目錄

入門 **2**

連線設定 4

MySQL 一般設定 *6*

Oracle 一般設定 *7*

Basic 連線一般設定 8

TNS 連線一般設定 9

PostgreSQL 一般設定 *10*

SQLite 一般設定 *12*

SQL Server 一般設定 *13*

SSH 設定 (只限於 MySQL、Oracle、PostgreSQL 及 SQL Server 並只支援 SSH2 通訊協定) *14*

SSH 通道的好處 15

密碼驗證 16

公開金鑰驗證 18

HTTP 設定(只限於 MySQL、PostgreSQL 及 SQLite) *20*

SSL 設定(只限於 MySQL 及 PostgreSQL) *21*

安裝 OpenSSL 及 MySQL 或 PostgreSQL 22

設定 MySQL 或 PostgreSQL 的 SSL 憑證 23

設定 Navicat 的用戶端憑證 26

進階設定 *28*

入門

要在 Navicat Report Viewer 開始檢視報表，你要首先用連線視窗建立一個或多個連線。如果你是一個 MySQL、Oracle、PostgreSQL、SQLite 或 SQL Server 新手和不肯定如何工作，你可能想看看：

- MySQL 使用者手冊
- Oracle 資料庫文件
- PostgreSQL 使用者手冊
- SQLite 使用者手冊
- SQL Server MSDN Library

