Securing the SQL Anywhere Monitor With Transport Layer Security (TLS)

A whitepaper from Sybase iAnywhere

Date: May 2009

This whitepaper was written in the context of SQL Anywhere 11.
However, its content may be applicable to future releases.

CONTENTS

INTRODUCTION

The SQL Anywhere Monitor, also referred to as the Monitor, is a web-browser-based administration tool that provides you with information about the health and availability of SQL Anywhere databases and MobiLink servers.

This whitepaper explains how to secure communication between the SQL Anywhere Monitor server and the browser when using the Monitor. The SQL Anywhere Monitor web server supports HTTPS connections using SSL version 3.0 and TLS version 1.0. It is possible to operate with a verified certificate from a certificate authority (CA) or a self-signed certificate. To set up transport-layer security for the SQL Anywhere Monitor, you must:

1. Obtain digital certificates
2. Start the web server with transport-layer security
3. Configure web clients

These steps are explained in detail in this whitepaper and require SQL Anywhere 11.0.1.2052 or later. You must update all shortcuts and/or bookmark locations to reflect the new URL.

OBTAIN DIGITAL CERTIFICATES

If you have a digital certificate from a certificate authority (CA), you can skip to the next section. VeriSign and Thawte offer free trial certificates or you can create a self-signed certificate with the Certificate Creation utility (createcert).

CREATE A SELF-SIGNED CERTIFICATE

The Certificate Creation utility (createcert) that is bundled with SQL Anywhere 11 generates a self-signed certificate. This tool is located in the `Bin32` directory of the install.

Here is an example of the process. For more information about using the createcert utility, see the SQL Anywhere documentation:

```
C:\cert>createcert
SQL Anywhere X.509 Certificate Generator Version 11.0.1.2197
Choose encryption type ((R)SA or (E)CC): R
Enter RSA key length (512-16384): 1024
Generating key pair...
Country Code: CA
State/Province: ON
Locality: Waterloo
Organization: Sybase
Organizational Unit: PM
Common Name: localhost
Enter file path of signer's certificate:
Certificate will be a self-signed root
Serial number [generate GUID]:
Generated serial number: a1424ed9bc7844fd8b4c9fbeae820af1
Certificate valid for how many years (1-100): 10
Certificate Authority (Y/N) [N]: Y
1.  Digital Signature
2.  Nonrepudiation
3.  Key Encipherment
4.  Data Encipherment
5.  Key Agreement
6.  Certificate Signing
7.  CRL Signing
8.  Encipher Only
9.  Decipher Only
Key Usage [6,7]:
Enter file path to save certificate: cert.cer
Enter file path to save private key: key.cer
Enter password to protect private key: pwd
Enter file path to save identity: identity.pem
```

Note: The Common Name should be the host name of your server.

START THE WEB SERVER WITH TRANSPORT-LAYER SECURITY

To start the web server with TLS, you need to alter the server startup string of the SQL Anywhere Monitor. This whitepaper uses the self-signed certificates generated by the createcert utility. However, the process is the same with a verified certificate. The most important thing is to ensure that the port number, location of the identity file, and identity password are correct. They are declared in the HTTPS protocol of the –xs database server option. For more information on the –xs server option, see the SQL Anywhere documentation.

ALTERING THE MONITOR'S STARTUP STRING (WINDOWS)

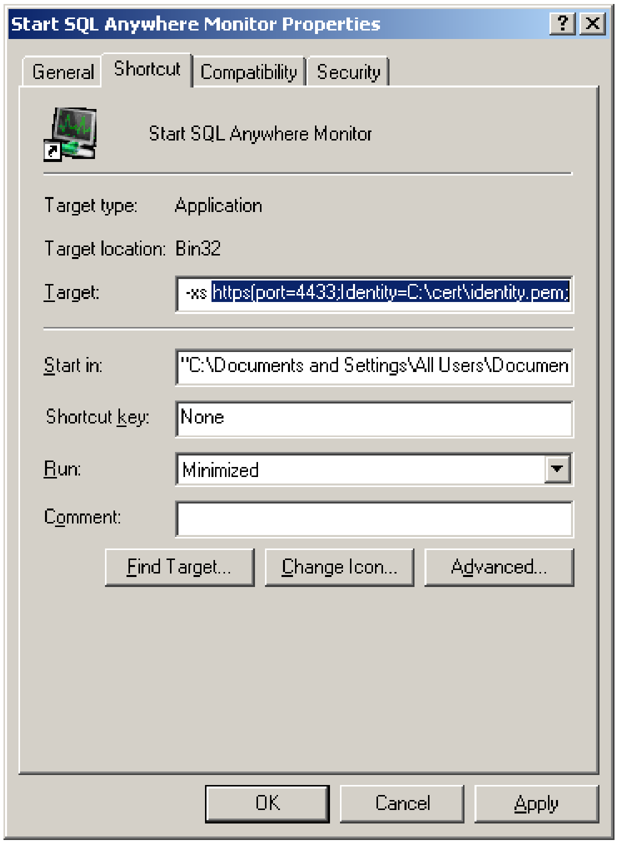
There are two possible ways to run the SQL Anywhere Monitor. You may choose to run it on the same computer where SQL Anywhere is running for development purposes, or as a standalone service on a separate computer. Follow the appropriate steps according to the installation and operating system of your setup.

TO ALTER THE STARTUP STRING ON THE SAME COMPUTER THAT SQL ANYWHERE IS RUNNING ON

1. From the Start menu, choose All Programs > SQL Anywhere 11 > SQL Anywhere Monitor > Start SQL Anywhere Monitor.
2. Right-click the icon and choose Properties.
   The default location for the shortcut is:

   ```
   "C:\Documents and Settings\All Users\Start Menu\Programs\SQL Anywhere
   11\SQL Anywhere Monitor\Start SQL Anywhere Monitor"
   ```

   You should see this prompt:



Start SQL Anywhere Monitor

Target type: Application

Target location: Bin32

Target: -xs https(port=4433;Identity=C:\cert\identity.pem;

Start in: "C:\Documents and Settings\All Users\Documen

Shortcut key: None

Run: Minimized

Comment:

3. Edit the Target value of the shortcut:

   From:

   ```
   "C:\Program Files\SQL Anywhere 11\Bin32\dbsrv11.exe" -sb 0 -xd -qi -n
   SAMonitor_COMPUTERNAME -xs http(port=4950) "C:\Documents and
   Settings\All Users\Documents\SQL Anywhere 11\Monitor\\samonitor.db"
   ```

