



Table of Contents

CONNECTION SETTINGS	2
GENERAL SETTINGS	4
SSH SETTINGS (SUPPORTS SSH2 PROTOCOL ONLY)	6
<i>Benefit of SSH Tunneling</i>	7
<i>Password Authentication</i>	8
<i>Public Key Authentication</i>	11
HTTP SETTINGS	14
<i>Uploading the Tunneling Script</i>	15
<i>Setting up HTTP Tunnel</i>	18
SSL SETTINGS	20
<i>Installation of OpenSSL and MySQL</i>	21
<i>Setting up SSL Certificate for MySQL</i>	22
<i>Setting up Client Certificate for Navicat</i>	24
ADVANCED SETTINGS	26
<i>Setting Advanced Database Properties</i>	28

Connection Settings

Navicat for MySQL assembles utilitarian tools to manage your databases. To start managing your databases in Navicat for MySQL, the first thing you require to do is to establish your MySQL Server connection.

Create Connection

Navicat for MySQL provides three typical approaches to establish your connection, click  or choose File ->  **New Connection** to start the setup.

- [General Settings](#)
- [SSH Settings](#)
- [HTTP Settings](#)


Note: A commonly-used protocol - **Secure Sockets Layer (SSL)** is employed for managing the security of a message transmission on the Internet (see [SSL Settings](#) for details).

Navicat for MySQL provides an evaluated account for testing purpose ("Select" privilege only). The remote connection settings are:

- Host name/IP address: server1.navicat.com
- Port: 4406
- User name: navicat
- Password: testnavicat


Note: Navicat for MySQL authorizes you to make connection to remote MySQL Server running on different platform, i.e. Windows, Mac, Linux and UNIX.

To create a new connection with the same properties as one of the existing connection has

- Right-click the connection in the navigation pane and choose  **Duplicate Connection**.
- The newly created connection will be named as "connectionname_copy".

Delete Connection

To delete a connection

- Right-click the connection in the navigation pane and choose  **Delete Connection**.
- Confirm removing in the dialog window.


Open Connection

To open a connection

- Double-click the connection to open in the navigation pane.


Close Connection

To close a connection

- Right-click the connection in the navigation pane and choose  **Close Connection**.

Edit Connection

To edit a connection information

- Close the connection if it is being opened.
- Right-click the connection and choose  **Connection Properties**.

Achieve Connection Information

To achieve a connection information

- Open the connection in the navigation pane.
- Right-click the opened connection and choose **Connection Information**.

General Settings

The following instruction guides you through the process of creating a new connection. To successfully establish a new connection to local/ remote MySQL Server - no matter via SSL, SSH or HTTP, set the connection properties in the corresponding boxes: Connection name, Host name, Port number, User name, and Password.

Connection Name

A friendly name to best describe your connection.

Host name/IP address

A host name where the database is situated or the IP address of the server.

Port

A TCP/IP port for connecting to the database server.

User name

User name for connecting to the database server.

Password

Password for connecting to the server.

See also:

[Advanced Settings](#)

Related topics:

[SSH](#), [HTTP](#)

The image shows a screenshot of the 'Connection' dialog box in Navicat. The dialog has a title bar with a close button. Below the title bar are five tabs: 'General', 'Advanced', 'SSL', 'SSH', and 'HTTP'. The 'General' tab is selected. The dialog contains the following fields and options:

- Connection Name: Localhost
- Host name/IP address: localhost
- Port: 3306
- User name: root
- Password: (empty field)
- Save Password

At the bottom of the dialog, there are three buttons: 'Test Connection', 'OK', and 'Cancel'.

SSH Settings (supports SSH2 Protocol only)

Secure SHell (SSH) is a program to log in into another computer over a network, execute commands on a remote server, and move files from one machine to another. It provides strong authentication and secure encrypted communications between two hosts, known as **SSH Port Forwarding (Tunneling)**, over an insecure network. Typically, it is employed as an encrypted version of Telnet.

In a Telnet session, all communications, including username and password, are transmitted in plain-text, allowing anyone to listen-in on your session and steal passwords and other information. Such sessions are also susceptible to session hijacking, where a malicious user takes over your session once you have authenticated. SSH serves to prevent such vulnerabilities and allows you to access a remote server's shell without compromising security.

- [Benefit of SSH Tunneling.](#)

To ensure that the incoming connection request is from you, SSH can use a password, or public/private key pair (also called public key) authentication mechanism.

- [Password Authentication.](#)
- [Public Key Authentication.](#)

Note: Please make sure that the parameter - "AllowTcpForwarding" in the Linux Server must be set to value "yes", otherwise, the SSH port forwarding will be disabled. To look for the path: `/etc/ssh/sshd_config` .By default, the SSH port forwarding should be enabled. Please double check the value settings.

****** Even the server support SSH tunnel, however, if the port forwarding being disabled, Navicat for MySQL cannot connect via SSH Port 22.

See also:

[Advanced Settings](#)

Related topic:

[SSL](#)

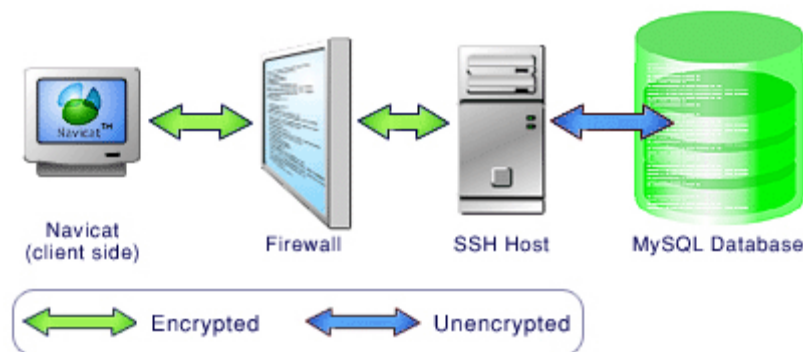
Benefit of SSH Tunneling

SSH has a wonderful feature called SSH Port Forwarding, sometimes called SSH Tunneling, which allows you to establish a secure SSH session and then tunnel arbitrary TCP connections through it. Tunnels can be created at any time, with almost no effort and no programming, which makes them very appealing. SSH Port Forwarding can be used for secure communications in a myriad of different ways.