按一下 或選擇檔案 -> **新增連線** 來設置連線內容。

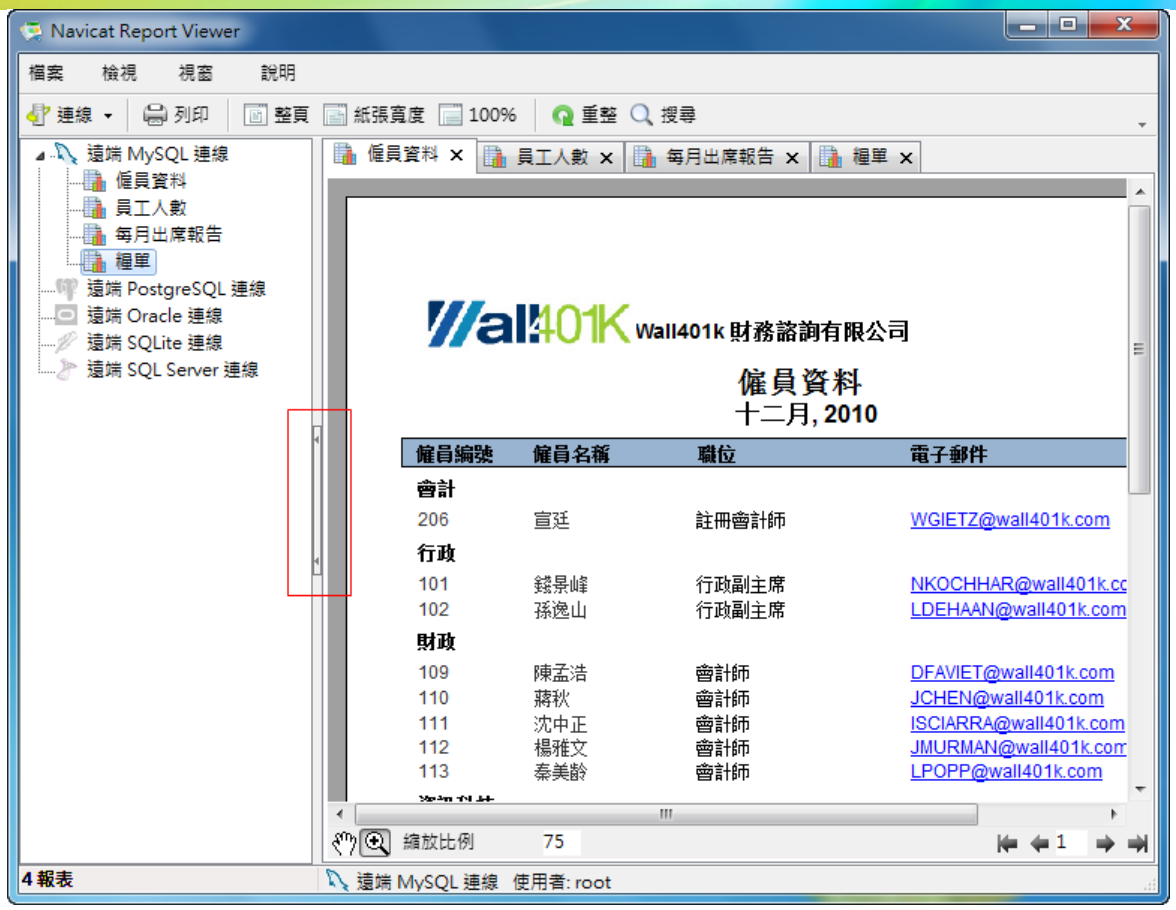
- 連線設定

在建立連線後，你需要在檢視及列印你的報表前放置報表檔案（.rtm）到報表位置。

Navicat Report Viewer 瀏覽器！

Navicat Report Viewer 視窗中包含一個物件窗格顯示已開啟的報表。左邊的是一個導覽窗格，幫助你瀏覽目前的報表。在視窗頂部的工具列提供其他控制項，你可以用它來操作你的報表。

要檢視或隱藏此視窗，按一下下面的紅色的標示。


Navicat Report Viewer
檔案 檢視 視窗 說明
連線 列印 整頁 紙張寬度 100% 重整 搜尋
遠端 MySQL 連線
僱員資料
員工人數
每月出席報告
糧單
遠端 PostgreSQL 連線
遠端 Oracle 連線
遠端 SQLite 連線
遠端 SQL Server 連線
僱員資料 員工人數 每月出席報告 糧單
Wall401k 財務諮詢有限公司
僱員資料
十二月, 2010
僱員編號 僱員名稱 職位 電子郵件
會計
206 宣廷 註冊會計師 WGIETZ@wall401k.com
行政
101 錢景峰 行政副主席 NKOCHHAR@wall401k.cc
102 孫逸山 行政副主席 LDEHAAN@wall401k.com
財政
109 陳孟浩 會計師 DFAVIET@wall401k.com
110 蔣秋 會計師 JCHEN@wall401k.com
111 沈中正 會計師 ISCIARRA@wall401k.com
112 楊雅文 會計師 JMURMAN@wall401k.com
113 秦美齡 會計師 LPOPP@wall401k.com
縮放比例 75
1
4 報表
遠端 MySQL 連線 使用者: root

連線設定

Navicat Report Viewer 集合實用工具來檢視及列印你的報表。要開始在 Navicat Report Viewer 管理你的報表，首先你需要做的是建立你的伺服器連線。

建立連線

Navicat Report Viewer 提供了三種典型的方法來建立你的連線， 按一下 或選擇檔案 -> **新增連線** 來開始設置。

- MySQL 一般設定
- Oracle 一般設定
- PostgreSQL 一般設定
- SQLite 一般設定
- SQL Server 一般設定
- SSH 設定（只限於 MySQL、Oracle、PostgreSQL 及 SQL Server）
- HTTP 設定（只限於 MySQL、PostgreSQL 及 SQLite）

注意：對於 MySQL 或 PostgreSQL 伺服器，一個常用的通訊協定 - **Secure Sockets Layer (SSL)** 是用於管理在網際網路上傳輸信息的安全（詳細資料請看 SSL 設定）。

注意：Navicat Report Viewer 能讓你連線到不同平台上的遠端伺服器，即 Windows、Mac、Linux 及 UNIX。

刪除連線

刪除一個連線

- 在導覽窗格中的連線上按一下滑鼠右鍵並選擇 **刪除連線**。
- 在對話視窗確認刪除。

開啟連線

開啟一個連線

- 在導覽窗格中按兩下以開啟連線。

關閉連線

關閉一個連線

- 在導覽窗格中的連線上按一下滑鼠右鍵並選擇 **關閉連線**。

編輯連線

編輯一個連線的資訊

- 如果連線開啟著，關閉它。
- 在連線上按一下滑鼠右鍵並選擇 **連線內容**。

取得連線資訊

取得一個連線的資訊

- 在導覽窗格中開啟連線。
- 在已開啟的連線上按一下滑鼠右鍵並選擇 **連線資訊**。

MySQL 一般設定

下列的指示引導你通過建立一個新連線的程序。要成功地建立一個新的連線到本機或遠端 MySQL - 不管通過 SSL、SSH 或 HTTP，都要在相應的框中設定連線內容：連線名稱、主機名稱、埠號碼、使用者名稱及密碼。