   To:

   ```
   "C:\Program Files\SQL Anywhere 11\Bin32\dbsrv11.exe" -sb 0 -xd -qi -n
   SAMonitor_COMPUTERNAME -xs
   https(port=4433;Identity=”C:\cert\identity.pem”;Identity_Password=pwd
   ) "C:\Documents and Settings\All Users\Documents\SQL Anywhere
   11\Monitor\\samonitor.db"
   ```

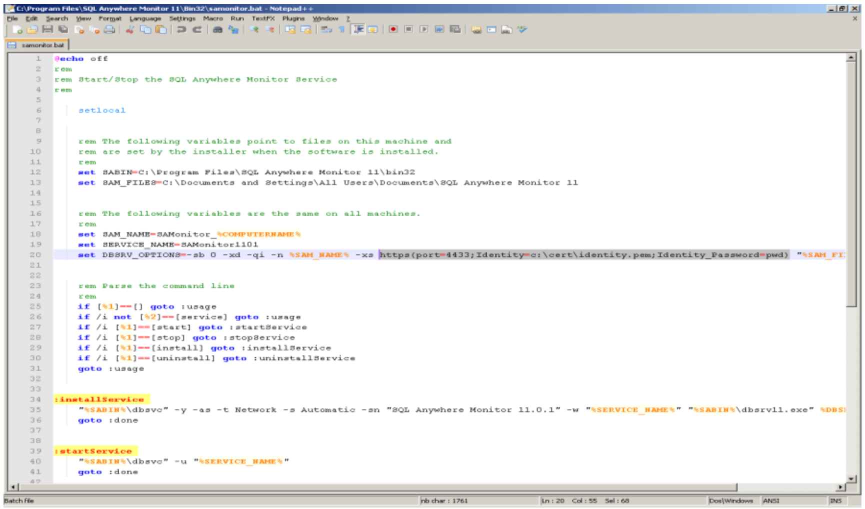
   - port is the port you want to start the server on.
   - Identity is the path and file name of the server identity.
   - Identity_Password is the password for the server private key. You specify this password when you create the server certificate.

4. Click Apply.
5. Click OK.

   Now, when you click Start SQL Anywhere Monitor, it starts using TLS.

TO ALTER THE STARTUP STRING ON A SEPARATE COMPUTER

1. Open the file: `C:\Program Files\SQL Anywhere Monitor 11\Bin32\samonitor.bat` in a text editor.
   This is the script that creates the SQL Anywhere Monitor service:



C:\Program Files\SQL Anywhere Monitor 11\Bin32\samonitor.bat - Notepad++
File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ?
samonitor.bat
Batch file
nb char : 1761
Ln : 20 Col : 55 Sel : 68
Dos\Windows
ANSI
INS

```
@echo off
rem
rem Start/Stop the SQL Anywhere Monitor Service
rem

    setlocal


    rem The following variables point to files on this machine and
    rem are set by the installer when the software is installed.
    rem
    set SABIN=C:\Program Files\SQL Anywhere Monitor 11\bin32
    set SAM_FILES=C:\Documents and Settings\All Users\Documents\SQL Anywhere Monitor 11


    rem The following variables are the same on all machines.
    rem
    set SAM_NAME=SAMonitor_%COMPUTERNAME%
    set SERVICE_NAME=SAMonitor1101
    set DBSRV_OPTIONS=-sb 0 -xd -qi -n %SAM_NAME% -xs https(port=4433;Identity=c:\cert\identity.pem;Identity_Password=pwd) "%SAM_FI


    rem Parse the command line
    rem
    if [%1]==[] goto :usage
    if /i not [%2]==[service] goto :usage
    if /i [%1]==[start] goto :startService
    if /i [%1]==[stop] goto :stopService
    if /i [%1]==[install] goto :installService
    if /i [%1]==[uninstall] goto :uninstallService
    goto :usage


:installService
    "%SABIN%\dbsvc" -y -as -t Network -s Automatic -sn "SQL Anywhere Monitor 11.0.1" -w "%SERVICE_NAME%" "%SABIN%\dbsrv11.exe" %DBS


:startService
    "%SABIN%\dbsvc" -u "%SERVICE_NAME%"
    goto :done
```

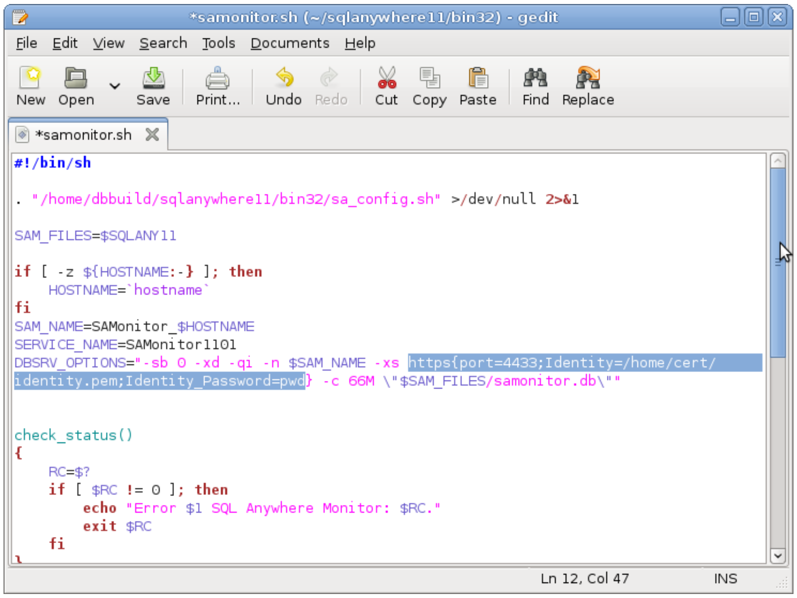
2. Edit the server options (DBSRV_OPTIONS) string:
   From:
   `set DBSRV_OPTIONS=-sb 0 -xd -qi -n %SAM_NAME% -xs http(port=4950) "%SAM_FILES%\samonitor.db"`
   To:
   `set DBSRV_OPTIONS=-sb 0 -xd -qi -n %SAM_NAME% -xs https(port=4433;Identity=”c:\cert\identity.pem”;Identity_Password=pwd) "%SAM_FILES%\samonitor.db"`
   - port is the port you’d like to start the server on.
   - Identity is the path and file name of the server identity.
   - Identity_Password is the password for the server private key. You specify this password when you create the server certificate.
3. Save the file.
4. At a command prompt, navigate to the `Bin32` folder of the SQL Anywhere Monitor directory where `samonitor.bat` resides. By default, this is `C:\Program Files\SQL Anywhere Monitor 11\Bin32`.
5. Uninstall the service.
   From the command prompt, run: `samonitor.bat uninstall service`
6. Install the service with the new parameters.
   From the command prompt, run: `samonitor.bat install service`
7. Start the service.
   From the command prompt, run: `samonitor.bat start service`

ALTERING THE MONITOR'S STARTUP STRING (LINUX)

There are two possible ways to run the SQL Anywhere Monitor. You may choose to run it on the same computer where SQL Anywhere is running for development purposes, or as a standalone service on a separate computer. Follow the appropriate steps according to the installation and operating system of your setup.