Many Hosting Companies that provide MySQL hosting will block access to the MySQL Server from outside the hosting company's network, and only grant access to users connecting from localhost.

There are several benefits to using SSH:



- Connection to a MySQL server from behind a firewall when the MySQL server port is blocked.
- Automatic authentication of users, no passwords sent in plain text to prevent the stealing of passwords.
- Multiple strong authentication methods that prevent such security threats as spoofing identity.
- Encryption and compression of data for security and speed.
- Secure file transfer.



Password Authentication

Using this mode, SSH is almost identical to the program telnet. When you make a connection, you are asked for your password. You type it in and you are either logged in or denied. Your password is first encrypted and then sent over the network and then decrypted at the remote host. This is the mode that most users will be encouraged to use, as it requires no additional setup or configuration.

The following instruction guides you through the process of configuring a SSH connection using Password Authentication. To successfully establish a SSH connection, set the SSH connection properties in the corresponding boxes: Host name/IP address, Port number, User name, Authentication Method and Password.

1. Click  or choose File ->  **New Connection** to set up the Connection Properties.
2. Select the SSH tab and enable **Use SSH Tunnel**.
3. Fill in the required information:

Host name/IP address

A host where SSH server is activated.

Port

A port where SSH server is activated, by default it is 22.

User name

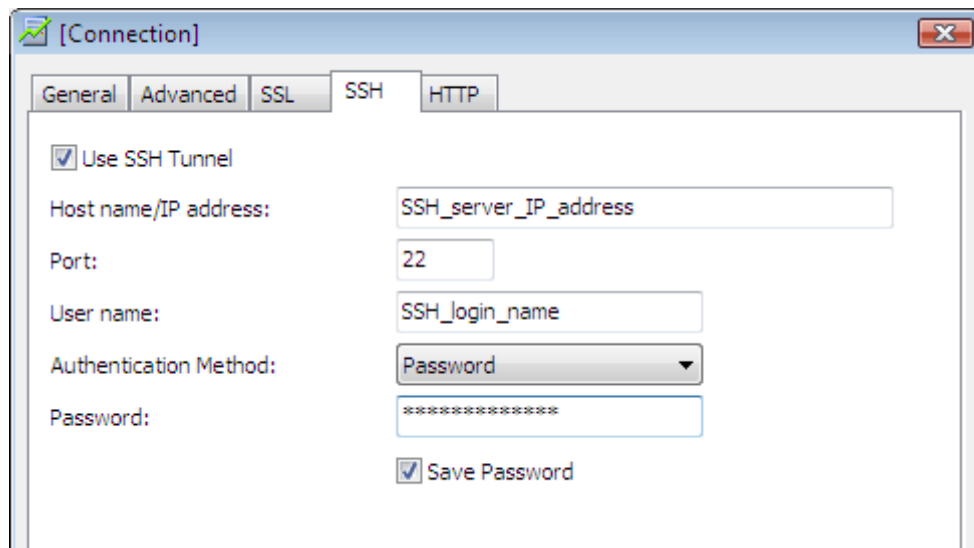
A user on Linux machine. (It is a Linux user. It is not a user of MySQL Server.)

Authentication Method

Choose between Password Authentication and [Public Key Authentication](#)

Password

It is a Linux user password.



- Navicat for MySQL host name at the General Settings page should be set relatively to the SSH server in this case. For example: host_of_mysqlatabase shown below is the host address, which provided by your hosting company, of your remote MySQL database.

Connection Name

A friendly name to best describe your connection.

Host name/IP address

A host where MySQL Server is located in point of view SSH server. If SSH and MySQL Server are on the same machine, it is equal to SSH Host, or may be 'localhost'.

Port

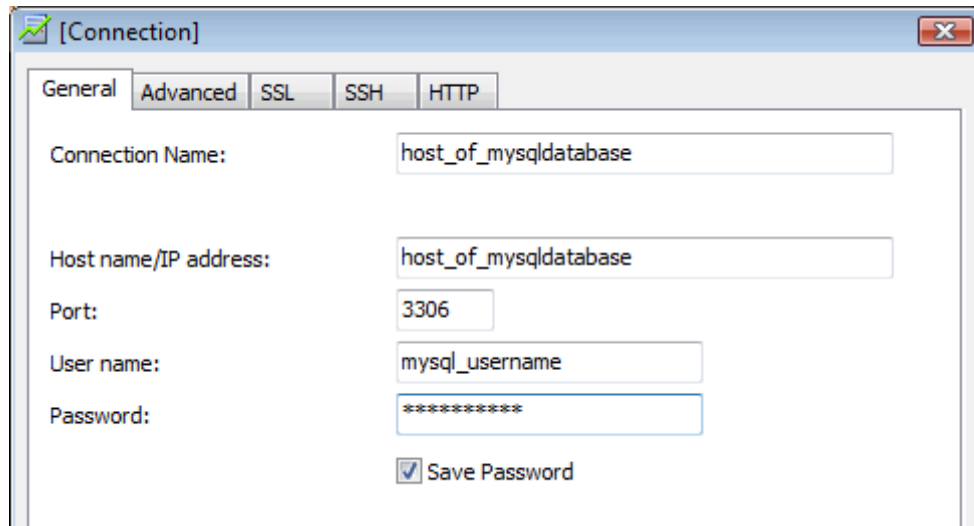
A port of MySQL Server on Remote Host, by default it is 3306.

