Installing an all-in-one vSphere server

VSPHERE 5.1.0A | IPV6-ONLY | USING TRUSTED SSL CERTIFICATES

Contents

1. Request SSL Certificates for vSphere ESXi Hosts and vCenter Server

(This document and the included scripts are also available online at http://terena.org/ipv6)

The first step of having a secure VMware vSphere installation is making sure there are valid SSL certificates available. This guide will make provide all configurations and commands necessary to create a default – yet secure installation of VMware vSphere.

1.1. Create RSA keys

To create RSA keys a system with OpenSSL is required. This guide will provide commands that can either be run on a Linux machine, or in the Windows cmd prompt (given you have navigated to the /bin folder of the installation path of OpenSSL-win32).

The folder structure of /home/user/certs – or C:\certs should be like this:

```
--- catbert
       --- OpenSSL.cfg
--- dogbert
       --- OpenSSL.cfg
--- InventoryService
       --- inventoryservice.cfg
--- LogBrowser
       --- LogBrowser.cfg
--- ratbert
       --- OpenSSL.cfg
--- SSO
       --- sso.cfg
--- vCenter
       --- vcenter.cfg
--- UpdateManager
       --- UpdateManager.cfg
--- WebClient
       --- webclient.cfg
```

In each of the ESXi host folders (for my specific setup, these folders are catbert, dogbert and ratbert), type in the command to generate a new 2048 bit RSA key, and generate a Certificate Signing Request (CSR)

```
openssl genrsa 2048 > rui.key
openssl req -out rui.csr -key rui.key -new -config OpenSSL.cfg
```

Because the names of the vSphere components do not change, there exist scripts that automate this otherwise repetitive job:

Create_CSR.sh (run in /home/user/certs) or Create_CSR.bat

1.2. Requesting SSL Certificates

The next part is to request SSL Certificates. To do this, log in to the Certificate Authority of choice, and request new certificates. The CA will then ask CSR files, which are in the folders created in chapter 1.1.

2. Installing SSL Certificates on ESXi hosts

Since the VMware knowledgebase article is very clear on this step, here is a quote of their article which can be found here: http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2015499

Note: make sure to put the certificates received from your CA in their respective folder. Make sure the certificate chain is Base-64 encoded (in most cases it is), if it is not, transcode it. Rename the certificate chain to Root64.cer, and put it in the /home/user/certs or C:\certs folder.

"

Installing and configuring the certificate on the ESXi host

After the certificate is created, complete the installation and configuration of the certificate on the ESXi 5.x host:

1. Log in to vCenter Server
2. Put the host into **Maintenance Mode**.
3. Navigate to the console of the server to enable SSH on the ESXi 5.x host.
4. Press F2 to log in to the **Direct Console User Interface (DUCI)**.
5. Click **Troubleshooting options > Enable SSH**.
6. Log in to the host and then navigate to `/etc/vmware/ssl`.
7. Copy the files to a backup location, such as a VMFS volume.
8. Log in to the host with WinSCP and navigate to the `/etc/vmware/ssl` directory.
9. Delete the existing `rui.crt` and `rui.key` from the directory.
10. Copy the newly created `rui.crt` and `rui.key` to the directory using Text Mode or ASCII mode to avoid the issue of special characters (`^M`) appearing in the certificate file.
11. Type less `rui.crt` to validate that there are no extra characters.

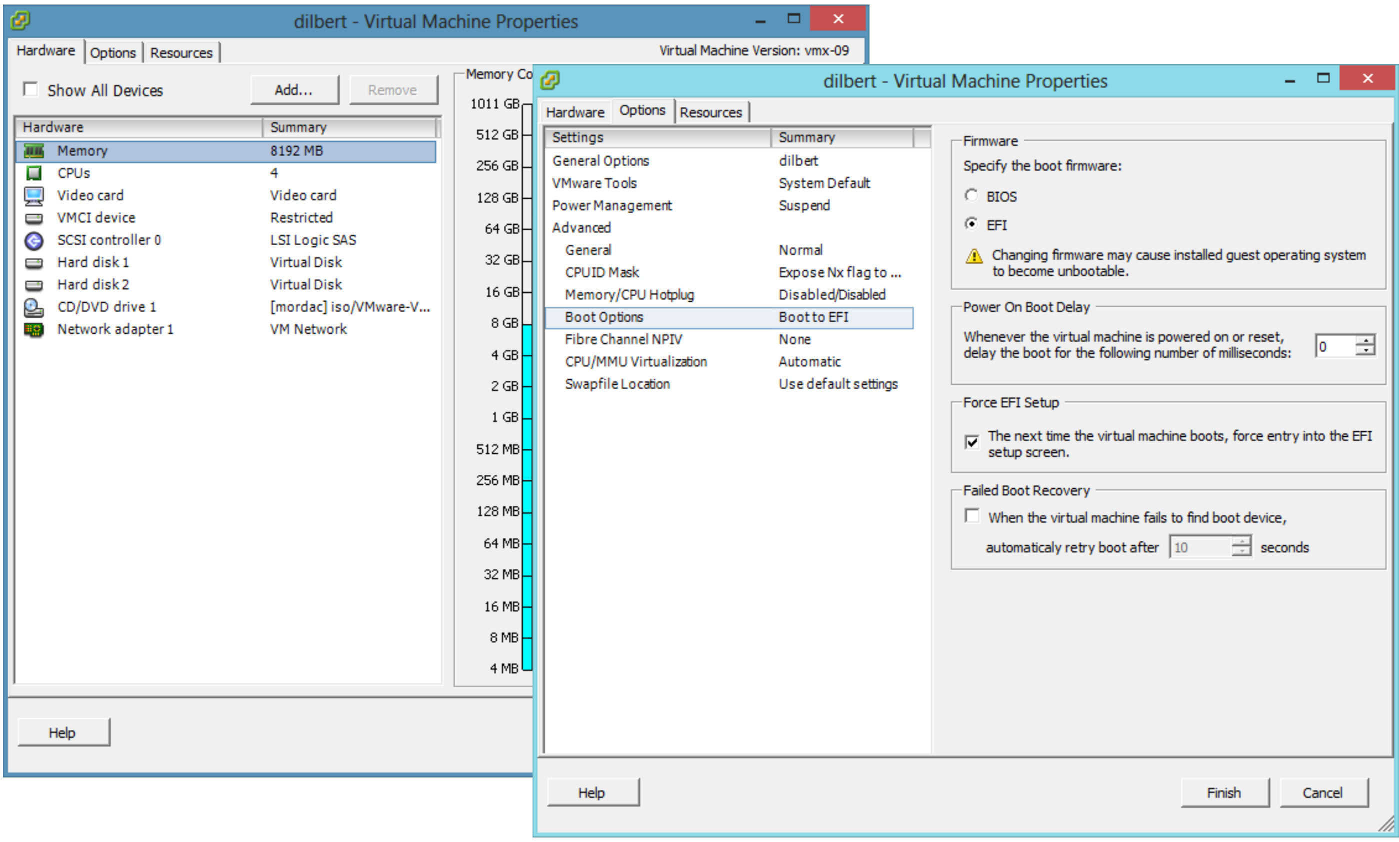
    Note: There should not be any erroneous ^M characters at the end of each line.

12. Switch back to the DCUI of the host and select **Troubleshooting Options > Restart Management Agents**.
13. When prompted press **F11** to restart the agents. Wait until they are restarted.
14. Press **ESC** several times until you logout of the DCUI.
15. Exit the host from **Maintenance Mode**.

"

3. Install an all-in-one vCenter Server

- Create a VM with at least 2 vCPUs and 4GB RAM, or install on a physical machine
- Recommended: a VM with 4vCPUs, 8GB RAM and EFI boot option (to make use of GPT disks, and thus 3TB+ volumes)



Virtual Machine Version: vmx-09

☐ Show All Devices

Hardware	Summary
Memory	8192 MB
CPUs	4
Video card	Video card
VMCI device	Restricted
SCSI controller 0	LSI Logic SAS
Hard disk 1	Virtual Disk
Hard disk 2	Virtual Disk
CD/DVD drive 1	[mordac] iso/VMware-V...
Network adapter 1	VM Network

Settings	Summary
General Options	dilbert
VMware Tools	System Default
Power Management	Suspend
Advanced	
General	Normal
CPUID Mask	Expose Nx flag to ...
Memory/CPU Hotplug	Disabled/Disabled
Boot Options	Boot to EFI
Fibre Channel NPIV	None
CPU/MMU Virtualization	Automatic
Swapfile Location	Use default settings

Firmware

Specify the boot firmware:

○ BIOS

● EFI

⚠ Changing firmware may cause installed guest operating system to become unbootable.

Power On Boot Delay

Whenever the virtual machine is powered on or reset, delay the boot for the following number of milliseconds: 0

Force EFI Setup

☑ The next time the virtual machine boots, force entry into the EFI setup screen.