TO ALTER THE STARTUP STRING ON THE SAME COMPUTER THAT SQL ANYWHERE IS RUNNING ON

1. Open the file: /***PATH-TO-SQL-ANYHWERE_11***/`bin32/samonitor.sh` in a text editor.
   By default: "`/opt/sqlanywhere11/bin32/samonitor.sh`"
   This is the script that starts the SQL Anywhere server:


*samonitor.sh (~/sqlanywhere11/bin32) - gedit
File Edit View Search Tools Documents Help
New
Open
Save
Print...
Undo
Redo
Cut
Copy
Paste
Find
Replace
*samonitor.sh
Ln 12, Col 47
INS

```
#!/bin/sh

. "/home/dbbuild/sqlanywhere11/bin32/sa_config.sh" >/dev/null 2>&1

SAM_FILES=$SQLANY11

if [ -z ${HOSTNAME:-} ]; then
    HOSTNAME=`hostname`
fi
SAM_NAME=SAMonitor_$HOSTNAME
SERVICE_NAME=SAMonitor1101
DBSRV_OPTIONS="-sb 0 -xd -qi -n $SAM_NAME -xs https{port=4433;Identity=/home/cert/
identity.pem;Identity_Password=pwd} -c 66M \"$SAM_FILES/samonitor.db\""


check_status()
{
    RC=$?
    if [ $RC != 0 ]; then
        echo "Error $1 SQL Anywhere Monitor: $RC."
        exit $RC
    fi
}
```

2. Edit the server options (DBSRV_OPTIONS) string:
   From:

```
DBSRV_OPTIONS="-sb 0 -xd -qi -n $SAM_NAME -xs http{port=4950} -c 66M
\"$SAM_FILES/samonitor.db\""
```

   To:

```
DBSRV_OPTIONS="-sb 0 -xd -qi -n $SAM_NAME -xs
https{port=4433;Identity=/home/cert/identity.pem;Identity_Password=pw
d} -c 66M \"$SAM_FILES/samonitor.db\""
```

   - port is the port you'd like to start the server on.
   - Identity is the path and file name of the server identity.
   - Identity_Password is the password for the server private key. You specify this password when you create the server certificate.

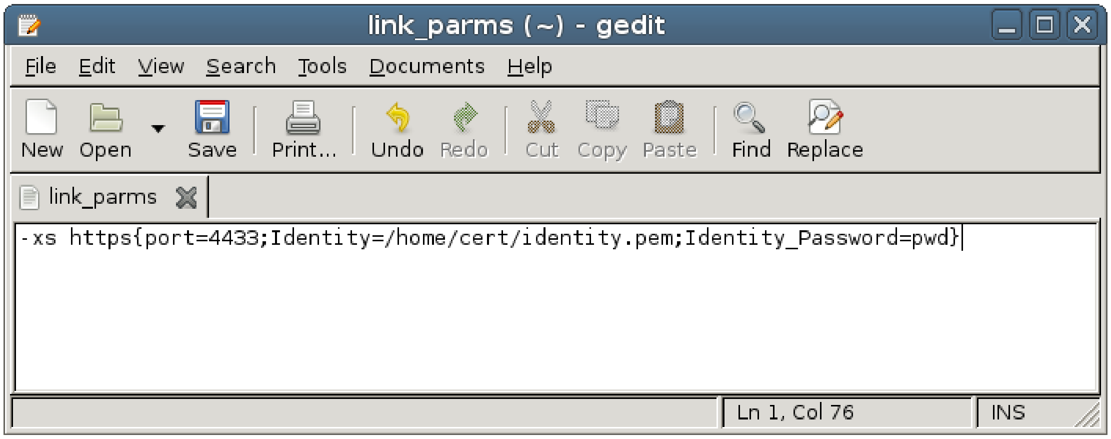
3. Save the file.
   Now, when you click Start SQL Anywhere Monitor, it starts using TLS.

TO ALTER THE STARTUP STRING ON A SEPARATE COMPUTER

1. Create a configuration file to store the link parameters for the –xs option. This is a text file that can be saved in any location on your computer. This whitepaper uses `/home/link_parms`. For more information on the configuration file usage, see the SQL Anywhere documentation.
2. Edit the contents of your configuration file (“`/home/link_parms`”) to read:

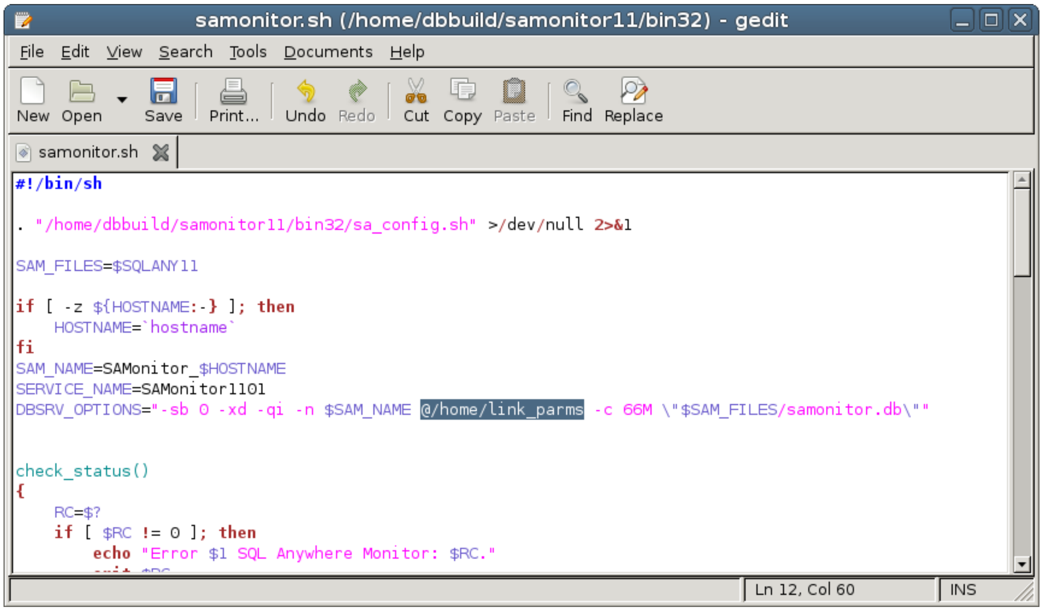
```
-xs https{port=4433;
Identity=/home/cert/identity.pem;Identity_Password=pwd}
```