預設情況下，MySQL 提供使用者名稱為「root」和密碼為空白。

連線名稱
一個最恰當地描述你的連線名稱。

主機名稱或 IP 位址
資料庫所在的主機名或伺服器的 IP 位址。

埠
連線到資料庫伺服器的 TCP/IP 埠。

使用者名稱
連線到資料庫伺服器的使用者名稱。

密碼
連線到伺服器的密碼。

如果你的互聯網服務供應商（ISP）不提供直接存取其伺服器，安全通道協定 SSH 或 HTTP 是另一種解決辦法。

參見：
進階設定

相關主題：
SSL、SSH、HTTP

Oracle 一般設定

下列的指示引導你通過建立一個新連線的程序。要成功地建立一個新的連線到本機或遠端 Oracle - 不管通過 SSH，都要在相應的框中設定連線內容：連線名稱、主機名稱、埠號碼、使用者名稱及密碼。

預設情況下，Oracle 在安裝時已建立一些使用者帳號。管理帳號：SYS、SYSTEM、SYSMAN 及 DBSNMP。樣本結構描述帳號：SCOTT、HR、OE、OC、PM、IX 及 SH。

Navicat 支援兩種 Oracle 伺服器連線：

- Basic 連線
- TNS 連線

如果你的互聯網服務供應商（ISP）不提供直接存取其伺服器，安全通道協定 SSH 是另一種解決辦法。

相關主題：
SSH

Basic 連線一般設定

連線名稱

一個最恰當地描述你的連線名稱。

連線類型

連線到伺服器的連線類型：**Basic** 或 TNS。

> **Basic**
> 在 Basic 模式中，Navicat Report Viewer 通過 Oracle Call Interface (OCI) 連線到 Oracle。OCI 是一個應用程式介面，讓程式開發人員使用第三代語言原有程序或函式呼叫來存取 Oracle 資料庫及控制全部 SQL 語句執行的階段。OCI 是一個標準資料庫存取的函式庫和動態連結庫形式檢索函式。

主機名稱或 IP 位址

資料所在的主機名或伺服器的 IP 位址。

埠

連線到資料庫伺服器的 TCP/IP 埠。

Service Name/SID

設定使用者連線時用的 Service Name/SID。選擇相應的選項按鈕。

使用者名稱

連線到資料庫伺服器的使用者名稱。

密碼

連線到伺服器的密碼。

參見：

進階設定

相關主題：

TNS、SSH

TNS 連線一般設定

連線名稱

一個最恰當地描述你的連線名稱。

連線類型

連線到伺服器的連線類型：Basic 或 **TNS**。

> **TNS**
>
> 在 TNS 模式中，Navicat Report Viewer 使用在 tnsnames.ora 檔案中的別名項目通過 Oracle Call Interface (OCI) 連線到 Oracle 伺服器。OCI 是一個應用程式介面，讓程式開發人員使用第三代語言原有程序或函式呼叫來存取 Oracle 資料庫及控制全部 SQL 語句執行的階段。OCI 是一個標準資料庫存取的函式庫和動態連結庫形式檢索函式。

網路服務名稱

網路服務的名稱。

使用者名稱

連線到資料庫伺服器的使用者名稱。

密碼

連線到伺服器的密碼。

參見：

進階設定

相關主題：

Basic、SSH

PostgreSQL 一般設定

下列的指示引導你通過建立一個新連線的程序。要成功地建立一個新的連線到本機或遠端 PostgreSQL - 不管通過 SSL、SSH 或 HTTP，都要在相應的框中設定連線內容：連線名稱、主機名稱、埠號碼、初始資料庫、使用者名稱及密碼。

預設情況下，PostgreSQL 提供使用者名稱為「postgres」和密碼為空白。

連線名稱
一個最恰當地描述你的連線名稱。

主機名稱或 IP 位址
資料庫所在的主機名或伺服器的 IP 位址。

埠
連線到資料庫伺服器的 TCP/IP 埠。

初始資料庫
連線時，使用者連線到的初始資料庫。

使用者名稱
連線到資料庫伺服器的使用者名稱。

密碼
連線到伺服器的密碼。

如果你的互聯網服務供應商（ISP）不提供直接存取其伺服器，安全通道協定 SSH 或 HTTP 是另一種解決辦法。

注意：基於安全原因，本機遠端直接連線到 PostgreSQL 伺服器已停用。因此，你可能不能使用 Navicat Premium 或其他類似的 PostgreSQL 管理應用程式直接連線到遠端伺服器。如需詳細資料，請參考下一段的伺服器管理。

伺服器管理：
預設情況下，PostgreSQL 只允許由使用 TCP/IP 連線的本機電腦連線。其他的機器將無法連接，除非你修改在 *postgresql.conf* 檔案中的 *listen_addresses*，通過修改 *$PGDATA/pg_hba.conf* 檔案啟用基於主機的認證及重開伺服器。欲了解更多信息：用戶端身份驗證

