MySQL Migration Toolkit

MySQL Migration Toolkit

Abstract

This is the MySQL Migration Toolkit Manual.

Document generated on: 2006-10-18 (revision: 3666)

Please email `<docs@mysql.com>` for more information or if you are interested in doing a translation.

Table of Contents

List of Figures

Chapter 1. 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.

The current version of MySQL Migration Toolkit is a release candidate. While every effort has been made to ensure it is free of bugs, it should not be used in a production environment. Always back up your data before using the MySQL Migration Toolkit. If you find it's lacking some feature important to you, or if you discover a bug, please use our MySQL Bug System [http://bugs.mysql.com] to request features or report problems.

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

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 [http://dev.mysql.com/downloads//gui-tools/5.0.html] page.

2.2. Installing Under Windows

MySQL GUI Tools run on recent 32-bit Windows NT based operating systems, including Windows 2000, XP, 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 [http://support.microsoft.com/default.aspx?scid=kb;EN-US;292539] 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 [http://dev.mysql.com/doc/refman/5.1/en/bug-reports.html].

To install MySQL GUI Tools, right click on the MSI file and select Install.

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.

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`, `MySQLQueryBrowser.exe`, and `MySQLWorkbench.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.

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 -xzvf 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.

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. On FC5 you may install this toolkit from the command line in the following way:

```
shell> yum -install gtkmm24
```

For installation on SuSE Linux:

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

> **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 -iv mysql-*.rpm
```

If you are upgrading to a newer version of MySQL GUI Tools type:

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

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

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

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 -iv 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 [http://dev.mysql.com/downloads//gui-tools/5.0.html] page for the most up-to-date listing of the various RPM packages available.

2.4. Installing Under Mac OS X

To install MySQL Administrator 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.

Chapter 3. Removing GUI Tools

Under Linux and 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 -U mysql-*.rpm
```