- port is the port you'd like to start the server on.
- Identity is the path and file name of the server identity.
- Identity_Password is the password for the server private key. You specify this password when you create the server certificate.


link_parms (~) - gedit
File Edit View Search Tools Documents Help
New Open Save Print... Undo Redo Cut Copy Paste Find Replace
link_parms
Ln 1, Col 76
INS

```
-xs https{port=4433;Identity=/home/cert/identity.pem;Identity_Password=pwd}
```

3. Save the file.
4. Open the file /***PATH-TO-SQL-ANYHWERE-MONITOR-11***/`bin32/samonitor.sh` in a text editor.
   By default this file is located in `/opt/samonitor11/bin32/samonitor.sh`.
5. This is the script that creates the SQL Anywhere Monitor service.


samonitor.sh (/home/dbbuild/samonitor11/bin32) - gedit
File Edit View Search Tools Documents Help
New Open Save Print... Undo Redo Cut Copy Paste Find Replace
samonitor.sh
Ln 12, Col 60
INS

```
#!/bin/sh

. "/home/dbbuild/samonitor11/bin32/sa_config.sh" >/dev/null 2>&1

SAM_FILES=$SQLANY11

if [ -z ${HOSTNAME:-} ]; then
    HOSTNAME=`hostname`
fi
SAM_NAME=SAMonitor_$HOSTNAME
SERVICE_NAME=SAMonitor1101
DBSRV_OPTIONS="-sb 0 -xd -qi -n $SAM_NAME @/home/link_parms -c 66M \"$SAM_FILES/samonitor.db\""


check_status()
{
    RC=$?
    if [ $RC != 0 ]; then
        echo "Error $1 SQL Anywhere Monitor: $RC."
```

6. Edit the server options (DBSRV_OPTIONS) string:

   From:

   `DBSRV_OPTIONS="-sb 0 -xd -qi -n $SAM_NAME -xs http{port=4950} -c 66M \"$SAM_FILES/samonitor.db\""`

   To:

   `DBSRV_OPTIONS="-sb 0 -xd -qi -n $SAM_NAME @/home/link_parms -c 66M \"$SAM_FILES/samonitor.db\""`

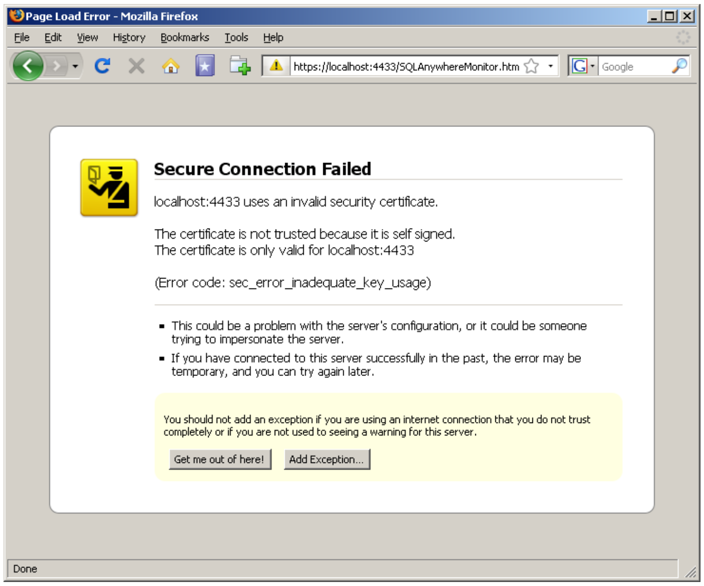
7. Enter the path to the configuration file that you created in Step 1 prefixed by @.
8. Save the file.
9. Open the terminal and navigate to the bin32 folder of the SQL Anywhere Monitor directory where `samonitor.sh` resides. By default the file is located in `/opt/samonitor11/bin32`.
10. Uninstall the service. From the terminal, run: `./samonitor.sh uninstall`
11. Install the service with the new parameters. From the terminal, run: `./samonitor.sh install`
12. Start the service. From terminal, run: `./samonitor.sh start`

CONFIGURE WEB CLIENTS

If you are using a self-signed certificate, depending on your browser, you will see one of the following security prompts. If you are using a certificate from a verified CA, it is likely that your CA is trusted by the browser and you will not see these prompts.

MOZILLA FIREFOX

You will see this warning:



Secure Connection Failed

localhost:4433 uses an invalid security certificate.

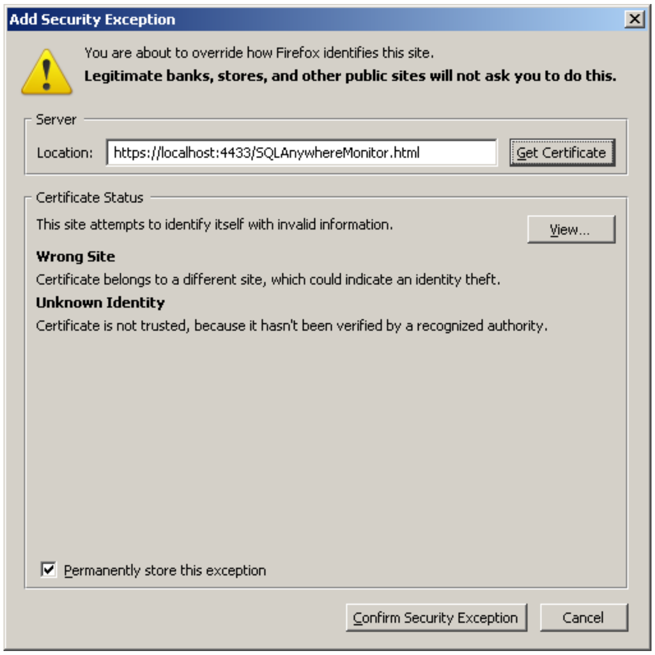
The certificate is not trusted because it is self signed.
The certificate is only valid for localhost:4433

(Error code: sec_error_inadequate_key_usage)

- This could be a problem with the server's configuration, or it could be someone trying to impersonate the server.
- If you have connected to this server successfully in the past, the error may be temporary, and you can try again later.