Failed Boot Recovery

☐ When the virtual machine fails to find boot device, automaticaly retry boot after 10 seconds

- Install Windows Server 2008 R2, as Server 2012 is not yet officially supported by vSphere 5.1.0A
- Setup passwords, network settings (untick IPv4), and remote desktop (if it is going to be used)
- Install VMware Tools

3.1. Create Personal Information Exchange (PFX) files

Note: Make sure to put the certificates received from your CA in their respective folder. Make sure the certificate chain is Base-64 encoded (in most cases it is), if it is not, transcode it. Rename the certificate chain to Root64.cer, and put it in the /home/user/certs or C:\certs folder.

Because the names of the vSphere components do not change, there exist scripts that automate this otherwise repetitive job:

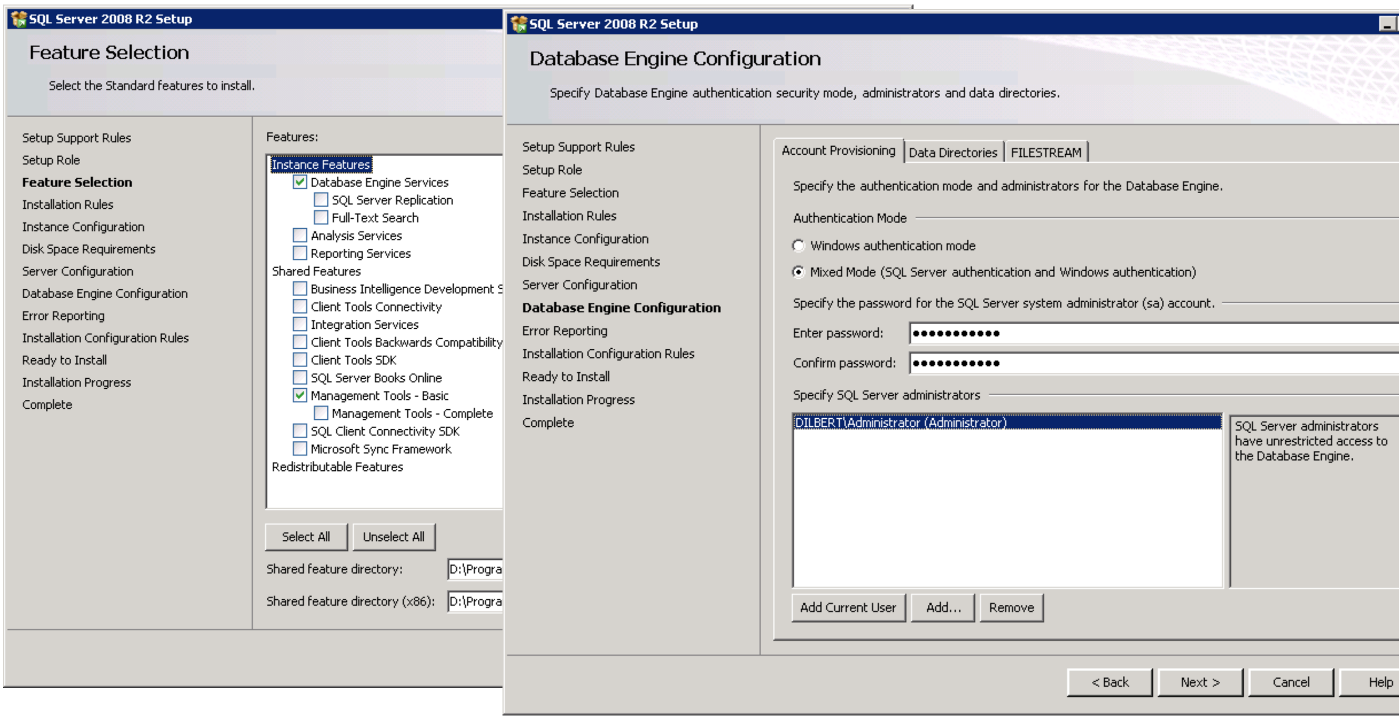
Create_PFX.sh (run in /home/user/certs) or Create_PFX.bat

3.2. Pre-stage certificates

The next step is to copy all the certificate files to C:\certs\ on the vCenter Server (VM). After the files are copied they can be put into place using the PreStage.bat script.

3.3. Start installing vCenter Server

- Install SQL Server 2008 R2, as SQL Server 2012 is not yet officially supported by vSphere 5.1.0A
- Select at least Database Engine Services and Management Tools – Basic
- Make sure to enable both username/password and Windows Authentication (mixed mode)



SQL Server 2008 R2 Setup

Feature Selection

Select the Standard features to install.

Features:

Instance Features
☑ Database Engine Services
□ SQL Server Replication
□ Full-Text Search
□ Analysis Services
□ Reporting Services
Shared Features
□ Business Intelligence Development S
□ Client Tools Connectivity
□ Integration Services
□ Client Tools Backwards Compatibility
□ Client Tools SDK
□ SQL Server Books Online
☑ Management Tools - Basic
□ Management Tools - Complete
□ SQL Client Connectivity SDK
□ Microsoft Sync Framework
Redistributable Features

Shared feature directory: D:\Progra

Shared feature directory (x86): D:\Progra

SQL Server 2008 R2 Setup

Database Engine Configuration

Specify Database Engine authentication security mode, administrators and data directories.

Specify the authentication mode and administrators for the Database Engine.

Authentication Mode

○ Windows authentication mode

● Mixed Mode (SQL Server authentication and Windows authentication)

Specify the password for the SQL Server system administrator (sa) account.

Enter password: ●●●●●●●●●●●

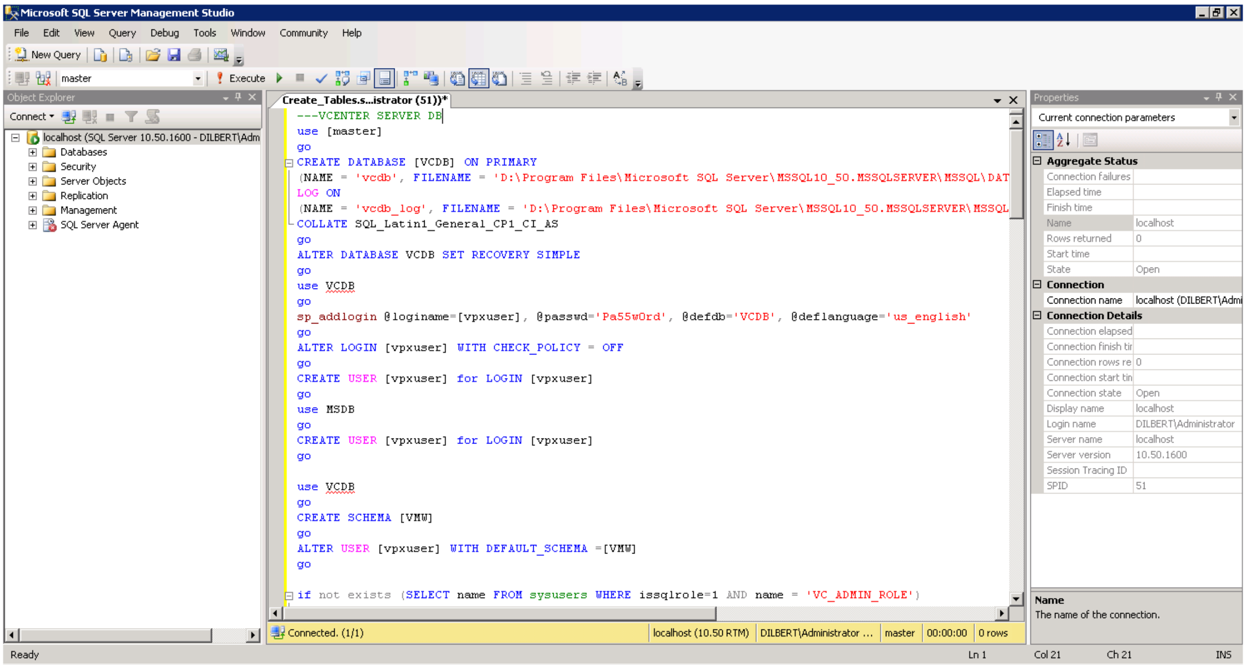
Confirm password: ●●●●●●●●●●●

Specify SQL Server administrators

DILBERT\Administrator (Administrator)

SQL Server administrators have unrestricted access to the Database Engine.

