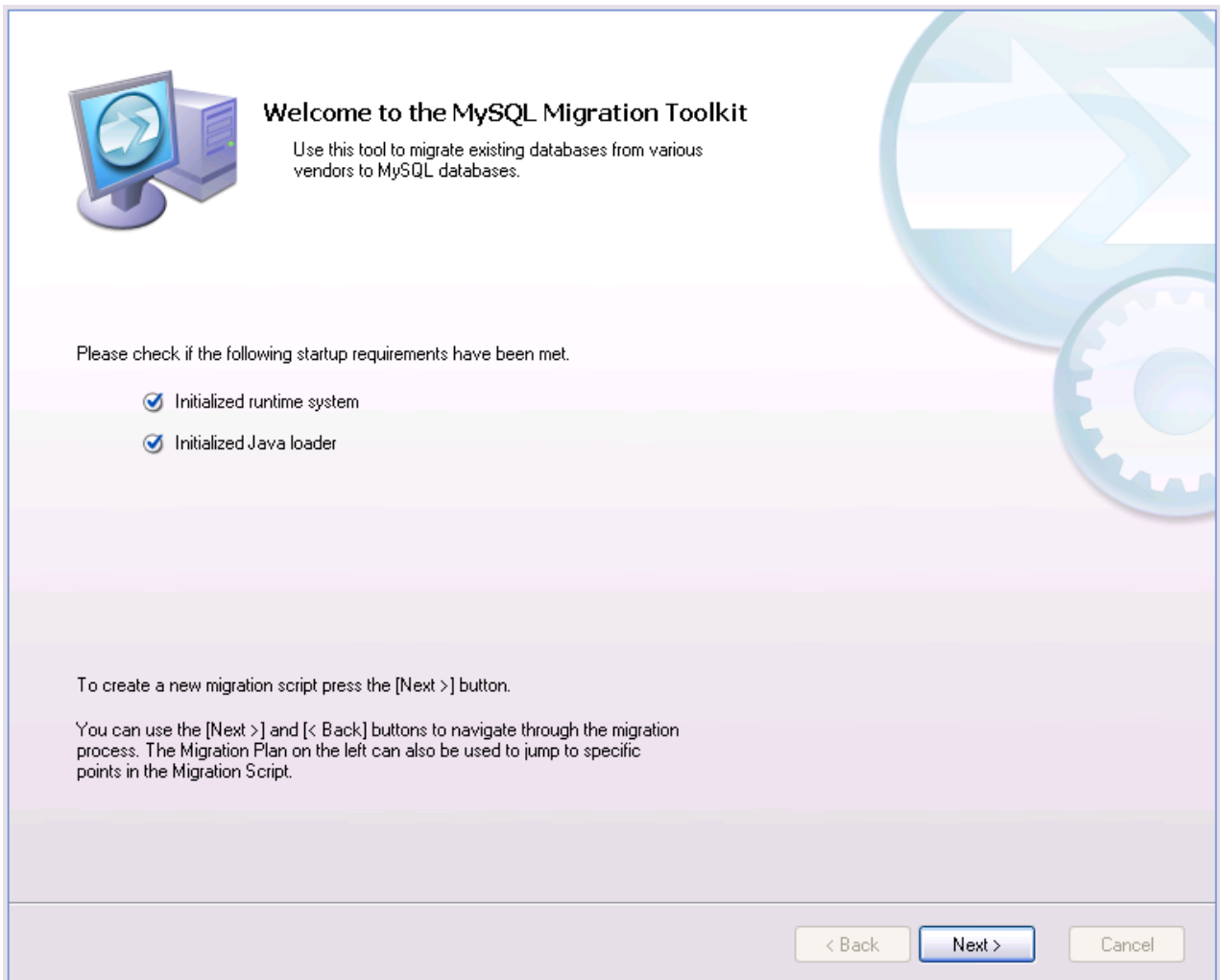
27.1. Introduction

In this chapter we will cover the different steps of the MySQL Migration Toolkit in depth. The different steps will be covered in the order that they appear in the MySQL Migration Toolkit.

27.2. The Welcome Screen

The first screen of the MySQL Migration Toolkit is the Welcome Screen:

Figure 27.1. The MySQL Migration Toolkit welcome screen

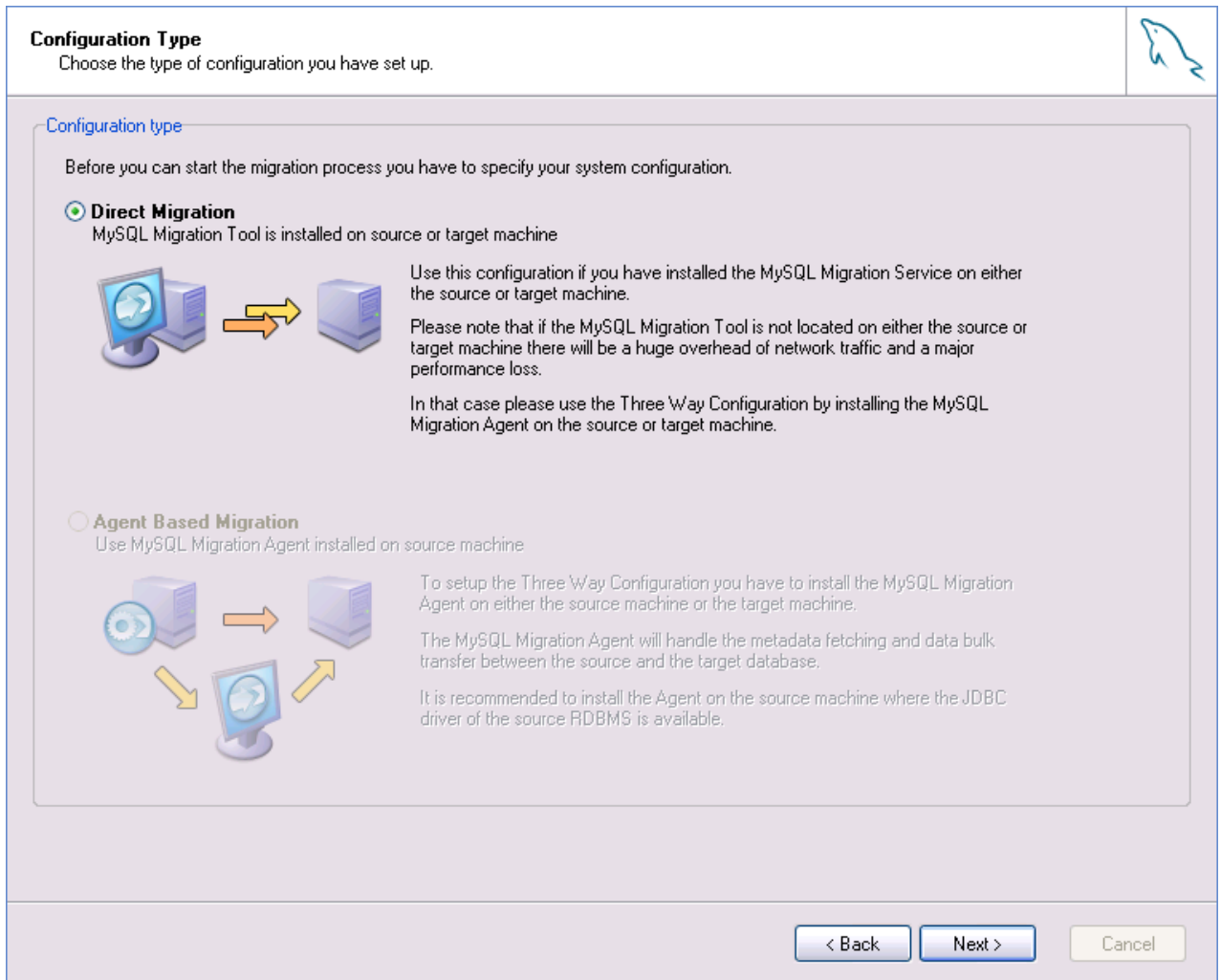


The welcome screen indicates the progress of loading the various components of the MySQL Migration Toolkit. If you encounter any errors on the welcome screen you should close the MySQL Migration Toolkit and confirm that you have properly installed the Java Runtime Environment. See [Chapter 25, Introduction to the MySQL Migration Toolkit](#) for information on downloading and installing the Java Runtime Environment.

27.3. The Configuration Type Screen

The Configuration Type screen allows you to choose between a Direct Migration and a Agent-Based Migration:

Figure 27.2. The Configuration Type screen



Use the [Direct Migration](#) if the MySQL Migration Toolkit is installed on either the source or target machine. The Direct Migration should not be used when the MySQL Migration Toolkit is not located on either the source or target machine, as it will create a large amount of network traffic and will result in decreased performance.

Use the [Agent-Based Migration](#) when migrating between two server machines that do not support the use of the MySQL Migration Toolkit. The MySQL Migration agent should be installed on the source machine before using the Agent-Based Migration.

27.4. The Source Database Screen

Use the Source Database screen to select the source RDBMS used in the migration and to specify the connection parameters.

The Source Database screen's appearance will vary depending on the type of source database selected.

All Data Source screens will feature a **DETAILS** button that can be used to expose the **ADVANCED SETTINGS** panel. The Advanced Settings panel can be used to manually specify a JDBC driver and JDBC connection string for your migration session.

27.4.1. Microsoft Access

The Source Database screen appears as follows when you select Microsoft Access as the source database:


Figure 27.3. Source database – Microsoft Access

Source Database
Select the source database you want to migrate from.

Source Database Connection

Database System: Select the target RDBMS you want to migrate from.

Connection Parameters

 **MS Access Server**
MS Access over JDBC-ODBC bridge

Connection: + - Select a stored connection or use [+] to store or [-] to remove.

Database File: ... MS Access database file.

Username: Name of the user to connect with.

Password: The user's password.

Details >> < Back Next > Cancel

Specify the path to the `.mdb` database file in the `Database File` field and specify the database user name and password information if applicable.

You must make special modifications to your Access database file before it can be used with MySQL Migration Toolkit. Please see [Chapter 30, Preparing a Microsoft Access Database for Migration](#) for further information.

27.4.2. Microsoft SQL Server


The Source Database screen appears as follows when you select `MS SQL Server` as the source database:

Figure 27.4. Source database – Microsoft SQL Server

Source Database Connection

Database System: Select the target RDBMS you want to migrate from.

Connection Parameters

 **MS SQL Server**
JDBC driver to connect to MS SQL Server 2000.

Connection: + - Select a stored connection or use [+] to store or [-] to remove.

Hostname: Port: Name or IP address of the server machine / TCP/IP port

Username: Name of the user to connect with.

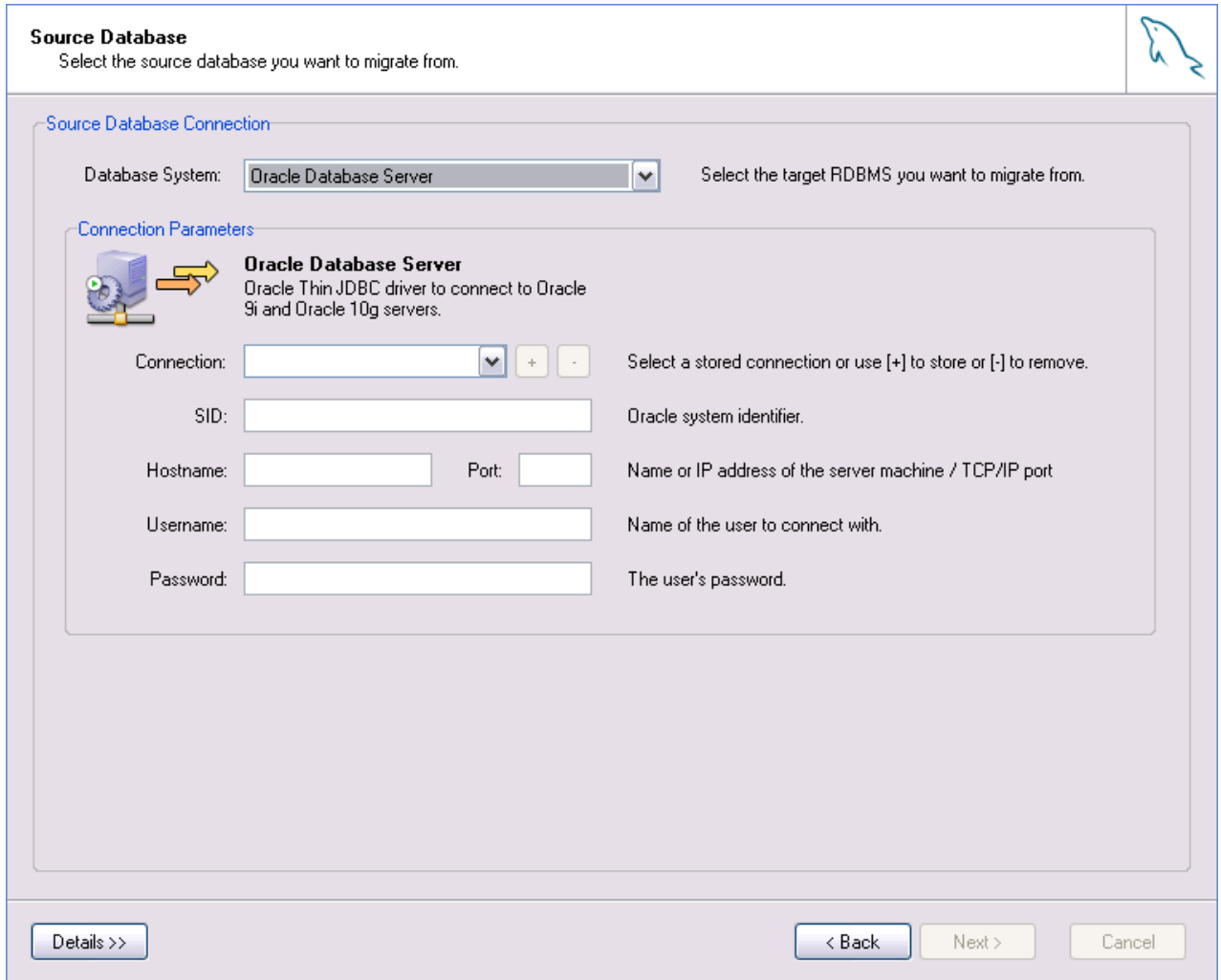
Password: The user's password.