參見：
進階設定

相關主題：

SSL、SSH、HTTP

SQLite 一般設定

下列的指示引導你通過建立一個新連線的程序。要成功地建立一個新的連線到本機或遠端 SQLite - 不管通過 HTTP，都要在相應的框中設定連線內容：連線名稱、類型及資料庫名稱。

連線名稱

一個最恰當地描述你的連線名稱。

資料庫檔案

指定初始資料庫檔案。如果 HTTP 通道 開啟，你需要輸入資料庫在你網絡伺服器的絕對檔案路徑。

參見：

進階設定

相關主題：

HTTP

SQL Server 一般設定

下列的指示引導你通過建立一個新連線的程序。要成功地建立一個新的連線到本機或遠端 SQL Server - 不管通過 SSH，都要在相應的框中設定連線內容：連線名稱、主機名稱及驗證類型。

連線名稱
一個最恰當地描述你的連線名稱。

主機名稱或 IP 位址
資料庫所在的主機名或伺服器的 IP 位址。

驗證
SQL Server 使用兩種方式來驗證連線到 SQL Server 資料庫：SQL Server Authentication 及 Windows Authentication。

SQL Server Authentication

SQL Server Authentication 使用登入記錄來驗證連線。每次使用者連線，都必須提供他們的使用者名稱及密碼。

使用者名稱
連線到資料庫伺服器的使用者名稱。

密碼
連線到伺服器的密碼。

Windows Authentication

當一個使用者通過一個視窗使用者帳號連線，SQL Server 會使用作業系統中 Windows 主要代符來驗證帳號名稱及密碼。這意味著使用者身份是由 Windows 確認。SQL Server 不會要求輸入密碼，並且不執行身份驗證。

初始資料庫
當建立連線時使用者初始連接的資料庫。

如果你的互聯網服務供應商（ISP）不提供直接存取其伺服器，安全通道協定 SSH 是另一種解決辦法。

參見：
進階設定

相關主題：
SSH

SSH 設定（只限於 MySQL、Oracle、PostgreSQL 及 SQL Server 並只支援 SSH2 通訊協定）

Secure SHell (SSH) 是一個通過網絡登入其他電腦的程式，在遠端伺服器執行命令，和從一台機器移動檔案到另一台。在不安全的網絡中，它提供兩台主機之間強大認證和安全加密的的通訊，被稱為 **SSH Port Forwarding (Tunneling)**。通常情況下，它是使用為一個 Telnet 的加密版本。

在一個 Telnet 階段作業，所有的通訊，包括使用者名稱和密碼，會用純文字傳輸，讓任何人都能監聽你的階段作業及竊取密碼或其他資訊。這種階段作業也容易受到階段作業劫持，一旦你驗證，惡意使用者就能接管這種階段作業。SSH 的目的是防止這種漏洞，並允許你在不影響安全性的情況下存取遠端伺服器的 shell。

- SSH 通道的好處。

為確保進來的連線請求是由你發出，SSH 能夠使用密碼，或公開及私密金鑰對（也稱為公開金鑰）驗證機制。

- 密碼驗證。
- 公開金鑰驗證。

注意：請確保在 Linux 伺服器的參數 -「AllowTcpForwarding」設定值為「yes」，否則，會停用 SSH port forwarding。要尋找路徑：/etc/ssh/sshd_config。在預設情況下，SSH port forwarding 應該已啟用。請仔細檢查該值的設定。

** 即使伺服器支援 SSH 通道，然而，如果 port forwarding 被停用，Navicat 就無法通過 SSH 埠 22 連線。

參見：
進階設定

相關主題：
SSL

SSH 通道的好處

SSH 有一個極好功能叫做 SSH Port Forwarding，有時也被稱為 SSH 通道，它允許你建立一個安全的階段作業，然後通過它打開隨心所欲的 TCP 連線。通道可以隨時建立，幾乎不需要任何努力及編製程式，這令到它們非常有吸引力。在無數不同的方式，SSH Port Forwarding 可以用於安全通訊。

許多提供伺服器託管的託管公司會封鎖託管公司網絡以外存取伺服器，以及只授予存取給本機（localhost）使用者連線。

使用 SSH 的多個好處：

- 當伺服器埠被封鎖時，連線到使用了防火牆的伺服器。
- 自動驗證使用者，沒有發送純文字的密碼，以防止竊取密碼。
- 多個強大的認證方法，防止安全威脅如欺騙的身份。
- 安全和快速的加密和壓縮資料。
- 安全檔案傳輸。