- Import Create_Tables.sql using SQL Server Management Studio
- Change the passwords (default Pa55w0rd) and file storage paths in the SQL statement
- Click Execute to install Databases, tables and users for SSO, vCenter and UpdateManager



Microsoft SQL Server Management Studio

Create_Tables.s...istrator (51))*

```
---VCENTER SERVER DB
use [master]
go
CREATE DATABASE [VCDB] ON PRIMARY
(NAME = 'vcdb', FILENAME = 'D:\Program Files\Microsoft SQL Server\MSSQL10_50.MSSQLSERVER\MSSQL\DAT
LOG ON
(NAME = 'vcdb_log', FILENAME = 'D:\Program Files\Microsoft SQL Server\MSSQL10_50.MSSQLSERVER\MSSQL
COLLATE SQL_Latin1_General_CP1_CI_AS
go
ALTER DATABASE VCDB SET RECOVERY SIMPLE
go
use VCDB
go
sp_addlogin @loginame=[vpxuser], @passwd='Pa55w0rd', @defdb='VCDB', @deflanguage='us_english'
go
ALTER LOGIN [vpxuser] WITH CHECK_POLICY = OFF
go
CREATE USER [vpxuser] for LOGIN [vpxuser]
go
use MSDB
go
CREATE USER [vpxuser] for LOGIN [vpxuser]
go

use VCDB
go
CREATE SCHEMA [VMW]
go
ALTER USER [vpxuser] WITH DEFAULT_SCHEMA =[VMW]
go

if not exists (SELECT name FROM sysusers WHERE issqlrole=1 AND name = 'VC_ADMIN_ROLE')
```

Connected. (1/1) | localhost (10.50 RTM) | DILBERT\Administrator ... | master | 00:00:00 | 0 rows

Properties — Current connection parameters

Aggregate Status	
Connection failures	
Elapsed time	
Finish time	
Name	localhost
Rows returned	0
Start time	
State	Open
Connection	
Connection name	localhost (DILBERT\Admi
Connection Details	
Connection elapsed	
Connection finish tir	
Connection rows re	0
Connection start tin	
Connection state	Open
Display name	localhost
Login name	DILBERT\Administrator
Server name	localhost
Server version	10.50.1600
Session Tracing ID	
SPID	51

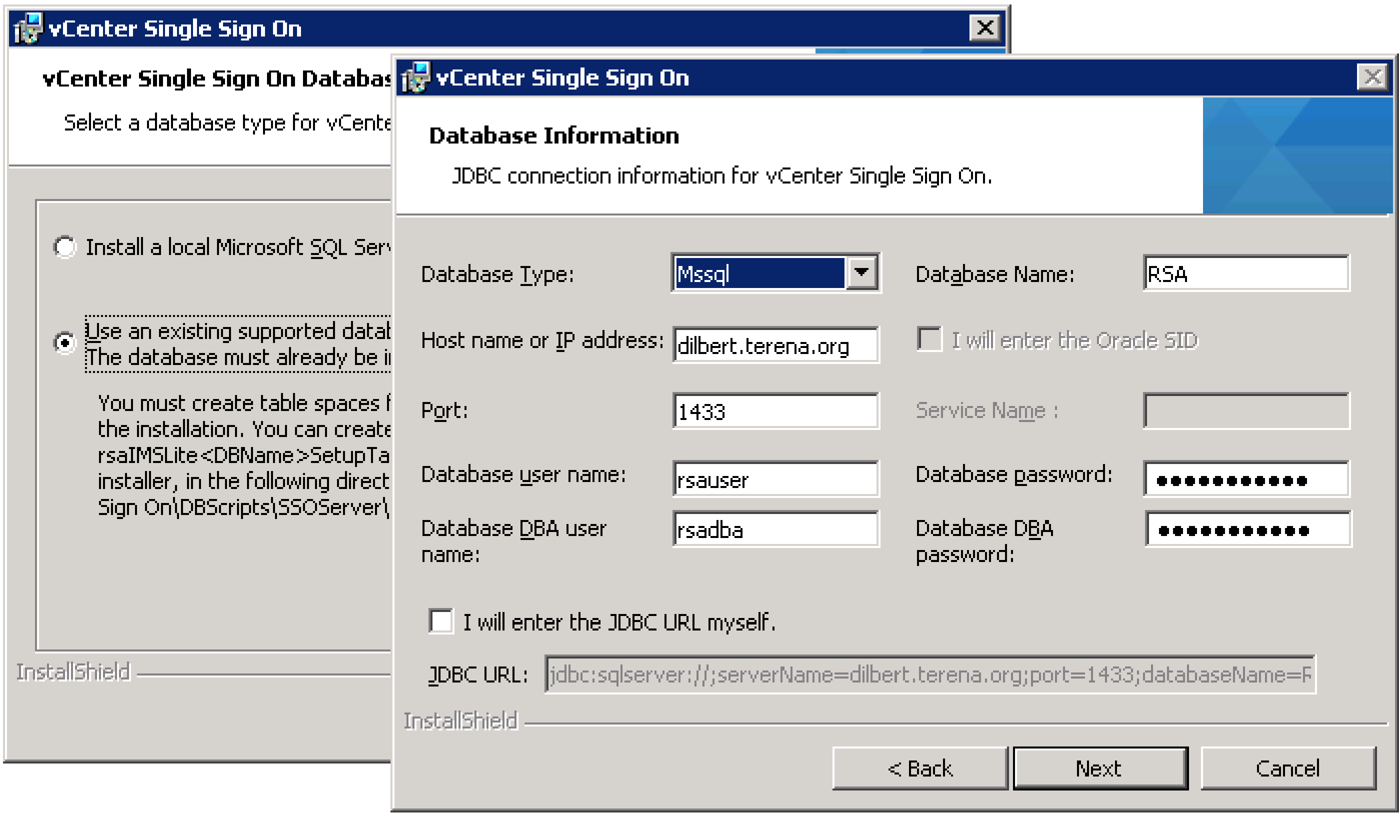
Name
The name of the connection.

Note: VMware installers will give errors about the FQDN cannot be resolved. This is due to VMware looking for an IPv4 address. This error can be safely ignored, be sure to double check for typing errors.

3.3.1. vCenter SSO

Install vCenter SSO

- Install the vCenter SSO by running the setup package



vCenter Single Sign On

vCenter Single Sign On Databas

Select a database type for vCente

○ Install a local Microsoft SQL Serv

● Use an existing supported datab
The database must already be i

You must create table spaces f
the installation. You can create
rsaIMSLite<DBName>SetupTa
installer, in the following direct
Sign On\DBScripts\SSOServer\

InstallShield

vCenter Single Sign On

Database Information

JDBC connection information for vCenter Single Sign On.

Database Type: Mssql

Database Name: RSA

Host name or IP address: dilbert.terena.org

□ I will enter the Oracle SID

Port: 1433

Service Name :

Database user name: rsauser

Database password: ●●●●●●●●●●●

Database DBA user name: rsadba

Database DBA password: ●●●●●●●●●●●

□ I will enter the JDBC URL myself.

JDBC URL: jdbc:sqlserver://;serverName=dilbert.terena.org;port=1433;databaseName=R

InstallShield

- Replace the SSO SSL certificates by running the ReplaceSSOcerts.bat script

3.3.2. VMware vCenter Inventory Service

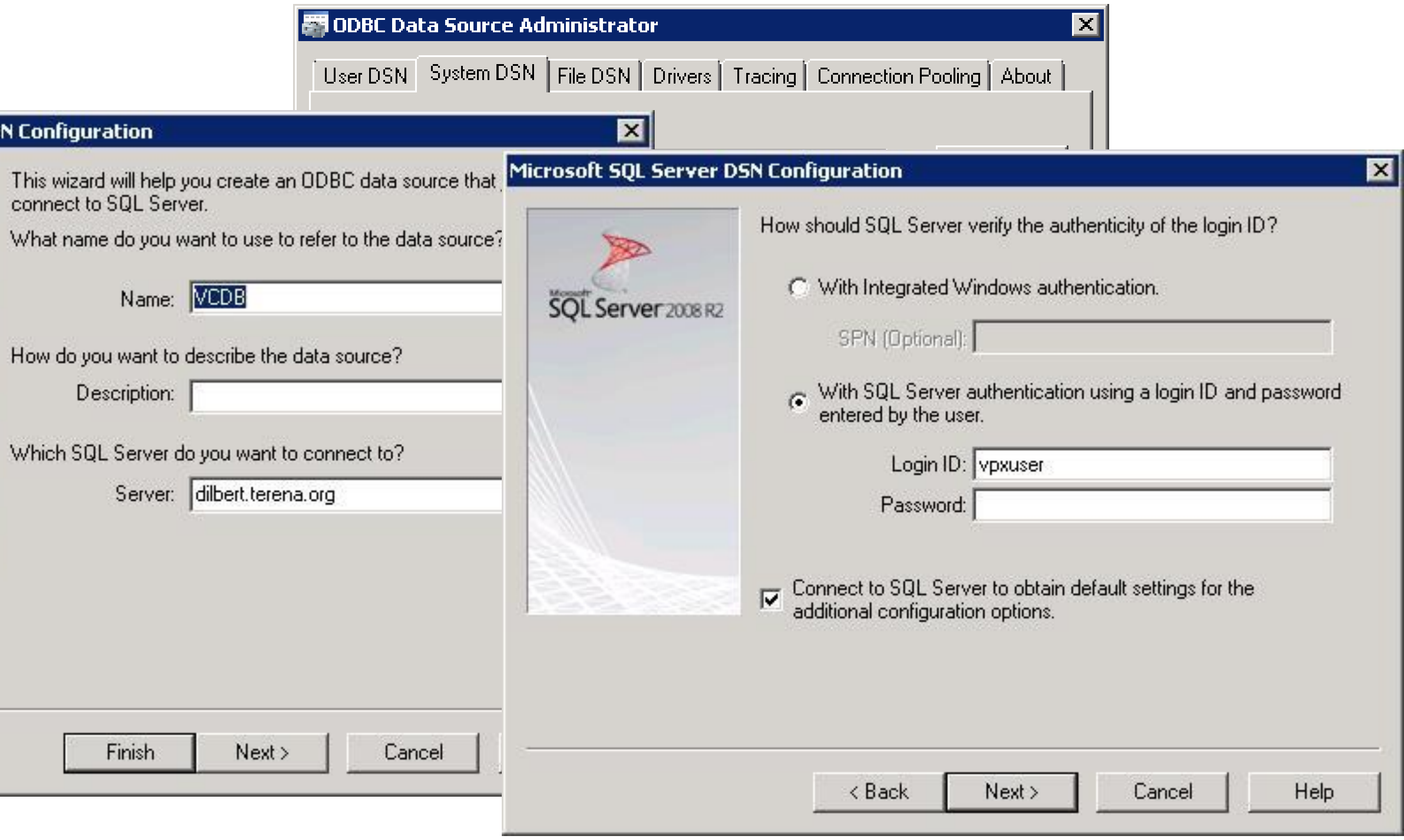
Install VMware vCenter Inventory Service