User name

It is a MySQL Server user name.

Password

It is a password of the MySQL user.



See also:

[Advanced Settings](#)

Related topics:

[Public Key Authentication](#)



Public Key Authentication

Public-key Authentication is based on the use of digital signatures and provides the best authentication security.

For Public Key Authentication to work four things are needed:

- the remote server(s) you are connecting must have your public key.
- the local client you are connecting from must have your private key.
- the remote server must be configured to allow you to login using your public key.
- the local client must be configured to use your private key while logging into remote server.

The following instruction guides you through the process of configuring a SSH connection using Public Key Authentication. To successfully establish a SSH connection , set the SSH connection properties in the corresponding boxes: Host name/IP address, Port number, User name, Authentication Method, Private Key and Passphrase.

1. Click  or choose File ->  **New Connection** to set up the Connection Properties.
2. Select the SSH tab and enable **Use SSH Tunnel**.
3. Fill in the required information:

Host name/IP address

A host where SSH server is activated.

Port

A port where SSH server is activated, by default it is 22.

User name

A user on Linux machine. (It is a Linux user. It is not a user of MySQL Server.)

Authentication Method

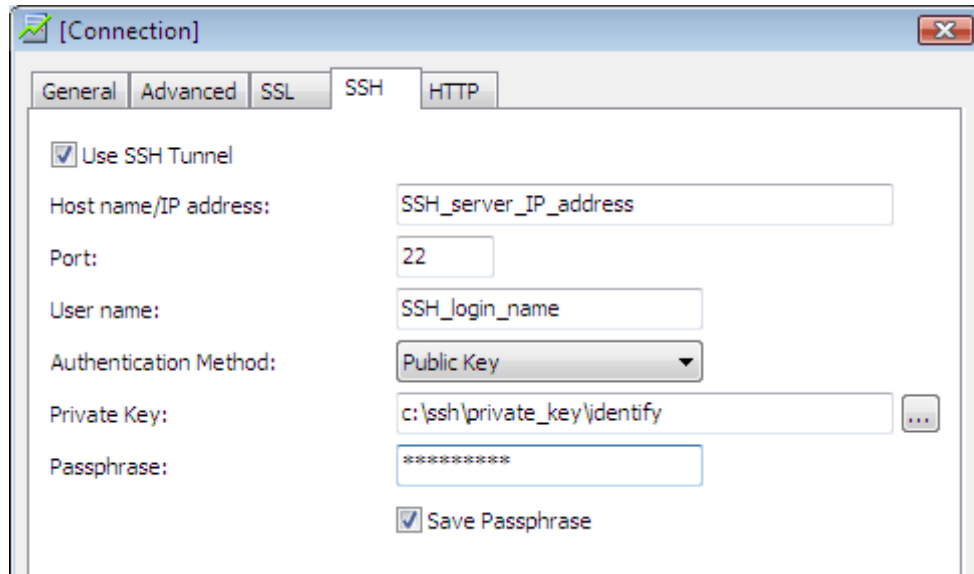
Choose between [Password Authentication](#) and Public Key Authentication

Private Key

It is used together with your public key. The private key should be readable only by you.

Passphrase

A passphrase is exactly like a password, except that it applies to the keys you are generating and not an account. The passphrase be any length under 1024 characters.



- Navicat for MySQL host name at the General Settings page should be set relatively to the SSH server in this case. For example: host_of_mysqlatabase shown below is the host address, which provided by your hosting company, of your remote MySQL database.

Connection Name

A friendly name to best describe your connection.

Host name/IP address

A host where MySQL Server is located in point of view SSH server. If SSH and MySQL Server are on the same machine, it is equal to SSH Host, or may be 'localhost'.

Port

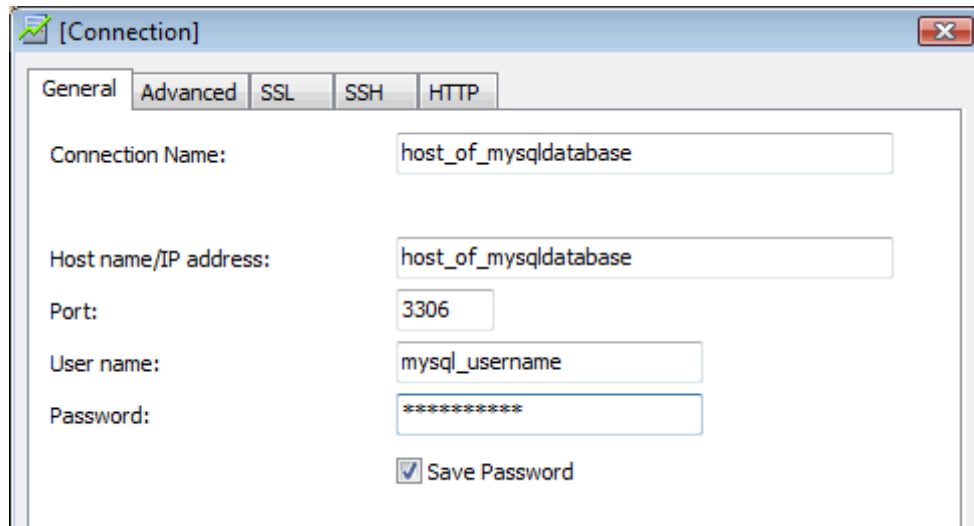
A port of MySQL Server on Remote Host, by default it is 3306.