相關主題：
密碼驗證、公開金鑰驗證

密碼驗證

使用這種模式下，SSH 是幾乎和 telnet 程式相同。當你進行連接時，它會詢問你的密碼。輸入它後，你會登入或被拒絕。你的密碼會首先加密並由網絡傳送，然後在遠端主機解密。這是鼓勵大多數使用者使用的模式，因為它不需要額外的設置或配置。

下列指示將引導你完成設定使用密碼驗證的 SSH 連線。要成功地建立一個 SSH 連線，在相應的框中設定 SSH 連線內容：主機名稱或 IP 位址、埠號碼、使用者名稱、驗證方法及密碼。

1. 按一下 或選擇檔案 -> **新增連線** 來設定連線內容。
2. 選擇 SSH 索引標籤並啟用 **使用 SSH 通道**。
3. 填寫所需的資訊：

主機名稱或 IP 位址
SSH 伺服器的主機。

埠
SSH 伺服器的埠，預設情況下它是 22。

使用者名稱
在 Linux 機器的使用者。（這是一個 Linux 使用者。它不是一個資料庫伺服器使用者。）

驗證方法
在密碼驗證和公開金鑰驗證間選擇。

密碼
這是一個 Linux 使用者的密碼。

4. 在一般設定頁面（MySQL、Oracle、PostgreSQL 或 SQL Server）的 Navicat 主機名是由你的資料庫託管公司提供，應設定和 SSH 伺服器相對。

參見：

進階設定

相關主題：

公開金鑰驗證

公開金鑰驗證

公開金鑰驗證是基於電子簽名的使用並提供最佳的安全驗證。

要使用公開金鑰驗證，有四項需要：

- 在你連線到的遠端伺服器必須有你的公開金鑰。
- 在你的本機用戶端必須有你的私密金鑰。
- 遠端伺服器必須配置為允許你使用你的公開金鑰登入。
- 本機用戶端必須配置為當你登入遠端伺服器時使用你的私密金鑰。

下列指示將引導你完成配置使用公開金鑰驗證的 SSH 連線。要成功地建立一個 SSH 連線，在相應的框中設定 SSH 連線內容：主機名稱或 IP 位址、埠號碼、使用者名稱、驗證方法、私密金鑰及密碼短語。

1. 按一下 或選擇檔案 -> **新增連線** 來設定連線內容。
2. 選擇 SSH 索引標籤並啟用 **使用 SSH 通道**。
3. 填寫所需的資訊：

主機名稱或 IP 位址
SSH 伺服器的主機。

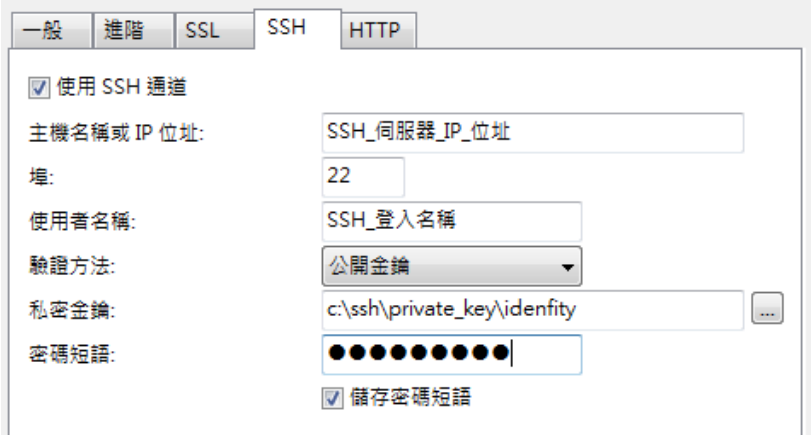
埠
SSH 伺服器的埠，預設情況下它是 22。

使用者名稱
在 Linux 機器的使用者。（這是一個 Linux 使用者。它不是一個資料庫伺服器使用者。）

驗證方法
在密碼驗證和公開金鑰驗證間選擇。

私密金鑰
這是和你的公開金鑰一起使用。私密金鑰應該只有你可讀。

密碼短語
密碼短語和密碼完全一樣，除了它應用於你產生的金鑰而不是帳號。密碼短語的長度是 1024 字元以下。



4. 在一般設定頁面（MySQL、Oracle、PostgreSQL 或 SQL Server）的 Navicat 主機名是由你的資料庫託管公司提供，應設定和 SSH 伺服器相對。