Specify the host name, user name, and password to connect to the source Microsoft SQL Server to connect and click NEXT.

27.4.3. Oracle

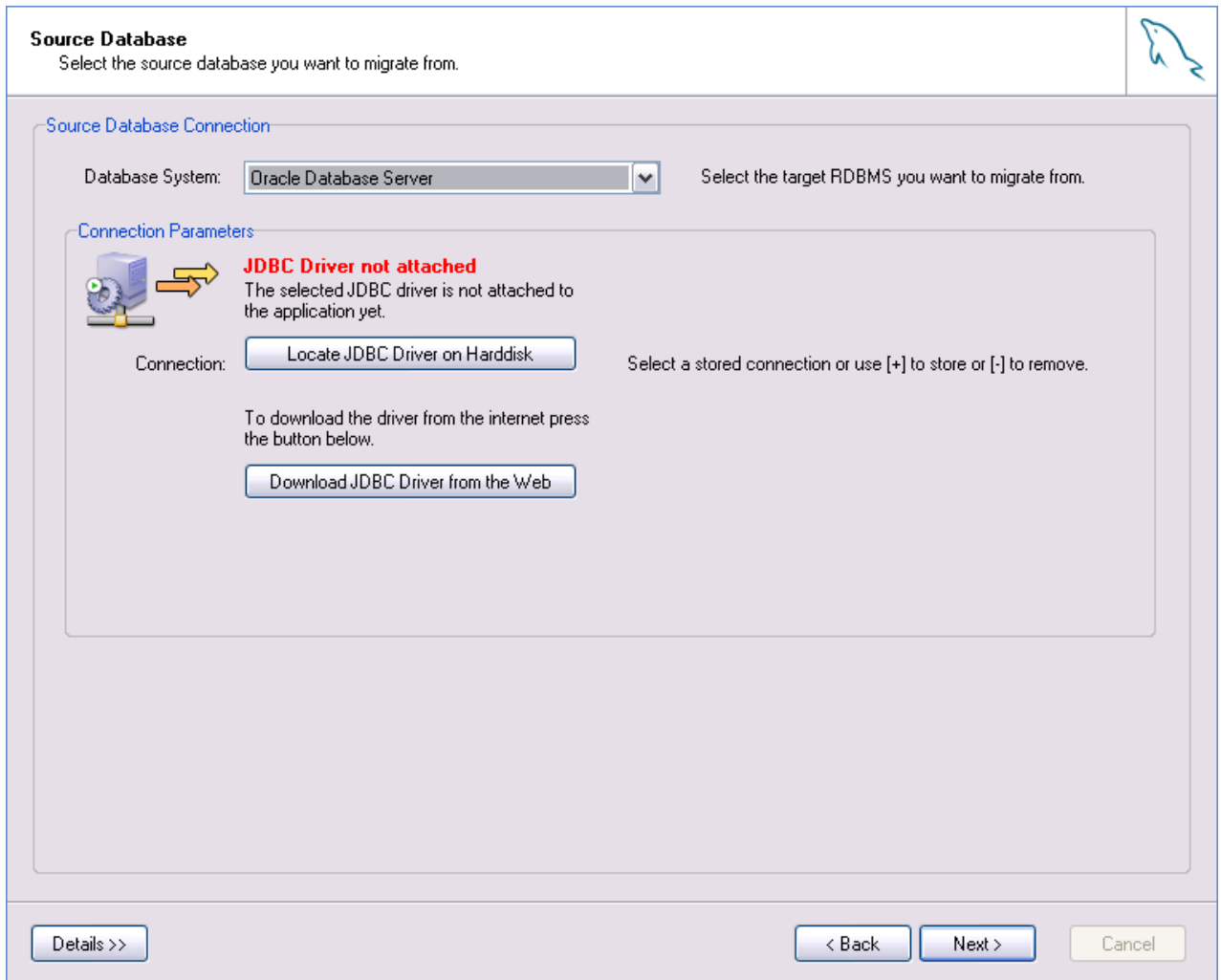
The Source Database screen appears as follows when you select Oracle as the source database:

Figure 27.5. Source database – Oracle



If you encounter the following database selection screen, it means that you do not have the appropriate JDBC driver for Oracle installed:

Figure 27.6. Oracle JDBC driver not attached



If the Oracle JDBC driver is present on your system, click the **LOCATE JDBC DRIVER ON HARDDISK** button to attach the driver.

If the Oracle JDBC driver is not present on your system, click the **DOWNLOAD JDBC DRIVER FROM THE WEB** button to download it. Download the [ojdbc14.jar](#) file and then attach it by clicking on the **LOCATE JDBC DRIVER ON HARDDISK** button.

After attaching the Oracle JDBC driver you need to restart the MySQL Migration Toolkit.

27.4.4. MySQL

The Source Database screen appears as follows when you select MySQL as the source database:

Figure 27.7. Source database – MySQL

Source Database
Select the source database you want to migrate from.

Source Database Connection

Database System: Select the target RDBMS you want to migrate from.

Connection Parameters

MySQL Server
MySQL JDBC driver to connect to MySQL 4.0, 4.1 and 5.0 servers.

Connection: + - Select a stored connection or use [+] to store or [-] to remove.

Hostname: Port: Name or IP address of the server machine / TCP/IP port

Username: Name of the user to connect with.

Password: The user's password.

27.4.5. Saving Connection Information

After entering the connection information for the source database, click the + button to save the connection information.

When prompted, enter a name for the connection information and click the OK button to save the connection information for later re-use.

You can discard saved connection information by selecting the saved connection from the **CONNECTION** drop-down list and clicking the – button.

27.5. The Target Database Screen

Use the Target Database screen to select the target RDBMS used in the migration and to specify the connection parameters.

The target Database screen uses an interface that is identical to that of the [Source Database screen](#):

Figure 27.8. Target Database – MySQL

Target Database
Select the destination database.

Target Database Connection

Database System: Select the target RDBMS you want to migrate to.

Connection Parameters

MySQL Server
MySQL JDBC driver to connect to MySQL 4.0, 4.1 and 5.0 servers.

Connection: + - Select a stored connection or use [+] to store or [-] to remove.

Hostname: Port: Name or IP address of the server machine / TCP/IP port

Username: Name of the user to connect with.

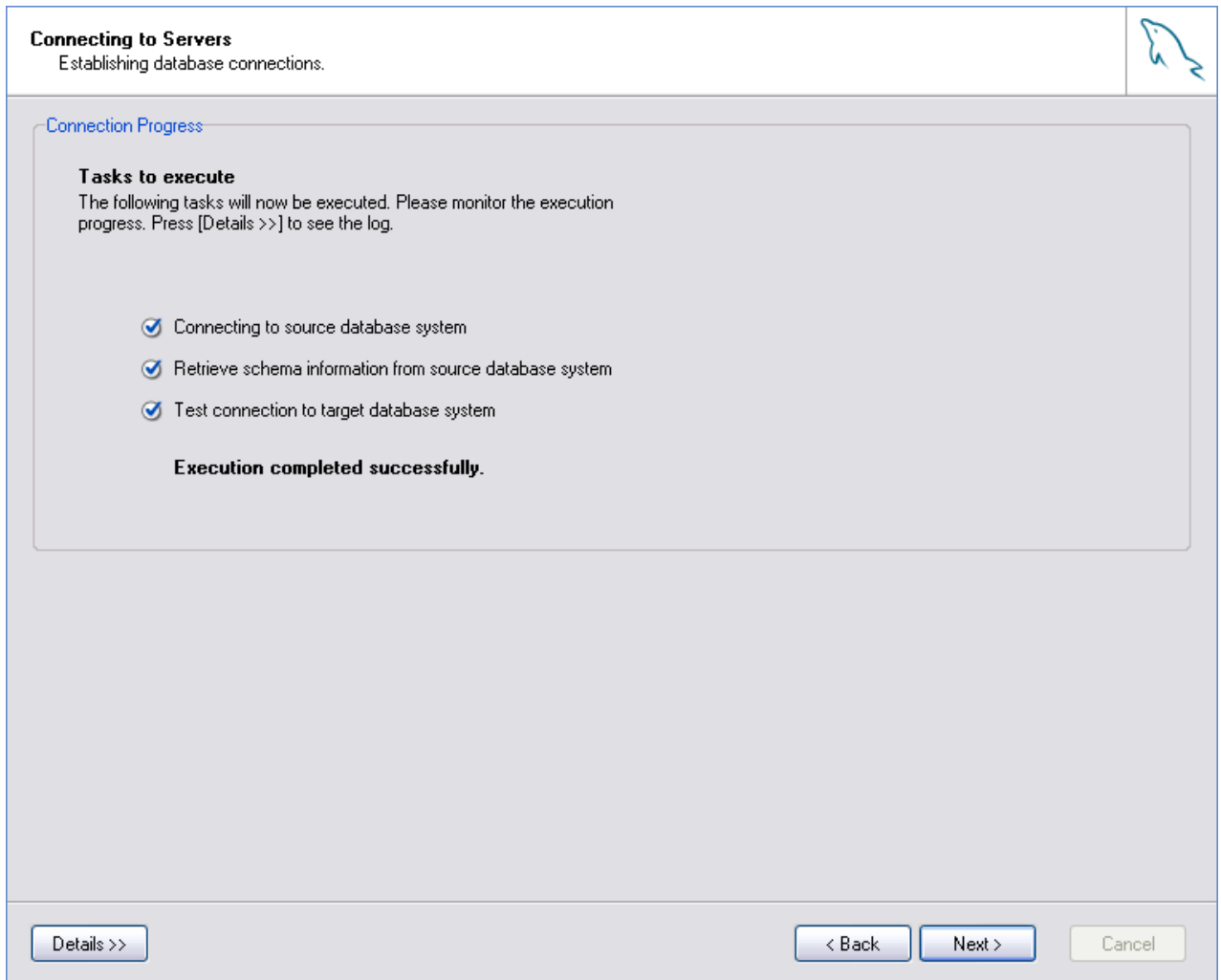
Password: The user's password.

Target database support for the MySQL Migration Toolkit is currently limited to MySQL 4.1 and MySQL 5.0.

27.6. The Connect to Server Screen

After you specify your source and target database servers, the MySQL Migration Toolkit will connect to each server and retrieve the schema information from the source server:

Figure 27.9. The Connect to Servers screen



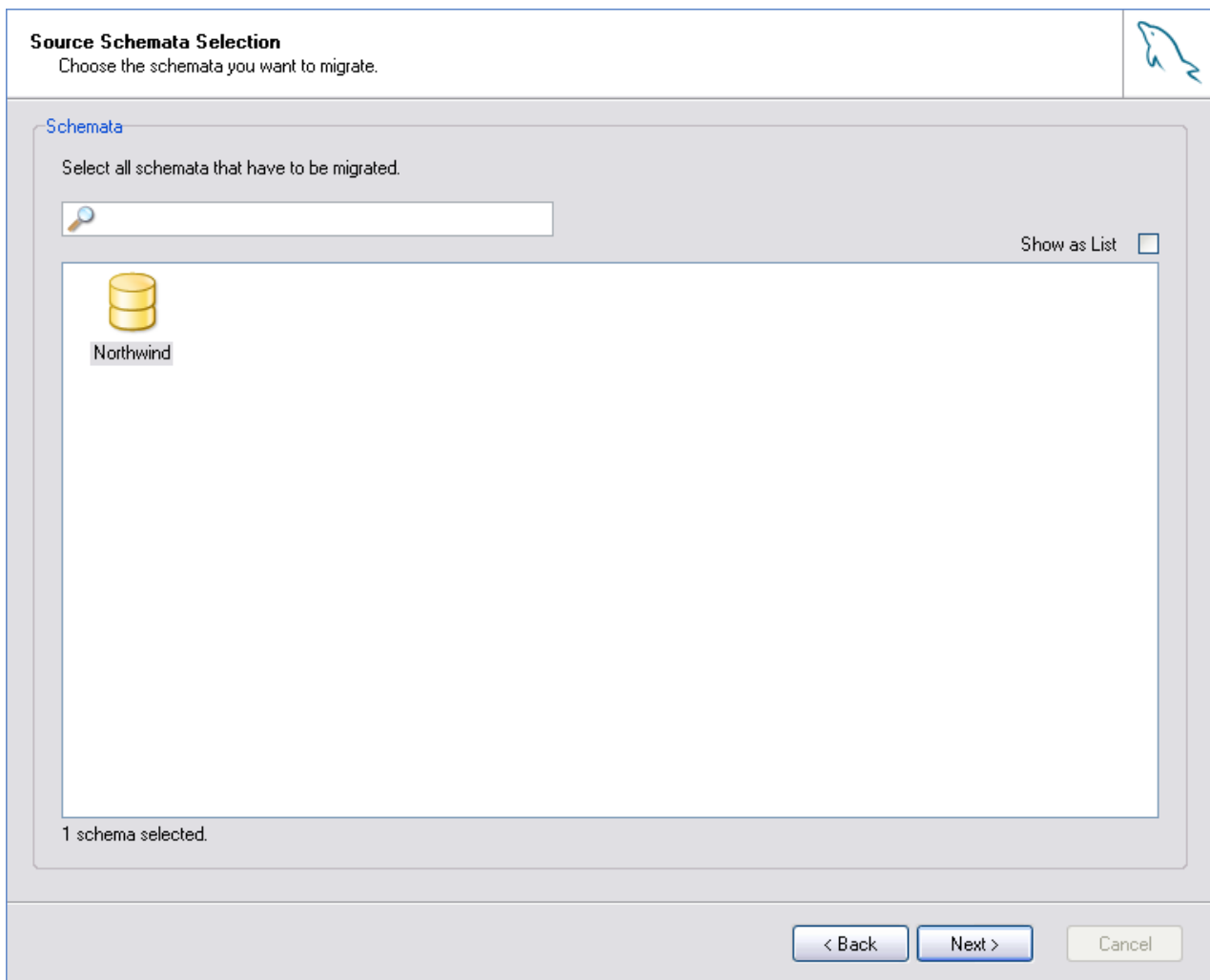
Click the DETAILS button to see a more detailed log of the connection process.