You should not add an exception if you are using an internet connection that you do not trust completely or if you are not used to seeing a warning for this server.

1. Click Add Exception.



You are about to override how Firefox identifies this site.
Legitimate banks, stores, and other public sites will not ask you to do this.

Location: https://localhost:4433/SQLAnywhereMonitor.html

This site attempts to identify itself with invalid information.

Wrong Site
Certificate belongs to a different site, which could indicate an identity theft.

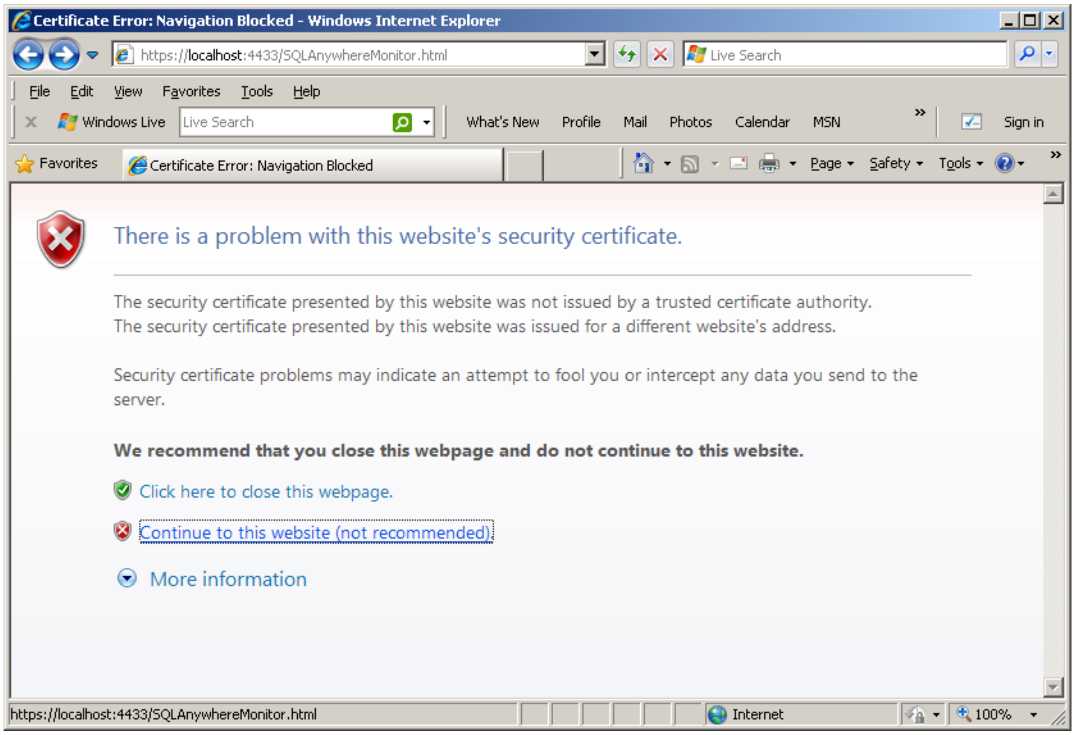
Unknown Identity
Certificate is not trusted, because it hasn't been verified by a recognized authority.

☑ Permanently store this exception

2. Click Get Certificate.
3. Click Confirm Security Exception.

MICROSOFT INTERNET EXPLORER

You will see this warning:



Certificate Error: Navigation Blocked - Windows Internet Explorer
https://localhost:4433/SQLAnywhereMonitor.html
Live Search
File Edit View Favorites Tools Help
Windows Live Live Search What's New Profile Mail Photos Calendar MSN Sign in
Favorites Certificate Error: Navigation Blocked
Page Safety Tools
https://localhost:4433/SQLAnywhereMonitor.html
Internet
100%

There is a problem with this website's security certificate.

The security certificate presented by this website was not issued by a trusted certificate authority.
The security certificate presented by this website was issued for a different website's address.

Security certificate problems may indicate an attempt to fool you or intercept any data you send to the server.

We recommend that you close this webpage and do not continue to this website.

Click here to close this webpage.

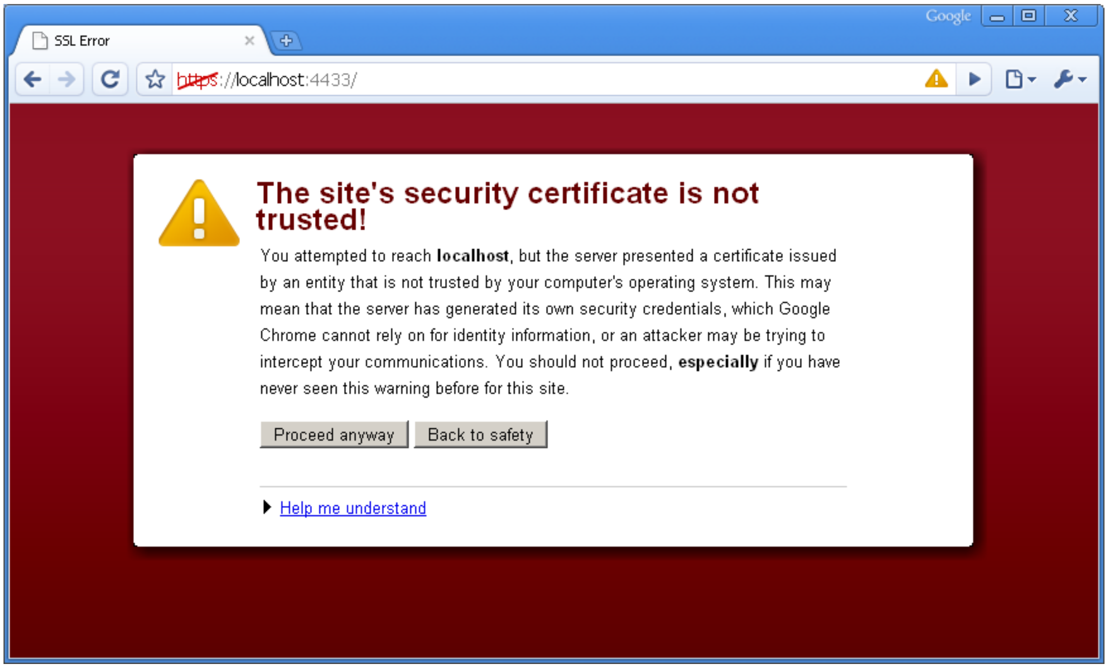
Continue to this website (not recommended).