參見：

進階設定

相關主題：

密碼驗證

HTTP 設定（只限於 MySQL、PostgreSQL 及 SQLite）

HTTP 通道是一種連線到伺服器的方法，使用與網絡伺服器相同的通訊協定（http://）和相同的埠（埠 80）。這是當你的互聯網服務供應商不允許直接連線，但允許建立 HTTP 連線時使用。

建立 HTTP 連線的步驟：

1. 上傳通道指令碼

   若要使用此連線方法，首先需要做的就是上傳通道指令碼到你的伺服器位於的網絡伺服器。

   注意：**ntunnel_mysql.php**、**ntunnel_pgsql.php** 或 **ntunnel_sqlite.php** 是在 Navicat 的安裝資料夾中。

2. 設定 **HTTP** 通道

   下列指示將引導你完成設定 HTTP 連線程序。

   i. 按一下 或選擇檔案 -> 新增連線 來設定連線內容。

   ii. 選擇 HTTP 索引標籤並啟用 使用 **HTTP** 通道。

   iii. 輸入通道指令碼的網址，例如：*http://www.navicat.com/ntunnel_mysql.php*。

   iv. 如果你的伺服器安裝了 ModSecurity，你可以勾選 用 **base64** 編碼傳出查詢選項。

   v. 如果通道指令碼是在一個密碼保護的伺服器或你要通過代理伺服器來存取網際網路，你可以在 驗證 或 代理伺服器 索引標籤提供所需的詳細驗證資料。

   vi. 在一般設定頁面的 Navicat 主機名是由你的資料庫託管公司提供，應設定和 HTTP 伺服器相對。

注意：HTTP 通道和 SSH 通道不能同時運作。當你選擇了 HTTP 通道，SSH 通道停用，反之亦然。

SSL 設定（只限於 MySQL 及 PostgreSQL）

Secure Sockets Layer(SSL) 是一個通過網際網路傳輸私人文件的協定。為了安全連線到 MySQL 或 PostgreSQL 伺服器，首先你需要做的是安裝 OpenSSL 函式庫和下載 MySQL 或 PostgreSQL 資料庫來源。

建立 MySQL 或 PostgreSQL 及 Navicat 的 SSL 連線步驟：

1. 安裝 OpenSSL 及 MySQL 或 PostgreSQL。
2. 設定 MySQL 或 PostgreSQL 的 SSL 憑證。
3. 設定 Navicat 的用戶端憑證。

注意：PostgreSQL 8.4 或以上版本支援。

參見：
進階設定

相關主題：
MySQL 一般設定、PostgreSQL 一般設定、SSH

安裝 OpenSSL 及 MySQL 或 PostgreSQL

安裝 OpenSSL

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

安裝 MySQL

1. 下載 MySQL - http://www.mysql.com
2. Linux 命令：[./configure --with -vio --with -openssl]
3. Linux 命令：[make]
4. Linux 命令：[make install]

注意：請確保 MySQL 伺服器支援 OpenSSL 使用查詢語句：[SHOW VARIABLES LIKE 'have_openssl';] - 回傳值 = YES

安裝 PostgreSQL

1. 下載 PostgreSQL - http://www.postgresql.org
2. Linux 命令：[./configure --with-openssl]
3. Linux 命令：[gmake]
4. Linux 命令：[gmake install]

注意：請確保 PostgreSQL 伺服器支援 OpenSSL 使用查詢語句：[SHOW ssl;] - 回傳值 = ON

參見
步驟 2：設定 MySQL 或 PostgreSQL 的 SSL 憑證

設定 MySQL 或 PostgreSQL 的 SSL 憑證

要建立伺服器或用戶端憑證，用 root 登入 Linux 伺服器並使用 Shell 命令如下：

MySQL

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. 產生 Certificate Authority (CA)

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

   注意：如果「PEM」是需要的，請通過以下步驟輸入另外的「PEM pass」。

7. 建立伺服器的請求和鑰匙

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

8. 從鑰匙移除密碼短語（非必須的）

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

9. 簽署伺服器憑證

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

10. 建立用戶端的請求和鑰匙

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

11. 從鑰匙移除密碼短語（非必須的）

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

12. 簽署用戶端憑證

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

13. 建立一個 **my.cnf** 檔案來測試憑證。 儲存於 /etc 或 MySQL 資料目錄（通常在來源安裝的 /usr/local/var）

my.cnf 範例內容：

```
[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. 要開始 MySQL 程式類型：

/usr/local/libexec/mysqld -u mysql &

or

/usr/local/sbin/mysqld -u &

PostgreSQL

1. 要建立一個伺服器的快速自簽署憑證，使用以下的 OpenSSL 命令：

openssl req -new -text -out server.reqm

2. 填寫 openssl 要求的資訊。請確認你輸入的本機主機名稱為「Common Name」；要求的密碼可以留空。程式會產生一個密碼保護的鑰匙；它不會接受一個少於 4 個字元長度的密碼短語。要移除密碼短語（你必須，如果你想要自動啟動的伺服器），執行命令：