27.7. The Source Schema Selection Screen

Use the Source Schema Selection screen to choose which databases from the source server you would like to migrate.

If there are a large number of databases to choose from, you can search for a specific database by entering the database name in the `schemata` textbox as shown in the following image.

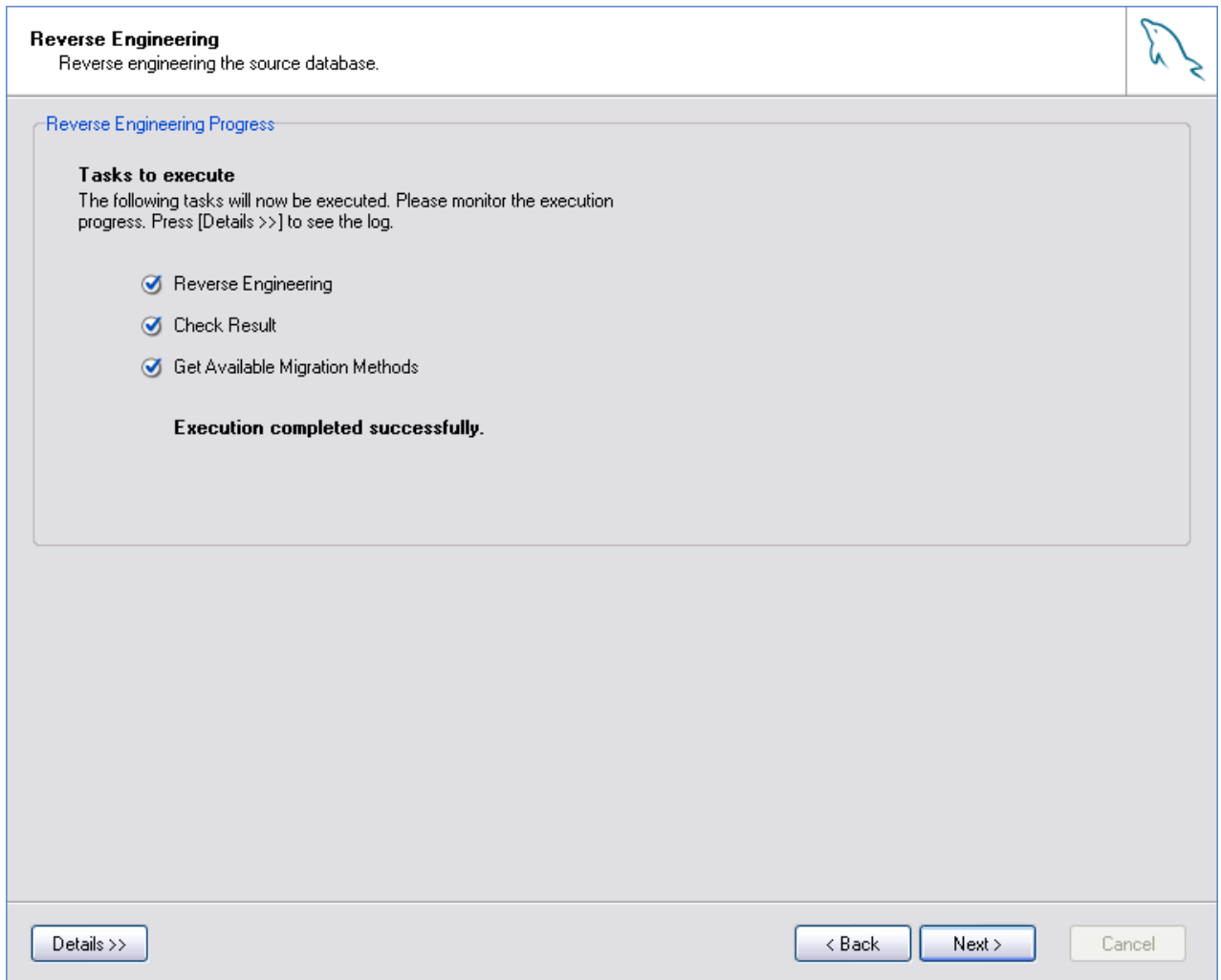
Figure 27.10. The Source Schema Selection screen



27.8. The Reverse Engineering Screen

Once you select the databases you wish to migrate, the MySQL Migration Toolkit begins the process of reverse engineering the source database:

Figure 27.11. The Reverse Engineering screen



The MySQL Migration Toolkit collects column and index information for each table in the source database, along with information on stored procedures and views.

Click the DETAILS button to see a detailed log of the reverse engineering process.

27.9. The Object Type Selection Screen

Use the Object Type Selection screen to choose which objects you wish to migrate:

Figure 27.12. The Object Type Selection screen

Object Type Selection
Select all object types that have to be migrated.

Migrate objects of type Table
Objects of type: **Table**
Number to migrate: **8 / 8**
If you do not want to migrate all objects use the [Detailed Selection] button.
Detailed selection >>

Migrate objects of type View
Objects of type: **View**
Number to migrate: **21 / 21**
If you do not want to migrate all objects use the [Detailed Selection] button.
Detailed selection >>

Migrate objects of type Stored Procedure
Objects of type: **Stored Procedure**
Number to migrate: **0 / 0**
If you do not want to migrate all objects use the [Detailed Selection] button.
Detailed selection >>

< Back Next > Cancel

Check the box next to the objects types (Tables, Views, Stored Procedures) that you wish to migrate.

27.9.1. Migrating a Sub-Set of Object Types

If you only wish to migrate a sub-set of the available object types, click the DETAILED SELECTION button next to the object type:

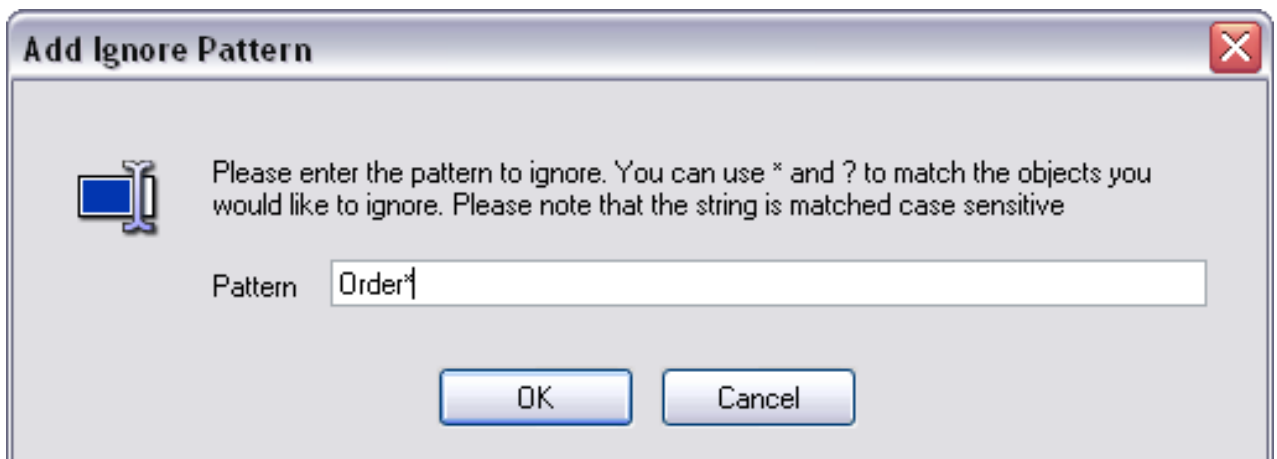
Figure 27.13. The detail view of the Object Type Selection screen



The objects in the left window will be migrated. To ignore an object, select it and click the > button. To move an object out of the ignore list, select it and click the < button.

To exclude objects based on pattern matching, click the + button:

Figure 27.14. The Add Ignore Pattern dialog



Patterns can include the * and ? characters, with * representing multiple characters (including no characters at all) and ? representing a single character.

27.10. The Object Mapping Screen

Use the Object Mapping screen to configure the methods used to migrate the database objects to MySQL. The MySQL Migration Toolkit comes with multiple migration methods that can be used to convert the database objects of an external RDBMS to MySQL.

In most cases the default settings will be adequate.

There are four areas of migration to be addressed: Generic RunTime (GRT) Object, Table, View, and Stored Procedures:

Figure 27.15. The Object Mapping screen

Object Mapping

Please define how to map the database objects.

Migration of type Schema

Migration method: **Generic** Generic method to migrate a schema to MySQL.

Parameter: **Latin1** Use this parameter group to use Latin1 as default character set for the schema.

Multilanguage Use this parameter group to use UTF8 as default character set for the schema.

User defined

Migration of type Table

Migration method: **Generic** Generic method to migrate a table to MySQL.

Parameter: **Data consistency** Standard parameter group. The migrated tables will use the InnoDB storage engine to offer transactional and foreign key support.

Statistical data Choose this parameter group for tables that contain lots of data which does not need transaction safety. This method is ideal for logging information or statistical data.

Data consistency / multilanguage The migrated tables will use the InnoDB storage engine to offer transactional and foreign key support and use UTF8 as default charset.

User defined

27.10.1. GRT Object

The GRT Object section of the Object Mapping screen dictates the properties of the database itself. By default a generic profile is used, with a **Latin1** character set.

To modify the character set used on the database level, click the SET PARAMETER button. Choose from **Latin1**, **Multilanguage**, or **User defined**.

27.10.2. Table Objects

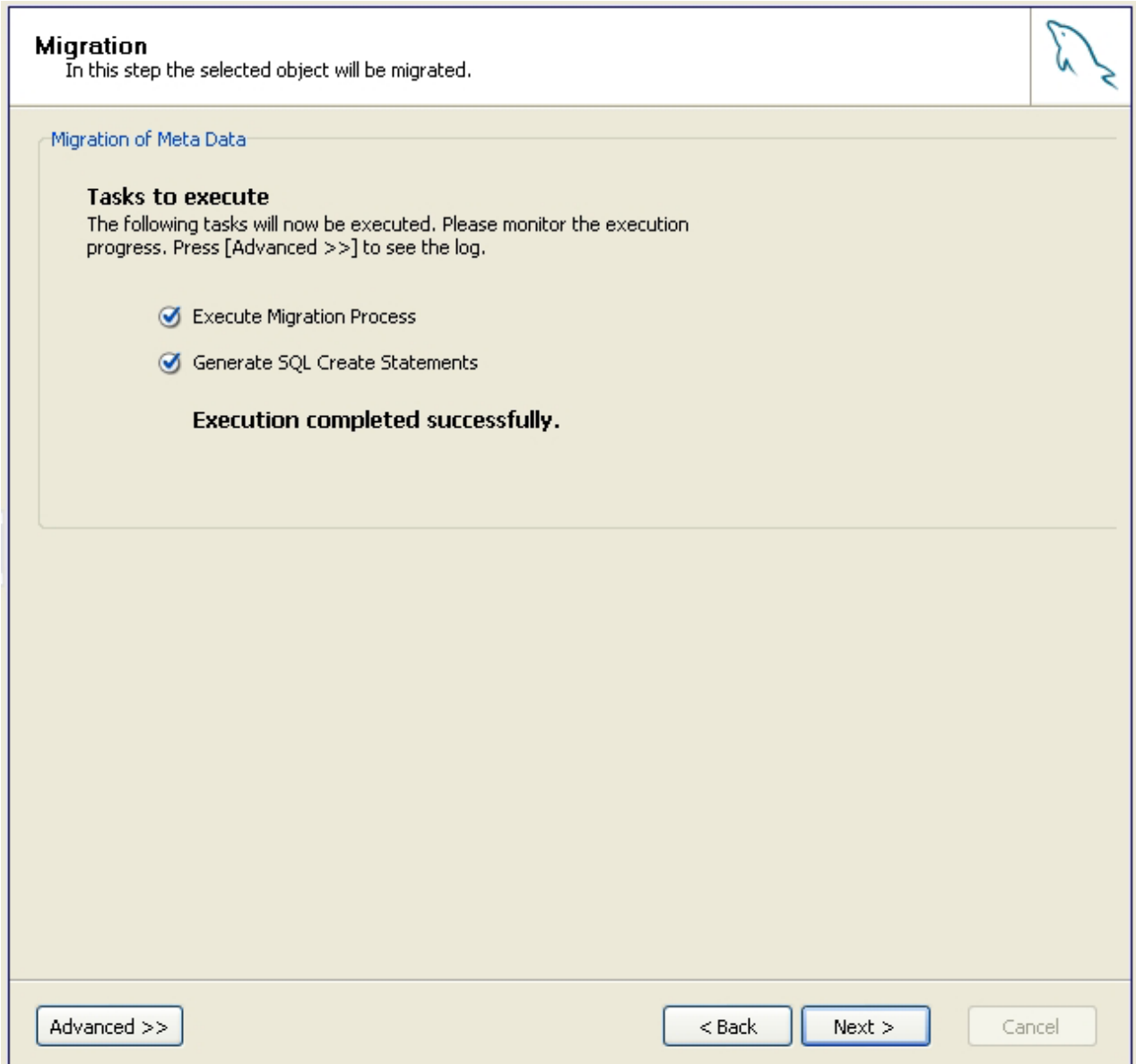
The Table section of the Object Mapping screen dictates the properties of the individual tables. By default a generic profile is used, with an **InnoDB** storage engine.