- Install the vCenter Inventory Service by running the setup package
- Use either the FQDN or the IPv6 address within brackets [] when registering the Inventory Service to the vCenter SSO

3.3.3. VMware vCenter Server

Create a 64-bit System DSN

- Create a 32-bit System DSN by running C:\Windows\System32\ocdbad32.exe
- Choose Add… > SQL Native Client 10.0
- Choose VCDB as database, and SQL Authentication as Authentication method
- Login ID: vpxuser,
- Password: the one used in Create_Tables.sql



This wizard will help you create an ODBC data source that connect to SQL Server.

What name do you want to use to refer to the data source?

Name: VCDB

How do you want to describe the data source?

Description:

Which SQL Server do you want to connect to?

Server: dilbert.terena.org

How should SQL Server verify the authenticity of the login ID?

○ With Integrated Windows authentication.

SPN (Optional):

● With SQL Server authentication using a login ID and password entered by the user.

Login ID: vpxuser

Password:

☑ Connect to SQL Server to obtain default settings for the additional configuration options.

- Start the Microsoft SQL Agent using the SQL Server Configuration Manager

Install VMware vCenter Server

- Install the vCenter Server by running the setup package
- Change SQL Agent Service to Automatic (Delayed Start), this is for scheduled database cleanup

3.3.4. VMware vSphere Update Manager

Note: hostupdate.vmware.com server has only an A record, so usage of Update Manager requires either native IPv4 access or NAT64

Create a 32-bit System DSN

- Create a 32-bit System DSN by running C:\Windows\SysWOW64\ocdbad32.exe
- Choose Add… > SQL Native Client 10.0
- Use Integrated Windows Authentication as Authentication method, then force UMDB as database

Install VMware vSphere Update Manager

- Install VMware vSphere Update Manager by running the setup package
- Enter 127.0.0.1 as IP address – update manager does not support connecting through IPv6 sockets

3.3.5. vSphere Web Client

Install vSphere Web Client

- Install the vSphere Web Client by running the setup package

Create LogBrowser keystore

- Open an elevated command prompt

Set the JAVA home path for the keytool:

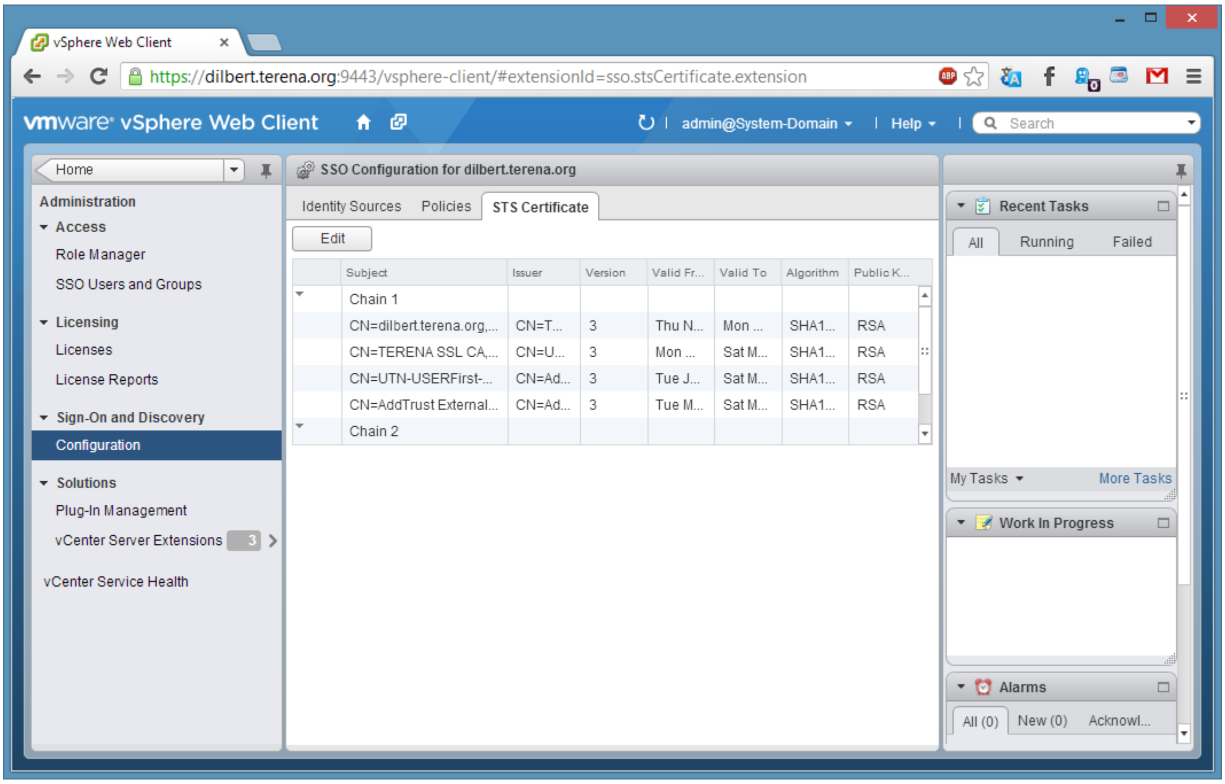
```
set JAVA_HOME=c:\Program Files\VMware\Infrastructure\JRE
```

Generate the Java Key Store:

```
cd /d “C:\Program Files\VMware\Infrastructure\jre\bin”
keytool -v -importkeystore -srckeystore C:\certs\LogBrowser\rui.pfx -srcstoretype
pkcs12 -srcstorepass testpassword -srcalias rui -destkeystore
C:\certs\LogBrowser\rui.jks -deststoretype JKS -deststorepass changeit -destkeypass
changeit
```

Import the LogBrowser keystore

- Copy JKS file to C:\Program Files\VMware\Infrastructure\SSOserver\security
- Import the JKS using the SSO STS Keystore configuration using the vSphere Web Client (Administration > Single-Sign-On and Discovery configuration > STS Certificates tab)



https://dilbert.terena.org:9443/vsphere-client/#extensionId=sso.stsCertificate.extension

SSO Configuration for dilbert.terena.org

Subject	Issuer	Version	Valid Fr...	Valid To	Algorithm	Public K...
Chain 1						
CN=dilbert.terena.org,...	CN=T...	3	Thu N...	Mon ...	SHA1...	RSA
CN=TERENA SSL CA,...	CN=U...	3	Mon ...	Sat M...	SHA1...	RSA
CN=UTN-USERFirst-...	CN=Ad...	3	Tue J...	Sat M...	SHA1...	RSA
CN=AddTrust External...	CN=Ad...	3	Tue M...	Sat M...	SHA1...	RSA
Chain 2						

Restart either the server or the VMware services in this order:

- Stop the VMware Log Browser service.
- Stop the VMware vSphere Web Client service.
- Stop the VMware VirtualCenter Server service.
- Stop the VMware vCenter Inventory service.
- Stop the vCenter Single Sign On service.
- Start the vCenter Single Sign On service.
- Start the VMware vCenter Inventory service.
- Start the VMware VirtualCenter Server service and the VMware VirtualCenter Management WebServices service.
- Start the VMware vSphere Web Client service.
- Start the VMware Log Browser service.