User name

It is a MySQL Server user name.

Password

It is a password of the MySQL user.



See also:

[Advanced Settings](#)

Related topics:

[Password Authentication](#)

HTTP Settings

HTTP Tunneling is a method for connecting to a MySQL server that uses the same protocol (http://) and the same port (port 80) as a webserver does. It is used while your ISPs do not allow direct connections to their MySQL server, but allows establishing HTTP connections.

Steps of setting up HTTP Connection for MySQL Server:

1. [Uploading the Tunneling Script.](#)
2. [Setting up HTTP Tunnel.](#)

Note: HTTP Tunnel and SSH Tunnel cannot function simultaneously. The SSH Tunnel is disabled when you select the HTTP Tunnel and vice versa.

See also:

[Advanced Settings](#)

Related topics:

[General Settings](#)

Uploading the Tunneling Script

To use this connection method, first thing you need to do is to upload the tunneling script - **ntunnel_mysql.php** to the webserver where MySQL Server is located.

Note: **ntunnel_mysql.php** is available in the Navicat installation folder.

The following instruction guides you through the process of uploading **ntunnel_mysql.php** to your webserver. In this tutorial, you require a FTP client - [WS FTP](#) to make connection to your webserver. You can use any FTP client you are familiar with.

1. Startup WS_FTP and configure Session Profile Information.

Profile Name

This is the name of the FTP session profile, and can be anything you desire to identify the connection you are creating.

Host name/Address

A fully qualified Internet host name or an IP address.

Host Type

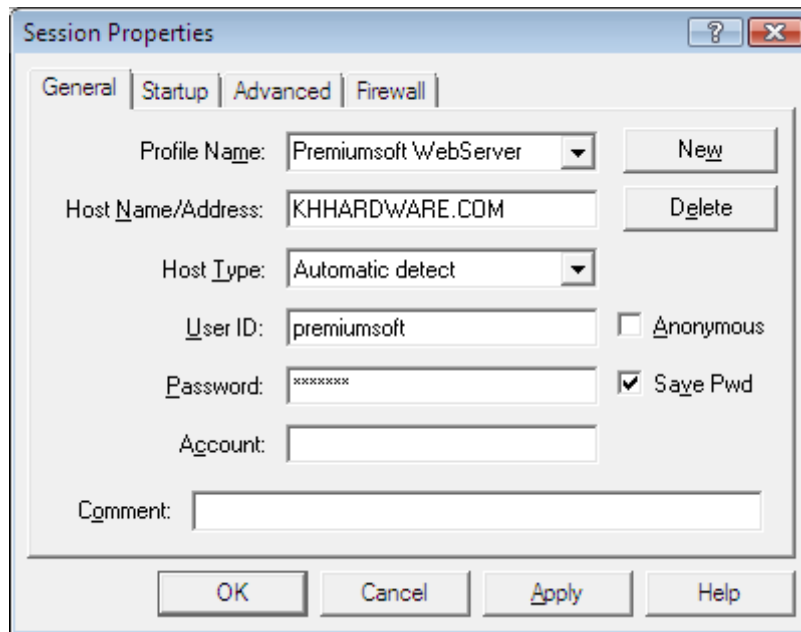
If you know the host type, select it from the drop down list. Otherwise, try **Automatic detect**.

User ID

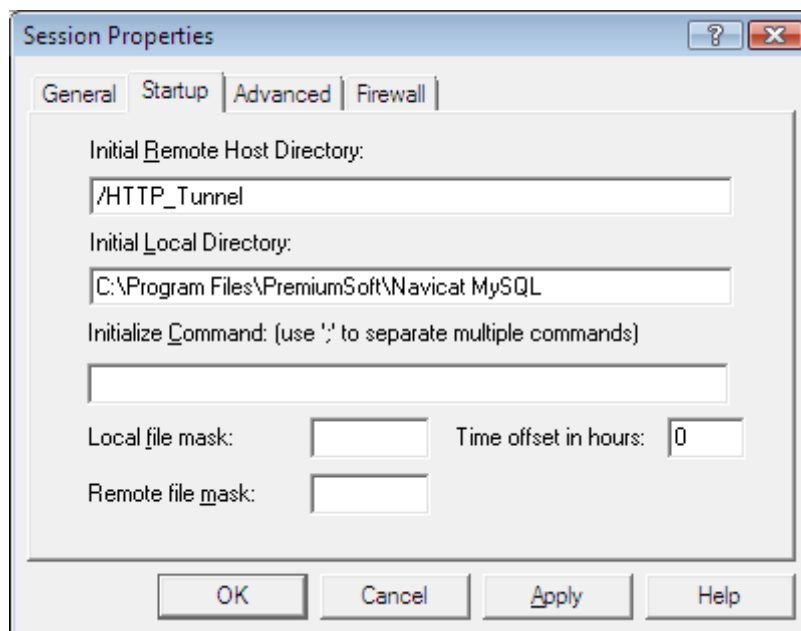
Enter the User ID you want to use for this session profile.

Password

Enter the Password you want to use for the User ID you entered.

