Red Hat Certificate System

Migration Guide: 7.x to 7.3

7.0

Matthew Harmsen

ISBN: N/A

Publication date: March 12, 2008

This migration guide provides in-depth procedures to migrate subsystems, user information, and certificate and key materials from Netscape Certificate Management System 7.0 and Red Hat Certificate System 7.1 and 7.2 to Red Hat Certificate System 7.3.

Red Hat Certificate System: Migration Guide: 7.x to 7.3

Author	Matthew Harmsen	<mharmsen@redhat.com>
Editor	Ella Deon Lackey	<dlackey@redhat.com>

The GPG fingerprint of the security@redhat.com key is:

CA 20 86 86 2B D6 9D FC 65 F6 EC C4 21 91 80 CD DB 42 A6 0E

1801 Varsity Drive
Raleigh, NC 27606-2072
USA
Phone: +1 919 754 3700
Phone: 888 733 4281
Fax: +1 919 754 3701
PO Box 13588
Research Triangle Park, NC 27709
USA

1. Introduction to Red Hat Certificate System Migration 1
1. Certificate System Migration Overview 1
1.1. Migration Scripts 2
1.2. Certificate System Subsystems 3
2. Considerations before Migration 4
2. Step 1: Preparing the Older Server Instance for Migration 7
3. Step 2: Installing the New Certificate System 9
4. Step 3: Stopping the New Certificate System Servers 11
5. Step 4: Migrating Security Databases 13
1. Certificate Authority (CA) Migration 13
1.1. Option 1: Security Databases to Security Databases Migration 13
1.2. Option 2: Security Databases to HSM Migration 15
1.3. Option 3: HSM to Security Databases Migration 18
1.4. Option 4: HSM to HSM Migration 21
2. Data Recovery Manager (DRM) Migration 23
2.1. Option 1: Security Databases to Security Databases Migration 24
2.2. Option 2: Security Databases to HSM Migration 25
2.3. Option 3: HSM to Security Databases Migration 29
2.4. Option 4: HSM to HSM Migration 33
3. Online Certificate Status Protocol Manager (OCSP) Migration 36
3.1. Option 1: Security Databases to Security Databases Migration 36
3.2. Option 2: Security Databases to HSM Migration 38
3.3. Option 3: HSM to Security Databases Migration 41
3.4. Option 4: HSM to HSM Migration 44
4. Token Key Service (TKS) Migration 47
4.1. Option 1: Security Databases to Security Databases Migration 47
4.2. Option 2: Security Databases to HSM Migration 49
4.3. Option 3: HSM to Security Databases Migration 54
4.4. Option 4: HSM to HSM Migration 58
6. Step 5: Migrating Password Cache Data 63
7. Step 6: Migrating Internal Databases 65
8. Step 7: Customizing User Data (Non-Console) 71
9. Step 8: Starting All Certificate System 7.3 Instances 73
10. Step 9: Generate New Certificate System Server Certificates 75
1. Self-Signing an SSL Server Certificate for a CA 75
2. Requesting a New SSL Server Certificate from a Third-Party CA 76
3. Generating a New DRM, OCSP, or TKS SSL Server Certificate 77
11. Step 10: Customizing User Data (Console) 79
12. Step 11: Verifying Migration 81

Introduction to Red Hat Certificate System Migration

Netscape Certificate Management System (CMS) versions 7.0, 7.1, and 7.2 can be migrated to Red Hat Certificate System version 7.3 using the Red Hat Certificate System migration utility.

Certificate System has the ability to extract data from the installation of a previous version and migrate this data to 7.3. Each Certificate System subsystem is migrated independently.

- Product *downgrade*, extracting information from a newer version to import into an older version, is not supported by the migration utility in any version of the Certificate System.
- Certificate System 7.3 does not support in-place upgrades. This is true even when the installations are on the same machine, running the same platform. For example, a Red Hat Certificate System 7.1 installation on a machine named `delta.example.com` running Red Hat Enterprise Linux has to be migrated to Certificate System 7.3 installation, even if the 7.3 installation is on the same machine named `delta.example.com` running Red Hat Enterprise Linux. This is accomplished simply by supplying a different installation directory for each instance.

NOTE

Throughout this manual, all system instances are referred to as *Certificate System*, unless the specific version or product name is required.

1. Certificate System Migration Overview

The migration process is staged to migrate the different subsets of Certificate System information separately. The general process is as follows:

1. Install the Certificate System 7.3 subsystem instances.
2. Migrate the Certificate Management System 7.x security databases, which contains the key and certificate materials for the server, to the Certificate System 7.3 instances.
3. Migrate the password database information for Certificate Management System 7.x to the Certificate System 7.3 password file.
4. Migrate the Certificate Management System 7.x internal databases, which contain user and group entries, to the Certificate System 7.3 server.
5. Migrate customized server configuration data from the 7.x server to Certificate System 7.3.

6. Renew all migrated certificates.

1.1. Migration Scripts

The Certificate System migration utility contains several separate platform-independent tools, but only two are required for migrating a Certificate System installation: one program to convert all of the data in an LDIF that was exported from the 7.x installation into a normalized LDIF text file, and another program to convert the normalized LDIF text file into an LDIF data file that can be imported into the newer Certificate System.

NOTE

The major version number of the migration export/import package is applied to all service packs for that version. (This also applies to installations which contain *hot-fixes*, individual bug fixes made after a service pack is released.)

Certificate System migration export utilities are files named *version*`ToTxt`. For migrating 7.0, 7.1, and 7.2 servers to Certificate System 7.3, there are three files which are used, depending on your 7.x version: `70ToTxt`, `71ToTxt`, and `72ToTxt`. Each export tool contains the following files:

- Two precompiled Java™ classes
- An tool shell script
- An tool batch script
- The Java™ source file that produced the precompiled Java™ classes
- A sample shell script that can compile the source file
- A sample batch script that can compile the source file

The export file to use is determined by the older version of the Certificate System being migrated.

NOTE

Each compilation batch and shell script is dependent on a specific version of the Java software development kit defined in its comments.

Certificate System migration import utilities are files named `TxtTo`*`version`*. To migrate the

Certificate Management System 7.x servers to Certificate System 7.3, use the `TxtTo73` script. The import tool contains the following files:

- Three precompiled Java™ classes
- An tool shell script
- An tool batch script
- The Java™ source file that produced the precompiled Java™ classes
- A sample shell script that can compile the source file
- A sample batch script that can compile the source file

Each compilation batch and shell script is dependent upon a specific version of the Java™ software development kit as defined in the comments.

1.2. Certificate System Subsystems

Certificate System installations may exist on different platforms. Additionally, each Certificate System installation may contain more than one type of subsystem or more than one instance of a type of subsystem. The following subsystems may be present in a Certificate System installation:

- Certificate Authority (CA)
- Data Recovery Manager (DRM)
- Online Certificate Status Protocol (OCSP) Manager
- Registration Authority (RA)
- Token Key Service (TKS)
- Token Processing System (TPS)

The following table defines the platforms and subsystems supported by different versions of Certificate System:

Product (including service packs and hot-fixes)	Subsystems	Platforms
Netscape Certificate Management System 7.0	CA DRM OCSP TKS TPS	Red Hat Enterprise Linux AS 2.1 Solaris 8

Product (including service packs and hot-fixes)	Subsystems	Platforms
Red Hat Certificate System 7.1	CA DRM OCSP TKS TPS	Red Hat Enterprise Linux AS/ES 3 Red Hat Enterprise Linux AS/ES 4 Solaris 9
Red Hat Certificate System 7.2	CA DRM OCSP TKS TPS	Red Hat Enterprise Linux AS/ES 4 (i386) Red Hat Enterprise Linux AS/ES 4 (AMD64, Intel EM64T) Solaris 9 (64-bit)
Red Hat Certificate System 7.3	CA DRM OCSP RA TKS TPS	Red Hat Enterprise Linux AS/ES 4 (i386) Red Hat Enterprise Linux AS/ES 4 (for AMD64 and Intel EM64T) Solaris 9 (64-bit)

Table 1.1. Certificate System Subsystem Types and Platforms

2. Considerations before Migration

Since all migrations are unique to a deployment, it is strongly recommended that the entire migration process be planned in advance. Here are some common issues related to migration.

The Migration Procedure.

- Not all steps in the migration processes apply to every deployment. Follow only the steps which apply to the deployment being migrated.
- Perform all steps in the migration procedure for every instance of each subsystem being migrated. Always follow the steps for migration in the order which they are presented in the manual, even if not every step needs to be performed.
- Perform each subsystem migration as a separate migration. Wait until one subsystem migration is complete before starting the migration of the next subsystem.

Setting File Permissions.

- On Linux and UNIX systems, make sure that the file owner (user and group) and the file permissions are correct when the file is copied between two instances. Also make sure that the target machine allows the file transfer.
- The `chmod` command used in the examples have the permissions `00600` (octal, no sticky bit permissions, user read/write permissions, no group permissions, no other permissions). These are the recommended permissions, but are not required.

Extracting Data from a Hardware Security Module.

While the migration procedure refers to extracting data from a hardware security module (HSM), no Certificate System tool can extract public/private key pairs from an HSM because of Federal Information Processing Standard (FIPS) 140-1, which protects the private key portion of an entry. Contact the HSM vendor for information on how to extract data from an old HSM. Extracted keys should not have any dependencies, such as nickname prefixes, on the old HSM.

Changing Subsystem Names and Port Numbers.

It is possible to change the names of migrated Certificate System subsystem instances, but greater care must be taken when extracting and renaming certain portions of the data. Because port numbers are stored in the `server.xml` file, which is unaffected by subsystem migration, port numbers can be changed between instances without difficulty.

About Usage Examples.

All examples assume that the new passwords are the same as the old passwords.

Subsystem and Version Related Considerations.

- It is not possible to migrate TPS data from Netscape Certificate Management System 7.0 to Red Hat Certificate System 7.3.
- The default key-splitting scheme used by the DRM subsystem in Certificate System 7.1 and later is not the scheme required by the DRM subsystem key recovery feature in Netscape Certificate Management System 7.0. There is no way to migrate from the old key-splitting scheme to the new scheme. Therefore, DRM instances in Certificate Management System 7.0 cannot be successfully migrated to version 7.3.

Step 1: Preparing the Older Server Instance for Migration

Before migrating a Certificate System instance, back up the Certificate System instance. When the backup process is complete, then make sure that the old Certificate System, Directory Server, and Administration Server instances have been stopped.

For commercial backup programs, do the following:

1. Stop all server instances running in the Certificate System, including all Certificate System subsystems; the Directory Server, including all internal databases; and the Administration Server instances on the local machine.
2. Run the commercial backup facility to back up all data associated with the old Certificate System installation.

For the Certificate System backup facilities, do the following:

NOTE

Back up utilities are not included on Certificate System versions 7.2 and later, but are available on previous versions of Certificate System.

1. Run the Certificate System backup facility to back up all data associated with this old installation based upon the version information below.
   - Certificate Management System 4.1, 4.2, 4.2 (SP 2), or 4.5

     For instructions to back up a Certificate Management System 4.1, 4.2, 4.2 (SP 2), or 4.5 instance, see the Certificate Management System 4.x *Certificate Management System Command-Line Tools Guide*. This information is located in the *server_root*`/manual/en/cert/tools/backup.htm` file.
   - iPlanet Certificate Management System 4.7

     For instructions to back up an iPlanet Certificate Management System 4.7 instance, refer to the documentation provided with the product.
   - Netscape Certificate Management System 6.0, 6.1, 6.2, or 7.0 and Red Hat Certificate System 7.1

     For instructions to back up a versions 6.0, 6.1, 6.2, 7.0, or 7.1, see chapter 8, "Backing Up and Restoring Data," in the Certificate Management System 6.x *Certificate Management*

System Command-Line Tools Guide.

2. Stop all server instances running in the Certificate System, including all Certificate System subsystems; the Directory Server, including all internal databases; and the Administration Server instances on the local machine.

Additionally, RA subsystems in older versions of Certificate System cannot be migrated to Certificate System Red Hat Certificate System 7.3. No upgrade path exists for the Registration Authority (RA) subsystem because this subsystem is no longer supported. To preserve the data in the RA subsystem, flush all data in RA request queues by processing it on the CA subsystem. Then stop the RA subsystem and database so that no further data can be gathered by it.

1. Stop the RA instance.

```
cd /usr/netscape/servers/cert-ra

./stop-cert
```

2. Stop the corresponding RA internal database instance.

```
cd /usr/netscape/servers/slapd-ra-db

./stop-slapd
```

Step 2: Installing the New Certificate System

Install a new Certificate System 7.3 instance. All subsystem instances are installed separately; make sure that every subsystem type which will be migrated has a corresponding new subsystem instance.

1. Obtain the appropriate packages either through the `up2date` command or by downloading the ISO image from the Certificate System 7.3 Red Hat Network channel.
2. If installing from an ISO image, run the installation utility to install the packages. This is done automatically when using `up2date`.

```
rhpki-install -pki_subsystem=subsystem_type -pki_package_path=/path/to/ISO
image -force
```

3. Go through the HTML configuration wizard for each subsystem.

   When the installation process is completed, the server returns a URL pointing to the configuration wizard. Click that link to open the configuration wizard. For example:

```
http://server.example.com:9080/ca/admin/console/config/login?pin=Yc6EuvuY2OeezKeX7REk
```

   The configuration wizard will fully configure the new subsystem instance and will generate all required certificates. Make sure to have all necessary information when going through this wizard. All subsystems require information to an external Red Hat Directory Server, including bind information. DRM, OCSP, TKS, and subordinate CAs require information for the CA which will generate their subsystem certificates.