More information

1. Click Continue To This Website (Not Recommended).

GOOGLE CHROME

You will see this warning:



The site's security certificate is not trusted!

You attempted to reach **localhost**, but the server presented a certificate issued by an entity that is not trusted by your computer's operating system. This may mean that the server has generated its own security credentials, which Google Chrome cannot rely on for identity information, or an attacker may be trying to intercept your communications. You should not proceed, **especially** if you have never seen this warning before for this site.

Proceed anyway   Back to safety

Help me understand

1. Click Proceed Anyway.

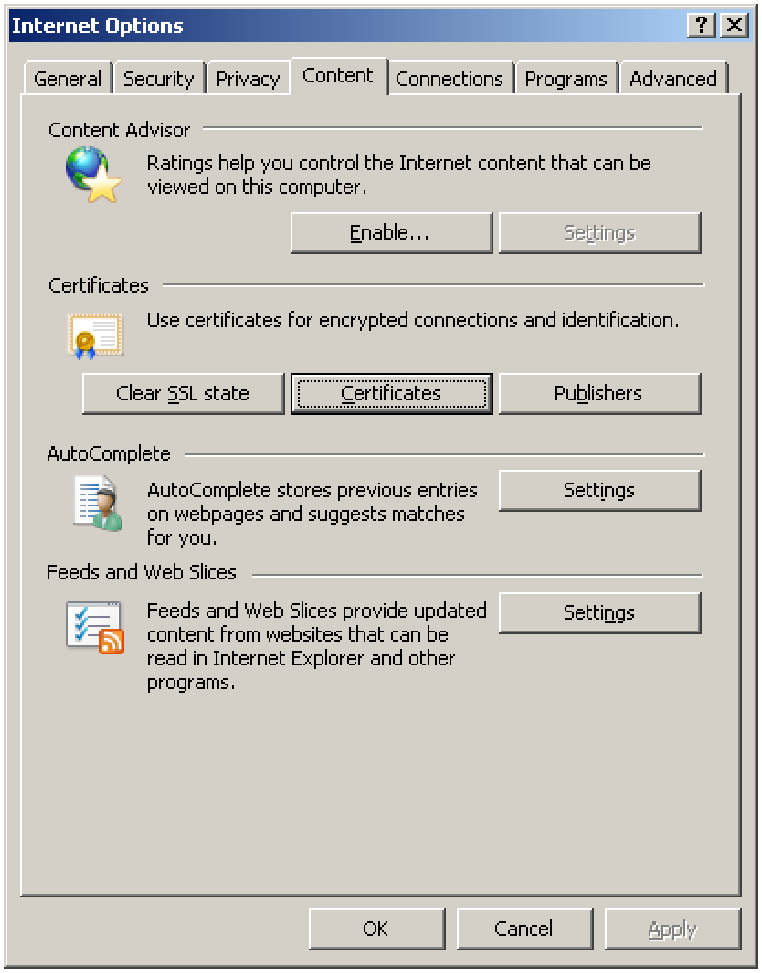
PERMANENT SETTINGS FOR INTERNET EXPLORER AND CHROME

To avoid being shown a security warning every time you open the SQL Anywhere Monitor, you must permanently trust a self-signed certificate. Mozilla Firefox does this automatically when you add the exception. Microsoft Internet Explorer and Google Chrome require you to import the public certificate.

Internet Explorer and Chrome share digital certificates. Therefore, if you add a self-signed certificate from either browser, it will be trusted on both browsers.

The following steps describe this process using Internet Explorer. The process is similar in Chrome.

1. From Internet Explorer, click Tools > Internet Options, or from the Control Panel double-click Internet Options.
2. Switch to the Content Tab.



Ratings help you control the Internet content that can be viewed on this computer.

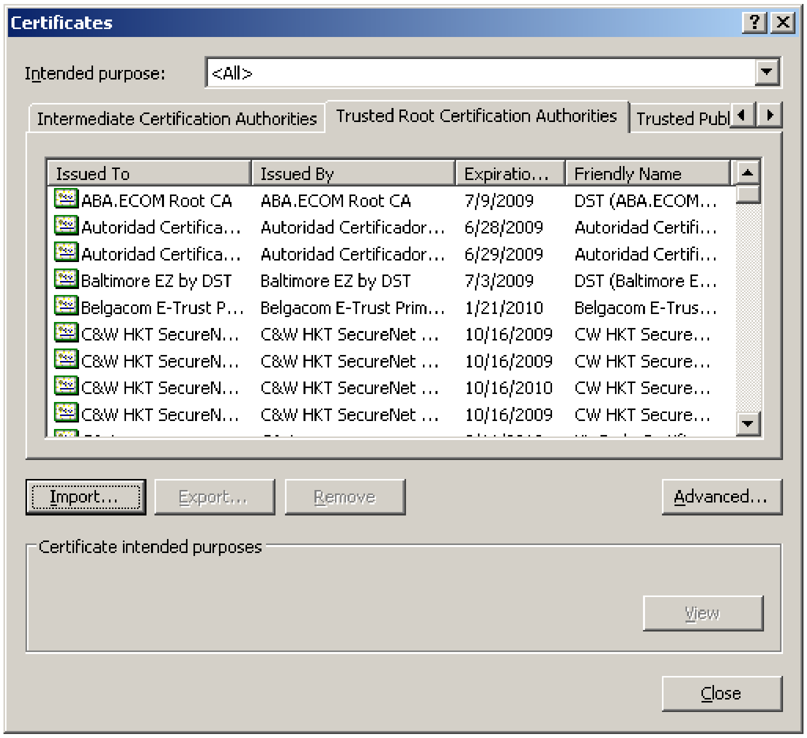
Use certificates for encrypted connections and identification.

AutoComplete stores previous entries on webpages and suggests matches for you.

Feeds and Web Slices provide updated content from websites that can be read in Internet Explorer and other programs.