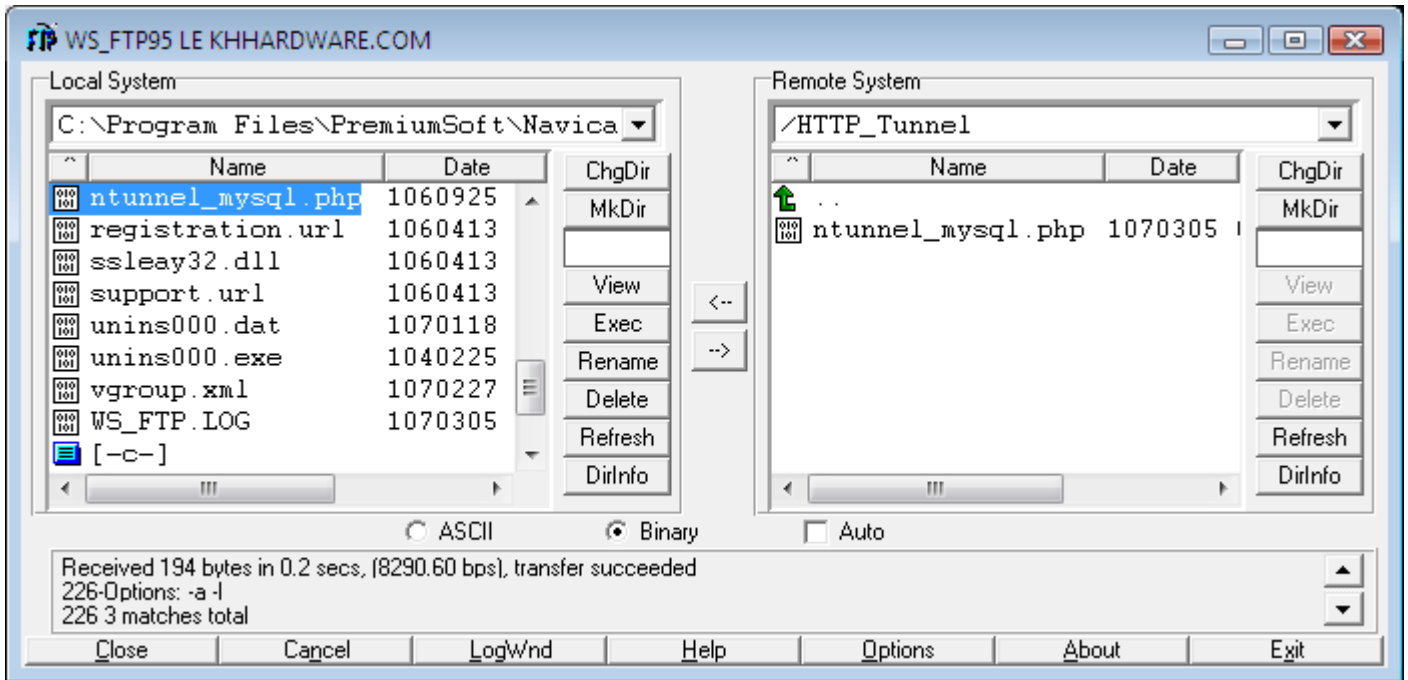


2. Select the Startup tab and enter Initial Directories for both Local and Remote system.



3. To upload the tunneling script:

1. Browse the ntunnel_mysql.php on the source system.
2. Open the directory to which you want to transfer files on the destination system.
3. Transfer the files using the left and right arrow buttons located between the list boxes, i.e. Click the right arrow button to transfer files from the local to the remote system.





See also:

[Advanced Settings](#)

Setting up HTTP Tunnel

The following instruction guides you through the process of configuring a HTTP connection.

1. Click  or choose File ->  **New Connection** to set up the Connection Properties.
2. Select the HTTP tab and enable **Use HTTP Tunnel**.
3. Enter URL of the tunneling script.



4. If the **ntunnel_mysql.php** is hosted in a password protected server or you have to access internet over a proxy server, you can provide the required authentication details.

Authentication Proxy

Use password authentication
 User name:
 Password:
 Save Password

Use certificate authentication
 Client Key ...
 Client Certificate ...
 CA Certificate ...
 Passphrase

Authentication Proxy

Use Proxy
 Host:
 Port:
 User name:
 Password:
 Save Password

5. Navicat for MySQL host name at the General Settings page should be set relatively to the HTTP server in this case. For example: `host_of_mysqlatabase` shown below is the host address, which provided by your hosting company, of your remote MySQL database.

Connection Name

A friendly name to best describe your connection.

Host name/IP address

A host where MySQL Server is located in point of view HTTP server.

Port

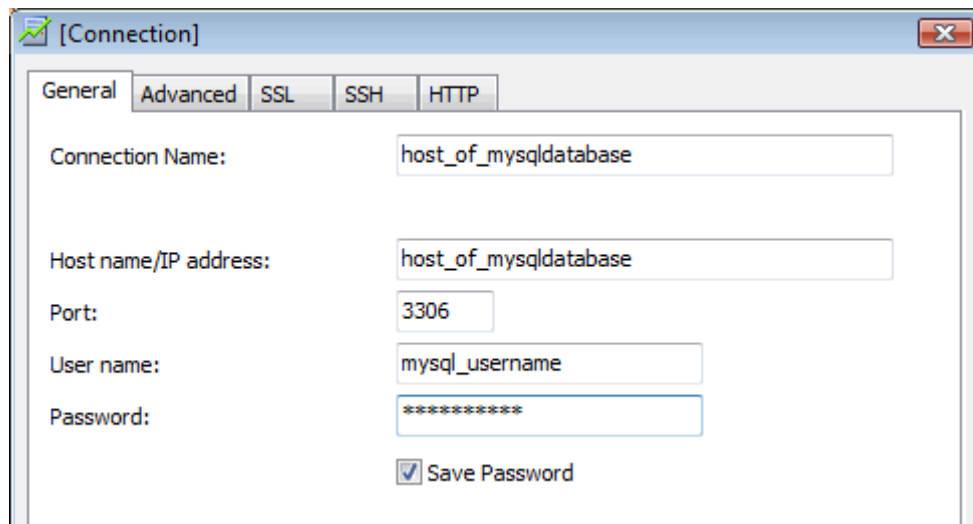
A port of MySQL Server on Remote Host, by default it is 3306.