To modify the storage engine used with the migrated tables, click the SET PARAMETER button. Choose the **Data consistency** option to use the InnoDB storage engine for transactional and foreign key support. Choose the **Statistical data** option to use the MyISAM storage engine with increased performance but no transaction safety. Choose the **Data consistency / multilanguage** option to use the InnoDB storage engine with UTF8 as the default charset. If none of the provided options meet your needs, choose the **User defined option** and **provide your own settings**.

27.11. The Meta Migration Screen

After you configure data object mapping, the MySQL Migration Toolkit performs the conversion of the database objects and generates SQL `CREATE` statements.

Figure 27.16. The Meta Migration screen



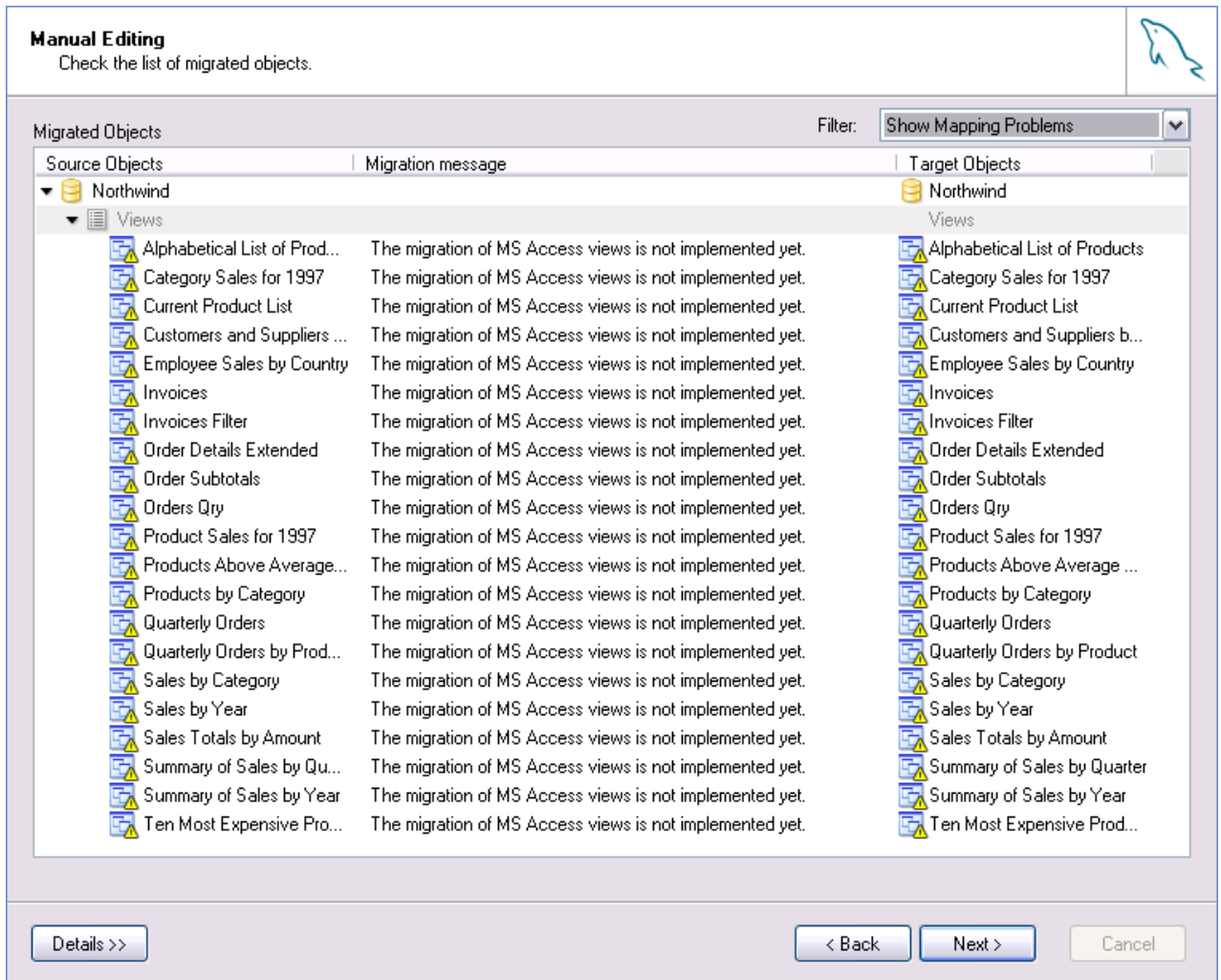
The SQL `CREATE` statements are not executed on the target server at this stage of the migration, but will be executed later.

Click the `DETAILS` button to view a detailed log of this stage of the migration process.

27.12. The Manual Editing Screen

Use the Manual Editing screen to review the SQL `CREATE` statements generated by the MySQL Migration Toolkit:

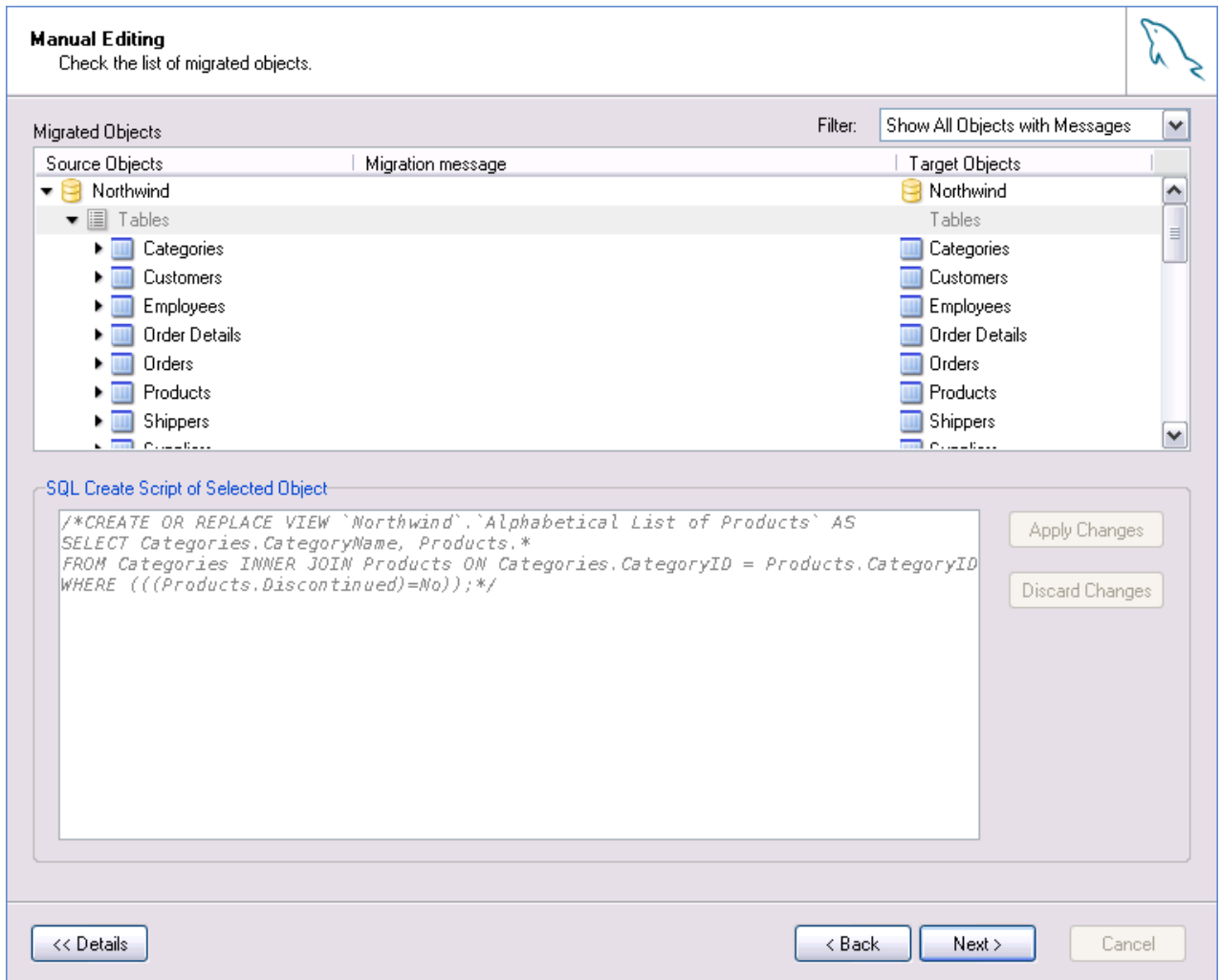
Figure 27.17. The Manual Editing screen



By default only objects that were not successfully converted are displayed in the Manual Editing screen. To view all objects select the **SHOW ALL OBJECTS** entry of the **FILTER** drop-down list. Choose the **SHOW ALL OBJECTS WITH MESSAGES** entry of the **FILTER** drop-down list to view all object with status messages.

To edit the SQL **CREATE** statements created by the MySQL Migration Toolkit, select the object and click the **DETAILS** button:

Figure 27.18. The Manual Editing screen – detailed view

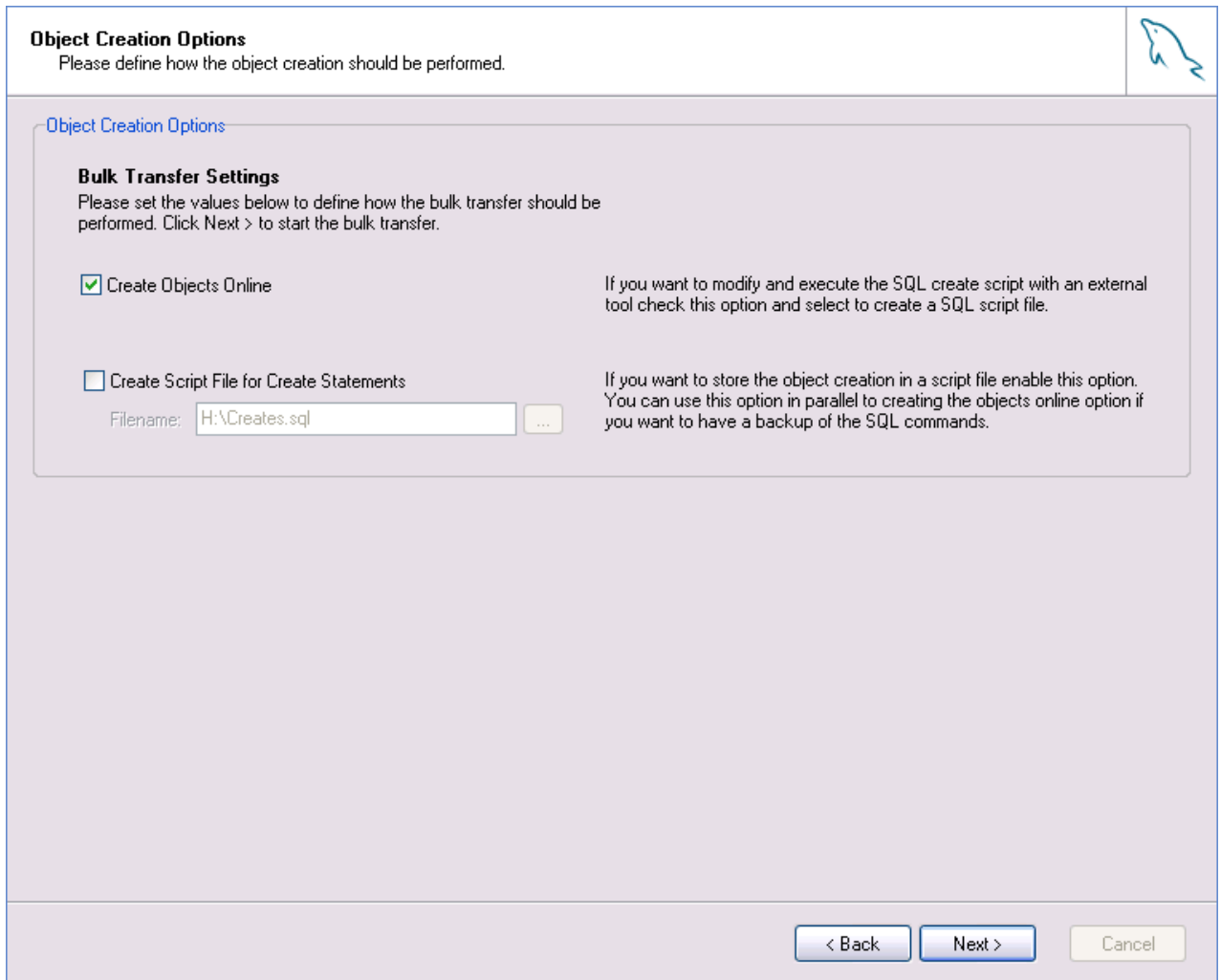


Make changes to the `CREATE` statement and click the `APPLY CHANGES` button. If you make a mistake while editing, click the `DISCARD CHANGES` button to undo the `CREATE` statement.