NOTE

It is possible to change the names of migrated Certificate System subsystem instances, but greater care must be taken when extracting and renaming certain portions of the data. Because port numbers are stored in the `server.xml` file, which is unaffected by subsystem migration, port numbers can be changed between instances without difficulty.

   For more information on the panels in the configuration wizard, see chapter 2, "Installation and Configuration," in the *Certificate System Administrator's Guide*.

Step 3: Stopping the New Certificate System Servers

1. First, stop all new Certificate System instances.

```
/etc/init.d/instance_ID stop
```

2. Then stop the Directory Server instance used by the Certificate System 7.3 servers.

```
cd /opt/redhat-ds/slapd-DS-instance

./stop-slapd
```

Step 4: Migrating Security Databases

For every Red Hat Certificate System subsystem instance migration, the data from the certificate (`cert8.db`) and key (`key3.db`) security databases for the Netscape Certificate Management System 7.0 or Red Hat Certificate System 7.1 or 7.2 instances must be extracted and copied into the Red Hat Certificate System 7.3 subsystem's `/alias` directory. Follow the migration procedure corresponding to the subsystem being migrated.

Four subsystems can be migrated from Certificate Management System 7.0 and Certificate System 7.1 and 7.2 to Certificate System 7.3 — the Certificate Authority (CA), the Data Recovery Manager (DRM), the Online Certificate Status Protocol Manager (OCSP), and the Token Key Service (TKS) — each with a different migration procedure.

- *Section 1, “Certificate Authority (CA) Migration”*
- *Section 2, “Data Recovery Manager (DRM) Migration”*
- *Section 3, “Online Certificate Status Protocol Manager (OCSP) Migration”*
- *Section 4, “Token Key Service (TKS) Migration”*

1. Certificate Authority (CA) Migration

Determine if the migration to be performed involves software security databases, an HSM, or both, and follow the appropriate process for the deployment scenario being migrated.

- *Section 1.1, “Option 1: Security Databases to Security Databases Migration”*
- *Section 1.2, “Option 2: Security Databases to HSM Migration”*
- *Section 1.3, “Option 3: HSM to Security Databases Migration”*
- *Section 1.4, “Option 4: HSM to HSM Migration”*

1.1. Option 1: Security Databases to Security Databases Migration

1. Remove all the security databases in the Certificate System 7.3 server which will receive migrated data.

```
rm /var/lib/instance_ID/alias/cert8.db

rm /var/lib/instance_ID/alias/key3.db
```

2. Copy the certificate and key security databases from the 7.x server to the 7.3 server.

```
cp old_server_root/alias/cert-old_CA_instance-cert8.db
/var/lib/instance_ID/alias/cert8.db

cp old_server_root/alias/cert-old_CA_instance-key3.db
/var/lib/instance_ID/alias/key3.db
```

3. Open the Certificate System `/alias` directory.

```
cd /var/lib/instance_ID/alias/
```

4. Log into the 7.3 server as `root`.

5. Set the file user and group to the Certificate System user and group.

```
# chown user:group cert8.db

# chown user:group key3.db
```

6. Log out as `root` and change to the Certificate System user login.

7. Set the file permissions.

```
chmod 00600 cert8.db

chmod 00600 key3.db
```

8. List the contents of the certificate database using the `certutil` tool. In this example, `-L` lists the certificates in the database.

```
certutil -L -d .

Server-Cert cert-old_CA_instance cu,cu,cu
caSigningCert cert-old_CA_instance cu,cu,cu
ocspSigningCert cert-old_CA_instance CTu,Cu,Cu
```

   subsystemCert cert-*old_CA_instance* cu,cu,cu

9. Open the `CS.cfg` configuration file in the `/var/lib/`*instance_ID*`/conf/` directory.

10. Edit the `ca.signing.cacertnickname` and `ca.ocsp_signing.cacertnickaname` attributes to reflect the 7.3 CA instance.

```
ca.signing.cacertnickname=caSigningCert cert-old_CA_instance
ca.ocsp_signing.cacertnickname=ocspSigningCert cert-old_CA_instance
```

11. If there is CA-DRM connectivity, then also modify the `ca.connector.KRA.nickname` attribute.

```
ca.connector.KRA.nickname=caSigningCert cert-old_CA_instance
```

12. In the same directory, edit the `serverCertNick.conf` file to contain the old certificate nickname. For example:

```
Server-Cert cert-old_CA_instance
```

1.2. Option 2: Security Databases to HSM Migration

1. Remove all the security databases in the Certificate System 7.3 which will receive migrated data.

```
rm /var/lib/instance_ID/alias/cert8.db

rm /var/lib/instance_ID/alias/key3.db
```

2. Copy the certificate and key security databases from the 7.x server to the 7.3 server.

```
cp old_server_root/alias/cert-old_CA_instance-cert8.db
/var/lib/instance_ID/alias/cert8.db

cp old_server_root/alias/cert-old_CA_instance-key3.db
/var/lib/instance_ID/alias/key3.db
```

3. Open the Certificate System `/alias` directory.

```
cd /var/lib/instance_ID/alias/
```

4. Log in as `root`.

5. Set the file user and group to the Certificate System user and group.

```
# chown user:group cert8.db

# chown user:group key3.db
```

6. Log out as `root`. As the Certificate System user, set the file permissions.

```
chmod 00600 cert8.db
```

```
chmod 00600 key3.db
```

7. List the certificates stored in the 7.x security databases by using the `certutil` command; `-L` lists the certificates.

```
certutil -L -d .

Server-Cert cert-old_CA_instance cu,cu,cu
caSigningCert cert-old_CA_instance cu,cu,cu
ocspSigningCert cert-old_CA_instance CTu,Cu,Cu
subsystemCert cert-old_CA_instance cu,cu,cu
```

8. Export the public/private key pairs of each entry in the Certificate System databases using the `pk12util` tool; `-o` exports the key pairs to file, and `-n` sets the name of the certificate and the old database prefix.

```
pk12util -o ServerCert.p12 -n "Server-Cert cert-old_CA_instance" -d .
 Enter Password or Pin for "NSS Certificate DB":********
 Enter password for PKCS12 file: ********
 Re-enter password: ********
 pk12util: PKCS12 EXPORT SUCCESSFUL

 pk12util -o caSigningCert.p12 -n "caSigningCert cert-old_CA_instance" -d .
 Enter Password or Pin for "NSS Certificate DB":********
 Enter password for PKCS12 file: ********
 Re-enter password: ********
 pk12util: PKCS12 EXPORT SUCCESSFUL

 pk12util -o ocspSigningCert.p12 -n "ocspSigningCert cert-old_CA_instance"
-d .
 Enter Password or Pin for "NSS Certificate DB":********
 Enter password for PKCS12 file: ********
 Re-enter password: ********
 pk12util: PKCS12 EXPORT SUCCESSFUL

 pk12util -o subsystemCert.p12 -n "subsystemCert cert-old_CA_instance" -d .
 Enter Password or Pin for "NSS Certificate DB":********
 Enter password for PKCS12 file: ********
 Re-enter password: ********
 pk12util: PKCS12 EXPORT SUCCESSFUL
```

NOTE

The 7.x security databases may contain additional public/private key pairs; these can also be extracted using `pk12util`.

9. Delete the 7.x security databases.

```
rm cert8.db

rm key3.db
```

10. Register the new HSM in the 7.3 token database.

```
modutil -nocertdb -dbdir . -add new_HSM_token_name -libfile
new_HSM_library_path/new_HSM_library
```

11. Identify the new HSM slot name.

```
modutil -dbdir . -nocertdb -list
```

12. Create new security databases.

```
certutil -N -d .
```

13. Import the public/private key pairs of each entry from the PKCS #12 files into the new HSM.

```
pk12util -i ServerCert.p12 -d . -h new_HSM_slot_name
 Enter Password or Pin for "new_HSM_slot_name":********
 Enter password for PKCS12 file: ********
 pk12util: PKCS12 IMPORT SUCCESSFUL

 pk12util -i caSigningCert.p12 -d . -h new_HSM_slot_name
 Enter Password or Pin for "new_HSM_slot_name":********
 Enter password for PKCS12 file: ********
 pk12util: PKCS12 IMPORT SUCCESSFUL

 pk12util -i ocspSigningCert.p12 -d . -h new_HSM_slot_name
 Enter Password or Pin for "new_HSM_slot_name":********
 Enter password for PKCS12 file: ********
 pk12util: PKCS12 IMPORT SUCCESSFUL

 pk12util -i subsystemCert.p12 -d . -h new_HSM_slot_name
 Enter Password or Pin for "new_HSM_slot_name":********
 Enter password for PKCS12 file: ********
 pk12util: PKCS12 IMPORT SUCCESSFUL
```

14. Optionally, delete the PKCS #12 files.

```
rm ServerCert.p12

rm caSigningCert.p12

rm ocspSigningCert.p12
```

```
rm subsystemCert.p12
```

15. Set the trust bits on the public/private key pairs that were imported into the new HSM.

```
certutil -M -n "new_HSM_slot_name:Server-Cert cert-old_CA_instance"
     -t "cu,cu,cu" -d . -h new_HSM_token_name

certutil -M -n "new_HSM_slot_name:caSigningCert cert-old_CA_instance"
     -t "CTu,CTu,CTu" -d . -h new_HSM_token_name

certutil -M -n "new_HSM_slot_name:ocspSigningCert cert-old_CA_instance"
     -t "CTu,Cu,Cu" -d . -h new_HSM_token_name

certutil -M -n "new_HSM_slot_name:subsystemCert cert-old_CA_instance"
     -t "cu,cu,cu" -d . -h new_HSM_token_name
```

16. Open the `CS.cfg` configuration file in the `/var/lib/`*instance_ID*`/conf/` directory.

17. Edit the `ca.signing.cacertnickaname` and `ca.ocsp_signing.cacertnickname` attributes to reflect the 7.3 CA instance.

```
ca.signing.cacertnickname=new_HSM_slot_name:caSigningCert
cert-old_CA_instance
ca.ocsp_signing.cacertnickname=new_HSM_slot_name:ocspSigningCert
cert-old_CA_instance
```

18. If there is CA-DRM connectivity, then also modify the `ca.connector.KRA.nickname` attribute.

```
ca.connector.KRA.nickname=new_HSM_slot_name:caSigningCert
cert-old_CA_instance
```

19. In the same directory, edit the `serverCertNick.conf` file to contain the old certificate nickname. For example:

```
new_HSM_slot_name:Server-Cert cert-old_CA_instance
```

1.3. Option 3: HSM to Security Databases Migration

1. Extract the public/private key pairs from the HSM. The format for the extracted key pairs should be portable, such as a PKCS #12 file.

The `pk12util` tool provided by Certificate System cannot extract public/private key pairs from an HSM because of requirements in the FIPS 140-1 standard which protect the private key. To extract this information, contact the HSM vendor. The extracted keys should not have any dependencies, such as nickname prefixes, on the HSM.

2. Copy the extracted key pairs from the 7.x server to the 7.3 server.

```
cp old_server_root/alias/ServerCert.p12
/var/lib/instance_ID/alias/ServerCert.p12

cp old_server_root/alias/caSigningCert.p12
/var/lib/instance_ID/alias/caSigningCert.p12

cp old_server_root/alias/ocspSigningCert.p12
/var/lib/instance_ID/alias/ocspSigningCert.p12

cp old_server_root/alias/subsystemCert.p12
/var/lib/instance_ID/alias/subsystemCert.p12
```

3. Open the Certificate System `/alias` directory.

```
cd /var/lib/instance_ID/alias/
```

4. Log in as `root`.

5. Set the file user and group to the Certificate System user and group.

```
# chown user:group ServerCert.p12

# chown user:group caSigningCert.p12

# chown user:group ocspSigningCert.p12

# chown user:group subsystemCert.p12
```

6. Log out as `root`, and log back into the system as the Certificate System user.

7. Set the file permissions.

```
chmod 00600 ServerCert.p12

chmod 00600 caSigningCert.p12

chmod 00600 ocspSigningCert.p12

chmod 00600 subsystemCert.p12
```

8. Import the public/private key pairs of each entry from the PKCS #12 files into the 7.3 security

databases.

```
pk12util -i ServerCert.p12 -d .
 Enter Password or Pin for "NSS Certificate DB":********
 Enter password for PKCS12 file: ********
 pk12util: PKCS12 IMPORT SUCCESSFUL

 pk12util -i caSigningCert.p12 -d .
 Enter Password or Pin for "NSS Certificate DB":********
 Enter password for PKCS12 file: ********
 pk12util: PKCS12 IMPORT SUCCESSFUL

 pk12util -i ocspSigningCert.p12 -d .
 Enter Password or Pin for "NSS Certificate DB":********
 Enter password for PKCS12 file: ********
 pk12util: PKCS12 IMPORT SUCCESSFUL

 pk12util -i subsystemCert.p12 -d .
 Enter Password or Pin for "NSS Certificate DB":********
 Enter password for PKCS12 file: ********
 pk12util: PKCS12 IMPORT SUCCESSFUL
```

9. Optionally, delete the PKCS #12 files.

```
rm ServerCert.p12

rm caSigningCert.p12

rm ocspSigningCert.p12

rm subsystemCert.p12
```

10. Set the trust bits on the public/private key pairs that were imported into the 7.3 security databases.

```
certutil -M -n "Server-Cert cert-old_CA_instance" -t "cu,cu,cu" -d .

certutil -M -n "caSigningCert cert-old_CA_instance" -t "CTu,CTu,CTu" -d .

certutil -M -n "ocspSigningCert cert-old_CA_instance" -t "CTu,Cu,Cu" -d .

certutil -M -n "subsystemCert cert-old_CA_instance" -t "cu,cu,cu" -d .
```

11. Open the `CS.cfg` configuration file in the `/var/lib/`*instance_ID*`/conf/` directory.

12. Edit the `ca.signing.cacertnickname` and `ca.ocsp_signing.cacertnickname` attributes to reflect the 7.3 CA instance.

```
ca.signing.cacertnickname=caSigningCert cert-old_CA_instance
ca.ocsp_signing.cacertnickname=ocspSigningCert cert-old_CA_instance
```

13. If there is CA-DRM connectivity, then also modify the `ca.connector.KRA.nickname` attribute.

```
ca.connector.KRA.nickname=caSigningCert cert-old_CA_instance
```

14. In the same directory, edit the `serverCertNick.conf` file to contain the old certificate nickname. For example:

```
Server-Cert cert-old_CA_instance
```

1.4. Option 4: HSM to HSM Migration

1. Extract the public/private key pairs from the HSM. The format for the extracted key pairs should be portable, such as a PKCS #12 file.