User name

It is a MySQL Server user name.

Password

It is a password of the MySQL user.



The screenshot shows the 'Connection' dialog box in Navicat, with the 'General' tab selected. The fields are filled with the following values:

Field	Value
Connection Name:	host_of_mysqlatabase
Host name/IP address:	host_of_mysqlatabase
Port:	3306
User name:	mysql_username
Password:	*****

The 'Save Password' checkbox is checked.

See also:

[Advanced Settings](#)

SSL Settings

Secure Sockets Layer(SSL) is a protocol for transmitting private documents via the Internet. To get a secure connection to work with MySQL Server, the first thing you need to do is to install OpenSSL Library and download MySQL Database Source.

Steps of setting up SSL Connection for MySQL Server and Navicat:

1. [Installation of OpenSSL and MySQL.](#)
2. [Setting up SSL Certificate for MySQL.](#)
3. [Setting up Client Certificate for Navicat.](#)

See also:

[Advanced Settings](#)

Related topics:

[General Settings](#), [SSH](#)

Installation of OpenSSL and MySQL

Installing OpenSSL

1. Download OpenSSL - <http://www.openssl.org>
2. Linux command : [zcat 0.96l.tar.gz | tar xvf -]
3. Linux command : [./config]
4. Linux command : [make]
5. Linux command : [make install]

Installing MySQL

1. Download MySQL - <http://www.mysql.com>
2. Linux command : [./configure --with -vio --with -openssl]
3. Linux command : [make]
4. Linux command : [make install]

Note: Please ensure if MySQL Server supports OpenSSL using query statement:
[SHOW VARIABLES LIKE 'have_openssl']; - Returns value = YES

See also:

Step 2: [Setting up SSL Certificate for MySQL](#)

Setting up SSL Certificate for MySQL

To create server/client side Certificate, login to the Linux Server as root and employ the Shell Command below:

1. `DIR=`pwd`/openssl`
2. `PRIV=$DIR/private`
3. `mkdir $DIR $PRIV $DIR/newcerts`
4. `cp /usr/share/ssl/openssl.cnf $DIR`
5. `replace ./demoCA $DIR -- $DIR/openssl.cnf`
6. Generation of Certificate Authority(CA)

```
/usr/local/ssl/bin/openssl req -new -x509 -keyout $PRIV/cakey.pem -out $DIR/cacert.pem -config $DIR/openssl.cnf
```

Note: If "PEM" is required, please enter different "PEM pass" via steps below.

7. Create server request and key

```
/usr/local/ssl/bin/openssl req -new -keyout $DIR/server-key.pem -out $DIR/server-req.pem -days 3600 -config $DIR/openssl.cnf
```

8. Remove the passphrase from the key (optional)

```
/usr/local/ssl/bin/openssl rsa -in $DIR/server-key.pem -out $DIR/server-key.pem
```

9. Sign server cert

```
/usr/local/ssl/bin/openssl ca -policy policy_anything -out $DIR/server-cert.pem -config $DIR/openssl.cnf -infiles $DIR/server-req.pem
```

10. Create client request and key

```
/usr/local/ssl/bin/openssl req -new -keyout $DIR/client-key.pem -out $DIR/client-req.pem -days 3600 -config $DIR/openssl.cnf
```

11. Remove a passphrase from the key (optional)

```
/usr/local/ssl/bin/openssl rsa -in $DIR/client-key.pem -out $DIR/client-key.pem
```

12. Sign client cert

```
/usr/local/ssl/bin/openssl ca -policy policy_anything -out $DIR/client-cert.pem  
-config $DIR/openssl.cnf -infile $DIR/client-req.pem
```

13. Create a **my.cnf** file for testing the Certificates. Store it either in **/etc** or MySQL data directory (typically **/usr/local/var** for source installation)

my.cnf example content:

```
[client]  
ssl-ca=$DIR/cacert.pem  
ssl-cert=$DIR/client-cert.pem  
ssl-key=$DIR/client-key.pem  
[mysqld]  
ssl-ca=$DIR/cacert.pem  
ssl-cert=$DIR/server-cert.pem  
ssl-key=$DIR/server-key.pem
```

14. To start MySQL daemon:


```
/usr/local/libexec/mysqld -u mysql &  
or  
/usr/local/sbin/mysqld -u &
```

See also:

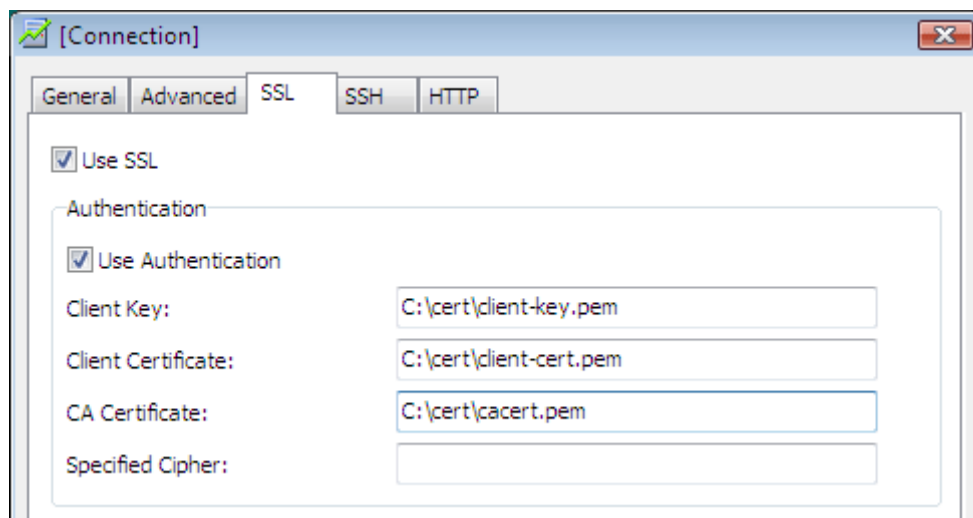
Step 3: [Setting up Client Certificate for Navicat](#)

Setting up Client Certificate for Navicat


The following instruction guides you through the process of configuring a connection between Navicat and MySQL Server using SSL. To successfully establish a SSL connection, please complete [Step 1: Installation of OpenSSL and MySQL](#) and [Step 2: Setting up SSL Certificate for MySQL](#) , and set the connection properties in the corresponding boxes: Client Key file, Client Certificate file, CA Certificate file and Specified Cipher.