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

```
shell> rpm -U 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 – OS X

Find the directory where you installed MySQL GUI Tools, and remove applications by moving their icons to the `Trash`.

Chapter 4. Running MySQL GUI Tools

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

4.1. Running Under Windows

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

- `MySQLAdministrator.exe`
- `MySQLQueryBrowser.exe`
- `MySQLWorkbench.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.

4.2. Running Under Linux

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

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

> **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.

4.3. Running On Mac OS X

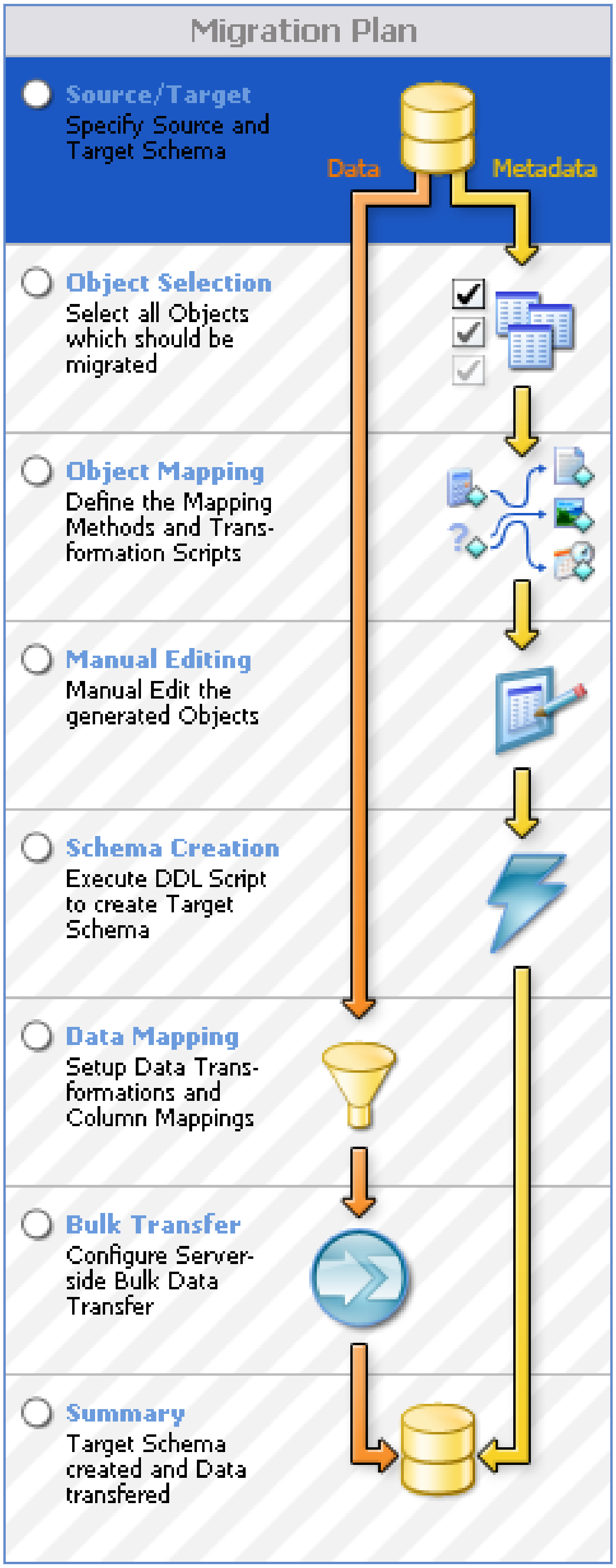
On Mac OS X, navigate to the MySQL GUI Tools installation directory and double-click on the application you wish to start.

> **Note**
>
> There is no OS X version of MySQL Migration Toolkit.

Chapter 5. An Overview of the Migration Process

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

Figure 5.1. The MySQL Migration Toolkit Migration Plan



Migration Plan

○ Source/Target
Specify Source and Target Schema

○ Object Selection
Select all Objects which should be migrated

○ Object Mapping
Define the Mapping Methods and Transformation Scripts

○ Manual Editing
Manual Edit the generated Objects

○ Schema Creation
Execute DDL Script to create Target Schema

○ Data Mapping
Setup Data Transformations and Column Mappings

○ Bulk Transfer
Configure Server-side Bulk Data Transfer

○ Summary
Target Schema created and Data transfered

- `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 6. The Migration Process In-Depth

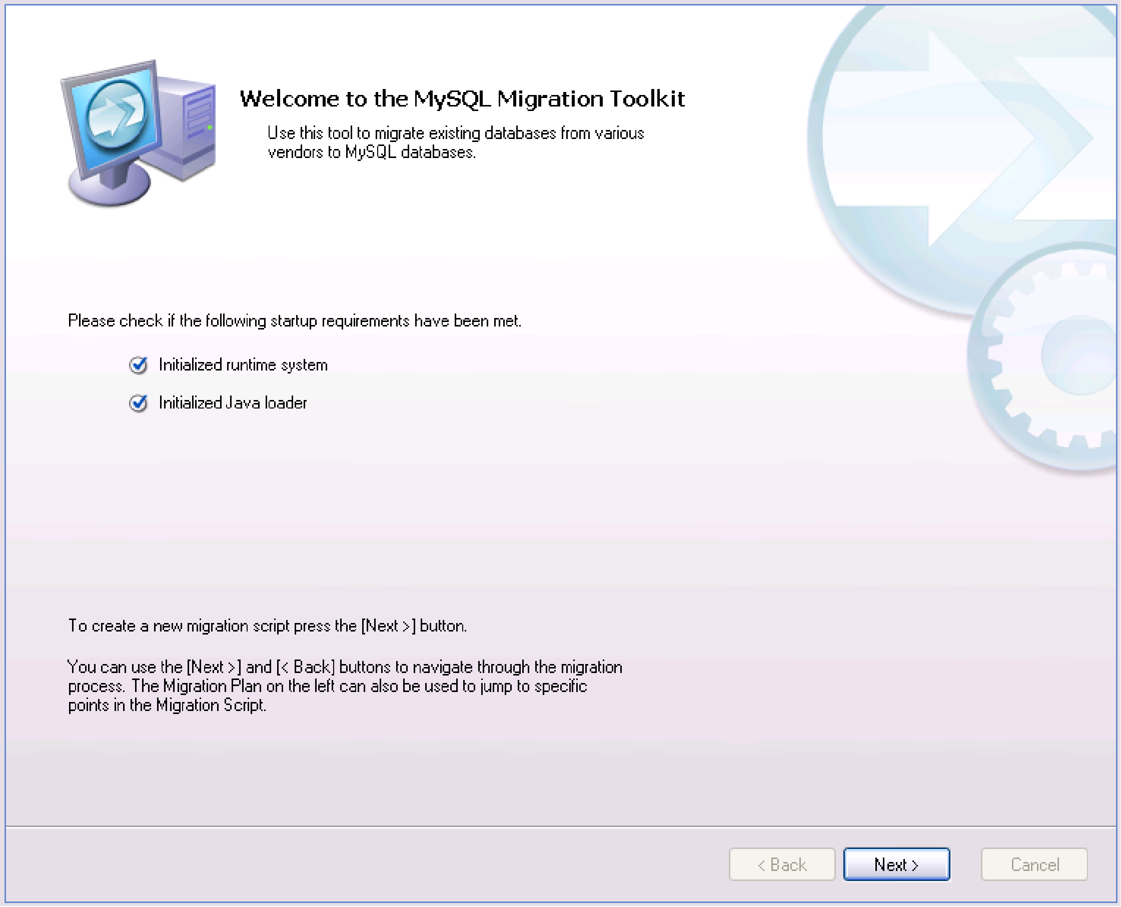
6.1. Introduction

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.

6.2. The Welcome Screen

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

Figure 6.1. The MySQL Migration Toolkit welcome screen



Welcome to the MySQL Migration Toolkit

Use this tool to migrate existing databases from various vendors to MySQL databases.

Please check if the following startup requirements have been met.

☑ Initialized runtime system

☑ Initialized Java loader

To create a new migration script press the [Next >] button.

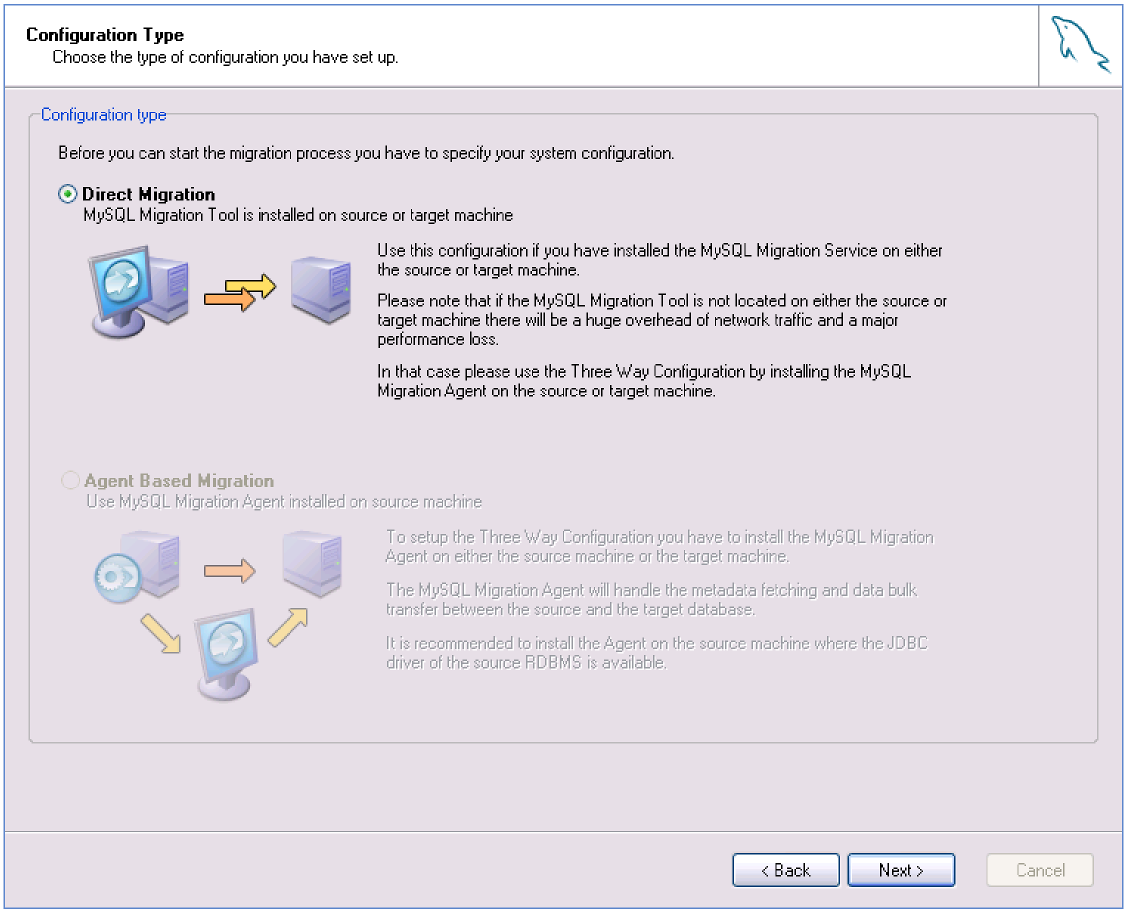
You can use the [Next >] and [< Back] buttons to navigate through the migration process. The Migration Plan on the left can also be used to jump to specific points in the Migration Script.

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 Introduction [http://dev.mysql.com/doc/migration-toolkit/en/mysql-migration-toolkit-installation-introduction.html] for information on downloading and installing the Java Runtime Environment.

6.3. The Configuration Type Screen

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

Figure 6.2. The Configuration Type screen



Configuration Type

Choose the type of configuration you have set up.

Configuration type

Before you can start the migration process you have to specify your system configuration.

● **Direct Migration**

MySQL Migration Tool is installed on source or target machine