   The `pk12util` tool provided by Certificate System cannot extract public/private key pairs from an HSM because of requirements in the FIPS 140-1 standard which protect the private key. To extract this information, contact the HSM vendor. The extracted keys should not have any dependencies, such as nickname prefixes, on the HSM.

2. Copy the extracted key pairs from the 7.x server to the 7.3 server.

```
cp old_server_root/alias/ServerCert.p12
/var/lib/instance_ID/alias/ServerCert.p12

cp old_server_root/alias/caSigningCert.p12
/var/lib/instance_ID/alias/caSigningCert.p12

cp old_server_root/alias/ocspSigningCert.p12
/var/lib/instance_ID/alias/ocspSigningCert.p12

cp old_server_root/alias/subsystemCert.p12
/var/lib/instance_ID/alias/subsystemCert.p12
```

3. Open the Certificate System `/alias` directory.

```
cd /var/lib/instance_ID/alias/
```

4. Log in as `root`.

5. Set the file user and group to the Certificate System user and group.

```
# chown user:group ServerCert.p12

# chown user:group caSigningCert.p12
```

```
# chown user:group ocspSigningCert.p12
```

6. Log out as `root`, and log back into the system as the Certificate System user.

7. Set the file permissions.

```
chmod 00600 ServerCert.p12

chmod 00600 caSigningCert.p12

chmod 00600 ocspSigningCert.p12

chmod 00600 subsystemCert.p12
```

8. Register the new HSM in the 7.3 token database.

```
modutil -nocertdb -dbdir . -add new_HSM_token_name -libfile
     new_HSM_library_path/new_HSM_library
```

9. Identify the new HSM slot name.

```
modutil -dbdir . -nocertdb -list
```

10. Import the public/private key pairs of each entry from the PKCS #12 files into the new HSM.

```
pk12util -i ServerCert.p12 -d . -h new_HSM_slot_name
 Enter Password or Pin for "new_HSM_slot_name":********
 Enter password for PKCS12 file: ********
 pk12util: PKCS12 IMPORT SUCCESSFUL

 pk12util -i caSigningCert.p12 -d . -h new_HSM_slot_name
 Enter Password or Pin for "new_HSM_slot_name":********
 Enter password for PKCS12 file: ********
 pk12util: PKCS12 IMPORT SUCCESSFUL

 pk12util -i ocspSigningCert.p12 -d . -h new_HSM_slot_name
 Enter Password or Pin for "new_HSM_slot_name":********
 Enter password for PKCS12 file: ********
 pk12util: PKCS12 IMPORT SUCCESSFUL

 pk12util -i subsystemCert.p12 -d . -h new_HSM_slot_name
 Enter Password or Pin for "new_HSM_slot_name":********
 Enter password for PKCS12 file: ********
 pk12util: PKCS12 IMPORT SUCCESSFUL
```

11. Optionally, delete the PKCS #12 files.

```

```

```
rm ServerCert.p12

rm caSigningCert.p12

rm ocspSigningCert.p12

rm subsystemCert.p12
```

12. Set the trust bits on the public/private key pairs that were imported into the new HSM.

```
certutil -M -n "new_HSM_slot_name:Server-Cert cert-old_CA_instance"
    -t "cu,cu,cu" -d . -h new_HSM_token_name

certutil -M -n "new_HSM_slot_name:caSigningCert cert-old_CA_instance"
    -t "CTu,CTu,CTu" -d . -h new_HSM_token_name

certutil -M -n "new_HSM_slot_name:ocspSigningCert cert-old_CA_instance"
    -t "CTu,Cu,Cu" -d . -h new_HSM_token_name

certutil -M -n "new_HSM_slot_name:subsystemCert cert-old_CA_instance"
    -t "cu,cu,cu" -d . -h new_HSM_token_name
```

13. Open the `CS.cfg` configuration file in the `/var/lib/`*instance_ID*`/conf/` directory.

14. Edit the `ca.signing.cacertnickname` and `ca.ocsp_cacertnickname` attributes to reflect the 7.3 CA instance.

```
ca.signing.cacertnickname=new_HSM_slot_name:caSigningCert
cert-old_CA_instance
ca.ocsp_signing.cacertnickname=new_HSM_slot_name:ocspSigningCert
cert-old_CA_instance
```

15. If there is CA-DRM connectivity, then also modify the `ca.connector.KRA.nickname` attribute.

```

ca.connector.KRA.nickname=new_HSM_slot_name:caSigningCert
cert-old_CA_instance
```

16. In the same directory, edit the `serverCertNick.conf` file to contain the old certificate nickname. For example:

```
new_HSM_slot_name:Server-Cert cert-old_CA_instance
```

2. Data Recovery Manager (DRM) Migration

Determine if the migration to be performed involves software security databases, an HSM, or both, and follow the appropriate process for the deployment scenario being migrated.

- *Section 2.1, "Option 1: Security Databases to Security Databases Migration"*
- *Section 2.2, "Option 2: Security Databases to HSM Migration"*
- *Section 2.3, "Option 3: HSM to Security Databases Migration"*
- *Section 2.4, "Option 4: HSM to HSM Migration"*

NOTE

Archived keys stored in a 7.0 DRM cannot be migrated to Certificate System 7.3 because the old key-splitting scheme is not supported in Certificate System 7.3. To be able to recover these keys, obtain a migration patch from Red Hat services.

2.1. Option 1: Security Databases to Security Databases Migration

1. Remove all the security databases in the Certificate System 7.3 server which will receive migrated data.

```
rm /var/lib/instance_ID/alias/cert8.db

rm /var/lib/instance_ID/alias/key3.db
```

2. Copy the certificate and key security databases from the 7.x server to the 7.3 server.

```
cp old_server_root/alias/cert-old_DRM_instance-cert8.db
/var/lib/instance_ID/alias/cert8.db

cp old_server_root/alias/cert-old_DRM_instance-key3.db
/var/lib/instance_ID/alias/key3.db
```

3. Open the Certificate System `/alias` directory.

```
cd /var/lib/instance_ID/alias/
```

4. Log in as `root`.

5. Set the file user and group to the Certificate System user and group.

```
# chown user:group cert8.db

# chown user:group key3.db
```

6. Log out as `root`, and log back into the system as the Certificate System user.

7. Set the file permissions.

```
chmod 00600 cert8.db

chmod 00600 key3.db
```

8. List the certificates in the 7.x security databases by using the `certutil` tool; `-L` lists the certificates.

```
certutil -L -d .

Server-Cert cert-old_DRM_instance cu,cu,cu
caSigningCert cert-old_DRM_instance CT,c,
kraStorageCert cert-old_DRM_instance u,u,u
kraTransportCert cert-old_DRM_instance u,u,u
```

9. Open the `CS.cfg` configuration file in the `/var/lib/`*instance_ID*`/conf/` directory.

10. Edit the `kra.storageUnit.nickname` and `kra.transportUnit.nickname` attribute to reflect the 7.3 DRM instance.

```
kra.storageUnit.nickname=kraStorageCert cert-old_DRM_instance
kra.transportUnit.nickname=kraTransportCert cert-old_DRM_instance
```

NOTE

The `caSigningCert` is not referenced in the `CS.cfg` file.

11. In the same directory, edit the `serverCertNick.conf` file to contain the old certificate nickname. For example:

```
Server-Cert cert-old_DRM_instance
```

2.2. Option 2: Security Databases to HSM Migration

1. Remove all the security databases in the Certificate System 7.3 which will receive migrated data.

```
rm /var/lib/instance_ID/alias/cert8.db

rm /var/lib/instance_ID/alias/key3.db
```

2. Copy the certificate and key security databases from the 7.x server to the 7.3 server.

```
cp old_server_root/alias/cert-old_DRM_instance-cert8.db
/var/lib/instance_ID/alias/cert8.db

cp old_server_root/alias/cert-old_DRM_instance-key3.db
/var/lib/instance_ID/alias/key3.db
```

3. Open the Certificate System `/alias` directory.

```
cd /var/lib/instance_ID/alias/
```

4. Log in as `root`.

5. Set the file user and group to the Certificate System user and group.

```
# chown user:group cert8.db

# chown user:group key3.db
```

6. Log out as `root`, and log back into the system as the Certificate System user.

7. Set the file permissions.

```
chmod 00600 cert8.db

chmod 00600 key3.db
```

8. List the certificates stored in the 7.x security databases by using the `certutil` command; `-L` lists the certificates.

```
certutil -L -d .

Server-Cert cert-old_DRM_instance cu,cu,cu
caSigningCert cert-old_DRM_instance cT,c,
kraStorageCert cert-old_DRM_instance u,u,u
kraTransportCert cert-old_DRM_instance u,u,u
```

9. Export the public/private key pairs of each entry in the Certificate System databases using

the `pk12util` tool; `-o` exports the key pairs to a PKCS #12 file, and `-n` sets the name of the certificate and the old database prefix.

```
pk12util -o ServerCert.p12 -n "Server-Cert cert-old_DRM_instance" -d .
Enter Password or Pin for "NSS Certificate DB":********
Enter password for PKCS12 file: ********
Re-enter password: ********
pk12util: PKCS12 EXPORT SUCCESSFUL

 pk12util -o kraStorageCert.p12 -n "kraStorageCert cert-old_DRM_instance" -d
.
Enter Password or Pin for "NSS Certificate DB":********
Enter password for PKCS12 file: ********
Re-enter password: ********
pk12util: PKCS12 EXPORT SUCCESSFUL

 pk12util -o kraTransportCert.p12 -n "kraTransportCert
cert-old_DRM_instance" -d .
Enter Password or Pin for "NSS Certificate DB":********
Enter password for PKCS12 file: ********
Re-enter password: ********
pk12util: PKCS12 EXPORT SUCCESSFUL
```

NOTE

The 7.x security databases may contain additional public/private key pairs; these can also be extracted using `pk12util`.

10 Export the public key using the `certutil` tool; `-L` lists the named certificate, `-n` sets the name of the file and the old prefix, and `-a` outputs the information to a base-64 file.

```
certutil -L -n "caSigningCert cert-old_DRM_instance" -d . -a >
caSigningCert.b64
```

NOTE

The 7.x security databases may contain additional public keys; these can also be extracted using `certutil`.

11 Delete the 7.x security databases.

```
rm cert8.db

rm key3.db
```

12. Register the new HSM in the 7.3 token database.

```
modutil -nocertdb -dbdir . -add new_HSM_token_name -libfile
new_HSM_library_path/new_HSM_library
```

13. Identify the new HSM slot name.

```
modutil -dbdir . -nocertdb -list
```

14. Create new security databases.

```
certutil -N -d .
```

15. Import the public/private key pairs of each entry from the PKCS #12 files into the new HSM.

```
pk12util -i ServerCert.p12 -d . -h new_HSM_slot_name
Enter Password or Pin for "new_HSM_slot_name":********
Enter password for PKCS12 file: ********
pk12util: PKCS12 IMPORT SUCCESSFUL

 pk12util -i kraStorageCert.p12 -d . -h new_HSM_slot_name
Enter Password or Pin for "new_HSM_slot_name":********
Enter password for PKCS12 file: ********
pk12util: PKCS12 IMPORT SUCCESSFUL

 pk12util -i kraTransportCert.p12 -d . -h new_HSM_slot_name
Enter Password or Pin for "new_HSM_slot_name":********
Enter password for PKCS12 file: ********
pk12util: PKCS12 IMPORT SUCCESSFUL
```

16. Optionally, delete the PKCS #12 files.

```
rm ServerCert.p12

rm kraStorageCert.p12

rm kraTransportCert.p12
```

17. Set the trust bits on the public/private key pairs that were imported into the new HSM.

```
certutil -M -n "new_HSM_slot_name:Server-Cert cert-old_DRM_instance"
     -t "cu,cu,cu" -d . -h new_HSM_token_name
```

```
certutil -M -n "new_HSM_slot_name:kraStorageCert cert-old_DRM_instance"
     -t "u,u,u" -d . -h new_HSM_token_name

certutil -M -n "new_HSM_slot_name:kraTransportCert cert-old_DRM_instance"
     -t "u,u,u" -d . -h new_HSM_token_name
```

18. Import the public key from the base-64 file into the new HSM, and set the trust bits.

```
certutil -A -n "new_HSM_slot_name:caSigningCert cert-old_DRM_instance"
      -t "CT,c," -d . -h new_HSM_token_name -i caSigningCert.b64
```

19. Optionally, delete the base-64 file.

```
rm caSigningCert.b64
```

20. Open the `CS.cfg` configuration file in the `/var/lib/`*instance_ID*`/conf/` directory.

21. Edit the `kra.storageUnit.nickname` and `kra.transportUnit.nickname` attributes to reflect the 7.3 DRM information.

```
kra.storageUnit.nickname=new_HSM_slot_name:kraStorageCert
cert-old_DRM_instance
kra.transportUnit.nickname=new_HSM_slot_name:kraTransportCert
cert-old_DRM_instance
```

NOTE

The `caSigningCert` is not referenced in the `CS.cfg` file.