27.13. The Object Creation Options Screen

After performing manual object editing the MySQL Migration Toolkit is ready to create the database objects on the target server. You have the option of either creating the database objects directly on the target server or to create a script file of the `CREATE` statements for later execution:

Figure 27.19. The Object Creation Options screen

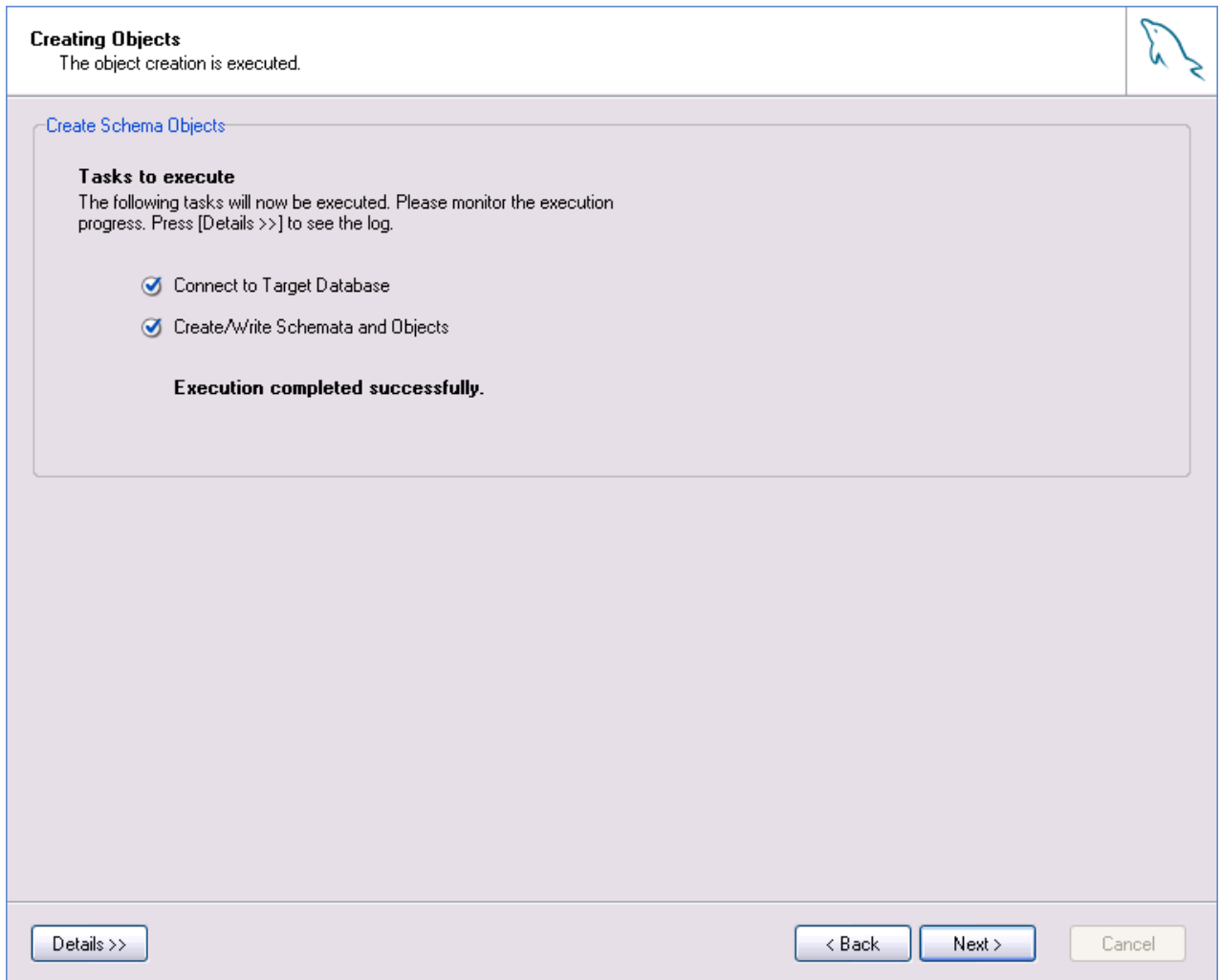


Select both options to create the target database objects and preserve a backup copy of the `CREATE` statements for later use.

27.14. The Creating Objects Screen

Once you choose object creation objects, the MySQL Migration Toolkit connects to the target server and creates the database objects (assuming you chose to have the MySQL Migration Toolkit connect to the target server to create the database objects).

Figure 27.20. The Creating Objects screen



Click the DETAILS button to view a detailed log of this stage of the migration process.

Choose the [Create Script File for Create Statements](#) check box to save a copy of the `CREATE` statements to disk.

27.15. The Data Mapping Options Screen

After the database objects are created on the target database server the MySQL Migration Toolkit is ready to move the server data to the target server. You have the option of either inserting the data directly on the target server or to create a script file of the `INSERT` statements for later execution:

Figure 27.21. The Data Mapping Options screen

Data Mapping Options
The selected object will now be migrated.

Standard Options

Bulk Transfer Settings
Please set the values below to define how the bulk transfer should be performed. Click Next > to start the bulk transfer.

Transfer Data Online
Keep this option selected to transfer the data now. If you want to transfer the data at a later point in time deactivate this option and enable the option to create a script file instead.

Create Script File for Insert Statements
If you want to store the data in a script file enable this option. You can use this option in parallel to the bulk transfer option if you want to have a backup of the transferred data.

Filename: ...

Details >> < Back Next > Cancel

Select both options to move the data and preserve a backup copy of the `INSERT` statements for later use.

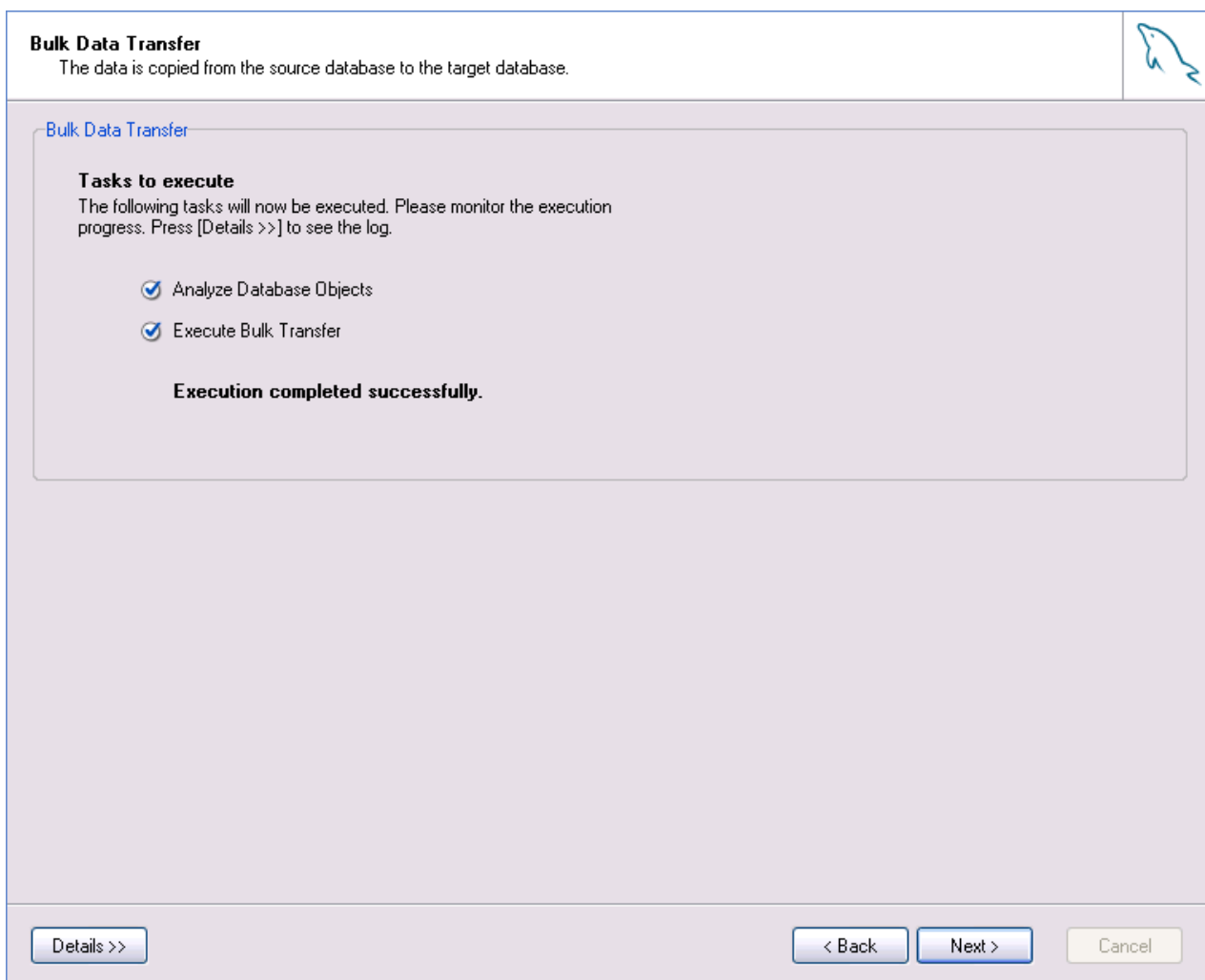
You can access additional options by clicking the `DETAILS` button. If you do not wish to move `BLOB` data to the target server, check the box next to the `EXCLUDE BLOB VALUES` option. The `BLOB` data will not be moved to the target server and will not be written to the script file. If you do not wish to move `BLOB` data to the target server, but wish to have the `BLOB` data written to the script file, also check the box next to the `WRITE BLOBS TO INSERT SCRIPT` option.

If you wish to limit the number of rows transferred to the target server, check the box next to the `MAXIMUM NUMBERS OF ROWS TO TRANSFER FOR A TABLE` option and enter the desired number of rows. This option can be useful when generating test data.

27.16. The Bulk Data Transfer Screen

Once the data mapping options are set, the MySQL Migration Toolkit will begin the bulk data transfer process. Data will be converted to a MySQL compatible format and inserted into the target database server using bulk `INSERT` statements. Data is typically inserted in batches of 15,000 rows at a time to maximize insertion speed.

Figure 27.22. The Bulk Data Transfer screen

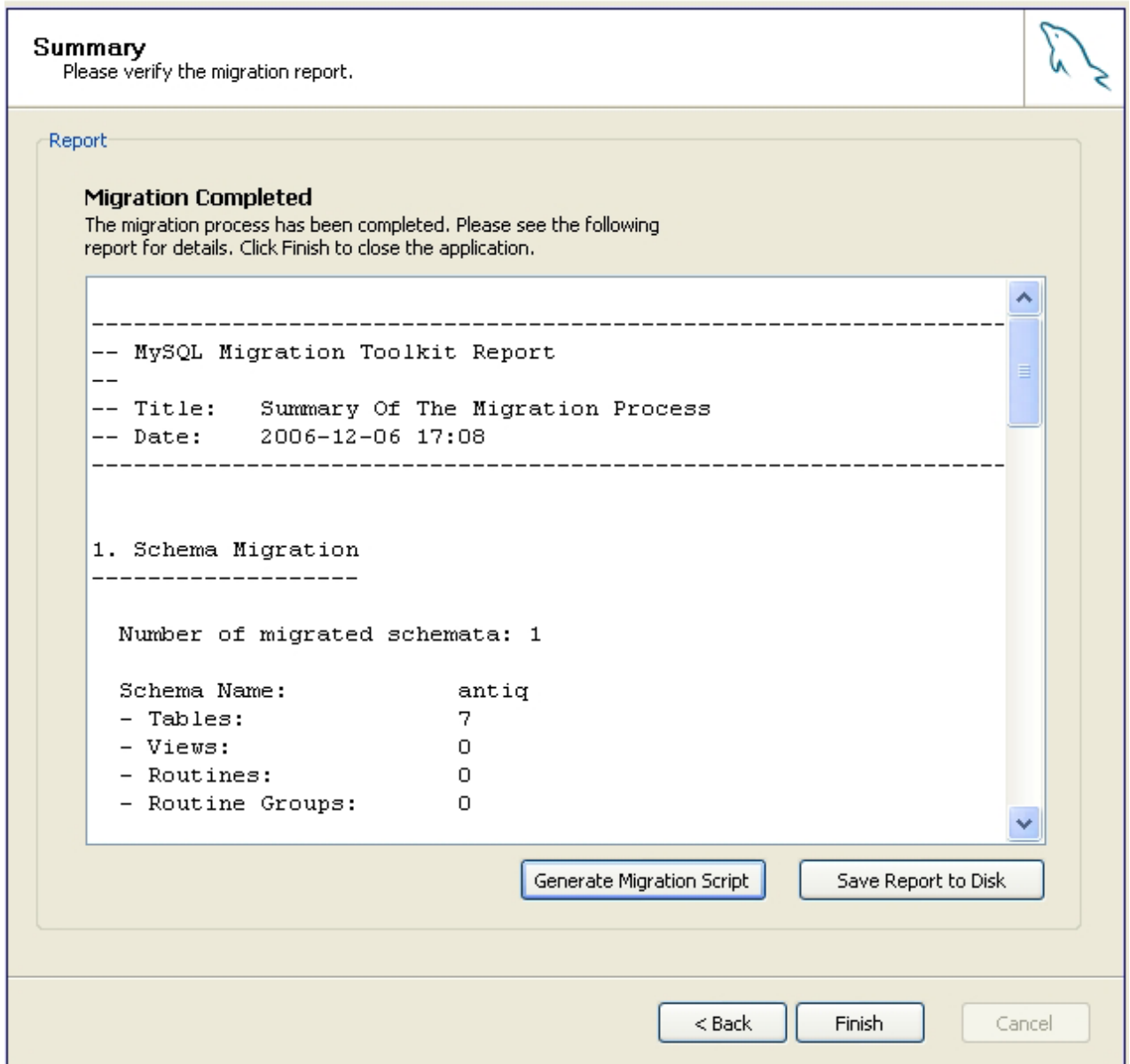


Click the DETAILS button to view a detailed log of this stage of the migration process.