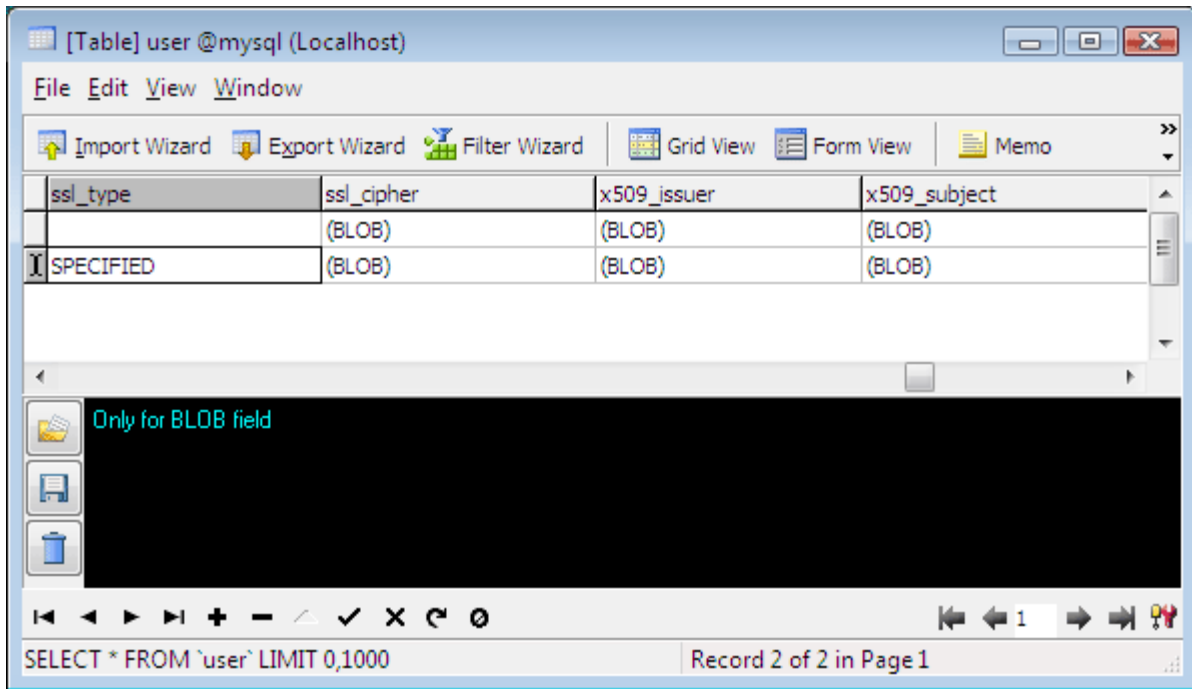
1. Click  or choose **Connection -> New Connection** to set up the Connection Properties.
2. Select the SSL tab and enable **Use SSL**.
3. To provide authentication details, fill in the required information:

Client Key file, **Client Certificate file** and **CA Certificate file** are usually stored in your Server - `/usr/local/openssl`. Please copy them from the remote server to local computer.



Specified Cipher (optional) is only required while **ssl_type** field has been set to "**SPECIFIED**" - [ssl_type can be found in a system database called "mysql" -> table called "user"]. Example of Specified Cipher is "EDH-RSA-DES-CBC3-SHA" which can be filled in either through the Connection Properties shown above or the "mysql" database -> "user" table -> "ssl_cipher" blob field shown below.

Note: You are allowed to store your Specified Cipher into a text file in order to load  into the "ssl_cipher" blob field.



See also:

[Advanced Settings](#)

Advanced Settings

Customize connection options according to your needs. The detailed description is given below:

Settings Save Path

When a new connection being established, Navicat for MySQL will create a subfolder under the Settings Save Path. Most files are stored within this subfolder:

Navicat Objects	File Extensions
Query	.sql
Export Query Result Profile	.npeq
Export View Result Profile	.npev
Backup	compressed (.psc), uncompressed (.psb)
Backup Profile	.npb
Report	.rtm
Import Wizard Profile	.npi
Export Wizard Profile	.npe

Other files are located in the **profiles** directory, which is a sub-directory of your Navicat installation folder, e.g. C:\Program Files\PremiumSoft\Navicat MySQL\profiles.

Other Profiles	File Extensions
Data Transfer	.npt
Data Synchronization	.npd
Structure Synchronization	.nps
Batch Job	.npj
Virtual Grouping	vgroup.xml - stores how the objects are categorized.

Hint: All your connection settings are stored in registry.

Encoding

Choose a codepage to communicate with MySQL Server while MySQL character set not being employed.

Keepalive Interval (sec)

This option allows you to keep the connection with the server alive by pinging it. You can set the period between pings in the edit field.