22. In the same directory, edit the `serverCertNick.conf` file to contain the old certificate nickname. For example:

```
new_HSM_slot_name:Server-Cert cert-old_DRM_instance
```

2.3. Option 3: HSM to Security Databases Migration

1. Extract the public/private key pairs from the HSM. The format for the extracted key pairs should be portable, such as a PKCS #12 file.

   The `pk12util` tool provided by Certificate System cannot extract public/private key pairs from an HSM because of requirements in the FIPS 140-1 standard which protect the private

key. To extract this information, contact the HSM vendor. The extracted keys should not have any dependencies, such as nickname prefixes, on the HSM.

2. Copy the extracted key pairs from the 7.x server to the 7.3 server.

```
cp old_server_root/alias/ServerCert.p12
/var/lib/instance_ID/alias/ServerCert.p12

cp old_server_root/alias/kraStorageCert.p12
/var/lib/instance_ID/alias/kraStorageCert.p12

cp old_server_root/alias/kraTransportCert.p12
/var/lib/instance_ID/alias/kraTransportCert.p12
```

3. Extract the public key of the CA signing certificate from the 7.x security databases and save the base-64 encoded output to a file called `caSigningCert.b64`.

   a. Open the Certificate Management System 7.x `/alias` directory.

```
cd old_server_root/alias
```

   b. Set the `LD_LIBRARY_PATH` environment variable to search the Certificate System libraries.

```
LD_LIBRARY_PATH=old_server_root/bin/cert/lib

export LD_LIBRARY_PATH
```

   c. Use the Certificate Management System 7.x `certutil` tool to identify the old HSM slot name.

```
old_server_root/bin/cert/tools/certutil -U -d .
```

   d. Use the Certificate Management System 7.x `certutil` tool to extract the public key from the security databases and save the base-64 output to a file.

```
old_server_root/bin/cert/tools/certutil -L
     -n "old_HSM_slot_name:caSigningCert cert-old_DRM_instance"
     -d . -h old_HSM_token_name -a > caSigningCert.b64
```

   e. Copy the key information from the 7.x server to the 7.3 server.

```
cp old_server_root/alias/caSigningCert.b64
/var/lib/instance_ID/alias/caSigningCert.b64
```

4. Open the Certificate System `/alias` directory.

```
cd /var/lib/instance_ID/alias/
```

5. Log in as `root`.

6. Set the file user and group to the Certificate System user and group.

```
# chown user:group ServerCert.p12

# chown user:group kraStorageCert.p12

# chown user:group kraTransportCert.p12

# chown user:group caSigningCert.b64
```

7. Log out as `root`, and log back into the system as the Certificate System user.

8. Set the file permissions.

```
chmod 00600 ServerCert.p12

chmod 00600 kraStorageCert.p12

chmod 00600 kraTransportCert.p12

chmod 00600 caSigningCert.b64
```

9. Import the public/private key pairs of each entry from the PKCS #12 files into the 7.3 security databases.

```
pk12util -i ServerCert.p12 -d .
Enter Password or Pin for "NSS Certificate DB":********
Enter password for PKCS12 file: ********
pk12util: PKCS12 IMPORT SUCCESSFUL

 pk12util -i kraStorageCert.p12 -d .
Enter Password or Pin for "NSS Certificate DB":********
Enter password for PKCS12 file: ********
pk12util: PKCS12 IMPORT SUCCESSFUL

 pk12util -i kraTransportCert.p12 -d .
Enter Password or Pin for "NSS Certificate DB":********
Enter password for PKCS12 file: ********
pk12util: PKCS12 IMPORT SUCCESSFUL
```

10. Optionally, delete the PKCS #12 files.

```
rm ServerCert.p12
```

```
rm kraStorageCert.p12

rm kraTransportCert.p12
```

11. Set the trust bits on the public/private key pairs that were imported into the 7.3 security databases.

```
certutil -M -n "Server-Cert cert-old_DRM_instance" -t "cu,cu,cu" -d .

certutil -M -n "kraStorageCert cert-old_DRM_instance" -t "u,u,u" -d .

certutil -M -n "kraTransportCert cert-old_DRM_instance" -t "u,u,u" -d .
```

12. Import the public key from the base-64 file, and set the trust bits.

```
certutil -A -n "caSigningCert cert-old_DRM_instance" -t "CT,c," -d . -i
caSigningCert.b64
```

13. Optionally, delete the base-64 file.

```
rm caSigningCert.b64
```

14. Open the `CS.cfg` configuration file in the `/var/lib/`*instance_ID*`/conf/` directory.

15. Edit the `kra.storageUnit.nickname` and `kra.transportUnit.nickname` attributes to reflect the 7.3 DRM instance.

```
kra.storageUnit.nickname=kraStorageCert cert-old_DRM_instance
kra.transportUnit.nickname=kraTransportCert cert-old_DRM_instance
```

NOTE

The `caSigningCert` is not referenced in the `CS.cfg` file.

16. In the same directory, edit the `serverCertNick.conf` file to contain the old certificate nickname. For example:

```
Server-Cert cert-old_DRM_instance
```

2.4. Option 4: HSM to HSM Migration

1. Extract the public/private key pairs from the HSM. The format for the extracted key pairs should be portable, such as a PKCS #12 file.

   The `pk12util` tool provided by Certificate System cannot extract public/private key pairs from an HSM because of requirements in the FIPS 140-1 standard which protect the private key. To extract this information, contact the HSM vendor. The extracted keys should not have any dependencies, such as nickname prefixes, on the HSM.

2. Copy the extracted key pairs from the 7.x server to the 7.3 server.

```
cp old_server_root/alias/ServerCert.p12
/var/lib/instance_ID/alias/ServerCert.p12

cp old_server_root/alias/kraStorageCert.p12
/var/lib/instance_ID/alias/kraStorageCert.p12

cp old_server_root/alias/kraTransportCert.p12
/var/lib/instance_ID/alias/kraTransportCert.p12
```

3. Extract the public key of the CA signing certificate from the 7.x security databases and save the base-64 encoded output to a file called `caSigningCert.b64`.

   a. Open the Certificate Management System 7.x `/alias` directory.

```
cd old_server_root/alias
```

   b. Set the `LD_LIBRARY_PATH` environment variable to search the Certificate System libraries.

```
LD_LIBRARY_PATH=old_server_root/bin/cert/lib

export LD_LIBRARY_PATH
```

   c. Use the Certificate Management System 7.x `certutil` tool to identify the old HSM slot name.

```
old_server_root/bin/cert/tools/certutil -U -d .
```

   d. Use the Certificate Management System 7.x `certutil` tool to extract the public key from the security databases and save the base-64 output to a file.

```
old_server_root/bin/cert/tools/certutil -L
     -n "old_HSM_slot_name:caSigningCert cert-old_DRM_instance"
     -d . -h old_HSM_token_name -a > caSigningCert.b64
```

```
```

e. Copy the key information from the 7.x server to the 7.3 server.

```
cp old_server_root/alias/caSigningCert.b64
/var/lib/instance_ID/alias/caSigningCert.b64
```

4. Open the Certificate System `/alias` directory.

```
cd /var/lib/instance_ID/alias/
```

5. Log in as `root`.

6. Set the file user and group to the Certificate System user and group.

```
# chown user:group ServerCert.p12

# chown user:group kraStorageCert.p12

# chown user:group kraTransportCert.p12

# chown user:group caSigningCert.b64
```

7. Log out as `root`, and log back into the system as the Certificate System user.

8. Set the file permissions.

```
chmod 00600 ServerCert.p12

chmod 00600 kraStorageCert.p12

chmod 00600 kraTransportCert.p12

chmod 00600 caSigningCert.b64
```

9. Register the new HSM in the 7.3 token database.

```
modutil -nocertdb -dbdir . -add new_HSM_token_name -libfile
new_HSM_library_path/new_HSM_library
```

10. Identify the new HSM slot name.

```
modutil -dbdir . -nocertdb -list
```

11. Import the public/private key pairs of each entry from the PKCS #12 files into the new HSM.

```
pk12util -i ServerCert.p12 -d . -h new_HSM_slot_name
Enter Password or Pin for "new_HSM_slot_name":********
Enter password for PKCS12 file: ********
pk12util: PKCS12 IMPORT SUCCESSFUL

 pk12util -i kraStorageCert.p12 -d . -h new_HSM_slot_name
Enter Password or Pin for "new_HSM_slot_name":********
Enter password for PKCS12 file: ********
pk12util: PKCS12 IMPORT SUCCESSFUL

 pk12util -i kraTransportCert.p12 -d . -h new_HSM_slot_name
Enter Password or Pin for "new_HSM_slot_name":********
Enter password for PKCS12 file: ********
pk12util: PKCS12 IMPORT SUCCESSFUL
```

12. Optionally, delete the PKCS #12 files.

```
rm ServerCert.p12

rm kraStorageCert.p12

rm kraTransportCert.p12
```

13. Set the trust bits on the public/private key pairs that were imported into the new HSM.

```
certutil -M -n "new_HSM_slot_name:Server-Cert cert-old_DRM_instance"
     -t "cu,cu,cu" -d . -h new_HSM_token_name

certutil -M -n "new_HSM_slot_name:kraStorageCert cert-old_DRM_instance"
     -t "u,u,u" -d . -h new_HSM_token_name

certutil -M -n "new_HSM_slot_name:kraTransportCert cert-old_DRM_instance"
     -t "u,u,u" -d . -h new_HSM_token_name
```

14. Import the public key from the base-64 file into the new HSM, and set the trust bits.

```
certutil -A -n "new_HSM_slot_name:caSigningCert cert-old_DRM_instance"
      -t "CT,c," -d . -h new_HSM_token_name -i caSigningCert.b64
```

15. Optionally, delete the base-64 file.

```
rm caSigningCert.b64
```

16. Open the `CS.cfg` configuration file in the `/var/lib/`*instance_ID*`/conf/` directory.

17. Edit the `kra.storageUnit.nickname` and `kra.transportUnit.nickname` attributes to reflect the 7.3 DRM information.

```
kra.storageUnit.nickname=new_HSM_slot_name:kraStorageCert
cert-old_DRM_instance
kra.transportUnit.nickname=new_HSM_slot_name:kraTransportCert
cert-old_DRM_instance
```

NOTE

The `caSigningCert` is not referenced in the `CS.cfg` file.

18. In the same directory, edit the `serverCertNick.conf` file to contain the old certificate nickname. For example:

```
new_HSM_slot_name:Server-Cert cert-old_DRM_instance
```

3. Online Certificate Status Protocol Manager (OCSP) Migration

Determine if the migration to be performed involves software security databases, an HSM, or both, and follow the appropriate process for the deployment scenario being migrated.

- *Section 3.1, "Option 1: Security Databases to Security Databases Migration"*
- *Section 3.2, "Option 2: Security Databases to HSM Migration"*
- *Section 3.3, "Option 3: HSM to Security Databases Migration"*
- *Section 3.4, "Option 4: HSM to HSM Migration"*

3.1. Option 1: Security Databases to Security Databases Migration

1. Remove all the security databases in the Certificate System 7.3 server which will receive migrated data.

```
rm /var/lib/instance_ID/alias/cert8.db

rm /var/lib/instance_ID/alias/key3.db
```

2. Copy the certificate and key security databases from the 7.x server to the 7.3 server.

```
cp old_server_root/alias/cert-old_OCSP_instance-cert8.db
/var/lib/instance_ID/alias/cert8.db

cp old_server_root/alias/cert-old_OCSP_instance-key3.db
/var/lib/instance_ID/alias/key3.db
```

3. Open the Certificate System `/alias` directory.

```
cd /var/lib/instance_ID/alias/
```

4. Log in as `root`.

5. Set the file user and group to the Certificate System user and group.

```
# chown user:group cert8.db

# chown user:group key3.db
```

6. Log out as `root`. As the Certificate System user, set the file permissions.

```
chmod 00600 cert8.db

chmod 00600 key3.db
```

7. List the certificates in the security databases using the `certutil` command; `-L` lists the certificates.

```
certutil -L -d .

Server-Cert cert-old_OCSP_instance cu,cu,cu
caSigningCert cert-old_OCSP_instance CT,c,
ocspSigningCert cert-old_OCSP_instance cu,cu,cu
```

8. Open the `CS.cfg` configuration file in the `/var/lib/`*instance_ID*`/conf/` directory.

9. Edit the `ocsp.signing.certnickname` attribute to reflect the 7.3 OCSP instance.

```
ocsp.signing.certnickname=ocspSigningCert cert-old_OCSP_instance
```

NOTE

The `caSigningCert` is not referenced in the `CS.cfg` file.