27.17. The Summary Screen

Once the bulk data transfer is complete, a summary report of the migration will be displayed:

Figure 27.23. The Summary screen



Click the **GENERATE MIGRATION SCRIPT** to create a `Lua` script file of the migration process. With this file you can script the migration of a database or customize it. For more information about this topic, see [Chapter 28, Scripted Migration](#).

Click the **FINISH** button to exit the MySQL Migration Toolkit.

27.18. Saving the Current Application State

If you need to close the MySQL Migration Toolkit part way through the migration process, you can save the state of the MySQL Migration Toolkit to disk and continue later.

To save the current state of the MySQL Migration Toolkit, choose the **STORE CURRENT APPLICATION STATE** option of the **FILE** menu.

To later retrieve the application state and resume configuring the migration, choose the **RELOAD STORED APPLICATION STATE** option of the **FILE** menu.

Saving the application state will prove useful when examining scripted migration in [Chapter 28, Scripted Migration](#).

Chapter 28. Scripted Migration

This section reviews a simple migration script and assumes some familiarity with the GRT shell. If you haven't yet done so, read [Chapter 31, The Generic Runtime Environment \(GRT\) Shell](#).

As noted earlier in [Section 27.17, "The Summary Screen"](#), if you choose, you can generate a migration script when migrating a database. Doing this creates a Lua script of the entire migration process. You can find out more about Lua by going to lua.org. However, if you have some familiarity with programming you should be able to make sense of this script without too much effort.

Note

The MySQL Migration Toolkit supports Lua version 5.0. Version 5.1 is not supported.

For a better understanding of the migration script, it is useful to have the GRT shell open within the MySQL Migration Toolkit. In this way you can examine the various objects created during migration by clicking on them in the Globals Tree panel. To save the application state and have access to these objects, migrate a database as described in [Chapter 27, The Migration Process In-Depth](#), and when you reach the [Summary](#) step, choose the menu options [FILE, STORE CURRENT APPLICATION STATE ...](#). This will save an XML file of the entire migration process. You can now reload the migration state whenever you wish.

28.1. The Steps for Scripted Migration

The migration script is conveniently divided up into six sections or checkpoints:

1. Set Source and Target Connection
2. Do the Reverse Engineering
3. Migration Methods and Ignore List
4. Set Object Mappings and Do Migration
5. Generate and Execute SQL Create Statements
6. Bulk Data Transfer

Each checkpoint will be examined in turn. To reload the application state and have access to the objects created during migration, choose the menu options [FILE, RELOAD STORED APPLICATION STATE ...](#). Find the XML file that you previously saved. Open this file and you will see how the various objects in the GRT Globals Tree are referenced from the shell. Press **F4** to open the GRT shell.

28.2. Setting the Source and Target Connection

With the GRT shell open and the **VALUES** tab selected, double click the `migration` object and find the `sourceConnection` object. Click this item and its parameters and the values of those parameters will be exposed in the frame beneath the Globals tree. The parameters are as follows:

- `_id`
- `driver`
- `modules`
- `name`
- `parameter values`

Note

If you cannot find a `sourceConnection` object then the application state has not been reloaded.

Open the Lua script that you generated during migration and find the `-- Set source connection` line. Immediately below this line is the code that defines the source connection of the schema that is being migrated.

The names of the parameters of the source connection in the Lua script should match the parameters shown in the frame below the Globals tree (though they do not appear in the same order). The value shown for the `_id` parameter is the value created by the `newGuid` method of the `grt` object. The Universally Unique Identifier (UUID) of the driver show in this frame should match the

value shown in your script.

The `name` is the name of the variable being created by the Lua script.

In the `Values` tab both the `modules` and `parameterValues` items show as objects of the `dict` type.

Click on `modules` to see the modules used during migration. The `MigrationModule` has a name specific to the migration source schema. If you are migrating from Access, for example, it is called `MigrationAccess`. Click on `parameterValues` and see the parameters you supplied for connecting to the source schema.

Below the `-- set struct and types` line find the definition of the data types used so far. A `sourceConnection` is a struct derived from the `db.mgmt.Connection` structure. `modules` and `parameterValues` are both strings.

The parameters and the modules used for a `targetConnection` are shown beneath the `--Set target connection` line. After examining a `sourceConnection` you should quickly be able to understand a `targetConnection`. The same applies to the structs and data types used by the `targetConnection`.

28.3. Reverse Engineering

The conversion of the source schema to a MySQL schema occurs in the lines immediately following the comment `-- Do the reverse engineering`. This calls the GRT reverse engineering module to convert from the current schema type to the target type.

Note

To convert GRT objects to Lua values requires using the `toLua` method. Future versions of the GRT may overload the assignment operator.

The reverse engineering modules used by the MySQL Migration Toolkit are also used by the MySQL Workbench.

28.4. Migration Methods

The code following the comment `-- Migration methods and ignore list` invokes the migration module appropriate to the source database.

If you are migrating from Access, for instance, the `MigrationAccess` module will be invoked. Find this module in the `modules` tab to examine its three methods:

- `migrationMethods`
- `migrate`
- `dataBulkTransfer`

Any schema objects that you chose not to migrate, show up in the `ignoreList` object. This object is found beneath the `migration` object in the `Globals` tree panel.

28.5. Map Objects and Migrate

Instead of finding an object in the `Values` tab of the `Objects` tree, you may query an object from the command line in the GRT shell. For example, in the code following the comment `-- Set object mappings and do migration` a `mappingDefaults` object is created by the `grtV.setGlobal` method.

To see that this method has executed successfully enter the following command in the GRT shell:

```
/ > print(grtV.getGlobal("/migration/mappingDefaults"))
```

This should output a listing of the mapping defaults as shown in your script file. The results will vary depending upon the objects you have chosen to migrate. However, in all cases you should see a schema mapping.

Beneath the `-- update _ids` comment the migration methods are copied to a local variable and then executed.

28.6. The SQL Create Statements

The code to create the target schema follows the `-- Generate and execute sql create statements` comment. If you chose the `Create Script File for Create Statements` option on the `OBJECT CREATION OPTIONS` screen then

a text file of the data definition statements required to create the migrated schema is written to a file. For information on setting this option see [Section 27.13, “The Object Creation Options Screen”](#).

The actual creation of the schema on the database server is done by the `transformationModule` module.

28.7. Bulk Data Transfer

The code to populate the target schema follows the `-- Bulk data transfer` comment. If you chose the `Create Script File for Insert Statements` option on the **DATA MAPPING OPTIONS** screen then a text file of the insert statements required to populate the migrated schema is written to a file. For more information on setting this option, see [Section 27.15, “The Data Mapping Options Screen”](#).

The actual creation of the data in the new schema is done by the `migrationModule` module.

Chapter 29. Extending The MySQL Migration Toolkit

29.1. Introduction

One of the key features of the MySQL Migration Toolkit is that it can be easily extended and customized to support new data sources. This is done through the use of its Generic RunTime (GRT) Environment.

Extending the MySQL Migration Toolkit to support a new RDBMS requires the creation of two new modules: one module that reverse engineers (ie., retrieves schema information) the source database objects and converts them into GRT objects, and one module that migrates the source GRT objects into MySQL GRT objects. The MySQL Migration Toolkit then converts the GRT MySQL objects into SQL statements that create the target MySQL server objects.

Existing modules can be easily expanded and customized to achieve the perfect migration for individual requirements. New migration “methods” that define how the migration is performed can be added easily. The new methods are listed and can be selected from the Wizard interface or used in migration scripts.

29.2. Architecture of the MySQL Migration Toolkit

The MySQL Migration Toolkit is built on three primary modules: reverse engineer modules, migration modules, and transformation modules.

Reverse engineering modules retrieve the schema information from the source database and return GRT objects that describe the schema. Reverse engineering modules will have a name similar to `ReverseEngineeringAccess`.

Migration modules convert the source database GRT objects to MySQL GRT objects and then handle the bulk data transfer between the source and MySQL databases. Migration modules will have a name similar to `MigrationAccess`.

Transformation modules convert the MySQL GRT objects into the actual SQL statements used to create objects such as tables and views on the target MySQL server. Transformation modules will have a name similar to `TransformationMySQL`. Transformation modules are supplied by the MySQL GUI team and need not be created to add support for a new source database.

All modules are derived from base classes whose methods can be rewritten to match the new source database.

29.3. The Modular Migration Process

From the point of view of modular development, the migration process is as follows:

1. The MySQL Migration Toolkit prompts the user for a source database and connection parameters. The user selection determines which modules will be used for the source database.
2. The MySQL Migration Toolkit calls the `testConnection` method of the reverse engineering module. The `testConnection` method returns success or failure to the MySQL Migration Toolkit.
3. The MySQL Migration Toolkit calls the `getSchemata` method of the reverse engineering module. The `getSchemata` method returns a list of the schemata available on the source RDBMS.
4. Once the user has selected one or more schemata to migrate, the MySQL Migration Toolkit calls the `reverseEngineer` method of the reverse engineering module. The `reverseEngineer` method converts all objects in the source RDBMS (tables, views, procedures) into a collection of GRT objects.
5. After the source database has been reverse engineered, the MySQL Migration Toolkit calls the `migrationMethods` method of the migration module. The `migrationMethods` method generates a list of available methods than can be selected by the user in the `Object Mapping Screen`.
6. Once the user has selected the migration methods, the MySQL Migration Toolkit calls the `migrate` method of the migration module. The `migrate` method converts the source RDBMS GRT objects into MySQL GRT objects by calling the selected migration method for each source schema object. If no explicit method is assigned, the default migration method for the object type will be used.
7. After the MySQL GRT objects have been created, the MySQL Migration Toolkit calls the transformation module to convert the MySQL GRT objects into SQL statements that will create the objects on the target MySQL server. The MySQL Migration Toolkit then either executes the SQL statements on the target MySQL server or writes them to a script file.
8. In the final step, the MySQL Migration Toolkit calls the `dataBulkTransfer` method of the migration module. The `dataBulkTransfer` method loops through the selected tables in the selected schema and migrates the rows of the tables to the target MySQL database or loads them into a script file, depending on the user preference.

29.4. Tools Required to Extend the MySQL Migration Toolkit

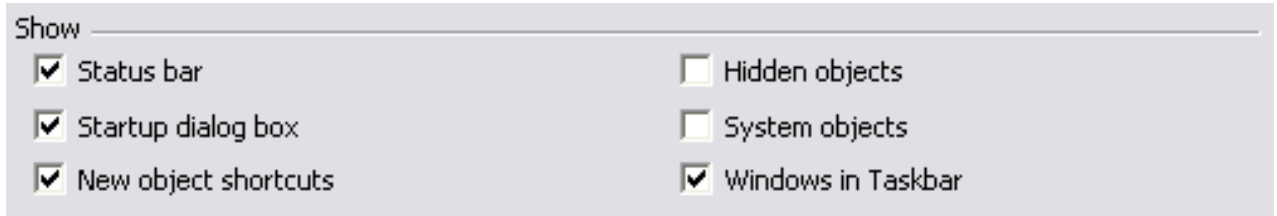
The MySQL AB GUI Team uses Eclipse for development of the MySQL Migration Toolkit modules and recommends Eclipse for use when developing modules for the MySQL Migration Toolkit. See <http://www.eclipse.org/> for more information.

Chapter 30. Preparing a Microsoft Access Database for Migration

The MySQL Migration Toolkit requires access to the system tables of a Microsoft Access database for the purpose of reverse-engineering. By default, read access to the system tables of an Access database is restricted for external applications.