4. Scripts and configuration files

```
OpenSSL.cfg
---
[ req ]
default_bits = 2048
default_keyfile = rui.key
distinguished_name = req_distinguished_name
encrypt_key = no
prompt = no
string_mask = nombstr
req_extensions = v3_req

[ v3_req ]
basicConstraints = CA:FALSE
keyUsage = digitalSignature, keyEncipherment, dataEncipherment
extendedKeyUsage = serverAuth
subjectAltName = DNS:catbert, DNS:catbert.terena.org

[ req_distinguished_name ]
countryName = NL
stateOrProvinceName = Noord-Holland
localityName = Amsterdam
0.organizationName = TERENA
organizationalUnitName = ESXi
commonName = catbert.terena.org
---

inventoryservice.cfg
---
[ req ]
default_bits = 2048
default_keyfile = rui.key
distinguished_name = req_distinguished_name
encrypt_key = no
prompt = no
string_mask = nombstr
req_extensions = v3_req

[ v3_req ]
basicConstraints = CA:FALSE
keyUsage = digitalSignature, keyEncipherment, dataEncipherment
extendedKeyUsage = serverAuth
subjectAltName = DNS:dilbert, DNS:dilbert.terena.org

[ req_distinguished_name ]
countryName = NL
stateOrProvinceName = Noord-Holland
localityName = Amsterdam
0.organizationName = TERENA
organizationalUnitName = vCenterInventoryService
commonName = dilbert.terena.org
---

sso.cfg
---
[ req ]
default_bits = 2048
default_keyfile = rui.key
distinguished_name = req_distinguished_name
encrypt_key = no
prompt = no
string_mask = nombstr
req_extensions = v3_req

[ v3_req ]
basicConstraints = CA:FALSE
keyUsage = digitalSignature, keyEncipherment, dataEncipherment
```

```
extendedKeyUsage = serverAuth
subjectAltName = DNS:dilbert, DNS:dilbert.terena.org

[ req_distinguished_name ]
countryName = NL
stateOrProvinceName = Noord-Holland
localityName = Amsterdam
0.organizationName = TERENA
organizationalUnitName = vCenterSSO
commonName = dilbert.terena.org
---

vcenter.cfg
---
default_bits = 2048
default_keyfile = rui.key
distinguished_name = req_distinguished_name
encrypt_key = no
prompt = no
string_mask = nombstr
req_extensions = v3_req

[ v3_req ]
basicConstraints = CA:FALSE
keyUsage = digitalSignature, keyEncipherment, dataEncipherment
extendedKeyUsage = serverAuth
subjectAltName = DNS:dilbert, DNS:dilbert.terena.org

[ req_distinguished_name ]
countryName = NL
stateOrProvinceName = Noord-Holland
localityName = Amsterdam
0.organizationName = TERENA
organizationalUnitName = vCenterServer
commonName = dilbert.terena.org
---

webclient.cfg
---
[ req ]
default_bits = 2048
default_keyfile = rui.key
distinguished_name = req_distinguished_name
encrypt_key = no
prompt = no
string_mask = nombstr
req_extensions = v3_req

[ v3_req ]
basicConstraints = CA:FALSE
keyUsage = digitalSignature, keyEncipherment, dataEncipherment
extendedKeyUsage = serverAuth
subjectAltName = DNS:dilbert, DNS:dilbert.terena.org

[ req_distinguished_name ]
countryName = NL
stateOrProvinceName = Noord-Holland
localityName = Amsterdam
0.organizationName = TERENA
organizationalUnitName = vCenterWebClient
commonName = dilbert.terena.org
---

LogBrowser.cfg
---
[ req ]
default_bits = 2048
default_keyfile = rui.key
```

```
distinguished_name = req_distinguished_name
encrypt_key = no
prompt = no
string_mask = nombstr
req_extensions = v3_req

[ v3_req ]
basicConstraints = CA:FALSE
keyUsage = digitalSignature, keyEncipherment, dataEncipherment
extendedKeyUsage = serverAuth
subjectAltName = DNS:dilbert, DNS:dilbert.terena.org

[ req_distinguished_name ]
countryName = NL
stateOrProvinceName = Noord-Holland
localityName = Amsterdam
0.organizationName = TERENA
organizationalUnitName = vCenterLogBrowser
commonName = dilbert.terena.org
---

UpdateManager.cfg
---
[ req ]
default_bits = 2048
default_keyfile = rui.key
distinguished_name = req_distinguished_name
encrypt_key = no
prompt = no
string_mask = nombstr
req_extensions = v3_req

[ v3_req ]
basicConstraints = CA:FALSE
keyUsage = digitalSignature, keyEncipherment, dataEncipherment
extendedKeyUsage = serverAuth
subjectAltName = DNS:dilbert, DNS:dilbert.terena.org

[ req_distinguished_name ]
countryName = NL
stateOrProvinceName = Noord-Holland
localityName = Amsterdam
0.organizationName = TERENA
organizationalUnitName = VMwareUpdateManager
commonName = dilbert.terena.org
```

```
Create_CSR.bat
---
CD /d C:\certs\InventoryService
C:\OpenSSL-Win32\bin\openssl genrsa 2048 > rui.key
C:\OpenSSL-Win32\bin\openssl req -out rui.csr -key rui.key -new -config
inventoryservice.cfg

CD /d C:\certs\SSO
C:\OpenSSL-Win32\bin\openssl genrsa 2048 > rui.key
C:\OpenSSL-Win32\bin\openssl req -out rui.csr -key rui.key -new -config sso.cfg

CD /d C:\certs\vCenter
C:\OpenSSL-Win32\bin\openssl genrsa 2048 > rui.key
C:\OpenSSL-Win32\bin\openssl req -out rui.csr -key rui.key -new -config vcenter.cfg

CD /d C:\certs\WebClient
C:\OpenSSL-Win32\bin\openssl genrsa 2048 > rui.key
C:\OpenSSL-Win32\bin\openssl req -out rui.csr -key rui.key -new -config
webclient.cfg

CD /d C:\certs\LogBrowser
C:\OpenSSL-Win32\bin\openssl genrsa 2048 > rui.key
C:\OpenSSL-Win32\bin\openssl req -out rui.csr -key rui.key -new -config
LogBrowser.cfg

CD /d C:\certs\UpdateManager
C:\OpenSSL-Win32\bin\openssl genrsa 2048 > rui.key
C:\OpenSSL-Win32\bin\openssl req -out rui.csr -key rui.key -new -config
UpdateManager.cfg
---
```

```
Create_CSR.sh
#!/bin/sh
cd InventoryService
openssl genrsa 2048 > rui.key
openssl req -out rui.csr -key rui.key -new -config inventoryservice.cfg
cd ..
cd SSO
openssl genrsa 2048 > rui.key
openssl req -out rui.csr -key rui.key -new -config sso.cfg
cd ..
cd vCenter
openssl genrsa 2048 > rui.key
openssl req -out rui.csr -key rui.key -new -config vcenter.cfg
cd ..
cd WebClient
openssl genrsa 2048 > rui.key
openssl req -out rui.csr -key rui.key -new -config webclient.cfg
cd ..
cd LogBrowser
openssl genrsa 2048 > rui.key
openssl req -out rui.csr -key rui.key -new -config LogBrowser.cfg
cd ..
cd UpdateManager
openssl genrsa 2048 > rui.key
openssl req -out rui.csr -key rui.key -new -config UpdateManager.cfg
cd ..
---
```

```
Create_PFX.bat
---
CD /d C:\certs\InventoryService
C:\OpenSSL-Win32\bin\openssl pkcs12 -export -in rui.crt -inkey rui.key -certfile
C:\certs\Root64.cer -name rui -passout pass:testpassword -out rui.pfx

CD /d C:\certs\SSO
C:\OpenSSL-Win32\bin\openssl pkcs12 -export -in rui.crt -inkey rui.key -certfile
C:\certs\Root64.cer -name rui -passout pass:testpassword -out rui.pfx

CD /d C:\certs\vCenter
C:\OpenSSL-Win32\bin\openssl pkcs12 -export -in rui.crt -inkey rui.key -certfile
C:\certs\Root64.cer -name rui -passout pass:testpassword -out rui.pfx

CD /d C:\certs\WebClient
C:\OpenSSL-Win32\bin\openssl pkcs12 -export -in rui.crt -inkey rui.key -certfile
C:\certs\Root64.cer -name rui -passout pass:testpassword -out rui.pfx

CD /d C:\certs\LogBrowser
C:\OpenSSL-Win32\bin\openssl pkcs12 -export -in rui.crt -inkey rui.key -certfile
C:\certs\Root64.cer -name rui -passout pass:testpassword -out rui.pfx

CD /d C:\certs\UpdateManager
C:\OpenSSL-Win32\bin\openssl pkcs12 -export -in rui.crt -inkey rui.key -certfile
C:\certs\Root64.cer -name rui -passout pass:testpassword -out rui.pfx
---
```

```
Create_PFX.sh
#!/bin/sh
cd InventoryService
openssl pkcs12 -export -in rui.crt -inkey rui.key -certfile ../Root64.cer -name rui
-passout pass:testpassword -out rui.pfx
cd ..
cd SSO
openssl pkcs12 -export -in rui.crt -inkey rui.key -certfile ../Root64.cer -name rui
-passout pass:testpassword -out rui.pfx
cd ..
cd vCenter
openssl pkcs12 -export -in rui.crt -inkey rui.key -certfile ../Root64.cer -name rui
-passout pass:testpassword -out rui.pfx
cd ..
cd WebClient
openssl pkcs12 -export -in rui.crt -inkey rui.key -certfile ../Root64.cer -name rui
-passout pass:testpassword -out rui.pfx
cd ..
cd LogBrowser
openssl pkcs12 -export -in rui.crt -inkey rui.key -certfile ../Root64.cer -name rui
-passout pass:testpassword -out rui.pfx
cd ..
cd UpdateManager
openssl pkcs12 -export -in rui.crt -inkey rui.key -certfile ../Root64.cer -name rui
-passout pass:testpassword -out rui.pfx
cd ..
---
```

```
PreStage.bat
---
mkdir C:\ProgramData\VMware\SingleSignOn\SSL
robocopy C:\certs\SSO\ c:\ProgramData\VMware\SingleSignOn\SSL\ /XF rui.csr sso.cfg

mkdir "C:\ProgramData\VMware\Infrastructure\Inventory Service\SSL"
robocopy C:\certs\Inventory\ "C:\ProgramData\VMware\Infrastructure\Inventory
Service\ssl" /XF rui.csr inventory.cfg

mkdir "C:\ProgramData\VMware\VMware VirtualCenter\SSL"
robocopy C:\certs\vCenter\ "C:\ProgramData\VMware\VMware VirtualCenter\SSL" /XF
rui.csr vcenter.cfg

mkdir "C:\ProgramData\VMware\vSphere Web Client\SSL"
robocopy C:\certs\WebClient\ "C:\ProgramData\VMware\vSphere Web Client\SSL" /XF