10. In the same directory, edit the `serverCertNick.conf` file to contain the old certificate nickname. For example:

```
Server-Cert cert-old_OCSP_instance
```

3.2. Option 2: Security Databases to HSM Migration

1. Remove all the security databases in the Certificate System 7.3 server which will receive migrated data.

```
rm /var/lib/instance_ID/alias/cert8.db

rm /var/lib/instance_ID/alias/key3.db
```

2. Copy the certificate and key security databases from the 7.x server to the 7.3 server.

```
cp old_server_root/alias/cert-old_OCSP_instance-cert8.db
/var/lib/instance_ID/alias/cert8.db

cp old_server_root/alias/cert-old_OCSP_instance-key3.db
/var/lib/instance_ID/alias/key3.db
```

3. Open the Certificate System `/alias` directory.

```
cd /var/lib/instance_ID/alias/
```

4. Log in as `root`.

5. Set the file user and group to the Certificate System user and group.

```
# chown user:group cert8.db

# chown user:group key3.db
```

6. Log out as `root`. As the Certificate System user, set the file permissions.

```
chmod 00600 cert8.db

chmod 00600 key3.db
```

7. List the certificates in the 7.x security databases using the `certutil` command; `-L` lists the certificates.

```
certutil -L -d .

Server-Cert cert-old_OCSP_instance cu,cu,cu
caSigningCert cert-old_OCSP_instance CT,c,
ocspSigningCert cert-old_OCSP_instance cu,cu,cu
```

8. Export the public/private key pairs of each entry in the Certificate System databases using the `pk12util` tool; `-o` exports the key pairs to a PKCS #12 file, and `-n` sets the name of the certificate and the old database prefix.

```
pk12util -o ServerCert.p12 -n "Server-Cert cert-old_OCSP_instance" -d .
Enter Password or Pin for "NSS Certificate DB":********
Enter password for PKCS12 file: ********
Re-enter password: ********
pk12util: PKCS12 EXPORT SUCCESSFUL

 pk12util -o ocspSigningCert.p12 -n "ocspSigningCert cert-old_OCSP_instance"
-d .
Enter Password or Pin for "NSS Certificate DB":********
Enter password for PKCS12 file: ********
Re-enter password: ********
pk12util: PKCS12 EXPORT SUCCESSFUL
```

NOTE

The 7.x security databases may contain additional public/private key pairs; these can also be extracted using `pk12util`.

9. Export the public key using the `certutil` tool; `-L` lists the named certificate, `-n` sets the name of the file and the old prefix, and `-a` outputs the information to a base-64 file.

```
certutil -L -n "caSigningCert cert-old_OCSP_instance" -d . -a >
caSigningCert.b64
```

NOTE

The 7.x security databases may contain additional public keys; these can also be exported using the `certutil` tool.

10. Delete the 7.x security databases.

```
rm cert8.db
```

```
rm key3.db
```

11. Register the new HSM in the 7.3 token database.

```
modutil -nocertdb -dbdir . -add new_HSM_token_name -libfile
new_HSM_library_path/new_HSM_library
```

12. Identify the new HSM slot name.

```
modutil -dbdir . -nocertdb -list
```

13. Create new security databases.

```
certutil -N -d .
```

14. Import the public/private key pairs of each entry from the PKCS #12 files into the new HSM.

```
pk12util -i ServerCert.p12 -d . -h new_HSM_slot_name
Enter Password or Pin for "new_HSM_slot_name":********
Enter password for PKCS12 file: ********
pk12util: PKCS12 IMPORT SUCCESSFUL

 pk12util -i ocspSigningCert.p12 -d . -h new_HSM_slot_name
Enter Password or Pin for "new_HSM_slot_name":********
Enter password for PKCS12 file: ********
pk12util: PKCS12 IMPORT SUCCESSFUL
```

15. Optionally, delete the PKCS #12 files.

```
rm ServerCert.p12

rm ocspSigningCert.p12
```

16. Set the trust bits on the public/private key pairs that were imported into the new HSM.

```
certutil -M -n "new_HSM_slot_name:Server-Cert cert-old_OCSP_instance" -t
"cu,cu,cu" -d .
     -h new_HSM_token_name

certutil -M -n "new_HSM_slot_name:ocspSigningCert cert-old_OCSP_instance" -t
"cu,cu,cu" -d .
     -h new_HSM_token_name
```

17. Import the public key from the base-64 file into the new HSM, and set the trust bits.

```
certutil -A -n "new_HSM_slot_name:caSigningCert cert-old_OCSP_instance"
     -t "CT,c," -d . -h new_HSM_token_name -i caSigningCert.b64
```

18. Optionally, delete the base-64 file.

```
rm caSigningCert.b64
```

19. Open the `CS.cfg` configuration file in the `/var/lib/`*instance_ID*`/conf/` directory.

20. Edit the `ocsp.signing.certnickname` attribute to reflect the 7.3 OCSP instance.

```
ocsp.signing.certnickname=new_HSM_slot_name:ocspSigningCert
cert-old_OCSP_instance
```

NOTE

The `caSigningCert` is not referenced in the `CS.cfg` file.

21. In the same directory, edit the `serverCertNick.conf` file to contain the old certificate nickname. For example:

```
new_HSM_slot_name:Server-Cert cert-old_OCSP_instance
```

3.3. Option 3: HSM to Security Databases Migration

1. Extract the public/private key pairs from the HSM. The format for the extracted key pairs should be portable, such as a PKCS #12 file.

   The `pk12util` tool provided by Certificate System cannot extract public/private key pairs from an HSM because of requirements in the FIPS 140-1 standard which protect the private key. To extract this information, contact the HSM vendor. The extracted keys should not have any dependencies, such as nickname prefixes, on the HSM.

2. Copy the extracted key pairs from the 7.x server to the 7.3 server.

```
cp old_server_root/alias/ServerCert.p12
/var/lib/instance_ID/alias/ServerCert.p12

cp old_server_root/alias/ocspSigningCert.p12
/var/lib/instance_ID/alias/ocspSigningCert.p12
```

```
```

3. Extract the public key of the CA signing certificate from the 7.x security databases and save the base-64 encoded output to a file called `caSigningCert.b64`.

   a. Open the Certificate Management System 7.x `/alias` directory.

```
cd old_server_root/alias
```

   b. Set the `LD_LIBRARY_PATH` environment variable to search the Certificate System libraries.

```
LD_LIBRARY_PATH=old_server_root/bin/cert/lib

export LD_LIBRARY_PATH
```

   c. Use the Certificate Management System 7.x `certutil` tool to identify the old HSM slot name.

```
old_server_root/bin/cert/tools/certutil -U -d .
```

   d. Use the Certificate Management System 7.x `certutil` tool to extract the public key from the security databases and save the base-64 output to a file.

```
old_server_root/bin/cert/tools/certutil -L -n
"old_HSM_slot_name:caSigningCert
      cert-old_OCSP_instance" -d . -h old_HSM_token_name -a >
caSigningCert.b64
```

   e. Copy the key information from the 7.x server to the 7.3 server.

```
cp old_server_root/alias/caSigningCert.b64
/var/lib/instance_ID/alias/caSigningCert.b64
```

4. Open the Certificate System `/alias` directory.

```
cd /var/lib/instance_ID/alias/
```

5. Log in as `root`.

6. Set the file user and group to the Certificate System user and group.

```
# chown user:group ServerCert.p12

# chown user:group ocspSigningCert.p12
```

```
# chown user:group caSigningCert.b64
```

7. Log out as `root`. As the Certificate System user, set the file permissions.

```
chmod 00600 ServerCert.p12

chmod 00600 ocspSigningCert.p12

chmod 00600 caSigningCert.b64
```

8. Import the public/private key pairs of each entry from the PKCS #12 files into the 7.3 security databases.

```
pk12util -i ServerCert.p12 -d .
Enter Password or Pin for "NSS Certificate DB":********
Enter password for PKCS12 file: ********
pk12util: PKCS12 IMPORT SUCCESSFUL

 pk12util -i ocspSigningCert.p12 -d .
Enter Password or Pin for "NSS Certificate DB":********
Enter password for PKCS12 file: ********
pk12util: PKCS12 IMPORT SUCCESSFUL
```

9. Optionally, delete the PKCS #12 files.

```
rm ServerCert.p12

rm ocspSigningCert.p12
```

10. Set the trust bits on the public/private key pairs that were imported into the 7.3 security databases.

```
certutil -M -n "Server-Cert cert-old_OCSP_instance" -t "cu,cu,cu" -d .

certutil -M -n "ocspSigningCert cert-old_OCSP_instance" -t "cu,cu,cu" -d .
```

11. Import the public key from the base-64 file, and set the trust bits.

```
certutil -A -n "caSigningCert cert-old_OCSP_instance" -t "CT,c," -d . -i
caSigningCert.b64
```

12. Optionally, delete the base-64 file.

```
rm caSigningCert.b64
```

13. Open the `CS.cfg` configuration file in the `/var/lib/`*instance_ID*`/conf/` directory.

14. Edit the `ocsp.signing.certnickname` attribute to reflect the 7.3 OCSP instance.

```
ocsp.signing.certnickname=ocspSigningCert cert-old_OCSP_instance
```

NOTE

The `caSigningCert` is not referenced in the `CS.cfg` file.

15. In the same directory, edit the `serverCertNick.conf` file to contain the old certificate nickname. For example:

```
Server-Cert cert-old_OCSP_instance
```

3.4. Option 4: HSM to HSM Migration

1. Extract the public/private key pairs from the HSM. The format for the extracted key pairs should be portable, such as a PKCS #12 file.

   The `pk12util` tool provided by Certificate System cannot extract public/private key pairs from an HSM because of requirements in the FIPS 140-1 standard which protect the private key. To extract this information, contact the HSM vendor. The extracted keys should not have any dependencies, such as nickname prefixes, on the HSM.

2. Copy the extracted key pairs from the 7.x server to the 7.3 server.

```
cp old_server_root/alias/ServerCert.p12
/var/lib/instance_ID/alias/ServerCert.p12

cp old_server_root/alias/ocspSigningCert.p12
/var/lib/instance_ID/alias/ocspSigningCert.p12
```

3. Extract the public key of the CA signing certificate from the 7.x security databases and save the base-64 encoded output to a file called `caSigningCert.b64`.

   a. Open the Certificate Management System 7.x `/alias` directory.

```
cd old_server_root/alias
```

b. Set the `LD_LIBRARY_PATH` environment variable to search the Certificate System libraries.

```
LD_LIBRARY_PATH=old_server_root/bin/cert/lib

export LD_LIBRARY_PATH
```

c. Use the Certificate Management System 7.x `certutil` tool to identify the old HSM slot name.

```
old_server_root/bin/cert/tools/certutil -U -d .
```

d. Use the Certificate Management System 7.x `certutil` tool to extract the public key from the security databases and save the base-64 output to a file.

```
old_server_root/bin/cert/tools/certutil -L
     -n "old_HSM_slot_name:caSigningCert cert-old_OCSP_instance"
     -d . -h old_HSM_token_name -a > caSigningCert.b64
```

e. Copy the key information from the 7.x server to the 7.3 server.

```
cp old_server_root/alias/caSigningCert.b64
/var/lib/instance_ID/alias/caSigningCert.b64
```

4. Open the Certificate System `/alias` directory.

```
cd /var/lib/instance_ID/alias/
```

5. Log in as `root`.

6. Set the file user and group to the Certificate System user and group.

```
# chown user:group ServerCert.p12

# chown user:group ocspSigningCert.p12

# chown user:group caSigningCert.b64
```

7. Log out as `root`. As the Certificate System user, set the file permissions.

```
chmod 00600 ServerCert.p12

chmod 00600 ocspSigningCert.p12
```

```
chmod 00600 caSigningCert.b64
```

8. Register the new HSM in the new token database.

```
modutil -nocertdb -dbdir . -add new_HSM_token_name -libfile
new_HSM_library_path/new_HSM_library
```

9. Identify the new HSM slot name.

```
modutil -dbdir . -nocertdb -list
```

10. Import the public/private key pairs of each entry from the PKCS #12 files into the new HSM.

```
pk12util -i ServerCert.p12 -d . -h new_HSM_slot_name
Enter Password or Pin for "new_HSM_slot_name":********
Enter password for PKCS12 file: ********
pk12util: PKCS12 IMPORT SUCCESSFUL

 pk12util -i ocspSigningCert.p12 -d . -h new_HSM_slot_name
Enter Password or Pin for "new_HSM_slot_name":********
Enter password for PKCS12 file: ********
pk12util: PKCS12 IMPORT SUCCESSFUL
```

11. Optionally, delete the PKCS #12 files.

```
rm ServerCert.p12

rm ocspSigningCert.p12
```

12. Set the trust bits on the public/private key pairs that were imported into the new HSM.

```
certutil -M -n "new_HSM_slot_name:Server-Cert cert-old_OCSP_instance
     -t "cu,cu,cu" -d . -h new_HSM_token_name

certutil -M -n "new_HSM_slot_name:ocspSigningCert cert-old_OCSP_instance"
     -t "cu,cu,cu" -d . -h new_HSM_token_name
```

13. Import the public key from the base-64 file into the new HSM, and set the trust bits.

```
certutil -A -n "new_HSM_slot_name:caSigningCert cert-old_OCSP_instance"
     -t "CT,c," -d . -h new_HSM_token_name -i caSigningCert.b64
```

14. Optionally, delete the base-64 file.

```
rm caSigningCert.b64
```

15. Open the `CS.cfg` configuration file in the `/var/lib/`*instance_ID*`/conf/` directory.

16. Edit the `ocsp.signing.certnickname` attribute to reflect the 7.3 subsystem information.

```
ocsp.signing.certnickname=new_HSM_slot_name:ocspSigningCert
cert-old_OCSP_instance
```

NOTE

The `caSigningCert` is not referenced in the `CS.cfg` file.