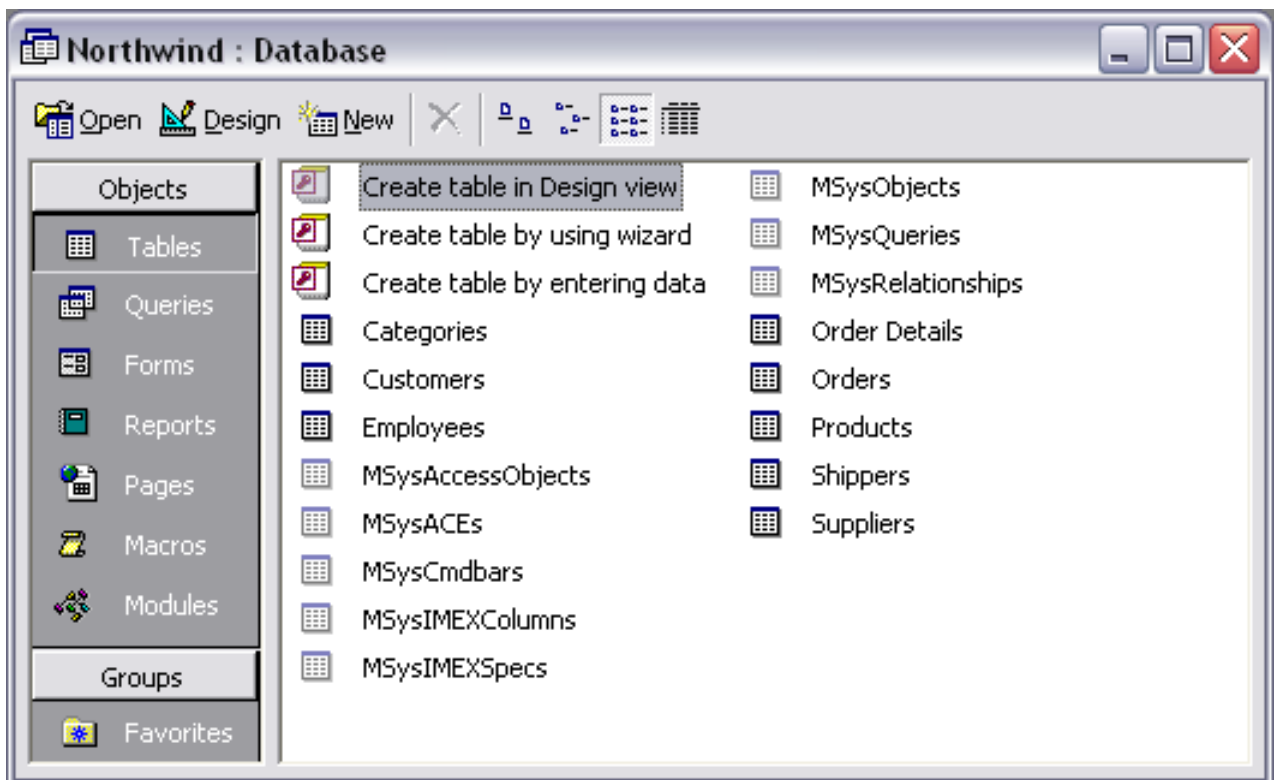
To enable access to the system tables by the MySQL Migration Toolkit, open the database in Microsoft Access and choose the **OPTIONS** entry of the **TOOLS** menu. Within the **Options** dialog, select the **VIEW** tab and look for the **Show** section:

Figure 30.1. The show section



Check the box next to the **SYSTEM OBJECTS** option and close the options dialog. System tables for the database should now be present:

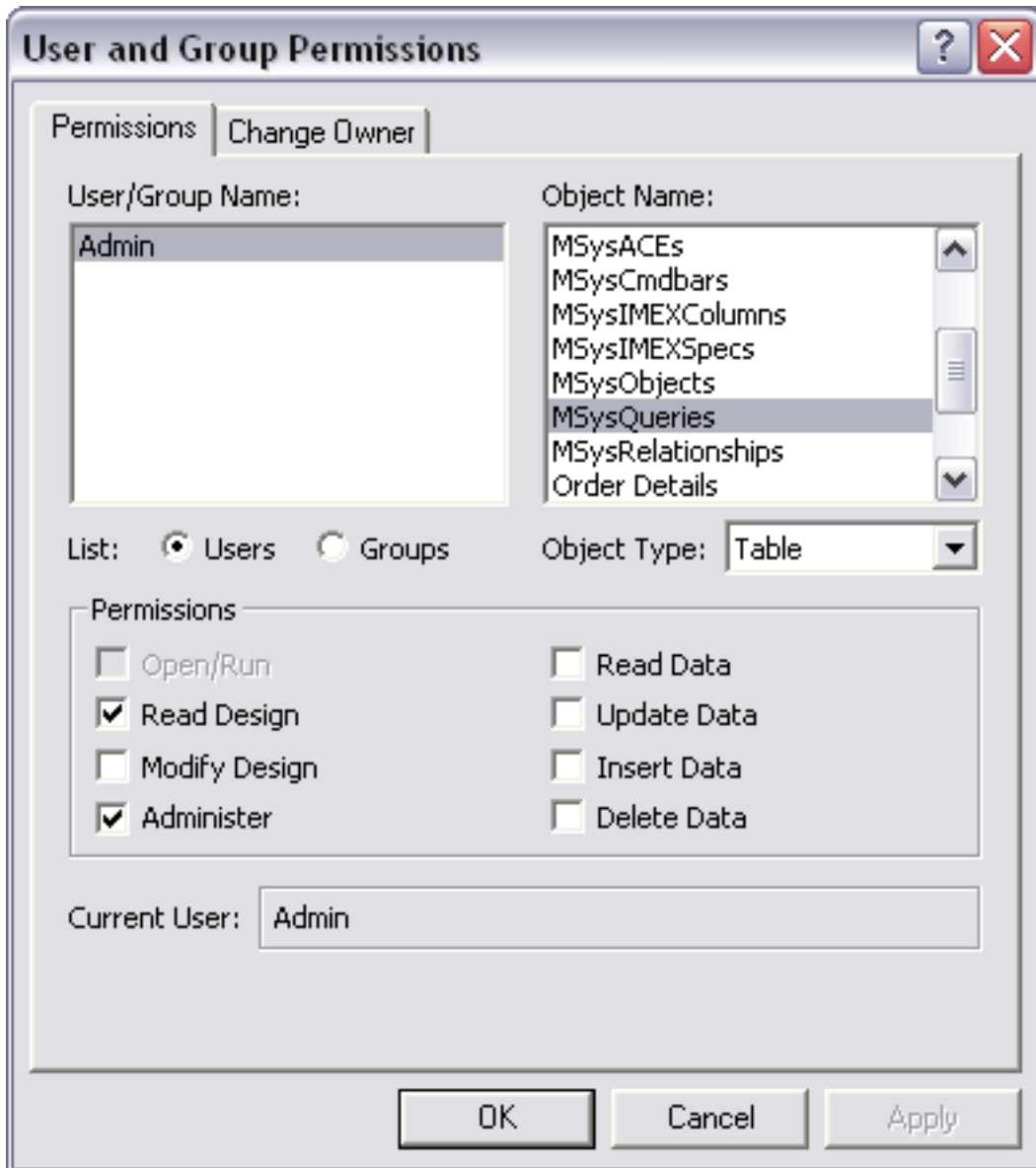
Figure 30.2. The system objects



Note the presence of the various **MSys** tables.

After you expose the system objects to the MySQL Migration Toolkit, you must also grant permission to access the objects. Choose the **USER AND GROUP PERMISSIONS** entry from the **SECURITY** section of the **TOOLS** menu:

Figure 30.3. Granting access to the system objects



Enable the [Administer](#) permission for both the [MSysObjects](#), [MSysQueries](#), and [MSysRelationships](#) tables. You will need to click the **APPLY** button after enabling each table and before moving to the next table.

After completing these steps, you can reverse engineer the schema in your Access database using the MySQL Migration Toolkit.

Chapter 31. The Generic Runtime Environment (GRT) Shell

31.1. Introduction

The GRT is a thin C layer, inspired by Objective C, which allows for dynamic typing and dynamic data objects. The GRT is used by the MySQL Migration Toolkit and provides a means for expanding these tools. Through the use of the GRT, these tools can support new behavior and data sources using code written in languages such as C, C++, Java, Python, and Lua with support for Mono forthcoming.

The GRT is not only useful for expanding MySQL GUI Tools. By using a script file from within the GRT shell you can perform repetitive tasks programmatically from the command line. Suppose, for example, that you have multiple schemata that you wish to migrate to MySQL. You can do this once using the graphical interface, in the process saving the procedure as a script file. You can then adapt this script file and run it unattended from the GRT shell.

The preferred development language is [Lua](#), a lightweight scripting language expressly designed for extending applications. For more information about this language see [lua.org](#).

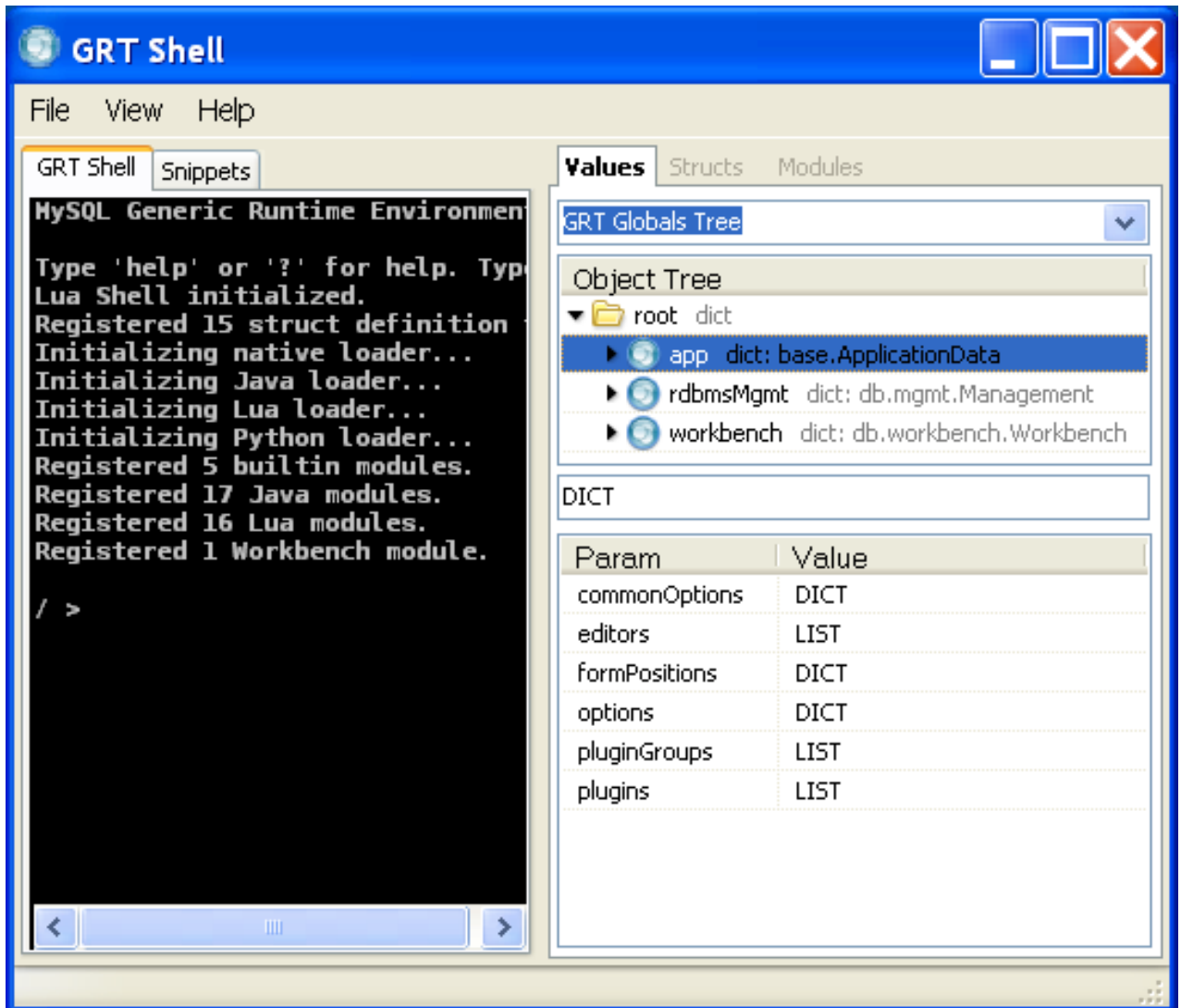
Note

The GRT supports Lua version 5.0. Version 5.1 is not supported.

31.2. Exploring the GRT Shell

To open the GRT shell from within the MySQL Migration Toolkit choose the GRT ENVIRONMENT SHELL option under the TOOLS menu. I

Figure 31.1. The GRT shell (Windows)



The GRT shell itself is the default tab on the left of the screen. Beside it is the [Snippets](#) tab, used for saving code snippets.

On the right, is the [GRT Globals Tree](#)—showing the various objects, variables, structures, and code modules used by the application. These objects are all directly accessible from the shell.

31.2.1. Menu Items

The menu runs across the top of the screen and varies significantly depending upon which OS you are using. Since the MySQL Migration Toolkit is currently only implemented on Windows, all references to invoking the GRT shell from this application apply only to the Windows OS.

31.2.1.1. The File or Shell Menu

31.2.1.1.1. Windows File Menu

Currently only two of the submenus under the [FILE](#) menu are implemented; [EXIT](#), which simply exits the GRT shell, and [OPEN SCRIPT](#). The [OPEN SCRIPT](#) option is for opening existing [Lua](#) scripts and is only implemented under Windows.

Sample scripts are found in the [Scripts](#) directory which is immediately below the installation directory. Opening a script from this menu option will create an additional tab labeled with the name of the script. Click on this tab to see the file contents. You may paste text between the [GRT Shell](#) tab and any script tab.

31.2.1.1.2. Linux Shell Menu

Under Linux the [SHELL](#) menu shows menu items for closing the shell, refreshing the objects listed in the Global tree or saving the object tree.

The [SAVE TREE...](#) menu option saves an XML file of all the data types and classes shown in the three tabs of the [Objects Tree](#).

Under Linux there is no menu option to open a script.

31.2.1.2. The View Menu

31.2.1.2.1. The View Menu: Windows

The [REFRESH](#) option of the [VIEW](#) menu refreshes the view of the objects shown in the object tree tabs on the right.