```
openssl rsa -in privkey.pem -out server.key
rm privkey.pem
```

3. 輸入舊的密碼短語來解開現有的鑰匙。現在做的：

```
openssl req -x509 -in server.req -text -key server.key -out server.crt
```

4. 將憑證變為自簽署憑證和複製鑰匙及憑證到伺服器將在那裏尋找他們。最後做：

```
chmod og-rwx server.key
```

相關主題：

步驟 3：設定 Navicat 的用戶端憑證

設定 **Navicat** 的用戶端憑證

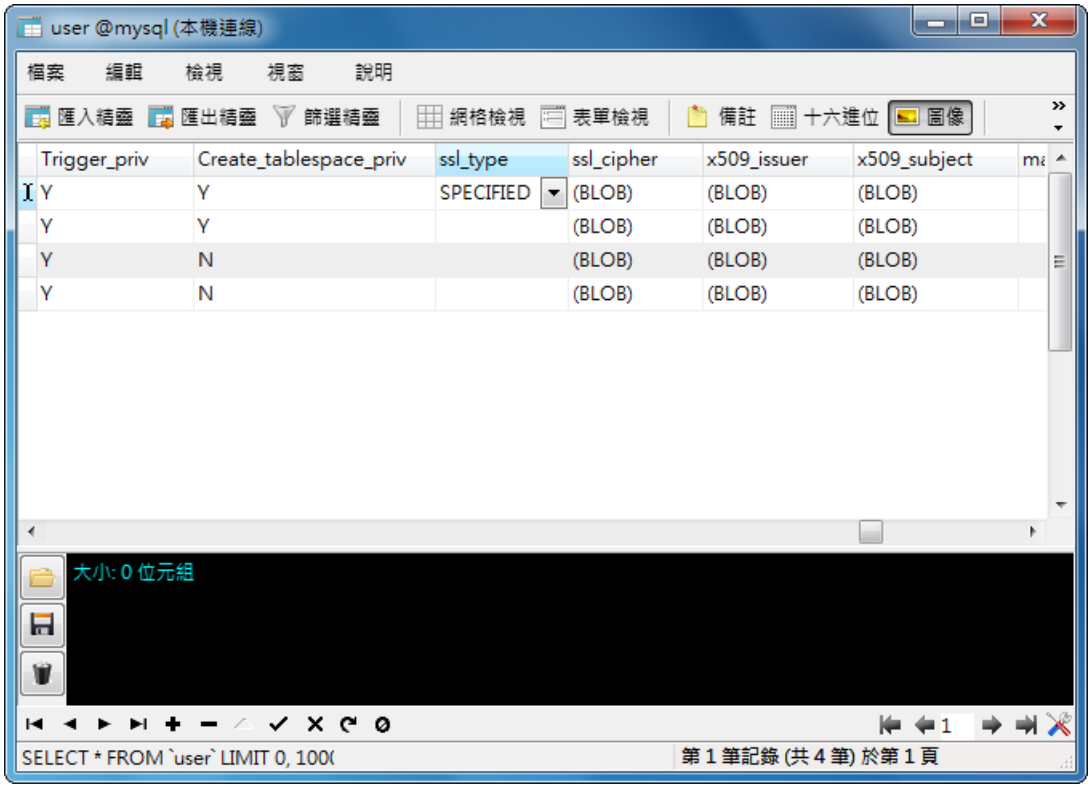
下列指示將引導你完成配置 Navicat 和 MySQL 或 PostgreSQL 伺服器之間使用 SSL 的連線程序。要成功地建立一個 SSL 連線，請填寫 步驟 1：安裝 OpenSSL 及 MySQL 或 PostgreSQL 及 步驟 2：設定 MySQL 或 PostgreSQL 的 SSL 憑證，並在相應的框中設定連線內容。

MySQL

1. 按一下 或選擇檔案 -> **新增連線** 來設定連線內容。
2. 選擇 SSL 索引標籤並啟用 **使用 SSL**。
3. 提供驗證資料，填寫所需的資訊：

用戶端鑰匙、**用戶端憑證** 及 **CA 憑證** 通常儲存在你的伺服器 - /usr/local/openssl。請從遠端伺服器複製他們到本機電腦。**指定密碼檢索表**（非必須的）只需要在 **ssl_type** 欄位設定為「**SPECIFIED**」- [ssl_type 可以在名為「mysql」的系統資料庫中找到 ->「user」資料表]。指定密碼檢索表的範例是「EDH-RSA-DES-CBC3-SHA」，可以填寫在上面顯示的連線內容或下面的「mysql」資料庫 -> 「user」資料表 -> 「ssl_cipher」blob 欄位。

注意：你可以儲存你的指定密碼檢索表在一個文字檔案，以便載入到到「ssl_cipher」blob 欄位。



PostgreSQL

1. >按一下 或選擇檔案 -> **新增連線** 來設定連線內容。
2. 選擇 SSL 索引標籤並啟用 **使用 SSL**。
3. 選擇 **SSL 模式**。
   require - 只嘗試一個 SSL 連線。
   verify-ca - 只嘗試一個 SSL 連線，並驗證伺服器憑證是由受信任的 CA 發行。
   verify-full - 只嘗試一個 SSL 連線，驗證伺服器憑證是由受信任的 CA 發行及伺服器主機名稱符合憑證內的。
4. 要提供驗證資料，啟用 **使用驗證** 並填寫所需的資訊：

   用戶端鑰匙、**用戶端憑證** 及 **CA 憑證** 通常儲存在你的伺服器 - /usr/local/openssl。請從遠端伺服器複製他們到本機電腦。