17. In the same directory, edit the `serverCertNick.conf` file to contain the old certificate nickname. For example:

```
new_HSM_slot_name:Server-Cert cert-old_OCSP_instance
```

4. Token Key Service (TKS) Migration

Determine if the migration to be performed involves software security databases, an HSM, or both; follow the appropriate process for the deployment scenario being migrated.

- *Section 4.1, “Option 1: Security Databases to Security Databases Migration”*
- *Section 4.2, “Option 2: Security Databases to HSM Migration”*
- *Section 4.3, “Option 3: HSM to Security Databases Migration”*
- *Section 4.4, “Option 4: HSM to HSM Migration”*

4.1. Option 1: Security Databases to Security Databases Migration

1. Remove all the security databases in the new Certificate System which will receive migrated data.

```
rm /var/lib/instance_ID/alias/cert8.db

rm /var/lib/instance_ID/alias/key3.db
```

2. Log into the 7.x server as the Certificate System user for that machine.
3. Migrate the master key from the 7.x TKS instance. (Depending on your installation, there may not be any master key information stored in the 7.x TKS instance.)
   a. Open the 7.x configuration file.
      - If the migration is from Certificate Management System 7.0, open the `CMS.cfg` in the `config` directory.
      - If the migration is from Certificate System 7.1, open the `CS.cfg` file in the Certificate System `config` directory.
      - If the migration is from Certificate System 7.2, open the `CS.cfg` file in the Certificate System `/var/lib/`*instance_ID*`/conf` directory.
   b. Write down the exact value for the `tks.mk_mappings.` line.

```
tks.mk_mappings.#tks_master_key_version_number#01=internal:tks_master_key_version_name
```

      A `tks.mk_mappings` value looks like the following:

```
tks.mk_mappings.#02#01=internal:tks_master_key_v2
```

      In this example, `02` is the *tks_master_key_version_ number*, and `tks_master_key_v2` is the *tks_master_key_version_name*.

4. Copy the certificate and key security databases from the 7.x server to the 7.3 server.

```
cp old_server_root/alias/cert-old_TKS_instance-cert8.db
/var/lib/instance_ID/alias/cert8.db

cp old_server_root/alias/cert-old_TKS_instance-key3.db
/var/lib/instance_ID/alias/key3.db
```

5. Open the Certificate System `alias/` directory.

```
cd /var/lib/instance_ID/alias/
```

6. Log in as `root`.
7. Set the file user and group to the Certificate System user and group.

```
# chown user:group cert8.db
```

```
# chown user:group key3.db
```

8. Log out as `root`. As the Certificate System user, change the permissions on the files.

```
chmod 00600 cert8.db

chmod 00600 key3.db
```

9. List the certificates in the security databases using the `certutil` command. In this example, `-L` lists the certificates.

```
certutil -L -d .

Server-Cert cert-old_TKS_instance cu,cu,cu
caSigningCert cert-old_TKS_instance CT,c,
tksTransportCert cert-old_TKS_instance CT,C,
```

10. Open the `CS.cfg` configuration file in the `/var/lib/`*instance_ID*`/conf/`.

11. If server-side keygen has been enabled, edit the `tks.drm_transport_cert_nickname` attribute to reflect the new TKS instance.

```
tks.drm_transport_cert_nickname=tksTransportCert cert-old_TKS_instance
```

12. If a master key was migrated from an 7.x TKS instance, edit the Certificate System 7.3 `CS.cfg`, and insert the "`tks.mk_mappings.#`*tks_master_key_version_number*`#01=internal:`*tks_master_key_version_name*" value from the Certificate System 7.x `CS.cfg` file. Be sure to use the proper *tks_master_key_version_number* and *tks_master_key_version_name* values.

NOTE

The `caSigningCert` is not referenced in the `CS.cfg` file.

13. In the same directory, edit the `serverCertNick.conf` file to contain the old certificate nickname. For example:

```
Server-Cert cert-old_TKS_instance
```

4.2. Option 2: Security Databases to HSM Migration

1. Remove all the security databases in the new Certificate System which will receive migrated data.

```
rm /var/lib/instance_ID/alias/cert8.db

rm /var/lib/instance_ID/alias/key3.db
```

2. Log into the 7.x server as the Certificate System user for that machine.

3. Migrate the master key from the 7.x TKS instance. (Depending on your installation, there may not be any master key information stored in the 7.x TKS instance.)

   a. Open the configuration file for the 7.x server instance being migrated.

   - If the migration is from Certificate Management System 7.0, open the `CMS.cfg` in the `config` directory.
   - If the migration is from Certificate System 7.1, open the `CS.cfg` file in the Certificate System `config` directory.
   - If the migration is from Certificate System 7.2, open the `CS.cfg` file in the Certificate System `/var/lib/`*instance_ID*`/conf` directory.

   b. Write down the exact value for the `tks.mk_mappings.` line.

```
tks.mk_mappings.#tks_master_key_version_number#01=internal:tks_master_key_version_name
```

   A `tks.mk_mappings` value looks like the following example:

```
tks.mk_mappings.#02#01=internal:tks_master_key_v2
```

   In this example, `02` is the *tks_master_key_version_ number*, and *tks_master_key_v2* is the *tks_master_key_version_name*.

4. Migrate symmetric keys from a 7.x TKS instance. Two things are necessary:

- A written copy of the original three session key shares to reproduce the symmetric transport key on the 7.x TKS instance.
- Copies of all files (there is at least one) containing the wrapped master keys for the 7.x security database; for example, `tks_master_key_v2.txt`.

NOTE

These files are created whenever the user generates a new master key using the `tksTool -W` option.

5. Copy the certificate and key security databases from the 7.x server to the 7.3 server.

```
cp old_server_root/alias/cert-old_TKS_instance-cert8.db
/var/lib/instance_ID/alias/cert8.db

cp old_server_root/alias/cert-old_TKS_instance-key3.db
/var/lib/instance_ID/alias/key3.db
```

6. Log into the new server as the Certificate System user, and open the Certificate System `alias/` directory.

```
cd /var/lib/instance_ID/alias/
```

7. Log in as `root`.

8. Set the file user and group to the Certificate System user and group.

```
# chown user:group cert8.db

# chown user:group key3.db
```

9. Log out as `root`. As the Certificate System user, change the permissions on the files.

```
chmod 00600 cert8.db

chmod 00600 key3.db
```

10. List the certificates stored in the 7.x security databases by using the `certutil` command. In this example, `-L` lists the certificates.

```
certutil -L -d .

Server-Cert cert-old_TKS_instance cu,cu,cu
caSigningCert cert-old_TKS_instance CT,c,
tksTransportCert cert-old_TKS_instance CT,C,C
```

11. Export the public/private key pairs of each entry in the Certificate System databases using the `pk12util` tool; `-o` exports the key pairs to a PKCS #12 file, and `-n` sets the name of the certificate and the old database prefix.

```
pk12util -o ServerCert.p12 -n "Server-Cert cert-old_TKS_instance" -d .
Enter Password or Pin for "NSS Certificate DB":********
Enter password for PKCS12 file: ********
Re-enter password: ********
pk12util: PKCS12 EXPORT SUCCESSFUL
```

12. Export the public key using the `certutil` tool; `-L` lists the named certificate, `-n` sets the name of the file and the old prefix, and `-a` outputs the information to a base-64 file.

```
certutil -L -n "caSigningCert cert-old_TKS_instance" -d . -a >
caSigningCert.b64

certutil -L -n "tksTransportCert cert-old_TKS_instance" -d . -a >
tksTransportCert.b64
```

13. Delete the 7.x security databases.

```
rm cert8.db

rm key3.db
```

14. Register the new HSM in the new token database.

```
modutil -nocertdb -dbdir . -add new_HSM_token_name -libfile
     new_HSM_library_path/new_HSM_library
```

15. Identify the new HSM slot name.

```
modutil -dbdir . -nocertdb -list
```

16. Create new security databases.

```
certutil -N -d .
```

17. Import the public/private key pair from the PKCS #12 file into the new HSM.

```
pk12util -i ServerCert.p12 -d . -h new_HSM_slot_name
Enter Password or Pin for "new_HSM_slot_name":********
Enter password for PKCS12 file: ********
pk12util: PKCS12 IMPORT SUCCESSFUL
```

18. Optionally, delete the PKCS #12 file:

```
rm ServerCert.p12
```

19. Set the trust bits on the public/private key pair that were imported into the new HSM.

```
certutil -M -n "new_HSM_slot_name:Server-Cert cert-old_TKS_instance"
     -t "cu,cu,cu" -d . -h <new_HSM_token_name
```

20. Import the public keys from the base-64 files into the new HSM, and set the trust bits.

```
certutil -A -n "new_HSM_slot_name:caSigningCert cert-old_TKS_instance"
     -t "CT,c," -d . -h new_HSM_token_name -i caSigningCert.b64

certutil -A -n "new_HSM_slot_name:tksTransportCert cert-old_TKS_instance"
     -t "CT,C,C" -d . -h new_HSM_token_name -i tksTransportCert.b64
```

21. Optionally, delete the base-64 files.

```
rm caSigningCert.b64

rm tksTransportCert.b64
```

22. Import the original symmetric transport key into the new HSM.

```
tksTool -I -d . new_HSM_token_name -n tks_transport_key_name
```

23. Type in the original three key session key shares (as prompted) to recreate the original transport key on the new HSM.

24. Log in as `root`.

25. Set the file user and group to the Certificate System user and group for each `wrapped_tks_master_key_file` file.

26. Unwrap and store all the original master keys in the new HSM.

```
tksTool -U -d . -h new_HSM_token_name -t tks_transport_key_name
 -n tks_master_key_version_name -i wrapped_tks_master_key_file
```

    Do this for every file containing a wrapped TKS master key.

27. Open the `CS.cfg` configuration file in the `/var/lib/`*instance_ID*`/conf/`.

28. If server-side keygen has been enabled, edit the `tks.drm_transport_cert_nickname` to reflect the new TKS information.

```
tks.drm_transport_cert_nickname=new_HSM_slot_name:tksTransportCert Cert
cert-old_TKS_instance
```

29. If a master key was migrated from the 7.x TKS instance, then also insert the `tks.mk_mappings.#` *tks_master_key_version_number*`#01=<` *new_HSM_slot_name:tks_master_key_version_name* line at the end of the `CS.cfg`. Be certain that the proper values for *tks_master_key_version_number*, *new_HSM_slot_name*,

and *tks_master_key_version_name* are set.

NOTE

The `caSigningCert` is not referenced in the `CS.cfg` file.

30. In the same directory, edit the `serverCertNick.conf` file to contain the old certificate nickname. For example:

```
new_HSM_slot_name:Server-Cert cert-old_TKS_instance
```

4.3. Option 3: HSM to Security Databases Migration

1. Extract the public/private key pairs from the HSM. The format for the extracted key pairs should be portable, such as a PKCS #12 file.

   The `pk12util` tool provided by Certificate System cannot extract public/private key pairs from an HSM because of requirements in the FIPS 140-1 standard which protect the private key. To extract this information, contact the HSM vendor. The extracted keys should not have any dependencies, such as nickname prefixes, on the HSM.

2. Log into the 7.x server as the Certificate System user for that machine.

3. Migrate the master key from the 7.x TKS instance. (Depending on your installation, there may not be any master key information stored in the 7.x TKS instance.)

   a. Open the Certificate System 7.x configuration file.

   - If the migration is from Certificate Management System 7.0, open the `CMS.cfg` in the `config` directory.
   - If the migration is from Certificate System 7.1, open the `CS.cfg` file in the Certificate System `config` directory.
   - If the migration is from Certificate System 7.2, open the `CS.cfg` file in the Certificate System `/var/lib/`*instance_ID*`/conf` directory.

   b. Write down the exact name-value pair for the `tks.mk_mappings.#`*tks_master_key_version_number*`#01=`*old_HSM_slot_name:tks_master_key_version_name* line. A `tks.mk_mappings` value looks like the following:

```
tks.mk_mappings.#02#01=mu:tks_master_key_v2
```

In this example, `02` is the *tks_master_key_version_ number*, `mu` is the *old_HSM_slot_name*, and `tks_master_key_v2` is the *tks_master_key_version_name*.