The [DISPLAY TYPE INFO](#) and [DISPLAY OBJECT VALUES](#) menu items toggle the view of the objects in the [Values](#) tab on the right. When checked, [DISPLAY TYPE INFO](#) shows the data type of objects and [DISPLAY OBJECT VALUES](#) shows their value.

The [DISPLAY OBJECT REFCOUNT](#) shows the current number of references to specific objects.

31.2.1.2.2. The View Menu: Linux

The [VIEW](#) menu has only one element, [STRUCTURE ONLY](#) and it is not yet activated.

31.2.1.3. Mac OS X Menu

Under Mac OS X there are only two menu options, [RELOAD](#) and [RELOAD SELECTED](#). [RELOAD](#) reloads all the objects in the Object Tree panel and is equivalent to the Windows [VIEW](#), [REFRESH](#) option. The [RELOAD SELECTED](#) option only reloads the selected object.

31.2.2. The Shell

The GRT shell is principally used for running Lua scripts or typing Lua commands directly. However, you can also access the GRT Scripting Library functions and global functions and objects. To see the available commands type “?”.

Some OS-specific commands are also available. For instance, under Windows you can clear the screen by typing `cls`. Unlike most shells, you can cut and paste text to and from the shell window.

Working from the command line is described in detail in [Section 31.3, “Using the GRT Shell”](#).

The [Snippets](#) tab functions as a scratch pad for saving code snippets. This makes it easy to reuse code and does away with the need to retype it at the command line.

If you have opened script files as described in [Section 31.2.1.1.1, “Windows File Menu” \[113\]](#), there may be any number of tabs to the right of the [Snippets](#) tab. These tabs will be labeled with the names of the script files. As with the [Snippets](#) tab you can cut and paste to or from any of the tabs. This gives you the opportunity to test code from the command line.

31.2.3. The Globals Tree Panel

The [Globals Tree](#) is found on the right side of the screen and is made up of three tabs, [Values](#), [Structs](#), and [Modules](#).

31.2.3.1. The Values Tab

If you are running MySQL Migration Toolkit, find a [migration](#) object beneath the [root](#) object. Both applications show the [rdBmsMgmt](#) object.

When the [Values](#) tab is selected right clicking an object in the Globals Tree panel opens a pop-up menu with the options:

- Refresh
- Remove Object
- Display Type Info
- Display Object Values
- Display Object RefCount

With the exception of [REMOVE OBJECT](#), these options are the same as those shown in [Section 31.2.1.2, “The View Menu”](#). You may remove any object except the [root](#) object.

■ Note

Note this pop-up menu only shows under Windows.

31.2.3.2. The **Struct** Tab

A **struct** is a user-defined data type formed by combining primitive data types. This tab shows the definitions of the structs used by the objects in the **Values** tab and the modules in the **Modules** tab.

When the **Structs** tab is selected right clicking a structure in the list opens a pop-up menu with the options:

- Order by Name
- Order by Hierarchy
- Order by Package

Note

Note this pop-up menu only shows under Windows.

The default view for this tab is by package, a grouping of elements by functionality. Double-click a package to show related structures. Under `db.mgmt`, for example, you should see elements you are already familiar with from the user interface, `Connection`, `Driver`, and so forth. If an element can be further decomposed, an arrow will show on its left. Double-click the item to reveal its constituent elements.

If you switch to the hierarchical view you'll find the `db.mgmt.driver` object under the **GRT Object** since this is the parent object from which it is derived.

Ordering by name simply shows all the different objects arranged alphabetically.

31.2.3.3. The **Modules** Tab

A module can be either a Python or Lua script or a Java class file. Information about modules appears in the window below the module tree. For example, go to the **Modules** tab and click on the `ReverseEngineeringGeneric` module. Double click a module and you will see its methods.

Double clicking a method name will copy it into the GRT shell window. You will see how useful this can be in [Section 31.3, "Using the GRT Shell"](#).

31.3. Using the GRT Shell

There are three built-in Lua modules that assist working from the GRT shell:

- `grtV` – for accessing any object/variable in the **Values** tab
- `grtS` – for viewing the structs defined in the **Structures** tab
- `grtM` – for accessing any object in the **Modules** tab

All items in all the tabs are accessible from the GRT shell.

The script example below uses the `getGlobal` method of the `grtV` object to return a list of databases and then iterates through this list.

```
dbs = grtV.getGlobal("/rdbmsMgmt/rdbms")
for i = 1, grtV.getn(dbs) do
  print(dbs[i].name)
end
```

The `getGlobal` method returns the object found at the path parameter passed to it. In this case, the object is a list that is traversed using a `for` loop controlled by the `getn` method which returns the number of elements in the database list.

Running this `for` loop outputs the names of the database formats supported by the MySQL Migration Toolkit:

```
"Oracle"
"MySql"
"MaxDB"
"GenericJdbc"
"Mssql"
```

```
"Access"
```

To discover all the methods available for a specific object, type the object name preceded by a “?”. For example typing `?grtV` shows:

```
GRT Value Management Library - grtV
-----
A library that contains functions to work with GRT values.

clearList          child          diffMake
diffApply          duplicate       fromXml
getContentTypes   getKey         getListItemByName
getListRefValueByObjName getn          getGlobal
insert            load           lookupAdd
lookupId          newDict        newList
newObj            remove         save
setContentTypes   setGlobal      toLua
toXml             typeOf

Type 'help grtV.<command>' to get help on a specific command.
```

31.4. Invoking the GRT Shell From the Command Line

Note

This capability is currently only available under Windows.

In addition to using the GRT shell from within the MySQL Migration Toolkit, you can invoke it directly from the command line. If the location of the MySQL GUI Tools is not included in the `PATH` variable, navigate to the installation directory and find the `grtsh.exe` file.

Execute this file by typing:

```
C:\> grtsh -?
```

Do this and you should see the following listing:

```
Usage: C:\Program Files\MySQL\MySQL Tools for 5.0\grtsh.exe [-classpath path] »
[-modulepath path] [-jvm library] [-d path] [-listen port] [-verbose] [-x] [luafile]
C:\Program Files\MySQL\MySQL Tools for 5.0\grtsh.exe -j structsfile outputdir
C:\Program Files\MySQL\MySQL Tools for 5.0\grtsh.exe -p structsfile outputdir

-lua ..... Use the Lua shell (default).
-py ..... Use the Python shell.
-classpath .. Sets the java classpath to the given value.
-modulepath .. Sets the location of the GRT module directory.
-jvm ..... The java virtual machine library to use (with absolute path).
-basedir ..... Path to the data files location.
-d path ..... Modules directory
-x ..... Exits the shell after running the specified file
luafile ..... File that is run at startup.

-listen port . Runs in 'remote agent' mode on the given port number.
-verbose ..... Prints detailed startup information.
-j ..... Generates Java classes from the given structs file.
-p ..... Generates PHP classes from the given structs file.
-D var=value . Sets a global shell variable to the given value.
Environment variables:
GRT_MODULE_PATH Equivalent to -modulepath, must point to the directory
                  where the grtsh binary resides
```

The default shell is the `Lua` shell and is indicated by the `/ >` prompt. Using the `-py` option opens a Python shell, indicated by the `/>>>` prompt.

If you wish to set the classpath for Java classes use the `classpath` option. You may also change the Java Virtual Machine (JVM) by using the `jvm` option with the absolute path to the JVM you wish to use.

The `modulepath` option sets the location of the `dll` files used with the GRT shell. These files are located in the same directory as the `grtsh.exe` file. You can also set this directory by defining the environment variable, `GRT_MODULE_PATH`.

The location of any data files you wish to use may be set using the `basedir` option.

To see the various modules that are loaded at startup use the `verbose` option. The java modules are stored in the `java\com\mysql\grt\modules` directory immediately below the installation directory and the `Lua` modules in the `lua` directory. Currently, importing Python modules is not supported.

To include modules other than the default modules, use the `d` option with a path.

It is also possible to use the GRT shell to convert XML files to Java or PHP class files, by opening the shell using the `j` or the `p` option and specifying the XML source file and the destination directory.

Use the `listen` option with a port number to run the GRT shell as a service that can be accessed from a remote location.

Perhaps most importantly, you can pass a `Lua` script to the shell on startup. This allows you to perform tasks using a script file without even opening the MySQL Migration Toolkit. This is an especially useful feature if you need to migrate the same database a number of times or you want to customize a migration. You can easily create a `Lua` script by clicking the GENERATE MIGRATION SCRIPT when migrating using the graphical interface. For information on creating a Lua script, see [Section 27.17, “The Summary Screen”](#). A Lua migration script is examined in detail in [Chapter 28, Scripted Migration](#).

Passing a `Lua` file to the shell is usually invoked using the `x` option so that the shell closes after the script has executed.

The appearance of the GRT shell run from the command line is identical to its appearance when run from within the MySQL Migration Toolkit. All the commands and options described in [Section 31.3, “Using the GRT Shell”](#) are available when the GRT shell is invoked from the command line.

Part VI. Appendixes

Table of Contents

A. Installing From Source	120
A.1. Introduction	120
A.2. Downloading The Source Code	120
A.3. Building from Source under Windows	120
A.3.1. Prerequisites	120
A.3.2. Compiling and Building	120
A.4. Building from Source under Linux	122
A.4.1. Prerequisites	122
A.5. Compiling and Building	123
A.5.1. Building Static Binaries	123
A.5.2. Building RPMs	123
B. Troubleshooting Application Errors	125
B.1. Troubleshooting Connection Errors	125
B.2. Troubleshooting Display Problems	125
C. How Connections Are Stored	126
D. XML Files Common to the MySQL GUI Applications	127
E. XML Files Used by MySQL Query Browser	128
F. XML Files Used by MySQL Administrator	129
G. The MySQL System Tray Monitor	131
H. Third-party Software Used by the MySQL GUI Tools	132
H.1. PCRE Library	132
H.2. PNG Support	132
I. Notes for Translators	133

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 (i.e. for MySQL Administrator this is `work\mysql-administrator`). There are three batch-files to start the build:

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

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

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

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+	glib2.0-dev	glib2-devel
gtk+	2.6+	gtk2.0-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	glibmm-2.4-dev	glibmm24-devel
gtkmm-2.4	2.4	gtkmm-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 libs and headers		libmysqlclient15-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 (i.e.: `mysql-administrator`, `mysql-query-browser`) in the same top-level directory.

First build `mysql-gui-common`:

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

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

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

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

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

A.5.1. Building Static Binaries

Building static binaries is non-trivial 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>
...
```

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. XML Files Used by MySQL Administrator

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

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

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

This excerpt demonstrates a number of things:

- A description (called *text*) does not necessarily have to be translated (see `status-var-Uptime`). If there is no translation, English will be used instead.
- Adding another translation is as simple as adding another `<trans xml:lang="language">` tag.

Consequently, translating the software is a relatively painless task; however, please look at the [Notes for Translators](#) below.

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

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

```
<trans xml:lang="de">Abbruch</trans>
</text>
...
```

- `mysqlx-user-connections.xml`: This file is described in [How MySQL Administrator Stores Connections](#).

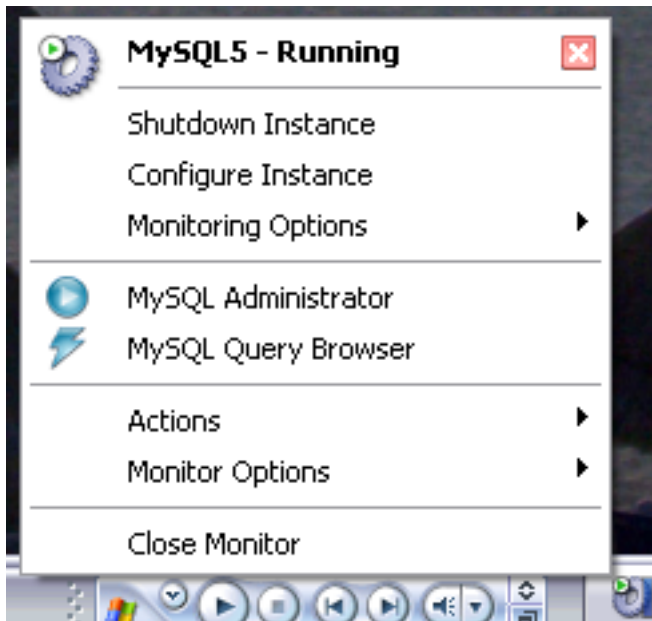
Appendix G. The MySQL System Tray Monitor

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

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

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

Figure G.1. The MySQL System Tray Monitor



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

- SHUTDOWN INSTANCE: this will begin the shutdown operation on the selected running service.
- START INSTANCE: this will start a currently stopped server instance.
- CONFIGURE INSTANCE: this will launch the MySQL Administrator in configure service mode for the currently selected instance. See [Configure Service Mode](#) for more details.
- MONITORING OPTIONS: currently the MONITORING OPTIONS submenu has one available option: IGNORE INSTANCE STATE. IGNORE INSTANCE STATE will cause the MySQL System Tray Monitor to stop monitoring the selected service.

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

The MANAGE MYSOQL INSTANCES option of the ACTIONS menu can be used to start MySQL Administrator in configure service mode so you can manage your server configurations. See [Configure Service Mode](#) for more details.

The MONITOR OPTIONS menu has two options, which affect the general behavior of the MySQL System Tray Monitor. DISPLAY CPU LOAD determines whether the CPU usage is displayed on the right of the MySQL System Tray Monitor. LAUNCH MONITOR AFTER LOGIN sets the MySQL System Tray Monitor to launch when you log into Windows.

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

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

H.1. PCRE Library

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

H.2. PNG Support

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

Appendix I. 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 [Documentation Team](#). 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.