Use this configuration if you have installed the MySQL Migration Service on either the source or target machine.

Please note that if the MySQL Migration Tool is not located on either the source or target machine there will be a huge overhead of network traffic and a major performance loss.

In that case please use the Three Way Configuration by installing the MySQL Migration Agent on the source or target machine.

○ **Agent Based Migration**

Use MySQL Migration Agent installed on source machine

To setup the Three Way Configuration you have to install the MySQL Migration Agent on either the source machine or the target machine.

The MySQL Migration Agent will handle the metadata fetching and data bulk transfer between the source and the target database.

It is recommended to install the Agent on the source machine where the JDBC driver of the source RDBMS is available.

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.

6.4. The Source Database Screen

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

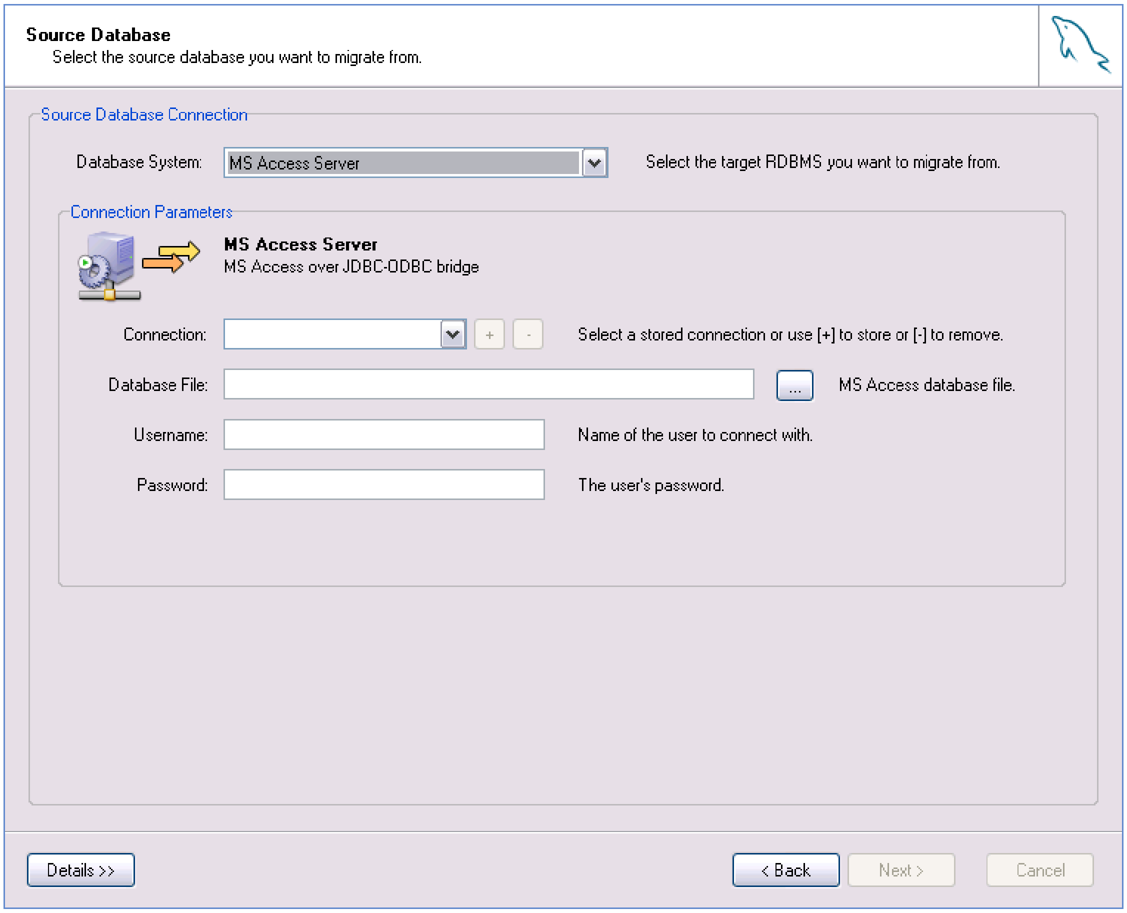
The Source Database screen's appearance will vary depending on the type of source database selected.

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.

6.4.1. Microsoft Access

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

Figure 6.3. Source database – Microsoft Access



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

Source Database Connection

Database System: MS Access Server — 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.

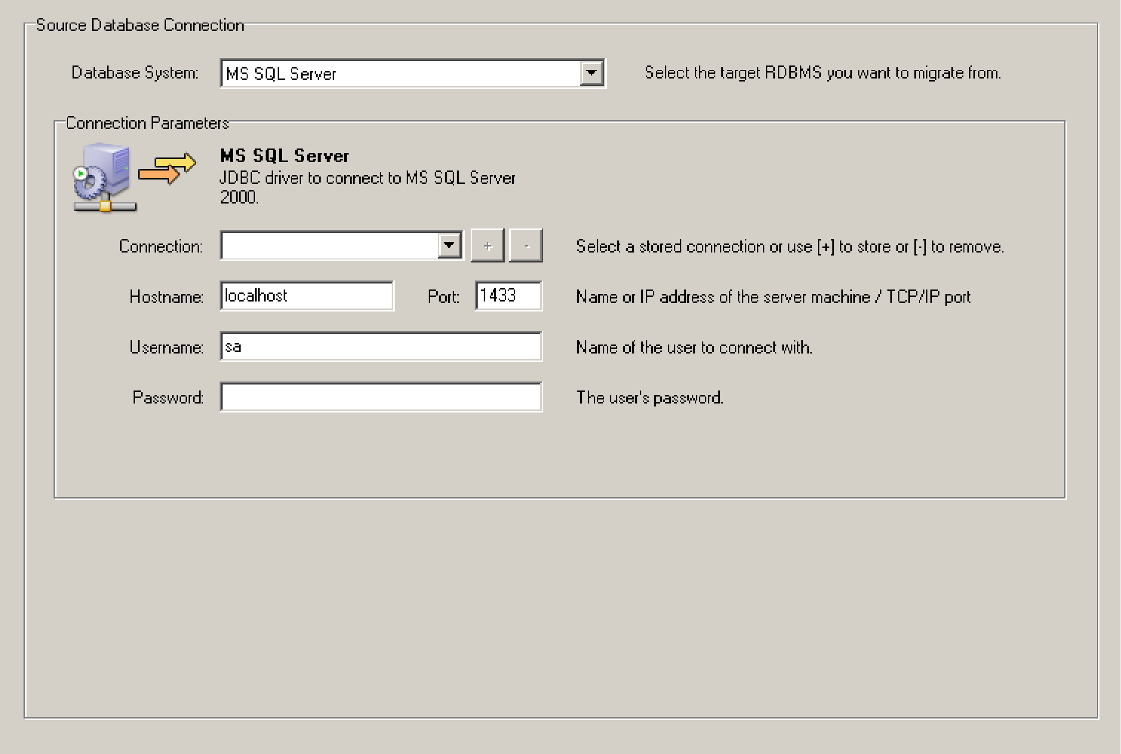
Specify the path to the `.mdb` database file in the `Database File` field and specify the database username and password information if applicable.

You must make special modifications to your Access database file before it can be used with MySQL Migration Toolkit. Please see Chapter 8, *Preparing a Microsoft Access Database for Migration* for further information.

6.4.2. Microsoft SQL Server

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

Figure 6.4. Source database – Microsoft SQL Server



Source Database Connection

Database System: MS SQL Server

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: localhost Port: 1433

Name or IP address of the server machine / TCP/IP port

Username: sa

Name of the user to connect with.

Password:

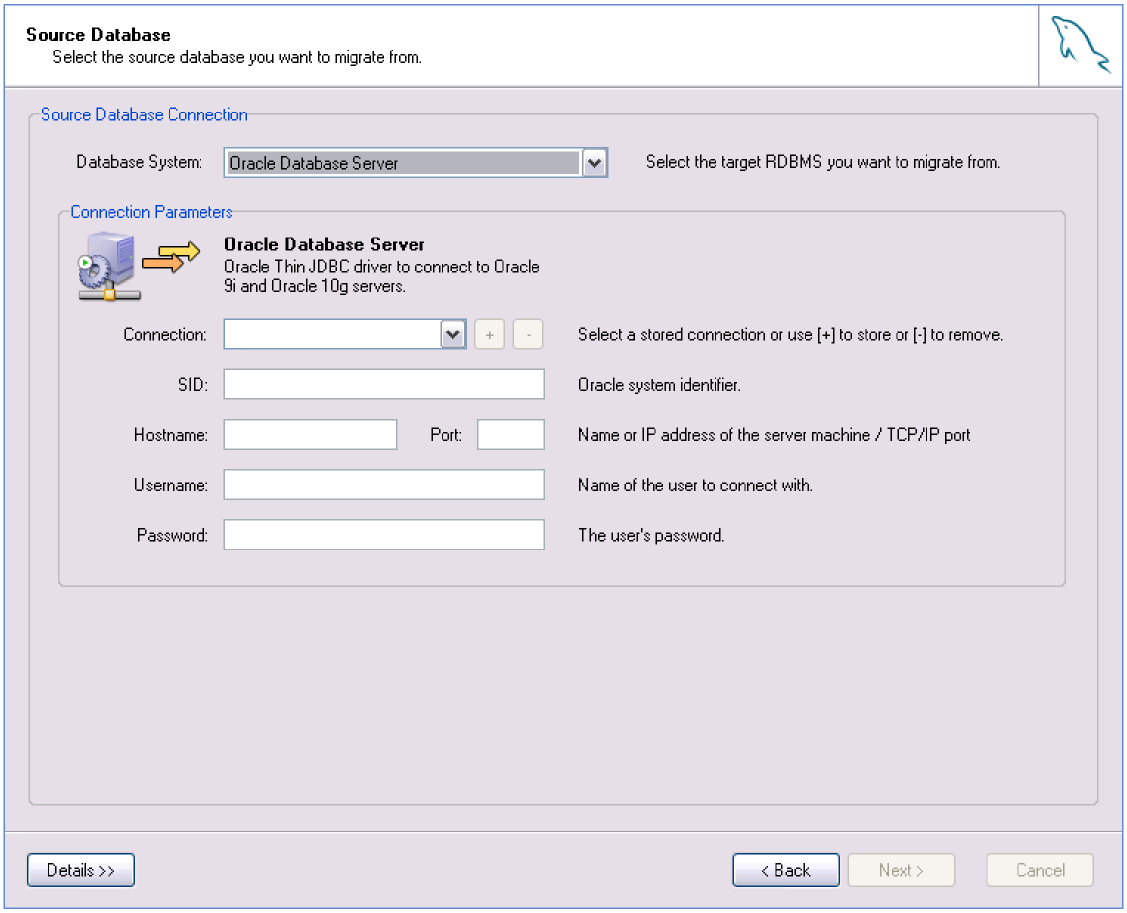
The user's password.

Specify the hostname, username, and password to connect to the source Microsoft SQL Server to connect and click `Next`.

6.4.3. Oracle

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

Figure 6.5. Source database – Oracle



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

Source Database Connection

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

Connection Parameters

Oracle Database Server
Oracle Thin JDBC driver to connect to Oracle 9i and Oracle 10g servers.

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

SID: — Oracle system identifier.

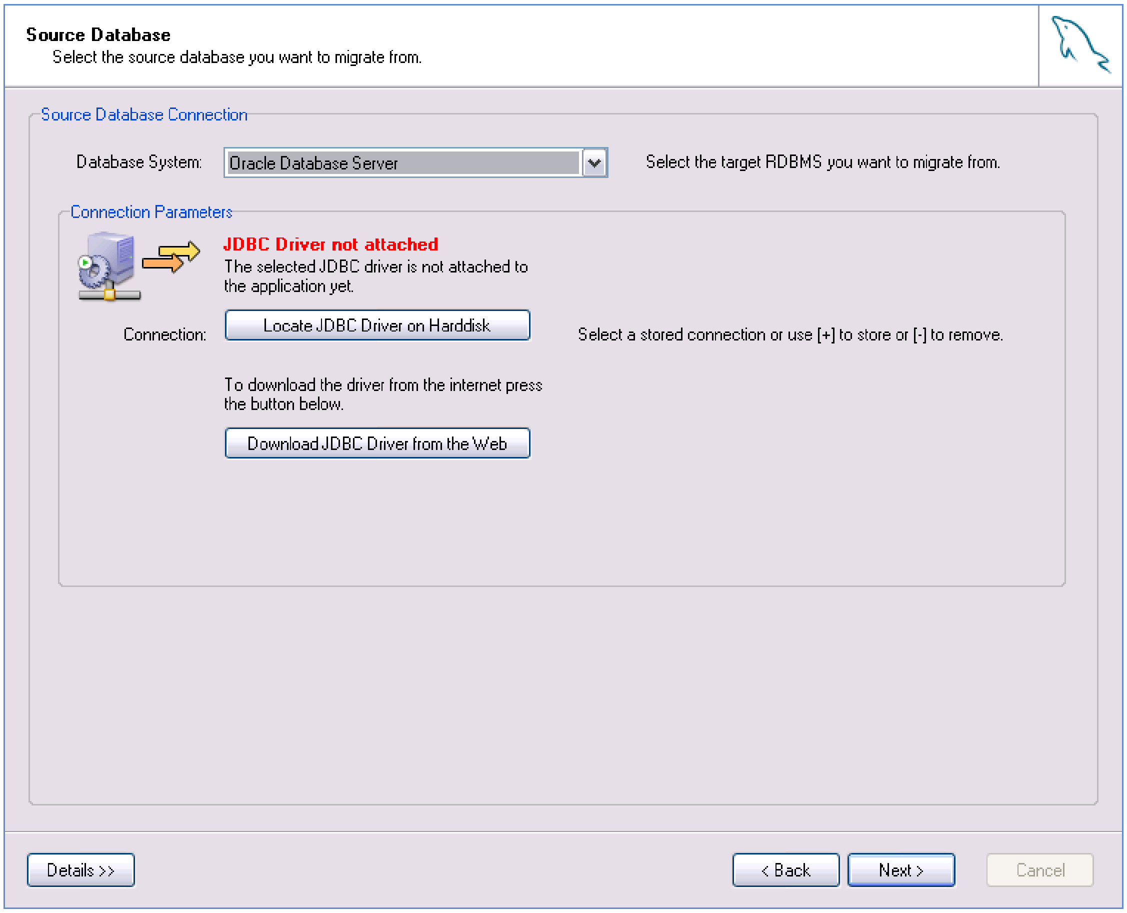
Hostname: Port: — Name or IP address of the server machine / TCP/IP port

Username: — Name of the user to connect with.

Password: — The user's password.

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