rui.csr webclient.cfg

mkdir "C:\Program Files (x86)\VMware\Infrastructure\Update Manager\SSL"
robocopy C:\certs\UpdateManager\ "C:\Program Files
(x86)\VMware\Infrastructure\Update Manager\SSL" /XF rui.csr UpdateManager.cfg
---
```

```
Create_Tables.sql
---
--- Create_Tables.sql is a modified compliation of VMware scripts.
--- Scripts used are:
--- rsaIMSLiteMSSQLSetupTablespaces.sql
--- rsaIMSLiteMSSQLSetupUsers.sql
--- DB_and_schema_creation_scripts_MSSQL.txt
--- VSPHERE SSO DATABASE
USE MASTER
GO
CREATE DATABASE RSA ON PRIMARY(
    NAME='RSA_DATA',
    FILENAME='D:\Program Files\Microsoft SQL
Server\MSSQL10_50.MSSQLSERVER\MSSQL\DATA\RSA_DATA.mdf',
    SIZE=10MB,
    MAXSIZE=UNLIMITED,
    FILEGROWTH=10%),
FILEGROUP RSA_INDEX(
    NAME='RSA_INDEX',
    FILENAME='D:\Program Files\Microsoft SQL
Server\MSSQL10_50.MSSQLSERVER\MSSQL\DATA\RSA_INDEX.ndf',
    SIZE=10MB,
    MAXSIZE=UNLIMITED,
    FILEGROWTH=10%)
LOG ON(
    NAME='translog',
    FILENAME='D:\Program Files\Microsoft SQL
Server\MSSQL10_50.MSSQLSERVER\MSSQL\DATA\translog.ldf',
    SIZE=10MB,
    MAXSIZE=UNLIMITED,
    FILEGROWTH=10% )
GO
EXEC SP_DBOPTION 'RSA', 'autoshrink', true
GO
EXEC SP_DBOPTION 'RSA', 'trunc. log on chkpt.', true
GO
CHECKPOINT
GO
--- VSPHERE SSO USERS
USE MASTER
GO
CREATE LOGIN RSA_DBA WITH PASSWORD = 'Pa55w0rd', DEFAULT_DATABASE = RSA
GO
CREATE LOGIN RSA_USER WITH PASSWORD = 'Pa55w0rd', DEFAULT_DATABASE = RSA
GO
USE RSA
GO
ALTER AUTHORIZATION ON DATABASE::RSA TO [RSA_DBA]
GO
CREATE USER RSA_USER FOR LOGIN [RSA_USER]
GO
CHECKPOINT
GO
--- VCENTER SERVER DATABASE AND VPXUSER
use [master]
go
CREATE DATABASE [VCDB] ON PRIMARY
(NAME = N'vcdb', FILENAME = N'D:\Program Files\Microsoft SQL
Server\MSSQL10_50.MSSQLSERVER\MSSQL\DATA\VCDB.mdf' , SIZE = 3000KB ,
FILEGROWTH = 10% )
```

```
LOG ON
(NAME = N'vcdb_log', FILENAME = N'D:\Program Files\Microsoft SQL
Server\MSSQL10_50.MSSQLSERVER\MSSQL\DATA\VCDB.ldf' , SIZE = 1000KB ,
FILEGROWTH = 10%)
COLLATE SQL_Latin1_General_CP1_CI_AS
go
use VCDB
go
sp_addlogin @loginame=[vpxuser], @passwd=N'Pa55w0rd', @defdb='VCDB',
@deflanguage='us_english'
go
ALTER LOGIN [vpxuser] WITH CHECK_POLICY = OFF
go
CREATE USER [vpxuser] for LOGIN [vpxuser]
go
CREATE SCHEMA [VMW]
go
ALTER USER [vpxuser] WITH DEFAULT_SCHEMA =[VMW]
go
if not exists (SELECT name FROM sysusers WHERE issqlrole=1 AND name =
'VC_ADMIN_ROLE')
CREATE ROLE VC_ADMIN_ROLE;
GRANT ALTER ON SCHEMA :: [VMW] to VC_ADMIN_ROLE;
GRANT REFERENCES ON SCHEMA :: [VMW] to VC_ADMIN_ROLE;
GRANT INSERT ON SCHEMA ::  [VMW] to VC_ADMIN_ROLE;
GRANT CREATE TABLE to VC_ADMIN_ROLE;
GRANT CREATE VIEW to VC_ADMIN_ROLE;
GRANT CREATE Procedure to VC_ADMIN_ROLE;
if not exists (SELECT name FROM sysusers WHERE issqlrole=1 AND name =
'VC_USER_ROLE')
CREATE ROLE VC_USER_ROLE
go
GRANT SELECT ON SCHEMA ::  [VMW] to VC_USER_ROLE
go
GRANT INSERT ON SCHEMA ::  [VMW] to VC_USER_ROLE
go
GRANT DELETE ON SCHEMA ::  [VMW] to VC_USER_ROLE
go
GRANT UPDATE ON SCHEMA ::  [VMW] to VC_USER_ROLE
go
GRANT EXECUTE ON SCHEMA :: [VMW] to VC_USER_ROLE
go
sp_addrolemember VC_ADMIN_ROLE , [vpxuser]
go
sp_addrolemember VC_USER_ROLE , [vpxuser]
go
use MSDB
go
CREATE USER [vpxuser] for LOGIN [vpxuser]
go
if not exists (SELECT name FROM sysusers WHERE issqlrole=1 AND name =
'VC_ADMIN_ROLE')
CREATE ROLE VC_ADMIN_ROLE;
go
grant select on msdb.dbo.syscategories to VC_ADMIN_ROLE
go
grant select on msdb.dbo.sysjobsteps to VC_ADMIN_ROLE
go
GRANT SELECT ON msdb.dbo.sysjobs to VC_ADMIN_ROLE
GO
GRANT EXECUTE ON msdb.dbo.sp_add_job TO VC_ADMIN_ROLE
```

```
go
GRANT EXECUTE ON msdb.dbo.sp_delete_job TO VC_ADMIN_ROLE
go
GRANT EXECUTE ON msdb.dbo.sp_add_jobstep TO VC_ADMIN_ROLE
go
GRANT EXECUTE ON msdb.dbo.sp_update_job TO VC_ADMIN_ROLE
go
GRANT EXECUTE ON msdb.dbo.sp_add_jobserver TO VC_ADMIN_ROLE
go
GRANT EXECUTE ON msdb.dbo.sp_add_jobschedule TO VC_ADMIN_ROLE
go
GRANT EXECUTE ON msdb.dbo.sp_add_category TO VC_ADMIN_ROLE
go
sp_addrolemember VC_ADMIN_ROLE , [vpxuser]
go
ALTER DATABASE VCDB SET RECOVERY SIMPLE
GO
--- VCENTER UPDATE MANAGER DATABASE
use master
go
CREATE DATABASE UMDB ON PRIMARY(
      NAME = 'umdb',
      FILENAME = 'D:\Program Files\Microsoft SQL
Server\MSSQL10_50.MSSQLSERVER\MSSQL\DATA\UMDB.mdf',
      SIZE = 3000KB, FILEGROWTH = 10%)
LOG ON(
      NAME = 'umdb_log',
      FILENAME = 'D:\Program Files\Microsoft SQL
Server\MSSQL10_50.MSSQLSERVER\MSSQL\DATA\UMDB.ldf',
      SIZE = 1000KB, FILEGROWTH = 10%)
COLLATE SQL_Latin1_General_CP1_CI_AS
GO
ALTER DATABASE UMDB SET RECOVERY SIMPLE
GO
---
```

```
ReplaceSSOcerts.cmd
---
@echo off
REM
***********************************************************************************
**********************
REM * Script Name: registerservices.cmd
REM * Script Usage: To register vCentre 5.1 SSO with CA signed certificates
REM * Written by: Simon Mann
REM * Date: 27/09/2012
REM
***********************************************************************************
**********************
REM * Version: 0.1
REM
***********************************************************************************
**********************
REM * Revisions:
REM * 27/09/2012 - [0.1] - Initial script version
REM
***********************************************************************************
**********************

REM Check if the correct number of variables have been passed
REM If not then redirect user to USAGE
IF "%1"=="" GOTO USAGE
IF "%2"=="" GOTO USAGE
IF "%3"=="" GOTO USAGE

REM Declare variables used throughout the script
Set FQDNVCENTRE=%1
Set CERTDIR=%2
Set PASSWORD=%3
Set JAVA_HOME_DIR=C:\Program Files\VMware\Infrastructure\jre
Set SSOCLIDIR="C:\Program Files\VMware\Infrastructure\SSOServer\ssolscli"
Set STSPROPERTIESFILE=sts.properties
Set GCPROPERTIESFILE=gc.properties
Set ADMINPROPERTIESFILE=admin.properties
Set SERVICESLISTFILE=services_list.txt
Set SERVICESFILE=services_file.txt
Set SERVICESIDFILE=services_id.txt
Set STSIDFILE=sts_id
Set GCIDFILE=gc_id
Set ADMINIDFILE=admin_id

REM Echo out the arguments passed to the script
echo INFO: vCentre FQDN is set to %FQDNVCENTRE%
echo INFO: Certificate directory is set to %CERTDIR%
echo INFO: SSOLSCLI directory is set to %SSOCLIDIR%

REM Check if the required certificate files exist
REM If not go to FILEERROR
SETLOCAL ENABLEDELAYEDEXPANSION

Set /a CERTCOUNT=0
IF EXIST %CERTDIR%\rui.crt Set /a CERTCOUNT+=1
IF EXIST %CERTDIR%\rui.key Set /a CERTCOUNT+=1
IF EXIST %CERTDIR%\rui.pfx Set /a CERTCOUNT+=1
IF !CERTCOUNT!==3 GOTO NOFILEERROR

SETLOCAL DISABLEDELAYEDEXPANSION

REM echo error to user and exit script
echo ERROR: Unable to locate required certificate files in %CERTDIR% (rui.crt,
rui.key, rui.pfx)
GOTO ENDWITHERROR

:NOFILEERROR
```

```
echo INFO: Found certificate files in %CERTDIR%
echo INFO: Setting JAVA_HOME to '%JAVA_HOME_DIR%'
REM check if javaw.exe is in the standard java bin directory
IF EXIST "%JAVA_HOME_DIR%\bin\javaw.exe" GOTO FOUNDJAVA
REM If javaw.exe not found then error
echo ERROR: Unable to locate %JAVA_HOME_DIR%\bin\javaw.exe
GOTO ENDWITHERROR

:FOUNDJAVA
echo INFO: Found javaw.exe OK
REM Set JAVA_HOME variable
SET JAVA_HOME=C:\Program Files\VMware\Infrastructure\jre
REM Change to SSOCLIDIR
echo INFO: Changing to %SSOCLIDIR%
cd /d %SSOCLIDIR%
REM  Using SSOLSCLI.cmd query the lookupservice to see what services are listed
echo INFO: Listing all services registered
cmd.exe /c ssolscli.cmd listServices https://%FQDNVCENTRE%:7444/lookupservice/sdk >
%CERTDIR%\%SERVICESLISTFILE%
If %ERRORLEVEL%==0 GOTO NOLISTSERVICESERROR
REM On error exit
echo ERROR: Encountered error %ERRORLEVEL% when running 'ssolscli.cmd listServices
https://%FQDNVCENTRE%:7444/lookupservice/sdk'
GOTO ENDWITHERROR

:NOLISTSERVICESERROR

echo INFO: Successfully ran ssolscli.cmd listServices
https://%FQDNVCENTRE%:7444/lookupservice/sdk

REM Delete any properties files which may have been left from a previous failed run
IF EXIST %CERTDIR%\%STSPROPERTIESFILE% del %CERTDIR%\%STSPROPERTIESFILE%
IF EXIST %CERTDIR%\%GCPROPERTIESFILE% del %CERTDIR%\%GCPROPERTIESFILE%
IF EXIST %CERTDIR%\%ADMINPROPERTIESFILE% del %CERTDIR%\%ADMINPROPERTIESFILE%

echo INFO: Creating %CERTDIR%\%STSPROPERTIESFILE%

REM Create STS properties file
echo [service]>> %CERTDIR%\%STSPROPERTIESFILE%
echo friendlyName=STS for Single Sign On>> %CERTDIR%\%STSPROPERTIESFILE%
echo version=1.0>> %CERTDIR%\%STSPROPERTIESFILE%
echo ownerId=>> %CERTDIR%\%STSPROPERTIESFILE%
echo type=urn:sso:sts>> %CERTDIR%\%STSPROPERTIESFILE%
echo description=The Security Token Service of the Single Sign On server.>>