Use MySQL character set

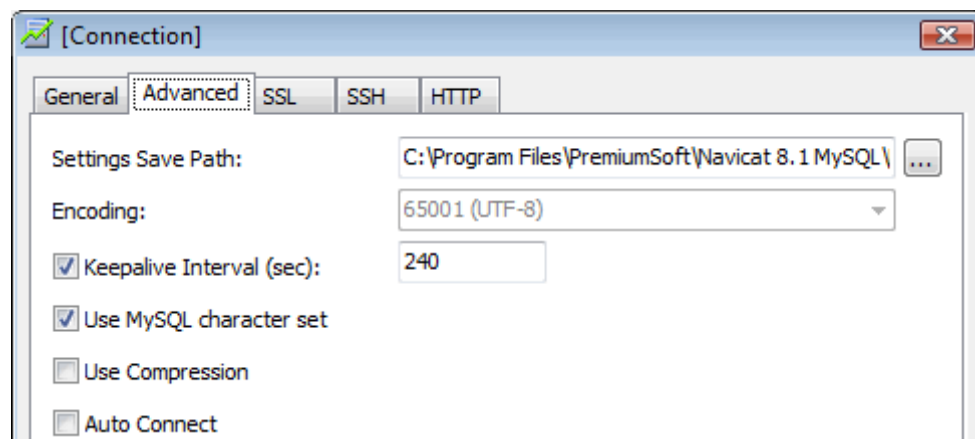
This option should be enabled if employing MySQL 4.1 or above.

Use Compression

This option allows you to use compression protocol. It is used if both client and server support zlib compression, and the client requests compression.

Auto Connect

With this option on, Navicat for MySQL automatically open connection with the registered database at application startup.



Setting Advanced Database Properties

Set the advanced database properties, which are not obligatory. To start working with advanced database settings, check the **Use Advanced Connections**. The detailed description is given below:

Show Selected Databases

To show the selected databases in the **close** state in the navigation pane

- Click the preferable databases in the Databases list box, the check box will show as

To show the selected databases in the **open** state in the navigation pane

- Double-click the preferable databases in the Databases list box, the check box will show as

Add Hidden Database

To add a hidden database

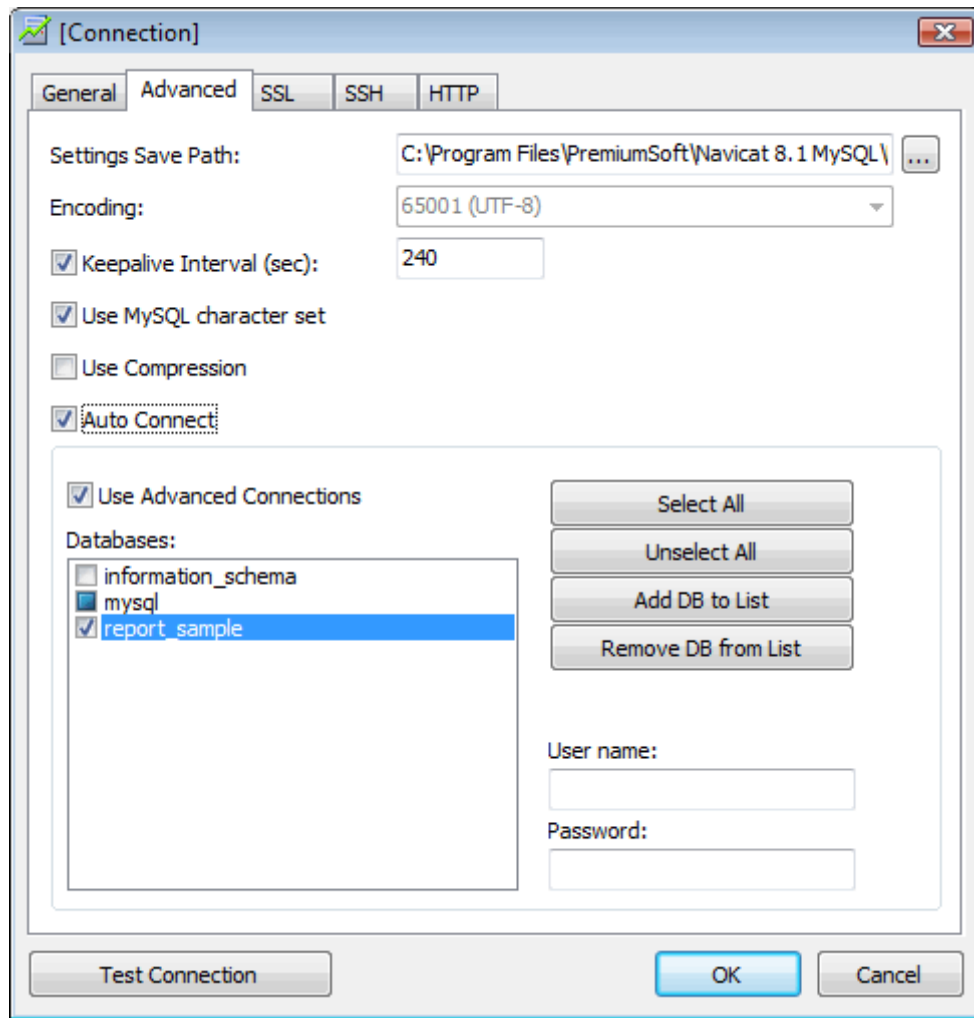
- Click **Add DB to List** button.
- Enter the database name.
- Select the newly added database in the Databases list box.
- Enter **User name** and **Password** which provide by your ISP.

Remove Database

To remove a database

- Select the database to remove in the Databases list box.
- Click **Remove DB from List** button.

Note: The database will be just removed from the Databases list box, it will still exist in the MySQL Server.



See also:

[Advanced Settings](#)

Related topic:

[Connection Settings](#)