Figure 6.6. Oracle JDBC driver not attached



Source Database

Select the source database you want to migrate from.

Source Database Connection

Database System: Oracle Database Server

Select the target RDBMS you want to migrate from.

Connection Parameters

JDBC Driver not attached

The selected JDBC driver is not attached to the application yet.

Connection:

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

To download the driver from the internet press the button below.

Locate JDBC Driver on Harddisk
Download JDBC Driver from the Web
Details >>
< Back
Next >
Cancel

If the Oracle JDBC driver is present on your system, click the `Locate JDBC Driver on Harddisk` button to attach the driver.

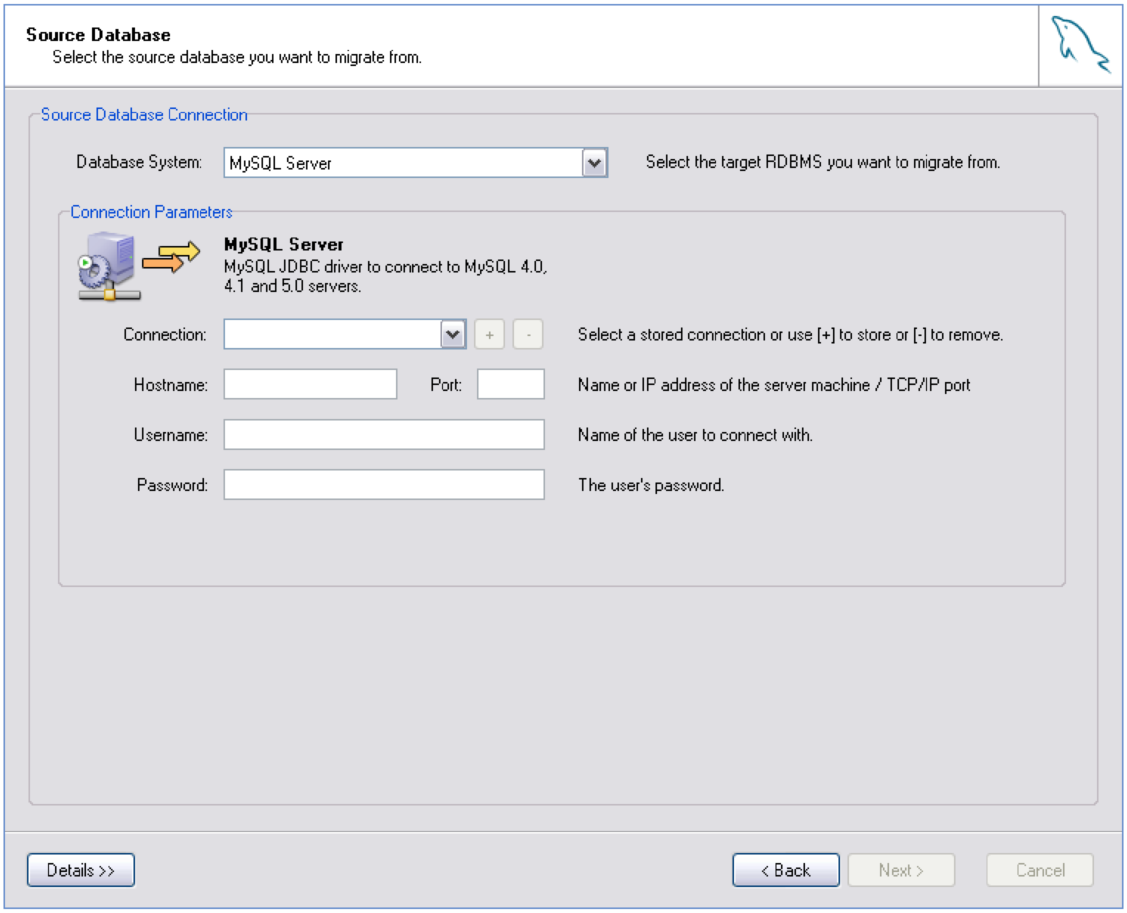
If the Oracle JDBC driver is not present on your system, click the `Download JDBC Driver from the Web` button to download it. Download the `ojdbc14.jar` file and then attach it by clicking on the `Locate JDBC Driver on Harddisk` button.

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

6.4.4. MySQL

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

Figure 6.7. Source database – MySQL



Source Database

Select the source database you want to migrate from.

Source Database Connection

Database System: MySQL Server — 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.

6.4.5. 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.

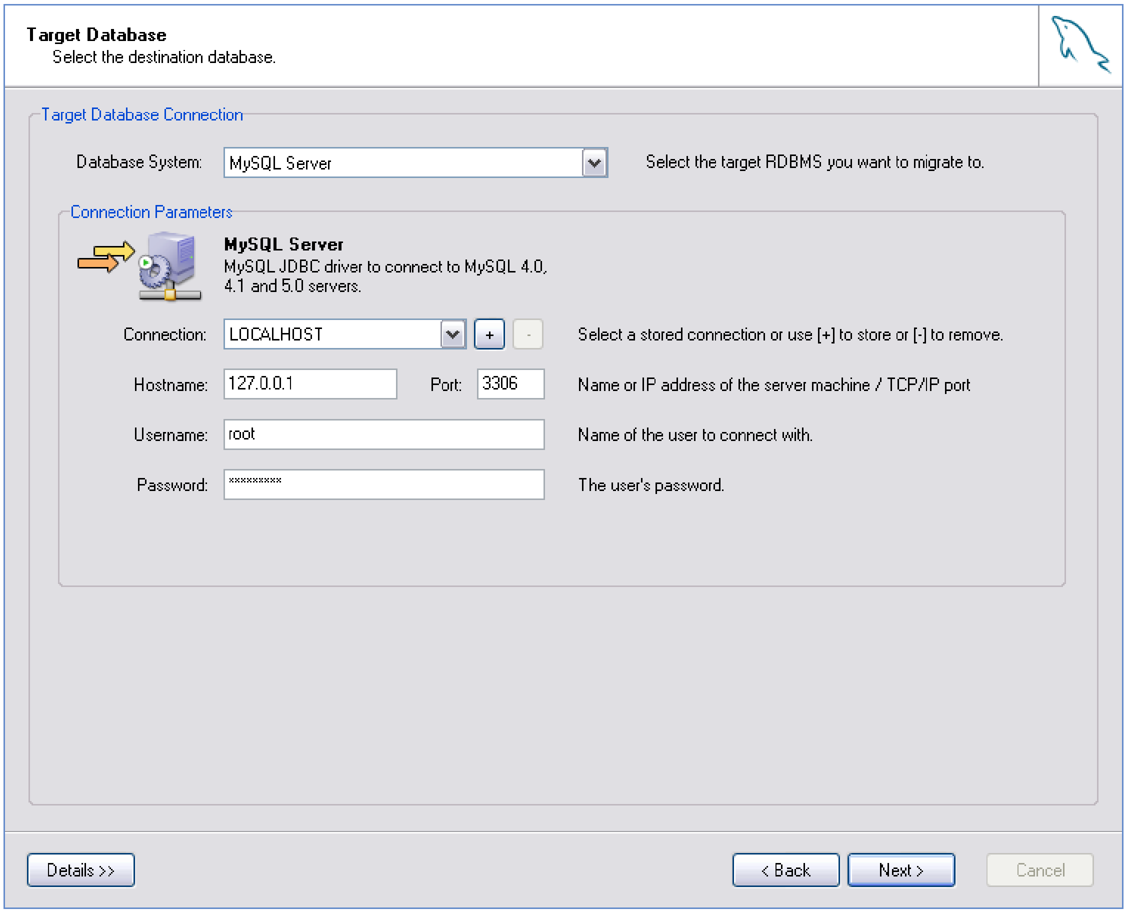
You can discard saved connection information by selecting the saved connection from the `Connection` drop-down list and clicking the `–` button.

6.5. 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 6.8. Target Database – MySQL



Target Database
Select the destination database.

Target Database Connection

Database System: MySQL Server — 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: LOCALHOST — Select a stored connection or use [+] to store or [-] to remove.

Hostname: 127.0.0.1 Port: 3306 — Name or IP address of the server machine / TCP/IP port

Username: root — Name of the user to connect with.

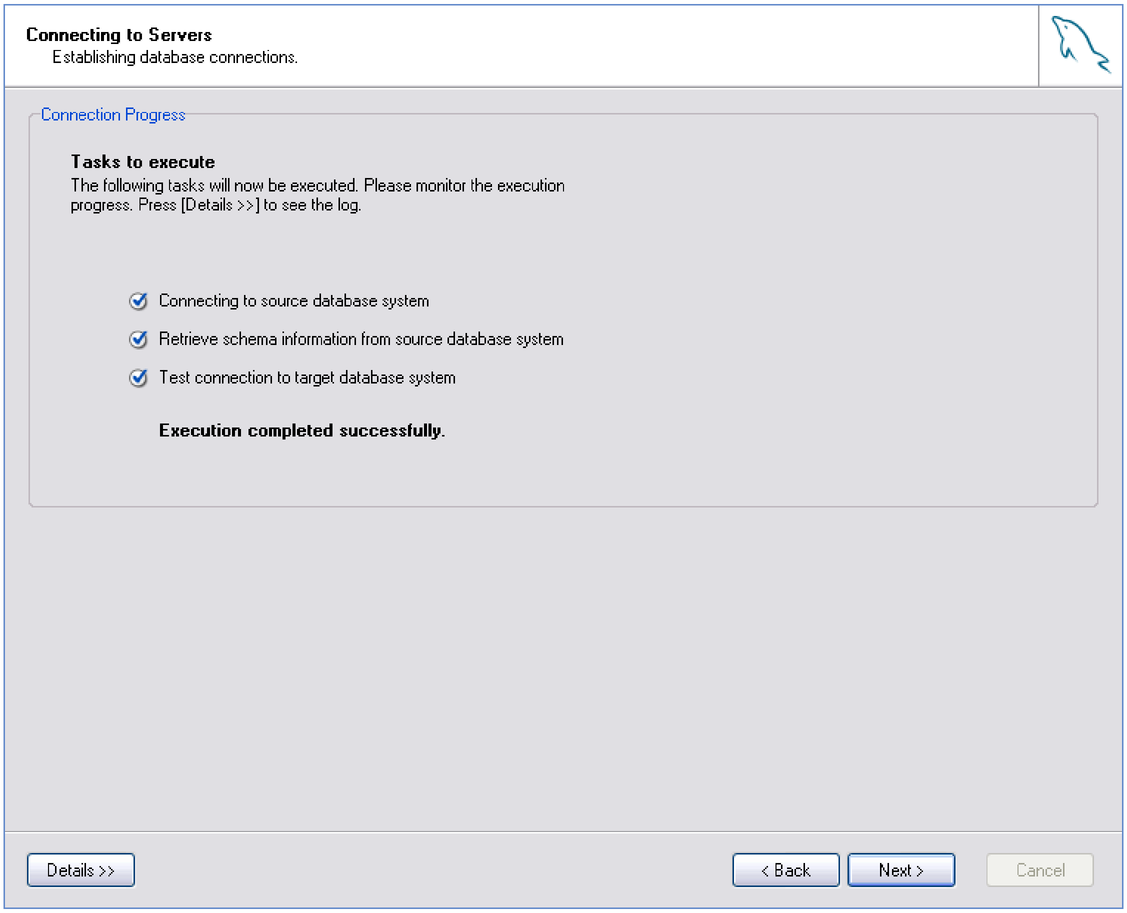
Password: ********** — The user's password.

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

6.6. 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 6.9. The Connect to Servers screen



Connecting to Servers
Establishing database connections.

Connection Progress

Tasks to execute
The following tasks will now be executed. Please monitor the execution progress. Press [Details >>] to see the log.

- ☑ Connecting to source database system
- ☑ Retrieve schema information from source database system
- ☑ Test connection to target database system

Execution completed successfully.

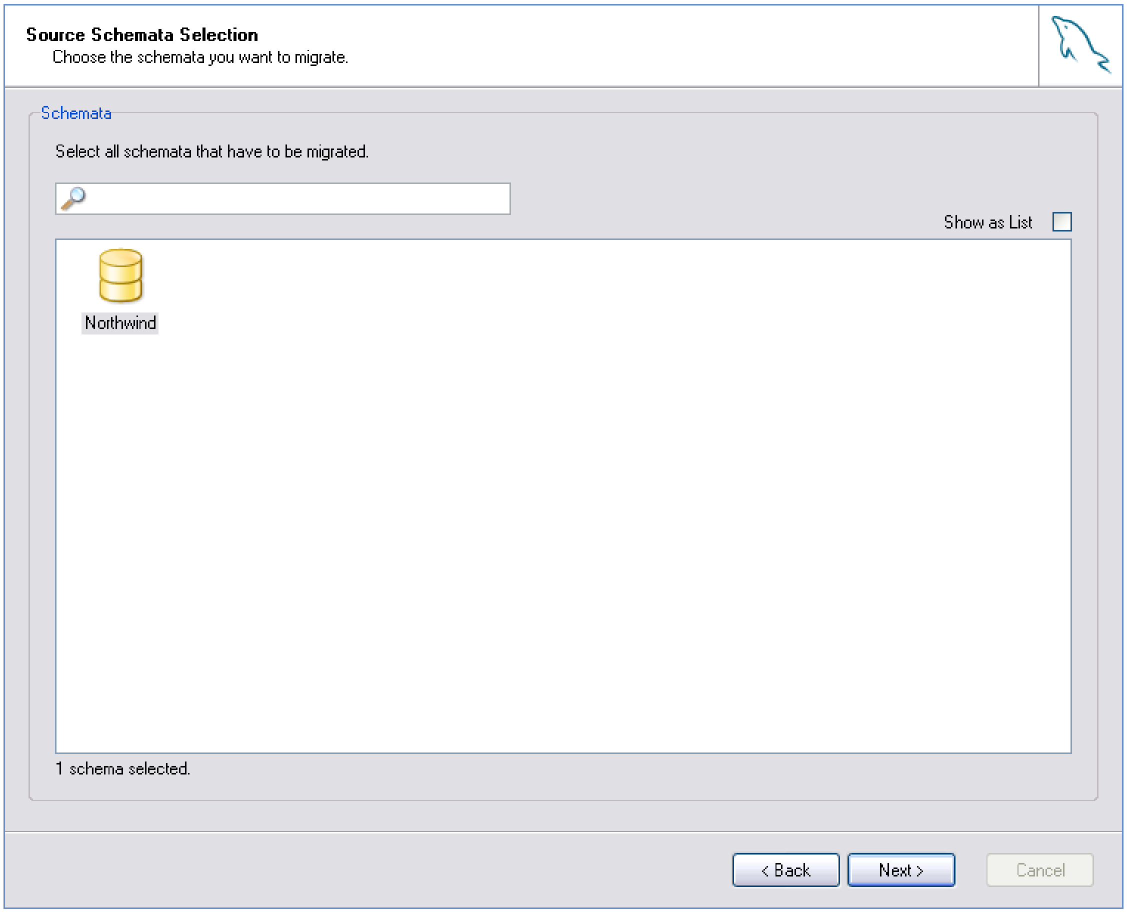
Click the `Details` button to see a more detailed log of the connection process.

6.7. 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 6.10. The Source Schema Selection screen



Source Schemata Selection

Choose the schemata you want to migrate.

Schemata

Select all schemata that have to be migrated.

Show as List □

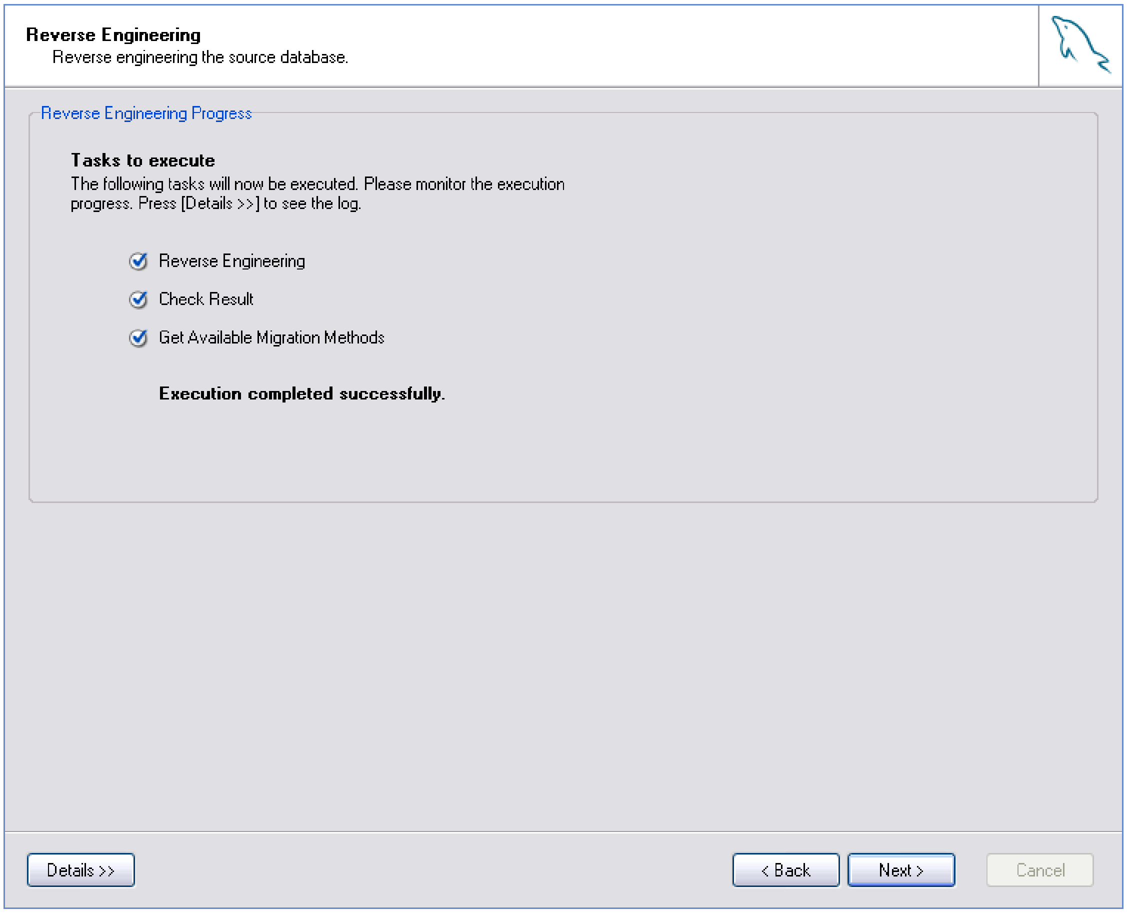
Northwind

1 schema selected.

6.8. 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 6.11. The Reverse Engineering screen