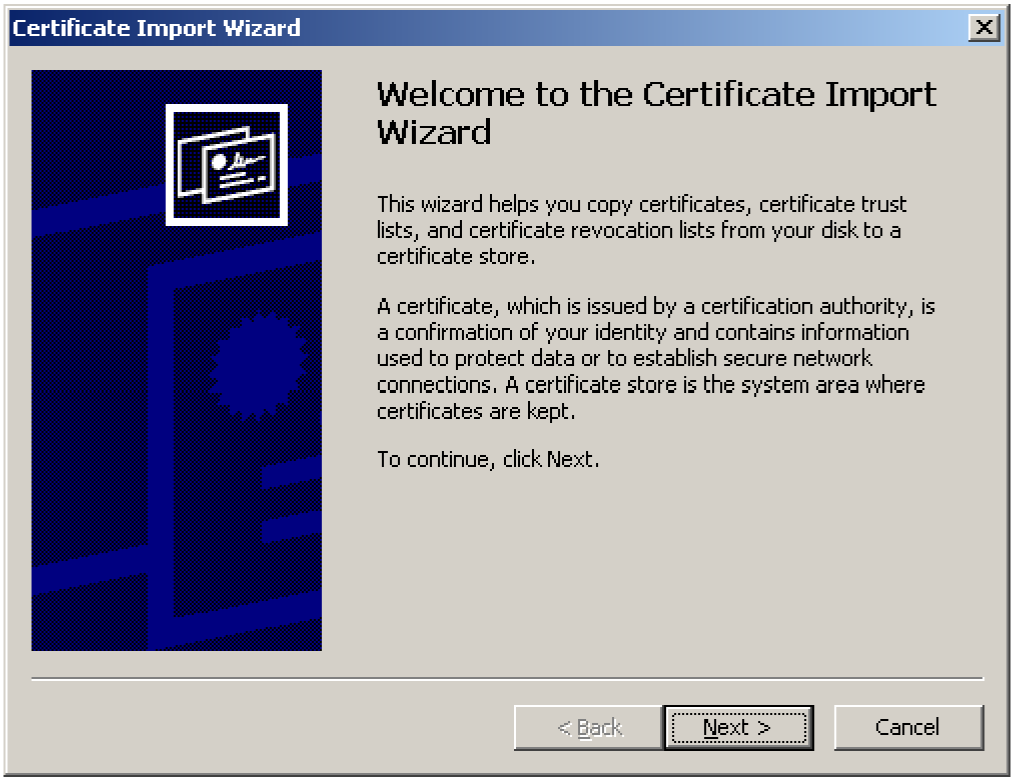
3. Click Certificates under the heading Certificates.

4. Switch to the Trusted Root Certification Authorities tab.


Certificates
? X
Intended purpose:
<All>
Intermediate Certification Authorities
Trusted Root Certification Authorities
Trusted Publ
Import...
Export...
Remove
Advanced...
Certificate intended purposes
View
Close

Issued To	Issued By	Expiratio...	Friendly Name
ABA.ECOM Root CA	ABA.ECOM Root CA	7/9/2009	DST (ABA.ECOM...
Autoridad Certifica...	Autoridad Certificador...	6/28/2009	Autoridad Certifi...
Autoridad Certifica...	Autoridad Certificador...	6/29/2009	Autoridad Certifi...
Baltimore EZ by DST	Baltimore EZ by DST	7/3/2009	DST (Baltimore E...
Belgacom E-Trust P...	Belgacom E-Trust Prim...	1/21/2010	Belgacom E-Trus...
C&W HKT SecureN...	C&W HKT SecureNet ...	10/16/2009	CW HKT Secure...
C&W HKT SecureN...	C&W HKT SecureNet ...	10/16/2009	CW HKT Secure...
C&W HKT SecureN...	C&W HKT SecureNet ...	10/16/2010	CW HKT Secure...
C&W HKT SecureN...	C&W HKT SecureNet ...	10/16/2009	CW HKT Secure...

5. Click Import.



Welcome to the Certificate Import Wizard

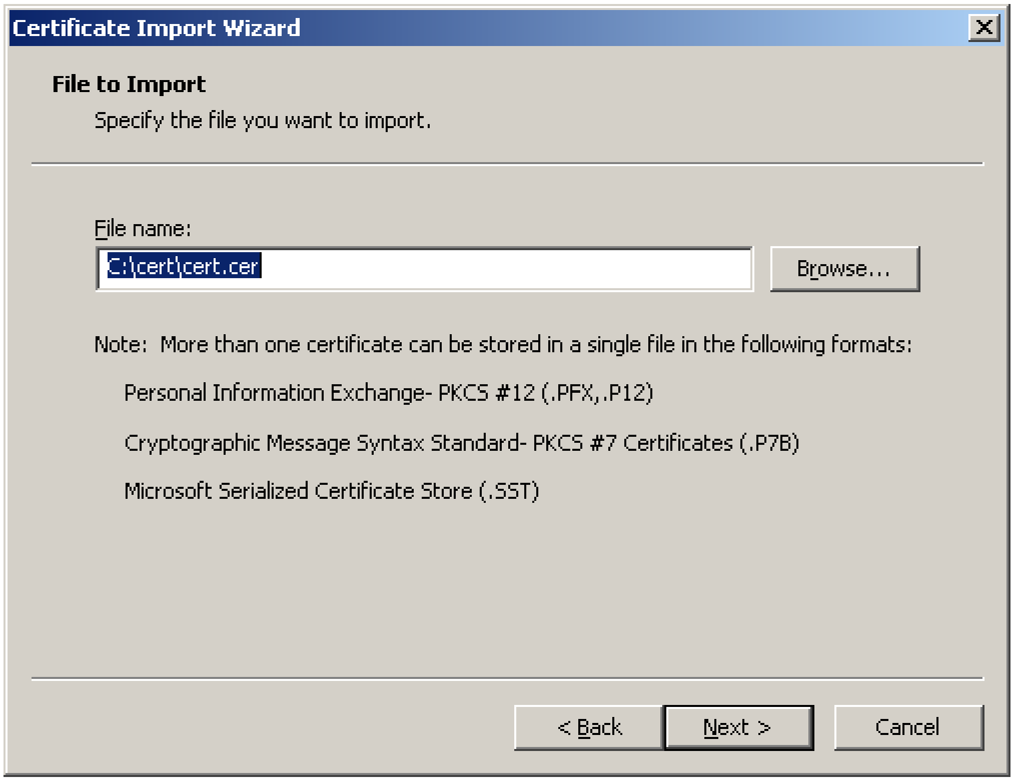
This wizard helps you copy certificates, certificate trust lists, and certificate revocation lists from your disk to a certificate store.

A certificate, which is issued by a certification authority, is a confirmation of your identity and contains information used to protect data or to establish secure network connections. A certificate store is the system area where certificates are kept.

To continue, click Next.