4. Migrate symmetric keys from the 7.x TKS instance. Two things are required:
   - A written copy of the original three session key shares to reproduce the symmetric transport key on the 7.x TKS instance.
   - Copies of all files (there is at least one) containing the wrapped master keys on the old HSM; for example, `tks_master_key_v2.txt`.

NOTE

These files are created whenever the user generates a new master key using the `tksTool -W` option.

5. Copy the extracted public/private key pairs from the 7.x server to the 7.3 server.

```
cp old_server_root/alias/ServerCert.p12
/var/lib/instance_ID/alias/ServerCert.p12
```

6. Extract the public key of the *"old_HSM_slot_name*`:caSigningCert cert-`*old_TKS_instance"* and *"old_HSM_slot_name*`:tksTransportCert"` `cert-`*old_TKS_instance"* from the 7.x security databases and save the base-64 encoded output to files called `caSigningCert.b64` and `tksTransportCert.b64`, respectively.
   a. Open the Certificate System 7.x `alias/` directory. cd *`old_server_root`*/alias
   b. Set the `LD_LIBRARY_PATH` environment variable to search the Certificate System libraries.

```
LD_LIBRARY_PATH=old_server_root/bin/cert/lib

export LD_LIBRARY_PATH
```

   c. Use the Certificate System 7.x `certutil` tool to identify the old HSM slot name.

```
old_server_root/bin/cert/tools/certutil -U -d .
```

   d. Use the Certificate System 7.x `certutil` tool to extract the public key of the following entries from the security databases and save each base-64 output to a separate file.

```
old_server_root/bin/cert/tools/certutil -L
      -n "old_HSM_slot_name:caSigningCert cert-old_TKS_instance"
```

```
      -d . -h old_HSM_token_name -a > caSigningCert.b64

 old_server_root/bin/cert/tools/certutil -L
      -n "old_HSM_slot_name:tksTransportCert cert-old_TKS_instance"
      -d . -h old_HSM_token_name -a > tksTransportCert.b64
```

e. Copy the key data from the 7.x server to the 7.3 server.

```
cp old_server_root/alias/caSigningCert.b64
/var/lib/instance_ID/alias/caSigningCert.b64

cp old_server_root/alias/tksTransportCert.b64
/var/lib/instance_ID/alias/tksTransportCert.b64
```

7. Log into the new server as the Certificate System user, and open the Certificate System `alias/` directory.

```
cd /var/lib/instance_ID/alias/
```

8. Log in as `root`.

9. Set the file user and group to the Certificate System user and group.

```
# chown user:group ServerCert.p12

# chown user:group caSigningCert.b64

# chown user:group tksTransportCert.b64
```

10. Log out as `root`. As the Certificate System user, change the permissions on the files.

```
chmod 00600 ServerCert.p12

chmod 00600 caSigningCert.b64

chmod 00600 tksTransportCert.b64
```

11. Import the public/private key pair from the PKCS #12 file into the new security databases.

```
pk12util -i ServerCert.p12 -d .
 Enter Password or Pin for "NSS Certificate DB":********
 Enter password for PKCS12 file: ********
 pk12util: PKCS12 IMPORT SUCCESSFUL
```

12. Optionally, delete the PKCS #12 file:

```
rm ServerCert.p12
```

13. Set the trust bits on the public/private key pairs that were imported into the new security databases.

```
certutil -M -n "Server-Cert cert-old_TKS_instance" -t "cu,cu,cu" -d .
```

14. Import the public keys from the base-64 files, and set the trust bits.

```
certutil -A -n "caSigningCert cert-old_TKS_instance"
      -t "CT,c," -d . -i caSigningCert.b64

certutil -A -n "tksTransportCert cert-old_TKS_instance"
      -t "CT,C,C" -d . -i tksTransportCert.b64
```

15. Optionally, delete the base-64 files.

```
rm caSigningCert.b64

rm tksTransportCert.b64
```

16. Import the original symmetric transport key into the new security databases.

```
tksTool -I -d . -n tks_transport_key_name
```

17. Type in the original three key session keyshares (as prompted) to recreate the original transport key in the new security databases.

18. Log in as `root`.

19. Set the file user and group to the Certificate System user and group for each `wrapped_tks_master_key_file` file.

20. Unwrap and store all the original master keys into the new security databases.

```
tksTool -U -d . -t tks_transport_key_name
      -n tks_master_key_version_name -i wrapped_tks_master_key_file
```

    Perform this step for each and every file containing a wrapped TKS master key.

21. Open the `CS.cfg` configuration file in the `/var/lib/`*instance_ID*`/conf/`.

22. If server-side keygen has been enabled, edit the `tks.drm_transport_cert_nickname` to reflect new TKS information.

```
tks.drm_transport_cert_nickname=tksTransportCert cert-old_TKS_instance
```

23. If a master key has been migrated from the 7.x TKS instance, insert the `"tks.mk_mappings.#` *tks_master_key_version_number*`#01=internal:` *tks_master_key_version_name"* line at the end of the `CS.cfg`. Be certain to use the proper values for *tks_master_key_version_number*, and *tks_master_key_version_name*.

NOTE

The `caSigningCert` is not referenced in the `CS.cfg` file.

24. In the same directory, edit the `serverCertNick.conf` file to contain the old certificate nickname. For example:

```
Server-Cert cert-old_TKS_instance
```

4.4. Option 4: HSM to HSM Migration

1. Extract the public/private key pairs from the HSM. The format for the extracted key pairs should be portable, such as a PKCS #12 file.

   The `pk12util` tool provided by Certificate System cannot extract public/private key pairs from an HSM because of requirements in the FIPS 140-1 standard which protect the private key. To extract this information, contact the HSM vendor. The extracted keys should not have any dependencies, such as nickname prefixes, on the HSM.

2. Log into the 7.x server as the Certificate System user for that machine.

3. Migrate the master key from the 7.x TKS instance. (Depending on your installation, there may not be any master key information stored in the 7.x TKS instance.)

   a. Open the Certificate System 7.x configuration file.

   - If the migration is from Certificate Management System 7.0, open the `CMS.cfg` in the `config` directory.
   - If the migration is from Certificate System 7.1, open the `CS.cfg` file in the Certificate System `config` directory.
   - If the migration is from Certificate System 7.2, open the `CS.cfg` file in the Certificate System `/var/lib/`*instance_ID*`/conf` directory.

   b. Write down or note the exact name-value pair for the `tks.mk_mappings.#`

tks_master_key_version_number`#01=` *old_HSM_slot_name:tks_master_key_version_name* line. A `tks.mk_mappings` value looks like the following:

```
tks.mk_mappings.#02#01=mu:tks_master_key_v2
```

In this example, `02` is the *tks_master_key_version_ number*, `mu` is the *old_HSM_slot_name*, and `tks_master_key_v2` is the *tks_master_key_version_name*.

4. Migrate symmetric keys from the 7.x TKS instance. Two things are required:
   - A written copy of the original three session key shares to reproduce the symmetric transport key on the 7.x TKS instance.
   - Copies of all files (there is at least one) containing the wrapped master keys on the old HSM; for example, `tks_master_key_v2.txt`.

NOTE

These files are created whenever the user generated a new master key using the `tksTool -W` option.

5. Copy the extracted public/private key pairs from the 7.x server to the 7.3 server.

```
cp old_server_root/alias/ServerCert.p12
/var/lib/instance_ID/alias/ServerCert.p12
```

6. Extract the public key of the *"old_HSM_slot_name*`:caSigningCert cert-`*old_TKS_instance"* and *"old_HSM_slot_name*`:tksTransportCert cert-`*old_TKS_instance"* from the 7.x security databases and save the base-64 encoded output to files called `caSigningCert.b64` and `tksTransportCert.b64`, respectively.

   a. Open the Certificate System 7.x `alias/` directory.

```
cd old_server_root/alias
```

   b. Set the `LD_LIBRARY_PATH` environment variable to search the Certificate System libraries.

```
LD_LIBRARY_PATH=old_server_root/bin/cert/lib

export LD_LIBRARY_PATH
```

   c. Use the Certificate System 7.x `certutil` tool to identify the old HSM slot name.

```
old_server_root/bin/cert/tools/certutil -U -d .
```

d. Use the Certificate System 7.x `certutil` tool to extract the public key of the following entries from the security databases and save each base-64 output to a separate file.

```
old_server_root/bin/cert/tools/certutil -L
      -n "old_HSM_slot_name:caSigningCert cert-old_TKS_instance"
      -d . -h old_HSM_token_name -a > caSigningCert.b64

 old_server_root/bin/cert/tools/certutil -L
      -n "old_HSM_slot_name:tksTransportCert cert-old_TKS_instance"
      -d . -h old_HSM_token_name -a > tksTransportCert.b64
```

e. Copy the key data from the 7.x server to the 7.3 server.

```
cp old_server_root/alias/caSigningCert.b64
/var/lib/instance_ID/alias/caSigningCert.b64

cp old_server_root/alias/tksTransportCert.b64
/var/lib/instance_ID/alias/tksTransportCert.b64
```

7. Log into the new server as the Certificate System user, and open the Certificate System `alias/` directory.

```
cd /var/lib/instance_ID/alias/
```

8. Log in as `root`.

9. Set the file user and group to the Certificate System user and group.

```
# chown user:group ServerCert.p12

# chown user:group caSigningCert.b64

# chown user:group tksTransportCert.b64
```

10. Log out as `root`. As the Certificate System user, change the permissions on the files.

```
chmod 00600 ServerCert.p12

chmod 00600 caSigningCert.b64

chmod 00600 tksTransportCert.b64
```

11. Register the new HSM in the new token database.

```
modutil -nocertdb -dbdir . -add new_HSM_token_name -libfile
     new_HSM_library_path/new_HSM_library
```

12. Identify the new HSM slot name.

```
modutil -dbdir . -nocertdb -list
```

13. Import the public/private key pair from the PKCS #12 file into the new HSM.

```
pk12util -i ServerCert.p12 -d . -h new_HSM_slot_name
 Enter Password or Pin for "new_HSM_slot_name":********
 Enter password for PKCS12 file: ********
 pk12util: PKCS12 IMPORT SUCCESSFUL
```

14. Optionally, delete the PKCS #12 file.

```
rm ServerCert.p12
```

15. Set the trust bits on the public/private key pair that was imported into the new HSM.

```
certutil -M -n "new_HSM_slot_name:Server-Cert cert-old_TKS_instance"
     -t "cu,cu,cu" -d . -h new_HSM_token_name
```

16. Import the public keys from the base-64 files, and set the trust bits.

```
certutil -A -n "new_HSM_slot_name:caSigningCert cert-old_TKS_instance"
     -t "CT,c," -d . -h new_HSM_token_name -i caSigningCert.b64

certutil -A -n new_HSM_slot_name:tksTransportCert cert-old_TKS_instance"
      -t "CT,C,C" -d . -h new_HSM_token_name -i tksTransportCert.b64
```

17. Optionally, delete the base-64 files.

```
rm caSigningCert.b64

rm tksTransportCert.b64
```

18. Import the original symmetric transport key into the new HSM.

```
tksTool -I -d . -h new_HSM_token_name -n tks_transport_key_name
```

19. Type in the original three key session key shares (as prompted) to recreate the original transport key on the new HSM.

20. Log in as `root`.

21. Set the file user and group to the Certificate System user and group for each `wrapped_tks_master_key_file` file.

22. Unwrap and store all the original master keys into the new HSM.

```
tksTool -U -d . -h new_HSM_token_name -t tks_transport_key_name
     -n tks_master_key_version_name -i wrapped_tks_master_key_file
```

    Do this for every file containing a wrapped TKS master key.

23. Open the `CS.cfg` configuration file in the `/var/lib/`*instance_ID*`/conf/`.

24. If server-side keygen has been enabled, edit the `tks.drm_transport_cert_nickname` attribute to reflect the new TKS information.

```
tks.drm_transport_cert_nickname=new_HSM_slot_name:tksTransport Cert
cert-old_TKS_instance
```

25. If a master key was migrated from the 7.x TKS instance, insert the `"tks.mk_mappings.#`*tks_master_key_version_number*`#01=`*new_HSM_slot_name:tks_master_key_version_name"* line at the end of the `CS.cfg` file. Fill in the proper values for the *tks_master_key_version_number*, *new_HSM_slot_name*, and *tks_master_key_version_name* variables.

NOTE

The `caSigningCert` is not referenced in the `CS.cfg` file.

26. In the same directory, edit the `serverCertNick.conf` file to contain the old certificate nickname. For example:

```
new_HSM_slot_name:Server-Cert cert-old_TKS_instance
```

Step 5: Migrating Password Cache Data

The password information for the Certificate System subsystems are saved in a special password file. In Certificate System 7.0 and 7.1, these were kept in the `pwcache.db` file. The contents of the password file must be decrypted and listed using the `PasswordCache` tool in the 7.x subsystem instance. Then, this information must be used to build the contents of the 7.3 `password.conf` file.

In Certificate System 7.2, passwords were kept in the `password.conf` file in the `/var/lib/`*instance_ID*`/conf/` directory, the same file that is used in Certificate System 7.3. This file contains the passwords for both internal database and the password for the `key3.db` file. The 7.2 `password.conf` file just needs to be copied over to the 7.3 `conf` directory.

NOTE

Only Certificate System 7.0 and 7.1 password cache data need to be migrated. For Certificate System 7.2, simply copy the `password.conf` file to the 7.3 instance directory.