Reverse Engineering
Reverse engineering the source database.

Reverse Engineering Progress

Tasks to execute
The following tasks will now be executed. Please monitor the execution progress. Press [Details >>] to see the log.

- ☑ Reverse Engineering
- ☑ Check Result
- ☑ Get Available Migration Methods

Execution completed successfully.

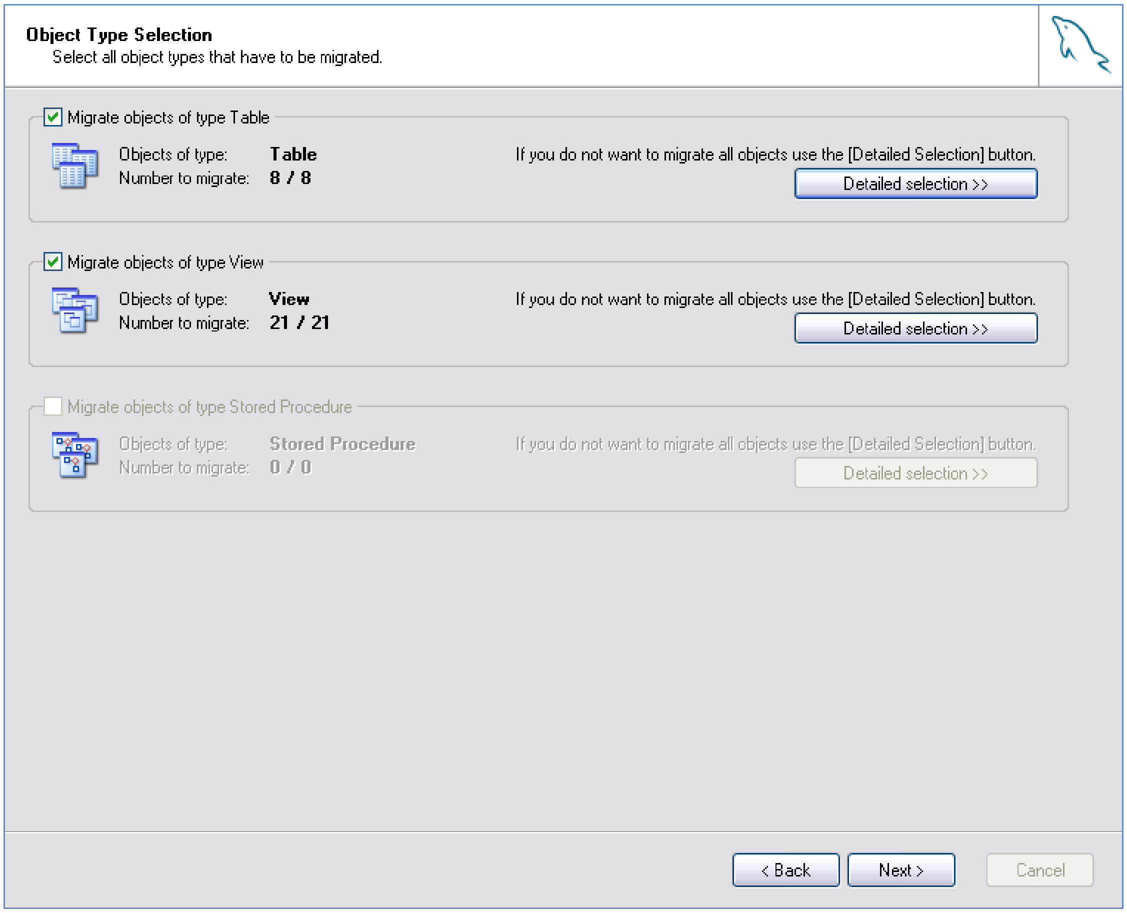
The MySQL Migration Toolkit collects column and index information for each table in the source database, along with information on stored procedures and views.

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

6.9. The Object Type Selection Screen

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

Figure 6.12. 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.

☑ 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.

☐ Migrate objects of type Stored Procedure

Objects of type: **Stored Procedure**
Number to migrate: **0 / 0**

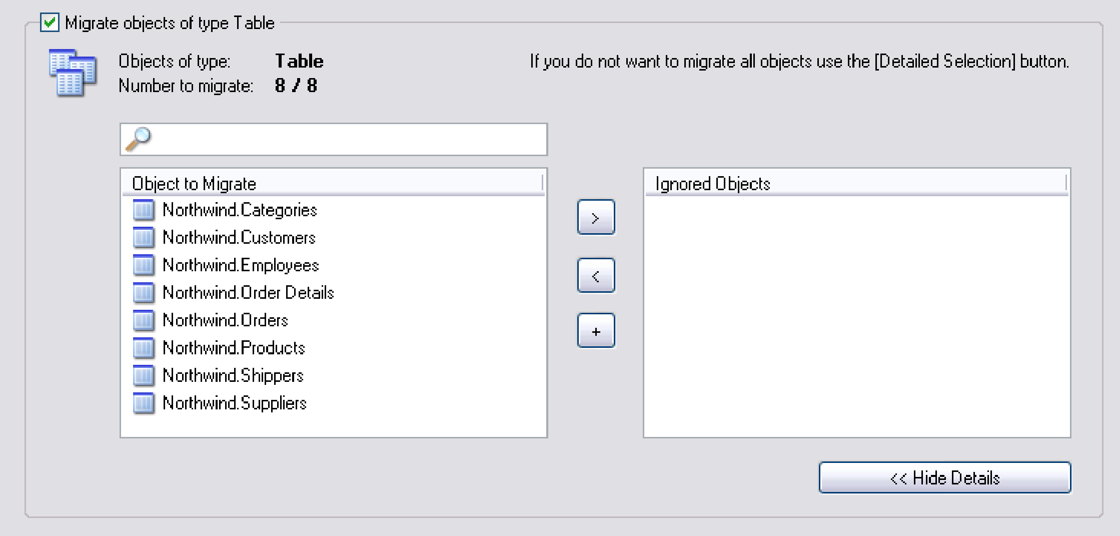
If you do not want to migrate all objects use the [Detailed Selection] button.

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

6.9.1. Migrating a Sub-Set of an Object Type

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 6.13. The detail view of the Object Type Selection screen



☑ 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.

Object to Migrate

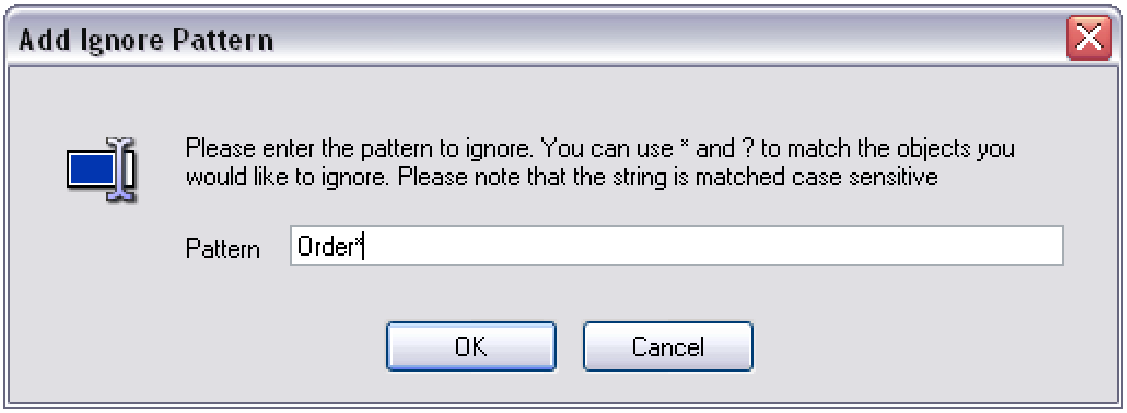
- Northwind.Categories
- Northwind.Customers
- Northwind.Employees
- Northwind.Order Details
- Northwind.Orders
- Northwind.Products
- Northwind.Shippers
- Northwind.Suppliers

Ignored Objects

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 6.14. The Add Ignore Pattern dialog



Add Ignore Pattern

Please enter the pattern to ignore. You can use * and ? to match the objects you would like to ignore. Please note that the string is matched case sensitive

Pattern: Order*

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

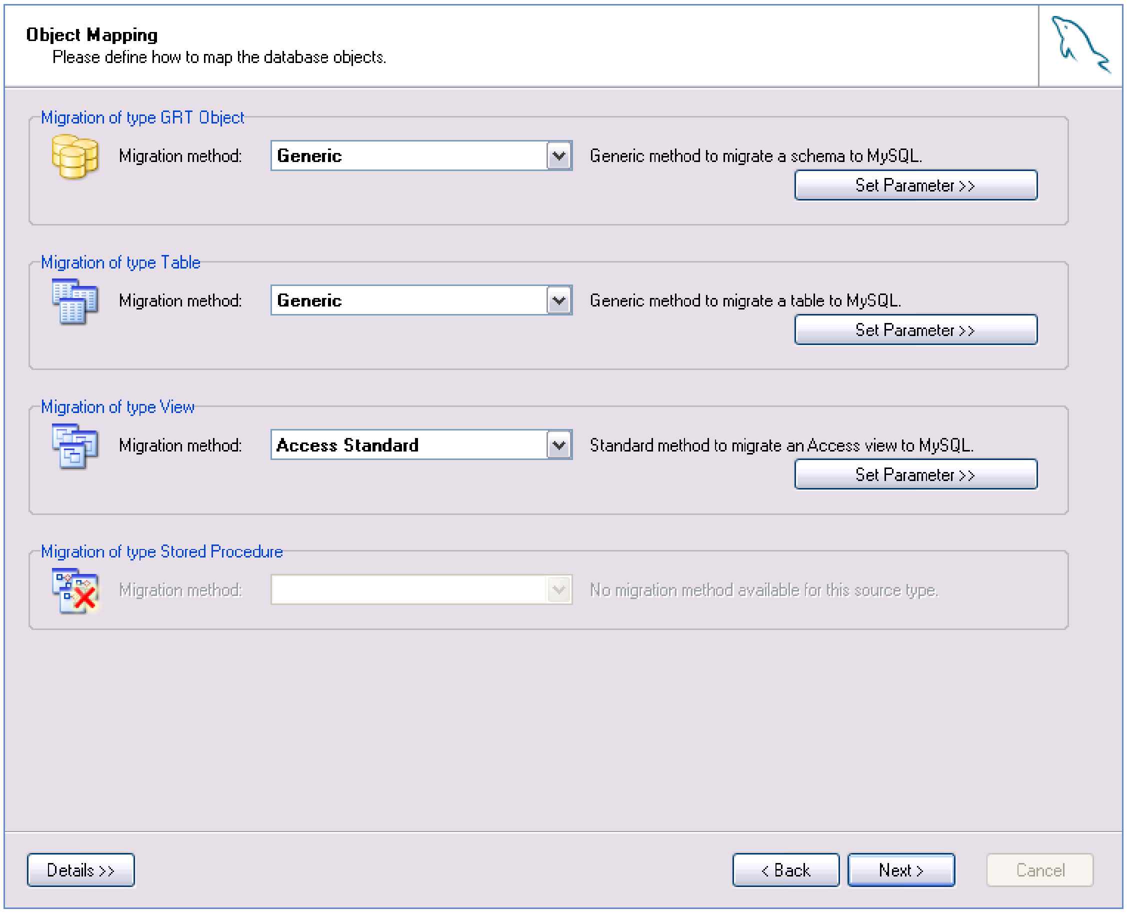
6.10. The Object Mapping Screen

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 6.15. The Object Mapping screen



Object Mapping

Please define how to map the database objects.

Migration of type GRT Object

Migration method: Generic

Generic method to migrate a schema to MySQL.

Migration of type Table

Migration method: Generic

Generic method to migrate a table to MySQL.

Migration of type View

Migration method: Access Standard

Standard method to migrate an Access view to MySQL.

Migration of type Stored Procedure

Migration method:

No migration method available for this source type.

6.10.1. 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`.

6.10.2. Table Objects

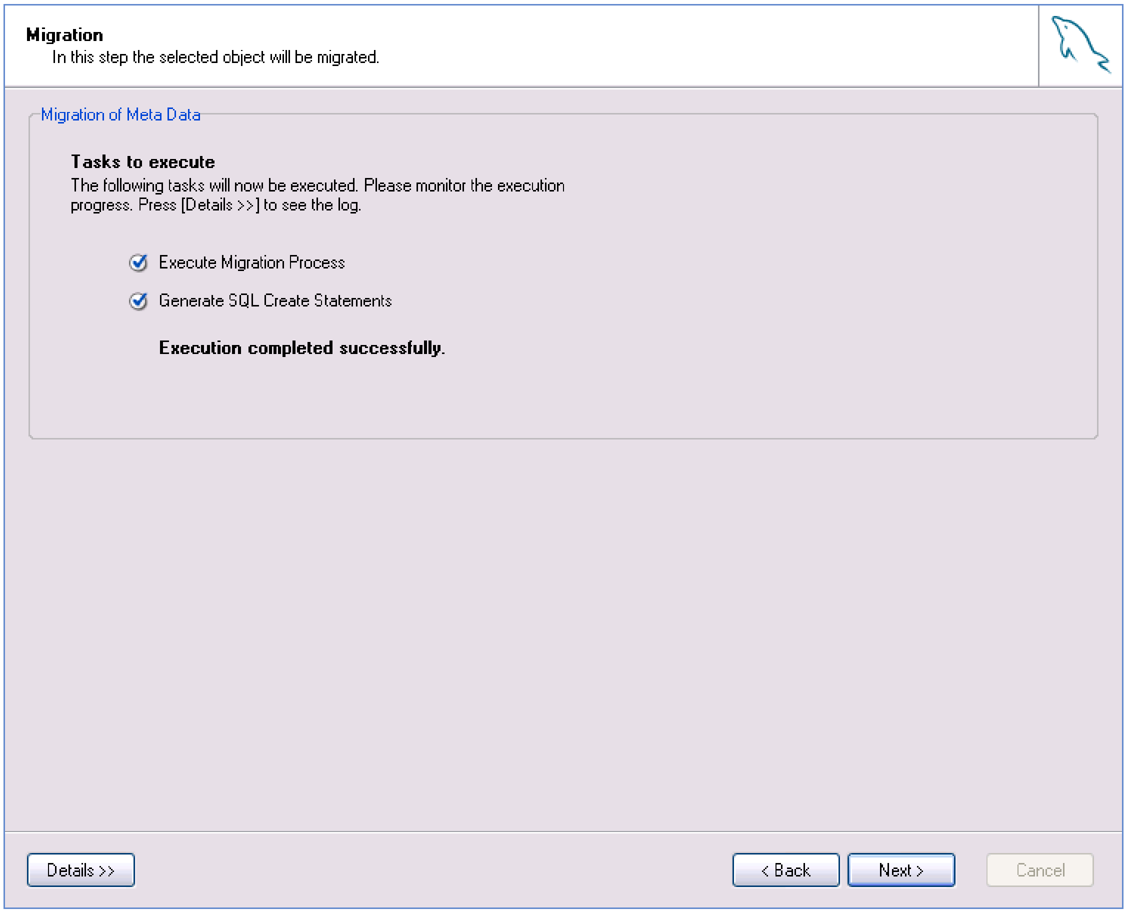
The Table section of the Object Mapping screen dictates the properties of the individual tables. By default a generic profile is used, with an `InnoDB` storage engine.

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.`

6.11. 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 6.16. The Meta Migration screen



Migration
In this step the selected object will be migrated.

Migration of Meta Data

Tasks to execute
The following tasks will now be executed. Please monitor the execution progress. Press [Details >>] to see the log.