%CERTDIR%\%STSPROPERTIESFILE%
echo [endpoint0]>> %CERTDIR%\%STSPROPERTIESFILE%
echo uri=https://%FQDNVCENTRE%:7444/ims/STSService>> %CERTDIR%\%STSPROPERTIESFILE%
echo ssl=%CERTDIR%\rui.crt>> %CERTDIR%\%STSPROPERTIESFILE%
echo protocol=wsTrust>> %CERTDIR%\%STSPROPERTIESFILE%

echo INFO: Creating %CERTDIR%\%GCPROPERTIESFILE%

REM Create GC properties file
echo [service]>> %CERTDIR%\%GCPROPERTIESFILE%
echo friendlyName=The group check interface of the SSO server>>
%CERTDIR%\%GCPROPERTIESFILE%
echo version=1.0>> %CERTDIR%\%GCPROPERTIESFILE%
echo ownerId=>> %CERTDIR%\%GCPROPERTIESFILE%
echo type=urn:sso:groupcheck>> %CERTDIR%\%GCPROPERTIESFILE%
echo description=The group check interface of the SSO server>>
%CERTDIR%\%GCPROPERTIESFILE%
echo [endpoint0]>> %CERTDIR%\%GCPROPERTIESFILE%
echo uri=https://%FQDNVCENTRE%:7444/sso-adminserver/sdk>>
%CERTDIR%\%GCPROPERTIESFILE%
echo ssl=%CERTDIR%\rui.crt>> %CERTDIR%\%GCPROPERTIESFILE%
echo protocol=vmomi>> %CERTDIR%\%GCPROPERTIESFILE%
echo INFO: Creating %CERTDIR%\%ADMINPROPERTIESFILE%

REM Create ADMIN properties file
echo [service]>>%CERTDIR%\%ADMINPROPERTIESFILE%
echo friendlyName=The administrative interface of the SSO
server>>%CERTDIR%\%ADMINPROPERTIESFILE%
echo version=1.0>>%CERTDIR%\%ADMINPROPERTIESFILE%
echo ownerId=>>%CERTDIR%\%ADMINPROPERTIESFILE%
echo type=urn:sso:admin>>%CERTDIR%\%ADMINPROPERTIESFILE%
echo description=The administrative interface of the SSO
server>>%CERTDIR%\%ADMINPROPERTIESFILE%
echo [endpoint0]>>%CERTDIR%\%ADMINPROPERTIESFILE%
echo uri=https://%FQDNVCENTRE%:7444/sso-
adminserver/sdk>>%CERTDIR%\%ADMINPROPERTIESFILE%
echo ssl=%CERTDIR%\rui.crt>>%CERTDIR%\%ADMINPROPERTIESFILE%
echo protocol=vmomi>>%CERTDIR%\%ADMINPROPERTIESFILE%

echo INFO: Getting ServiceID from %CERTDIR%\%SERVICESLISTFILE%

REM Delete any files which may exist from a previous failed run
IF EXIST %CERTDIR%\%SERVICESFILE% del %CERTDIR%\%SERVICESFILE%
IF EXIST %CERTDIR%\%SERVICESIDFILE% del %CERTDIR%\%SERVICESIDFILE%

REM Search %SERVICESLISTFILE% for {} to set %SERVICEID%
for /f "tokens=2 delims={}" %%a in ('findstr "serviceId="
%CERTDIR%\%SERVICESLISTFILE%') do (set SERVICEID={%%a})
echo INFO: ServiceID has been set to %SERVICEID%

REM Search %SERVICESLISTFILE% for a list of registered services and output service
name to %SERVICESFILE%
echo INFO: Getting registered services and dumping to %CERTDIR%\%SERVICESFILE%
for /f %%a in ('findstr "type=urn:" %CERTDIR%\%SERVICESLISTFILE%') do echo %%a >>
%CERTDIR%\%SERVICESFILE%

REM Search %SERVICESLISTFILE% for a list of registered services and output service
ID to %SERVICESIDFILE%
echo INFO: Getting registered services ID's and dumping to
%CERTDIR%\%SERVICESIDFILE%
for /f "tokens=2 delims=:" %%a in ('findstr "serviceId="
%CERTDIR%\%SERVICESLISTFILE%') do echo %%a >> %CERTDIR%\%SERVICESIDFILE%


REM Now we have a file with each service name and another file with a service ID so
we need to match them up
Set /a COUNTER=0

REM As we require the COUNTER and COUNTERID variables to increment during a for
loop we ENABLEDELAYEDEXPANSION
SETLOCAL ENABLEDELAYEDEXPANSION

REM Search and find out which line number in %SERVICESFILE% contains
'urn:sso:admin' and find the SERVICEID in %SERVICESIDFILE% for the same line number
for /f %%a in (%CERTDIR%\%SERVICESFILE%) do (set /a COUNTER+=1 & IF
%%a==^type=urn:sso:admin for /f %%b in (%CERTDIR%\%SERVICESIDFILE%) do Set /a
COUNTERID+=1 & IF !COUNTER!==!COUNTERID! SET ADMINID=%%b)
echo INFO: sso:admin is set to %SERVICEID%:%ADMINID%
echo %SERVICEID%:%ADMINID% > %CERTDIR%\%ADMINIDFILE%

REM Search and find out which line number in %SERVICESFILE% contains
'urn:sso:groupcheck' and find the SERVICEID in %SERVICESIDFILE% for the same line
number
for /f %%a in (%CERTDIR%\%SERVICESFILE%) do (set /a COUNTER+=1 & IF
%%a==^type=urn:sso:groupcheck for /f %%b in (%CERTDIR%\%SERVICESIDFILE%) do Set /a
COUNTERID+=1 & IF !COUNTER!==!COUNTERID! SET GROUPCHECKID=%%b)
echo INFO: sso:groupcheck is set to %SERVICEID%:%GROUPCHECKID%
echo %SERVICEID%:%GROUPCHECKID% > %CERTDIR%\%GCIDFILE%
```

```
REM Search and find out which line number in %SERVICESFILE% contains 'urn:sso:sts'
and find the SERVICEID in %SERVICESIDFILE% for the same line number
for /f %%a in (%CERTDIR%\%SERVICESFILE%) do (set /a COUNTER+=1 & IF
%%a==^type=urn:sso:sts for /f %%b in (%CERTDIR%\%SERVICESIDFILE%) do Set /a
COUNTERID+=1 & IF !COUNTER!==!COUNTERID! SET STSID=%%b)
echo INFO: sso:sts is set to %SERVICEID%:%STSID%
echo %SERVICEID%:%STSID% > %CERTDIR%\%STSIDFILE%

REM ENABLEDELAYEDEXPANSION no longer required
SETLOCAL DISABLEDELAYEDEXPANSION

echo INFO: Stopping "vCenter Single Sign On"
REM Stop "vCenter Single Sign On" via sc.exe
cmd.exe /c start /wait sc stop ssotomcat
If %ERRORLEVEL%==0 GOTO NOERRORSTOPPINGSERVICE
REM On error exit
echo ERROR: %ERRORLEVEL% when running 'cmd.exe /c start /wait sc stop ssotomcat'
Please stop this service manually and try again.
GOTO ENDWITHERROR

:NOERRORSTOPPINGSERVICE
echo INFO: Successfully stopped "vCenter Single Sign On"
echo INFO: Updating the certificate in the SSO store
REM Unfortunately, the ssocli configure-riat asks for a password...
echo ********************************************************************
echo INFO: When prompted enter the master password (admin@System-Domain)
echo ********************************************************************
cd /d C:\Program Files\VMware\Infrastructure\SSOServer\utils
cmd.exe /c ssocli configure-riat -a configure-ssl --keystore-file %CERTDIR%\rui.pfx
--keystore-password testpassword
echo %ERRORLEVEL%
If %ERRORLEVEL%==0 GOTO NOERRORSSOUPDATE
REM On error exit
echo ERROR: Failed to run 'cmd.exe /c ssocli configure-riat -a configure-ssl --
keystore-file %CERTDIR%\rui.pfx --keystore-password testpassword'
GOTO ENDWITHERROR

:NOERRORSSOUPDATE
echo INFO: Successfully updated the SSO certificate
REM Start "vCenter Single Sign On"
cmd.exe /c start /wait sc start ssotomcat
If %ERRORLEVEL%==0 echo INFO: Successfully started "vCenter Single Sign On"
echo INFO: Waiting for "vCenter Single Sign On" service to initialise fully
REM If this timeout doesn't exist then the next update commands can sometimes fails
timeout /t 10
cd /d %SSOCLIDIR%

REM Update sts certificate using ssolscli
cmd.exe /c ssolscli.cmd updateService -d
https://%FQDNVCENTRE%:7444/lookupservice/sdk -u admin@System-Domain -p %PASSWORD% -
si %CERTDIR%\%STSIDFILE% -ip %CERTDIR%\%STSPROPERTIESFILE%
If %ERRORLEVEL%==0 GOTO NOERRORSTSUPDATE
echo ERROR: Failed to run 'cmd.exe /c ssolscli.cmd updateService -d
https://%FQDNVCENTRE%:7444/lookupservice/sdk -u admin@System-Domain -p %PASSWORD% -
si %CERTDIR%\%STSIDFILE% -ip %CERTDIR%\%STSPROPERTIESFILE%'
GOTO ENDWITHERROR

:NOERRORSTSUPDATE
echo INFO: Successfully updated the STS certificate
REM Update gc certificate using ssolscli
cmd.exe /c ssolscli.cmd updateService -d
https://%FQDNVCENTRE%:7444/lookupservice/sdk -u admin@System-Domain -p %PASSWORD% -
si %CERTDIR%\%GCIDFILE% -ip %CERTDIR%\%GCPROPERTIESFILE%
If %ERRORLEVEL%==0 GOTO NOERRORGCUPDATE
echo ERROR: Failed to run 'cmd.exe /c ssolscli.cmd updateService -d
https://%FQDNVCENTRE%:7444/lookupservice/sdk -u admin@System-Domain -p %PASSWORD% -
si %CERTDIR%\%GCIDFILE% -ip %CERTDIR%\%GCPROPERTIESFILE%'
GOTO ENDWITHERROR
```

```
:NOERRORGCUPDATE
echo INFO: Successfully updated the GC certificate
REM Update admin certificate using ssolscli
cmd.exe /c ssolscli.cmd updateService -d
https://%FQDNVCENTRE%:7444/lookupservice/sdk -u admin@System-Domain -p %PASSWORD% -
si %CERTDIR%\%ADMINIDFILE% -ip %CERTDIR%\%ADMINPROPERTIESFILE%
If %ERRORLEVEL%==0 GOTO NOERRORGCUPDATE
echo ERROR: Failed to run 'cmd.exe /c ssolscli.cmd updateService -d
https://%FQDNVCENTRE%:7444/lookupservice/sdk -u admin@System-Domain -p %PASSWORD% -
si %CERTDIR%\%ADMINIDFILE% -ip %CERTDIR%\%ADMINPROPERTIESFILE%'
GOTO ENDWITHERROR

:NOERRORGCUPDATE
echo INFO: Successfully updated the ADMIN certificate
GOTO ENDWITHOUTERRORS

:ENDWITHERROR
echo ERROR: Script failed to complete successfully.  Please investigate and run
again.
GOTO END

:ENDWITHOUTERRORS
echo INFO: Successfully changed certificates without any issues
GOTO END

:USAGE
echo ERROR: Wrong number of arguments passed to the script
echo.
echo.
echo                 SCRIPT USAGE
echo.
echo ReplaceSSOCerts.cmd FQDNVCENTRE CERTDIR ADMINPASSWORD
echo.
echo *ADMIN-PASSWORD is the password used for admin@System-Domain
echo.
echo Example:
echo.
echo ReplaceSSOCerts.cmd vcentre01.foo.com c:\temp\certs Pa55w0rd
echo.

:END
---
```