   憑證撤銷清單 指定 SSL 憑證撤消清單（certificate revocation list - CRL）的檔案路徑。
   對於 PostgreSQL 伺服器，OpenSSL 支援範圍廣泛不同強度的密文和驗證演算法。當密文清單可以在 OpenSSL 設定檔案中指定，你可以透過修改 postgresql.conf 中的 ssl_ciphers 指定資料庫要使用的密文。

參見：
進階設定

進階設定

根據你的需要自訂連線選項。詳細介紹如下：

報表位置

當建立一個新的連線，Navicat Report Viewer 將建立一個子檔案夾在報表位置內。你該將你的所有報表檔案 (.rtm) 放在此子檔案夾。

提示：可以變更到任何你想要的位置。

MySQL

編碼

當使用 MySQL 字元集沒有使用，選擇一個代碼頁來和 MySQL 伺服器溝通。

☑保持連線間隔 (秒)

此選項允許你通過 ping 來保持連線。你可以在編輯欄位設定 ping 與 ping 之間的間隔。

☑使用 MySQL 字元集

如果使用 MySQL 4.1 或以上的版本，應該啟用此選項。

☑使用壓縮

此選項讓你使用壓縮通訊協定。使用它當用戶端和伺服器支援 zlib 壓縮，以及用戶端需要壓縮。

☑自動連接

勾選了這個選項，當程式啟動時，Navicat Report Viewer 會自動開啟連線和顯示登記的資料庫。

☑使用名稱管道、通訊端

勾選了這個選項，Navicat Report Viewer 使用本機連線的通訊端檔案。

Oracle

角色

指示資料庫使用者用 **Default**、**SYSOPER** 或 **SYSDBA** 系統權限連線。

☑作業系統驗證

勾選了這個選項，Oracle 資料庫使用 Windows 使用者登入證書來驗證資料庫使用者。

☑自動連接

勾選了這個選項，當程式啟動時，Navicat Report Viewer 會自動開啟連線和顯示登記的資料庫。

PostgreSQL

☑ **保持連線間隔（秒）**

此選項允許你通過 ping 來保持連線。你可以在編輯欄位設定 ping 與 ping 之間的間隔。

☑ **自動連接**

勾選了這個選項，當程式啟動時，Navicat Report Viewer 會自動開啟連線和顯示登記的資料庫。

SQLite

☑ **自動連接**

勾選了這個選項，當程式啟動時，Navicat Report Viewer 會自動開啟連線和顯示登記的資料庫。

☑ **已加密**

當連線到已加密的 SQLite 資料庫，啟用這個選項並提供 **密碼**。

已附加的資料庫

在連線中附加或卸除資料庫。

SQL Server

☑ **使用加密**

這選項讓你使用加密。

☑ **自動連接**

勾選了這個選項，當程式啟動時，Navicat Report Viewer 會自動開啟連線和顯示登記的資料庫。