☑ Execute Migration Process
☑ Generate SQL Create Statements

Execution completed successfully.

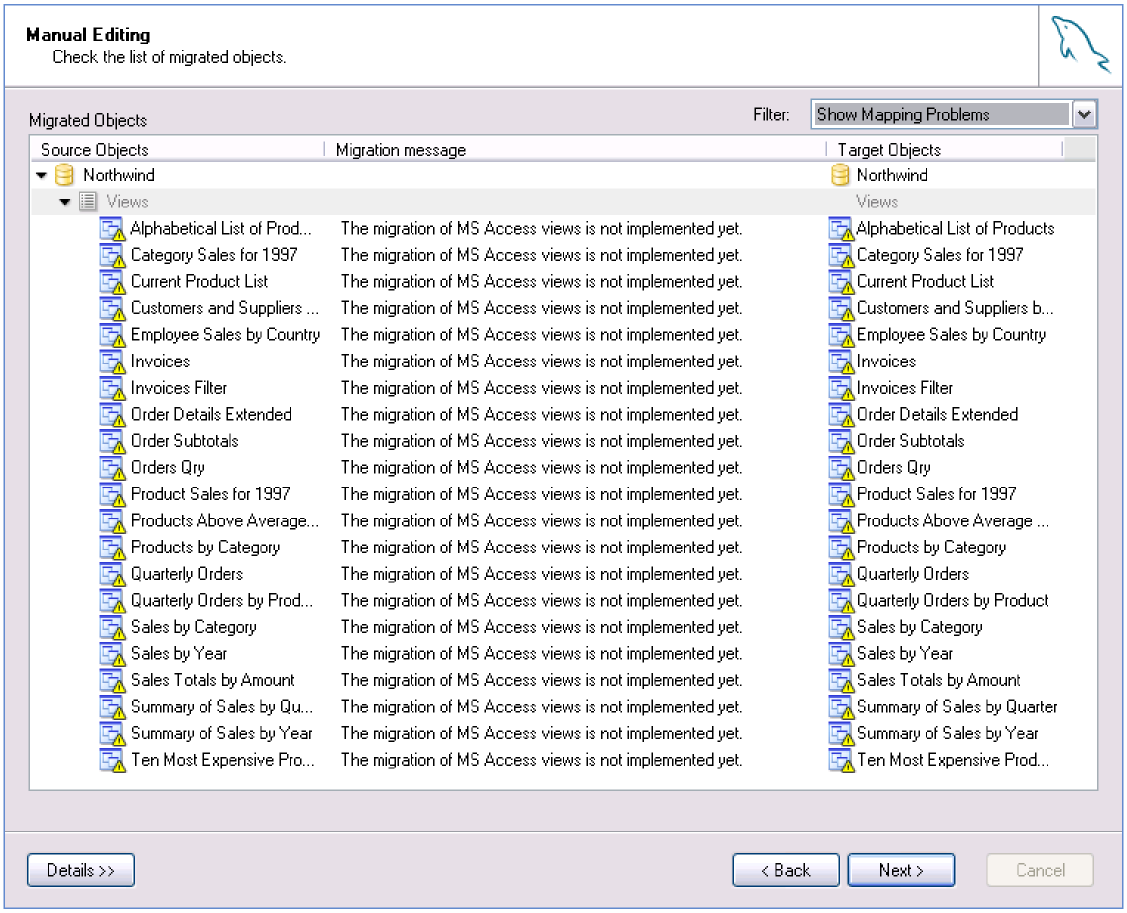
The SQL `CREATE` statements are not executed on the target server at this stage of the migration, but will be executed later.

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

6.12. The Manual Editing Screen

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

Figure 6.17. The Manual Editing screen



Manual Editing

Check the list of migrated objects.

Migrated Objects

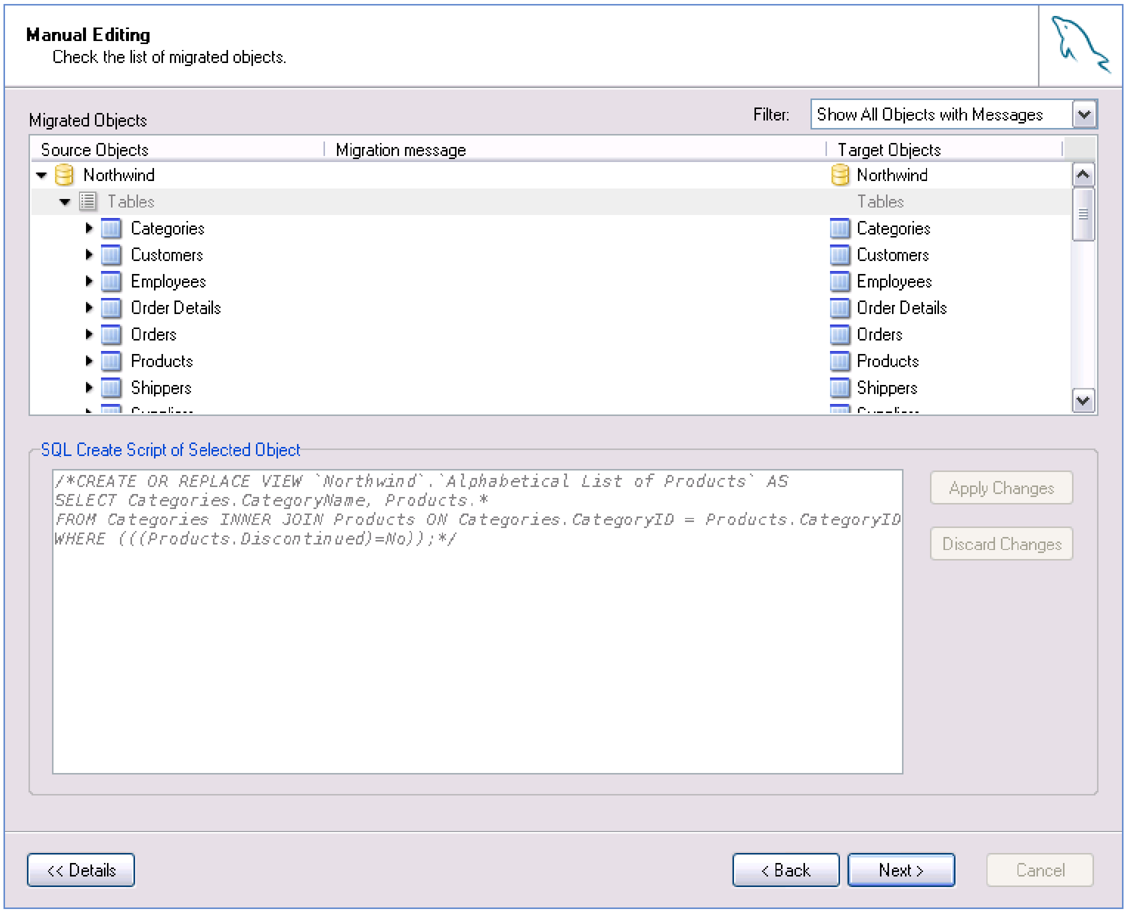
Filter: Show Mapping Problems

Source Objects	Migration message	Target Objects
Northwind		Northwind
Views		Views
Alphabetical List of Prod...	The migration of MS Access views is not implemented yet.	Alphabetical List of Products
Category Sales for 1997	The migration of MS Access views is not implemented yet.	Category Sales for 1997
Current Product List	The migration of MS Access views is not implemented yet.	Current Product List
Customers and Suppliers ...	The migration of MS Access views is not implemented yet.	Customers and Suppliers b...
Employee Sales by Country	The migration of MS Access views is not implemented yet.	Employee Sales by Country
Invoices	The migration of MS Access views is not implemented yet.	Invoices
Invoices Filter	The migration of MS Access views is not implemented yet.	Invoices Filter
Order Details Extended	The migration of MS Access views is not implemented yet.	Order Details Extended
Order Subtotals	The migration of MS Access views is not implemented yet.	Order Subtotals
Orders Qry	The migration of MS Access views is not implemented yet.	Orders Qry
Product Sales for 1997	The migration of MS Access views is not implemented yet.	Product Sales for 1997
Products Above Average...	The migration of MS Access views is not implemented yet.	Products Above Average ...
Products by Category	The migration of MS Access views is not implemented yet.	Products by Category
Quarterly Orders	The migration of MS Access views is not implemented yet.	Quarterly Orders
Quarterly Orders by Prod...	The migration of MS Access views is not implemented yet.	Quarterly Orders by Product
Sales by Category	The migration of MS Access views is not implemented yet.	Sales by Category
Sales by Year	The migration of MS Access views is not implemented yet.	Sales by Year
Sales Totals by Amount	The migration of MS Access views is not implemented yet.	Sales Totals by Amount
Summary of Sales by Qu...	The migration of MS Access views is not implemented yet.	Summary of Sales by Quarter
Summary of Sales by Year	The migration of MS Access views is not implemented yet.	Summary of Sales by Year
Ten Most Expensive Pro...	The migration of MS Access views is not implemented yet.	Ten Most Expensive Prod...

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 6.18. The Manual Editing screen – detailed view



Filter: Show All Objects with Messages

Source Objects	Migration message	Target Objects
Northwind		Northwind
Tables		Tables
Categories		Categories
Customers		Customers
Employees		Employees
Order Details		Order Details
Orders		Orders
Products		Products
Shippers		Shippers