5. Bibliography

Mann, S. (n.d.). *registeredservices.cmd*. Retrieved from http://communities.vmware.com/docs/DOC-20575

Seaman, D. (n.d.). *VMware vCenter 5.1 Installation - (part 1-14)*. Retrieved from http://derek858.blogspot.nl/2012/09/vmware-vcenter-51-installation-part-1.html

VMware. (n.d.). *Configuring CA signed certificates for ESXi 5.x hosts*. Retrieved from http://kb.vmware.com/selfservice/microsites/search.do?cmd=displayKC&externalId=2015499

VMware. (n.d.). *Configuring CA signed SSL certificates for the Inventory service in vCenter Server 5.1*. Retrieved from http://kb.vmware.com/selfservice/microsites/search.do?&cmd=displayKC&externalId=2035009

VMware. (n.d.). *Configuring CA signed SSL certificates for vCenter Server SSO in vCenter Server 5.1*. Retrieved from http://kb.vmware.com/selfservice/microsites/search.do?cmd=displayKC&externalId=2035011

VMware. (n.d.). *Configuring CA signed SSL certificates for vSphere Update Manager in vCenter Server 5.1*. Retrieved from http://kb.vmware.com/selfservice/microsites/search.do?&cmd=displayKC&externalId=2037581

VMware. (n.d.). *Creating certificate requests and certificates for vCenter Server 5.1 components*. Retrieved from http://kb.vmware.com/selfservice/microsites/search.do?cmd=displayKC&externalId=2037432

VMware. (n.d.). *Implementing CA signed SSL certificates with vSphere 5.1*. Retrieved from http://kb.vmware.com/selfservice/microsites/search.do?cmd=displayKC&externalId=2034833