6. Click Next.

7. Browse to the location of your public certificate.



Certificate Import Wizard

File to Import

Specify the file you want to import.

File name: C:\cert\cert.cer

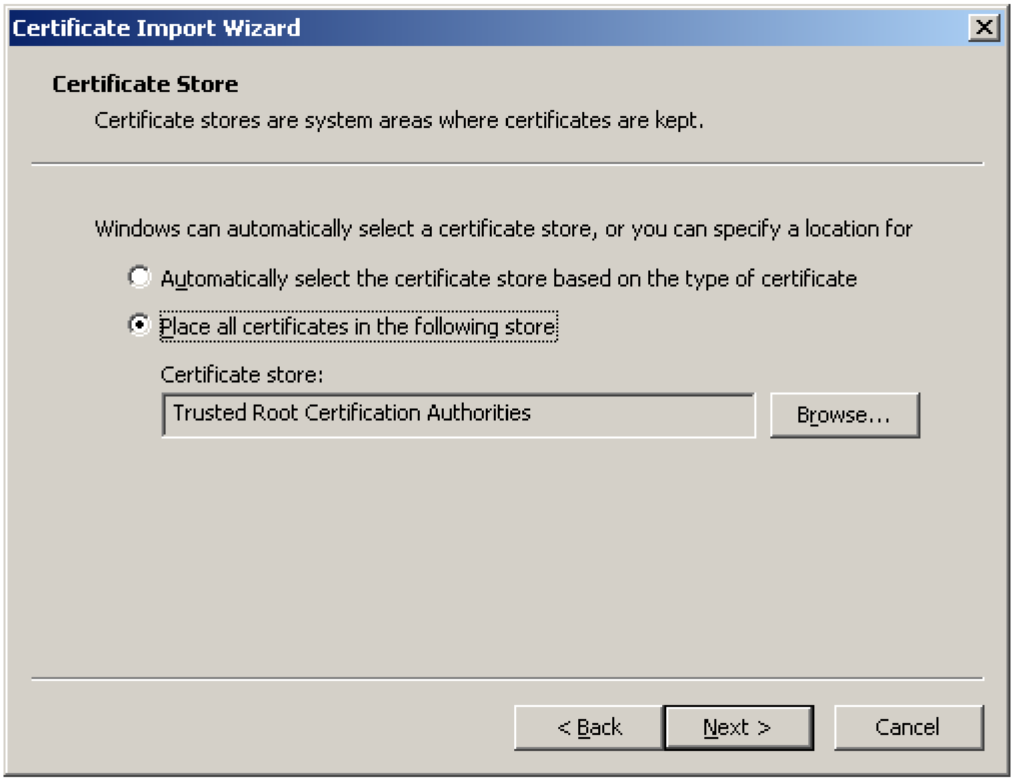
Note: More than one certificate can be stored in a single file in the following formats:

Personal Information Exchange- PKCS #12 (.PFX,.P12)

Cryptographic Message Syntax Standard- PKCS #7 Certificates (.P7B)

Microsoft Serialized Certificate Store (.SST)

8. Click Next.



Certificate Import Wizard

Certificate Store

Certificate stores are system areas where certificates are kept.

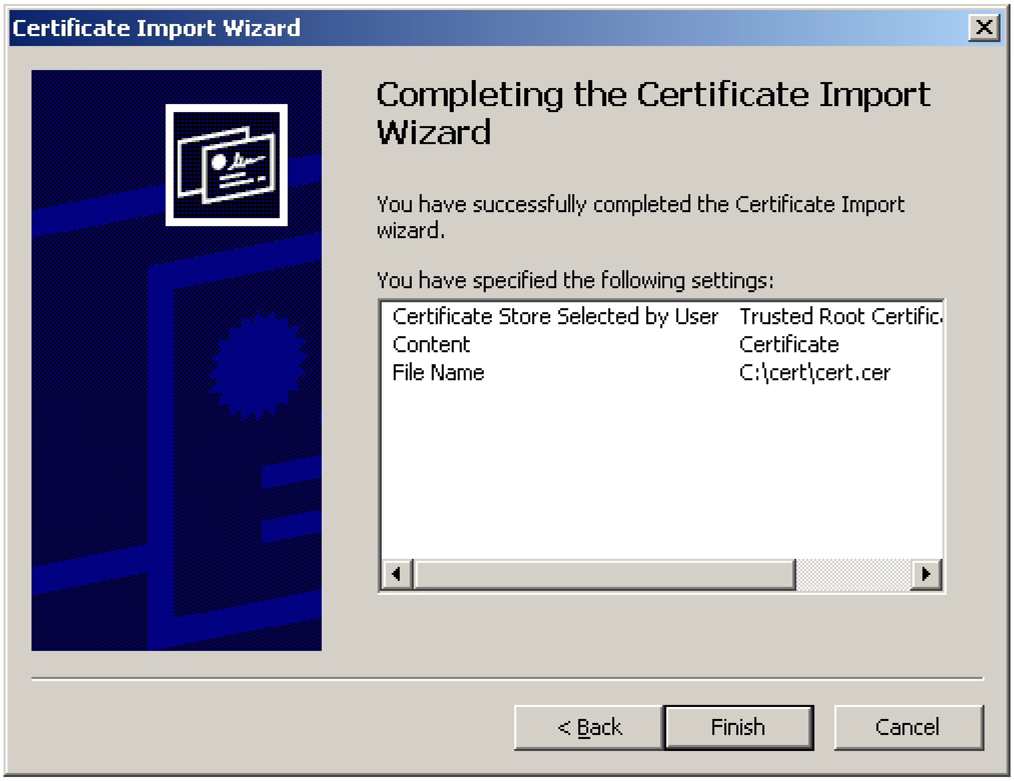
Windows can automatically select a certificate store, or you can specify a location for

○ Automatically select the certificate store based on the type of certificate

● Place all certificates in the following store

Certificate store: Trusted Root Certification Authorities

9. Accept default location and click Next.



Completing the Certificate Import Wizard

You have successfully completed the Certificate Import wizard.

You have specified the following settings:

Certificate Store Selected by User	Trusted Root Certific
Content	Certificate
File Name	C:\cert\cert.cer

10. Click Finish.

Once a certificate has been imported, you can access the SQL Anywhere Monitor securely without being warned every time.

SUMMARY

Securing communication between a browser and the SQL Anywhere Monitor is a fairly simple task. The SQL Anywhere HTTP server supports both third-party-signed and self-signed certificates. This process ensures that your data is encrypted and your identity is verified. For more information on securing communications, see the documentation.