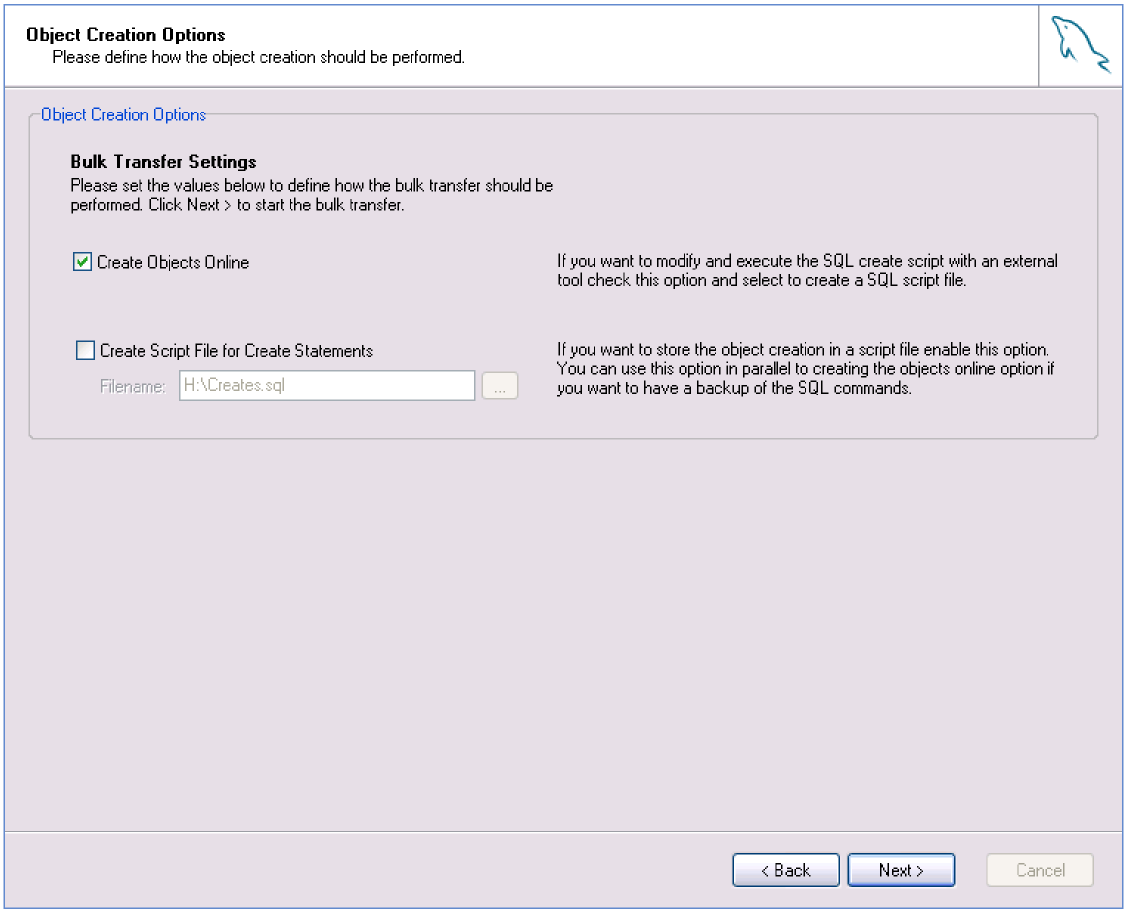
```
/*CREATE OR REPLACE VIEW `Northwind`.`Alphabetical List of Products` AS
SELECT Categories.CategoryName, Products.*
FROM Categories INNER JOIN Products ON Categories.CategoryID = Products.CategoryID
WHERE (((Products.Discontinued)=No));*/
```

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.

6.13. 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 6.19. The Object Creation Options screen



Object Creation Options

Please define how the object creation should be performed.

Object Creation 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.

☑ Create Objects Online

If you want to modify and execute the SQL create script with an external tool check this option and select to create a SQL script file.

□ Create Script File for Create Statements

Filename: H:\Creates.sql

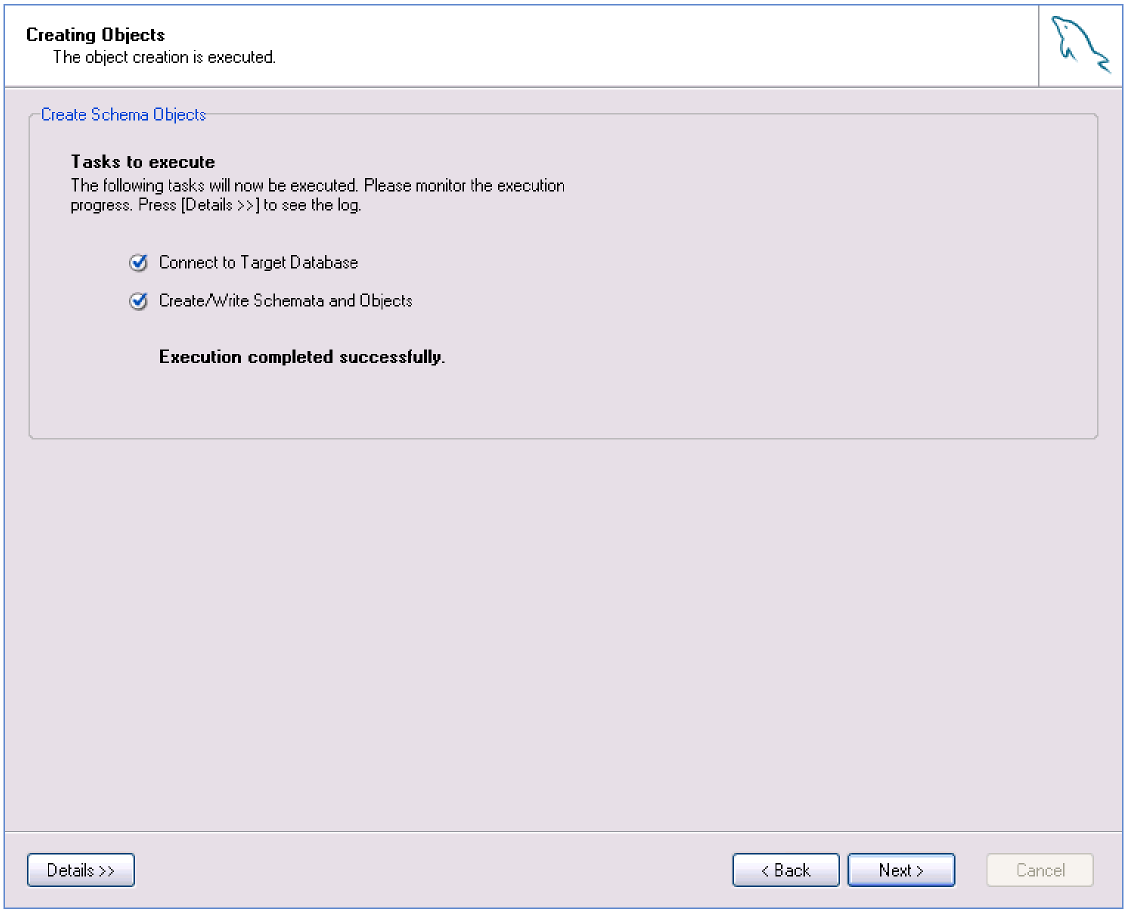
If you want to store the object creation in a script file enable this option. You can use this option in parallel to creating the objects online option if you want to have a backup of the SQL commands.

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

6.14. 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). If you chose to have a copy of the `CREATE` statements saved to disk this will be performed at this time.

Figure 6.20. The Creating Objects screen



Creating Objects
The object creation is executed.

Create Schema Objects

Tasks to execute
The following tasks will now be executed. Please monitor the execution progress. Press [Details >>] to see the log.

☑ Connect to Target Database

☑ Create/Write Schemata and Objects

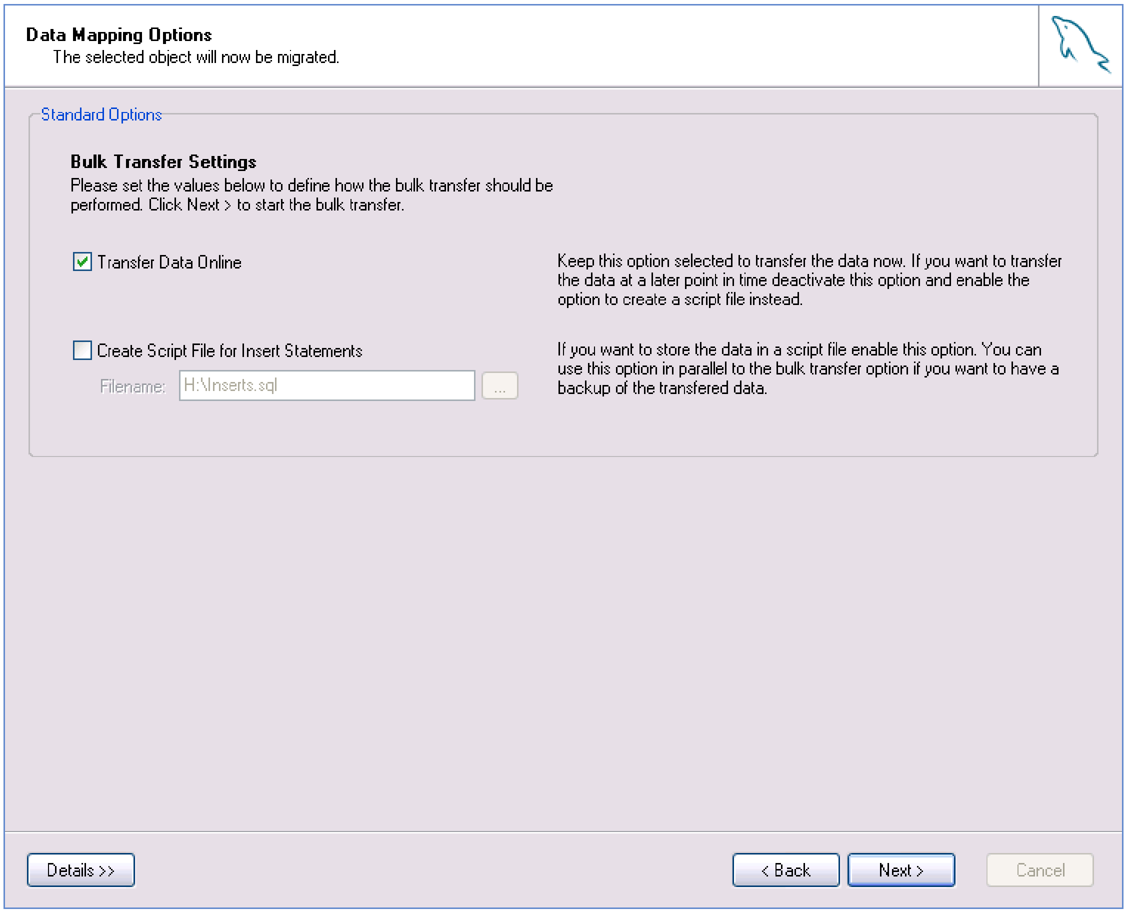
Execution completed successfully.

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

6.15. 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 6.21. 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

Filename: H:\Inserts.sql

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 transfered data.

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

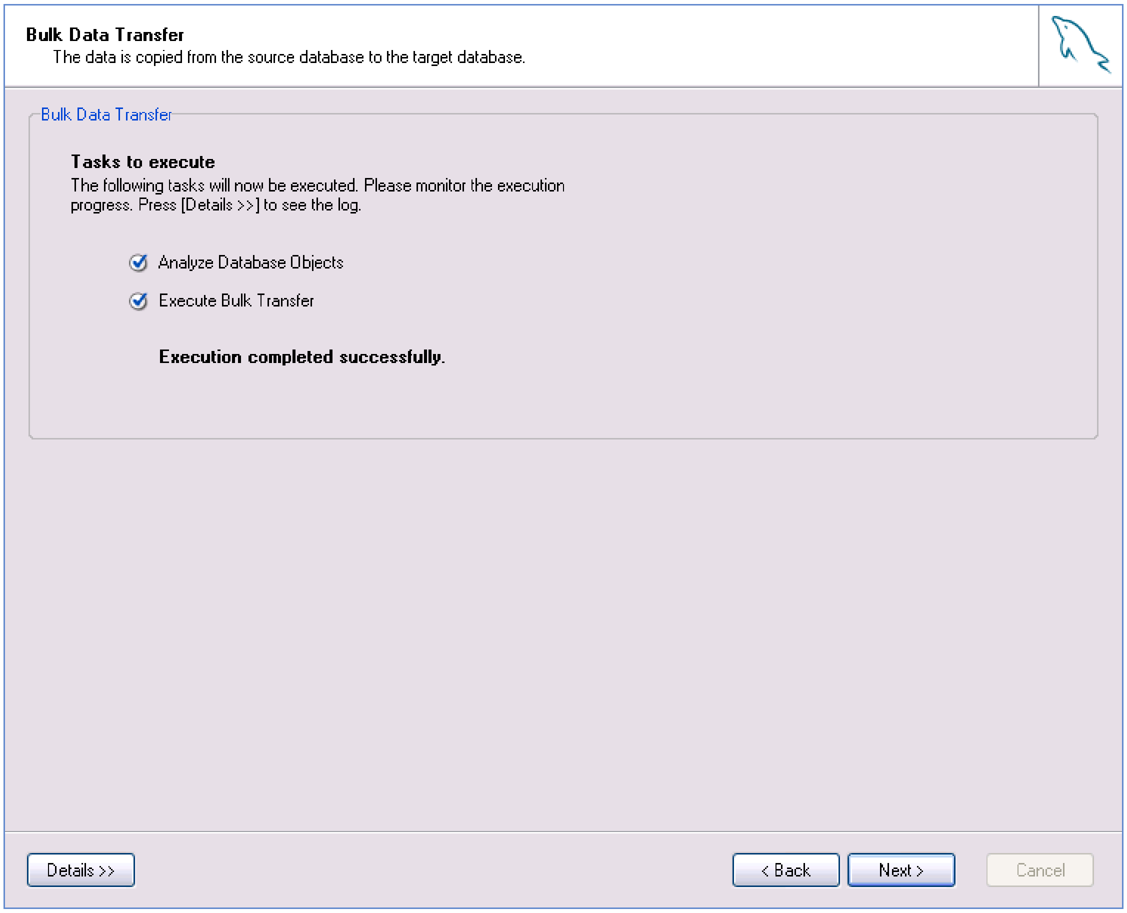
You can access additional options by clicking the `Details` button. If you do not wish to move BLOB data to the target server, check the box next to the `Exclude BLOB values` option. The BLOB data will not be moved to the target server and will not be written to the script file. If you do not wish to 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.

6.16. 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 6.22. The Bulk Data Transfer screen



Bulk Data Transfer

The data is copied from the source database to the target database.

Bulk Data Transfer

Tasks to execute

The following tasks will now be executed. Please monitor the execution progress. Press [Details >>] to see the log.

☑ Analyze Database Objects

☑ Execute Bulk Transfer

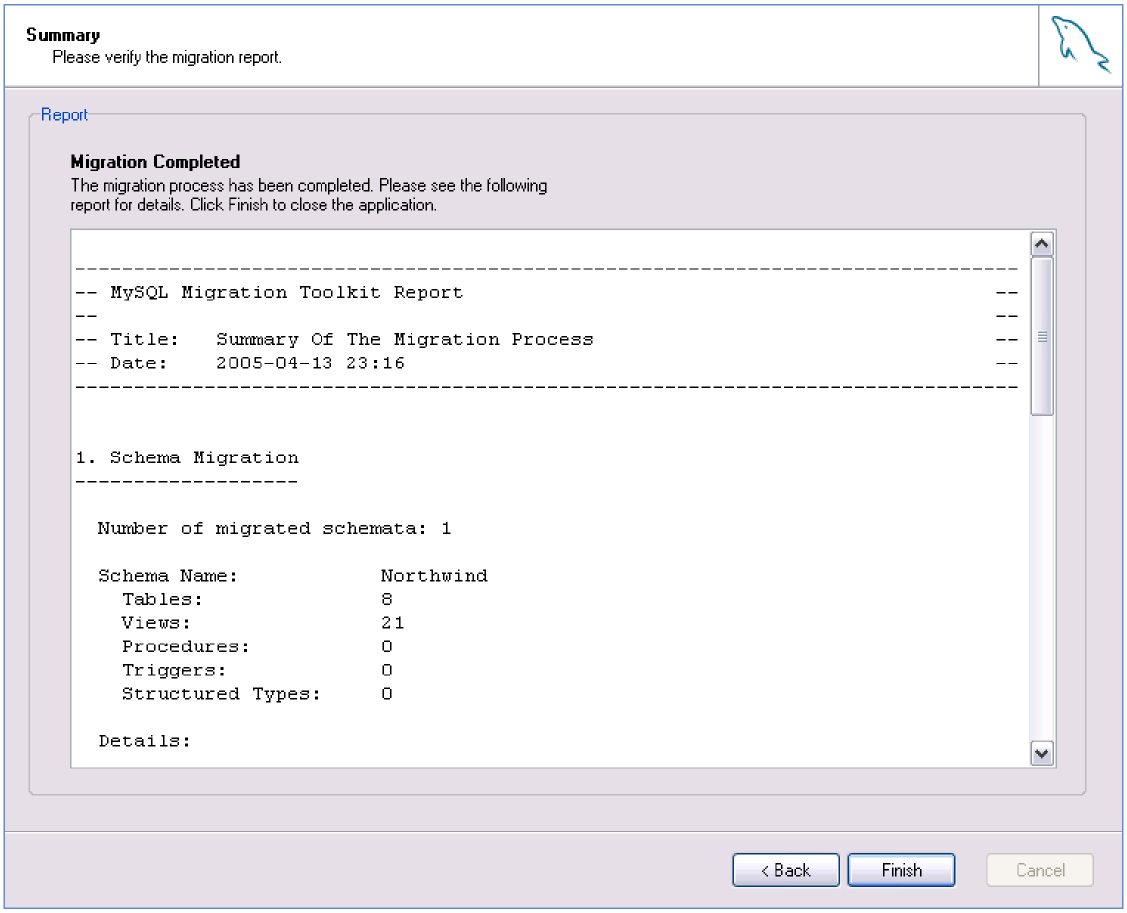
Execution completed successfully.

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

6.17. The Summary Screen

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

Figure 6.23. The Summary screen



Migration Completed

The migration process has been completed. Please see the following report for details. Click Finish to close the application.

```
---------------------------------------------------------------------------------
-- MySQL Migration Toolkit Report                                              --
--                                                                             --
-- Title:   Summary Of The Migration Process                                   --
-- Date:    2005-04-13 23:16                                                   --
---------------------------------------------------------------------------------



1. Schema Migration
-------------------

  Number of migrated schemata: 1

  Schema Name:            Northwind
    Tables:               8
    Views:                21
    Procedures:           0
    Triggers:             0
    Structured Types:     0

  Details:
```

Click the `Finish` button to exit the MySQL Migration Toolkit.

6.18. 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.

Chapter 7. Extending The MySQL Migration Toolkit

7.1. 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. Through the use of its Generic RunTime (`GRT`) Environment, the MySQL Migration Toolkit can support new behavior and data sources using code written in languages such as Java, PHP, and C.

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 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.

7.2. Architecture of the MySQL Migration Toolkit

The GRT is a thin C layer, inspired by Objective C, which allows for dynamic typing and dynamic data objects. The MySQL Migration Toolkit is written on top of this layer and supports the creation of new modules written in C, Java, PHP, and Lua, with support for Python and Mono forthcoming.

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

Reverse engineering modules retreive 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 re-written to match the new source database.

7.3. 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 schemas available on the source RDBMS for the user to choose from.

4. Once the user has selected one or more schema 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 moldule 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.

7.4. Tools Required to Extend the MySQL Migration Toolkit

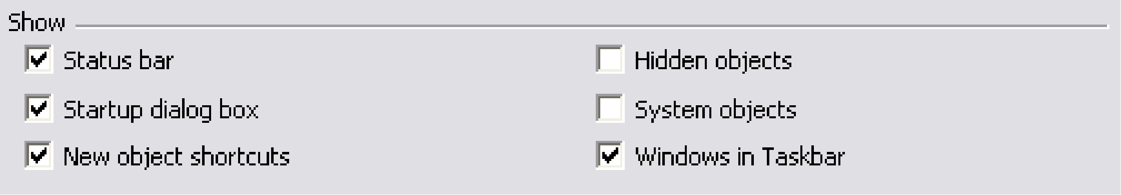
The MySQL AB GUI Team uses Eclipse for development of the MySQL Migration Toolkit modules and recommends Elcipse for use when developing modules for the MySQL Migration Toolkit. See http://www.eclipse.org/ for more information.

Chapter 8. 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 8.1. The show section



Show

☑ Status bar

☑ Startup dialog box

☑ New object shortcuts

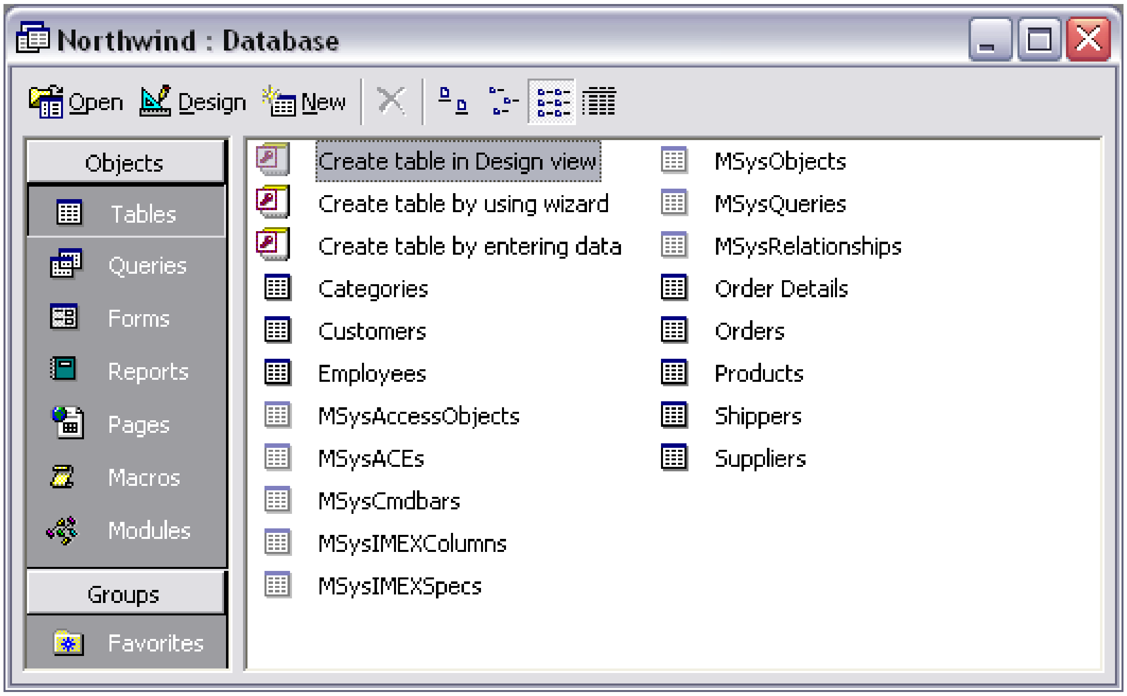
□ Hidden objects

□ System objects

☑ Windows in Taskbar

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

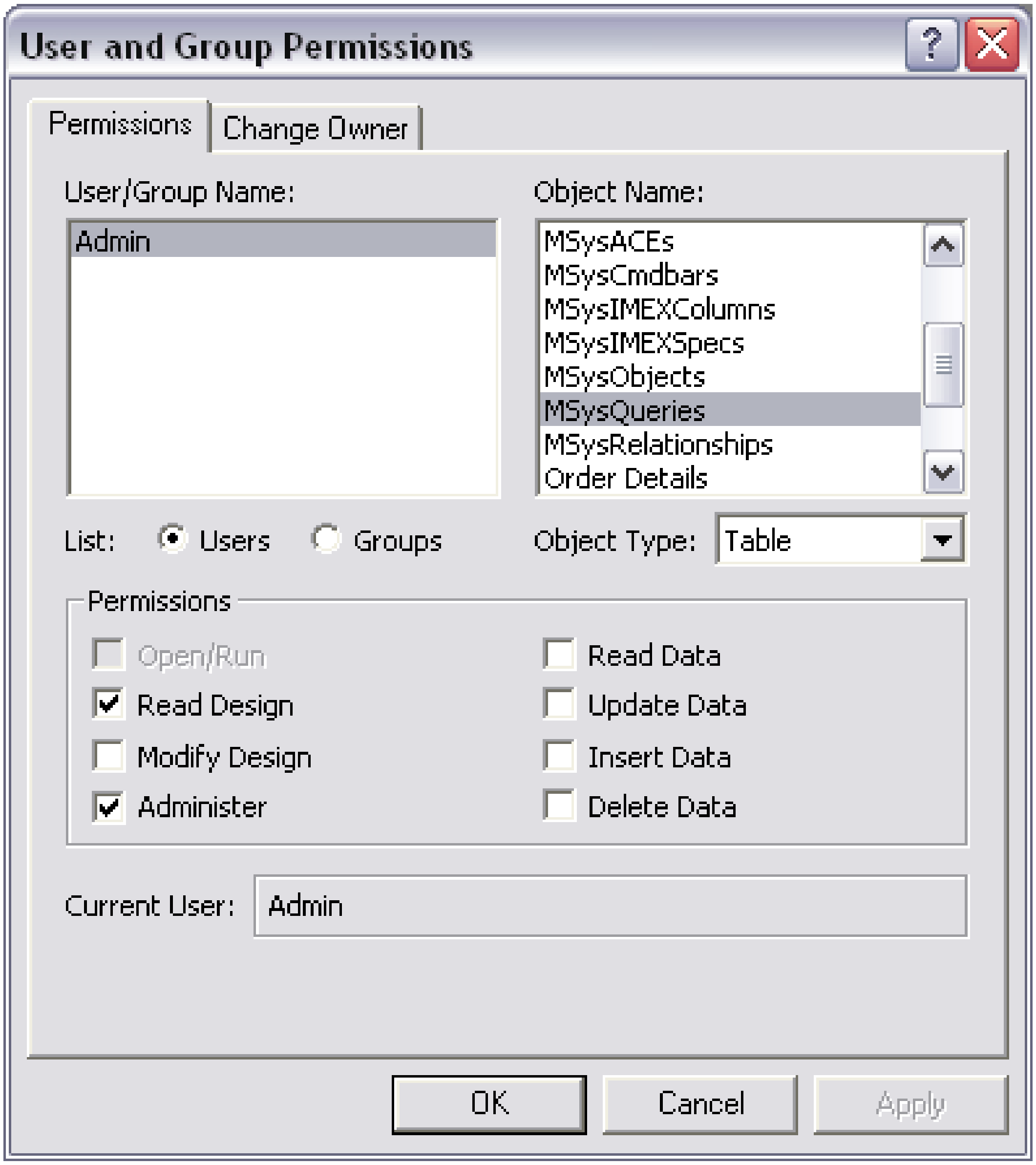
Figure 8.2. The system objects



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 8.3. Granting access to the system objects



User/Group Name:

Admin

Object Name:

MSysACEs
MSysCmdbars
MSysIMEXColumns
MSysIMEXSpecs
MSysObjects
MSysQueries
MSysRelationships
Order Details

List: ● Users ○ Groups

Object Type: Table

Permissions

☐ Open/Run
☑ Read Design
☐ Modify Design
☑ Administer
☐ Read Data
☐ Update Data
☐ Insert Data
☐ Delete Data

Current User: Admin

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.