Every 7.0 and 7.1 subsystem password file must be migrated separately, but the migration procedure is the same for all Certificate System subsystem instances.

1. Log into the 7.x server as the Certificate System user for that machine, and open the `config/` directory.

```
cd old_server_root/cert-old_instance/config/
```

2. Run the `PasswordCache` tool from the `tools` directory to retrieve the passwords from the database.

```
old_server_root/bin/cert/tools/PasswordCache old_passwordcache_password -d
old_server_root/alias
      -P cert-old_instance-old_hostname- -c pwcache.db list
```

   This lists the information stored in the password cache.

```
cert/key prefix = cert-old_instance-old_hostname-
path = old_server_root/alias
about to read password cache
----- Password Cache Content -----
internal : password
Internal LDAP Database : passwordldap
```

```
```

3. Use the listed tags and passwords to create the `password.conf` file. For example:

```
internal=password
Internal LDAP Database=passwordldap
```

4. If the 7.x server instance used the `password.conf` file to start the server instance automatically, then this file must also be migrated to the 7.3 server instance.

```
cp old_server_root/cert-old_instance/config/password.conf
/var/lib/instance_ID/conf/password.conf
```

5. Log into the 7.3 server as the Certificate System user, and open the Certificate System `conf/` directory.

```
cd /var/lib/instance_ID/conf/
```

6. Log in as `root`, and set the file user and group to the Certificate System user and group.

```
chown user:group password.conf
```

7. Log out as `root`. As the Certificate System user, change the permissions on the password file.

```
chmod 00600 password.conf
```

8. Copy the tags and passwords that were listed from the 7.x `pwdcache.db` file into the `password.conf` file.

Step 6: Migrating Internal Databases

Every Red Hat Certificate System 7.x subsystem contains LDIF data in an associated internal database which must be migrated to the corresponding Red Hat Certificate System 7.3 subsystem internal database. The procedure is the same for each subsystem type. The only difference between Certificate System 7.x versions is which import and export utility to use; these are version specific.

1. If you are migrating to a different platform or machine, then copy the newest version of the migration utility from the Certificate System 7.3 server to the Certificate System 7.x server.

   The migration utility is available as an independent RPM, which can be downloaded through the Certificate System Red Hat Network channel. The migration utilities are installed in the directory `/usr/share/rhpki/migrate`.

   a. Open the Certificate System instance directory. The migration utilities are in the `migrate` directory.

```
cd /usr/share/rhpki
```

   b. Package the latest version of the Certificate System migration utility using `zip` or `tar`.

```
tar -cvf migrate.tar migrate
```

NOTE

Regardless of the packaging tool used, the corresponding tool must be present on the 7.x server machine. If the platforms are identical and the `zip` utility is used, copy the `zip` tool from the 7.3 server to the 7.x server so that the `zip` and `unzip` versions match.

   c. Copy the package from the 7.3 server to the 7.x server, and then remove the package from the 7.3 server. (You can use any copy tool, such as `sftp`, `scp`, and `mv`.)

```
cp /usr/share/rhpki/migrate.tar old_server_root/bin/cert

rm /usr/share/rhpki/migrate.tar
```

   d. Log into the 7.x server as the Certificate System user for that machine, and open the Certificate System `bin/cert/` directory.

```
cd old_server_root/bin/cert
```

e. Log in as `root`, and set the file user and group to the Certificate System user and group.

```
# chown user:group migrate.tar
```

f. Log out as `root`. As the Certificate System user, change the permissions on the file.

```
chmod 00600 migrate.tar
```

g. Since the Certificate System 7.x migration utility will not be used, remove the Certificate System 7.x's `upgrade/` directory. This ensures that only the newest migration scripts, in the copied `migrate` directory, are available.

```
rm -rf old_server_root/bin/cert/upgrade
```

h. Unpackage the latest version of the migration utility using `unzip` or `tar`.

```
tar -xvf migrate.tar
```

i. Remove the migration utility package and any additional utilities, such as the `unzip` utility, that were copied to the Certificate System 7.x server.

```
rm migrate.tar
```

2. Log into the Directory Server for the Certificate System 7.3 instance, and export the internal database content to LDIF. The internal database name for the Certificate System instance is in the `internaldb.database` parameter in the `CS.cfg` file. Name the output file `new.ldif`.

   For example:

```
/opt/redhat-ds/slapd-DS-instance/db/db2ldif -n server.example.com-rhpki-ca
      -a /opt/redhat-ds/slapd-DS-instance/ldif/new.ldif
```

3. Log into the 7.x Certificate System instance, and export the database contents to LDIF. Name the output file `old.ldif`.

   For example:

```
cd old_server_root/slapd-old_instance-db/db/db2ldif -n userRoot
     -a old_server_root/slapd-old_instance-db/ldif/old.ldif
```

4. Modify the content of `old.ldif`.

NOTE

When using a text editor to perform the substitution instead of a script, use an editor that supports file sizes greater than 4 gigabytes, such as **vim**, because the LDIF files may be larger than 2 gigabytes and even 4 gigabytes in some deployments.

a. Open the Certificate System 7.x LDIF directory.

```
cd old_server_root/slapd-old_instance-db/ldif
```

b. Open the `old.ldif` file.

```
vi old.ldif
```

c. Replace `cn=aclResources` entry in the `old.ldif` file with the `cn=aclResources` entry from the `new.ldif` file. For example:

```
... replace with ...
dn: cn=aclResources,dc=server.example.com-rhpki-ca
resourceACLS: certServer.usrgrp.administration:read,modify:allow (read)
group=
      "Administrators" || group="Auditors" || group="Certificate Manager
Agents" |
      | group="Registration Manager Agents" || group="Data Recovery Manager
Agents
      " || group="Online Certificate Status Manager Agents";allow (modify)
group="
      Administrators":Administrators, auditors, and agents are allowed to
read user
      and group configuration but only administrators are allowed to modify
 ... list of ACLs ...
objectClass: top
objectClass: CertACLS
cn: aclResources
```

d. Add new groups for the the security domains to the `old.ldif` file. Security domains were not a feature in 7.0 or 7.1 versions of Certificate System, so it is necessary to add all of the group entries; for 7.2, it is only necessary to add the RA group entry.

```
dn: cn=Security Domain Administrators,ou=groups,basedn
description: People who are the Security Domain administrators
objectClass: top
objectClass: groupOfUniqueNames
cn: Security Domain Administrators
uniqueMember: uid=admin,ou=People,basedn
```

```
dn: cn=Enterprise CA Administrators,ou=groups,basedn
description: People who are the administrators for the security domain for
CA
objectClass: top
objectClass: groupOfUniqueNames
cn: Enterprise CA Administrators
uniqueMember: uid=admin,ou=People,basedn

dn: cn=Enterprise KRA Administrators,ou=groups,basedn
description: People who are the administrators for the security domain for
KRA
objectClass: top
objectClass: groupOfUniqueNames
cn: Enterprise KRA Administrators
uniqueMember: uid=admin,ou=People,basedn

dn: cn=Enterprise OCSP Administrators,ou=groups,basedn
description: People who are the administrators for the security domain for
OCSP
objectClass: top
objectClass: groupOfUniqueNames
cn: Enterprise OCSP Administrators
uniqueMember: uid=admin,ou=People,basedn

dn: cn=Enterprise TKS Administrators,ou=groups,basedn
description: People who are the administrators for the security domain for
TKS
objectClass: top
objectClass: groupOfUniqueNames
cn: Enterprise TKS Administrators
uniqueMember: uid=admin,ou=People,basedn

dn: cn=Enterprise RA Administrators,ou=groups,basedn
description: People who are the administrators for the security domain for
RA
objectClass: top
objectClass: groupOfUniqueNames
cn: Enterprise RA Administrators
uniqueMember: uid=admin,ou=People,basedn

dn: cn=Enterprise TPS Administrators,ou=groups,basedn
description: People who are the administrators for the security domain for
TPS
objectClass: top
objectClass: groupOfUniqueNames
cn: Enterprise TPS Administrators
uniqueMember: uid=admin,ou=People,basedn
```

5. Convert the `old.ldif` file to a text file.

   a. Open the version-to-text directory in the migration directory copied to the Certificate System 7.x server. Each 7.x major version has its own migration directory. For 7.0, use

`70ToTxt`; for 7.1, use `71ToTxt`; and for 7.2, use `72ToTxt`. For example:

```
cd old_server_root/bin/cert/migrate/71ToTxt
```

b. Edit the `run.sh` script to set the proper server and JRE information.

- For 7.0 and 7.1 migrations, uncomment and set the values for the following lines:

```
SERVER_ROOT=old_server_root
export SERVER_ROOT
INSTANCE=old_instance
export INSTANCE
```

- For versions 7.2 and later, make sure that the appropriate JRE root is given for the JRE version on the server. The default is:

```
JRE_ROOT=/usr/lib/jvm/jre-1.5.0
export JRE_ROOT
```

c. Run the `run.sh` to use the `old.ldif` file to create a text file.

```
run.sh old_server_root/slapd-old_instance-db/ldif/old.ldif >
     old_server_root/slapd-old_instance-db/ldif/old.txt
```

6. Open the Certificate System 7.x LDIF directory, and copy the `old.txt` file into the Certificate System 7.3 server instance's internal database LDIF directory. (You can use any copy tool, such as `sftp`, `scp`, and `mv`.)

```
cd old_server_root/slapd-old_instance-db/ldif

cp old_server_root/slapd-old_instance-db/ldif/old.txt
/opt/redhat-ds/slapd-DS-instance/ldif
```

7. Log into the 7.3 server as the Certificate System user, and open the Certificate System `ldif/` directory.

```
cd /opt/redhat-ds/slapd-DS-instance/ldif
```

8. Log in as `root`, and set the file user and group to the Certificate System user and group.

```
# chown user:group old.txt
```

9. Log out as `root`. As the Certificate System user, change the permissions on the file.

```
chmod 00600 old.txt
```

10. Convert the `old.txt` file to LDIF.

    a. Open the 7.3 text-conversion migration directory in the Certificate System 7.3 server.

```
cd /usr/share/rhpki/migrate/TxtTo73
```

    b. Edit the `run.sh` script; uncomment and set the values for the following lines. For example:

```
SERVER_ROOT=/var/lib
export SERVER_ROOT
INSTANCE=rhpki-ca
export INSTANCE
```

    c. Run `run.sh` to use `old.txt` to create an LDIF file.

```
run.sh /opt/redhat-ds/slapd-DS-instance/ldif/old.txt >
      /opt/redhat-ds/slapd-DS-instance/ldif/old.ldif
```

11. Import the `old.ldif` LDIF file into the Certificate System 7.3 server instance's internal database.

    a. Open the Certificate System 7.3 database directory.

```
cd /opt/redhat-ds/slapd-DS-instance/db
```

    b. Run the `ldif2db` tool to import the LDIF file into the Certificate System database. The internal database name for the Certificate System instance is in the `internaldb.database` parameter in the `CS.cfg` file. For example:

```
ldif2db -n server.example.com-rhpki-ca -i
/opt/redhat-ds/slapd-DS-instance/ldif/old.ldif
```

    c. Force the virtual list views (VLV) indexes to be re-indexed.

```
db2index
```

Step 7: Customizing User Data (Non-Console)

Copy all customized plug-ins, profiles, and forms to the Certificate System 7.3 server, and apply any hand-edited changes to the Certificate System 7.3 `CS.cfg` file.

In this example, the profile configuration in the *old_CA_instance* has been changed to enable S/MIME support. To migrate the configuration, make the same changes to the *new_CA_instance*. In Certificate Management System 7.x, S/MIME support is enabled by editing the `caTokenUserEncryptionKeyEnrollment` profile. Duplicate these changes over to the corresponding *new_CA_instance* profile.

1. Log into the 7.x server as the Certificate Management System user for that machine, and open the Certificate Management System `profiles/ca/` directory.
2. Copy the *p1* policy set in the `caTokenUserEncryptionKeyEnrollment.cfg` file, as shown:

```
policyset.set1.p1.constraint.class_id=noConstraintImpl
policyset.set1.p1.constraint.name=No Constraint
policyset.set1.p1.default.class_id=nsTokenUserKeySubjectNameDefaultImpl
policyset.set1.p1.default.name=nsTokenUserKeySubjectNameDefault
policyset.set1.p1.default.params.dnpattern=UID=$request.uid$,OU=Engineering,O=Example
policyset.set1.p1.default.params.ldap.enable=true
policyset.set1.p1.default.params.ldap.searchName=uid
policyset.set1.p1.default.params.ldapStringAttributes=uid,mail
policyset.set1.p1.default.params.ldap.basedn=dc=example,dc=com
policyset.set1.p1.default.params.ldap.maxConns=4
policyset.set1.p1.default.params.ldap.minConns=1
policyset.set1.p1.default.params.ldap.ldapconn.Version=2
policyset.set1.p1.default.params.ldap.ldapconn.host=ldaphostA.example.com
policyset.set1.p1.default.params.ldap.ldapconn.port=389
policyset.set1.p1.default.params.ldap.ldapconn.secureConn=false
```

   This configuration enables S/MIME support for services that use this profile to obtain certificates, such as token management systems.

3. Log into the new server as the Certificate System user, and open the Certificate System `profiles/ca/` directory.
4. Manually change the configuration in the *new_CA_instance* configuration to mimic the *old_CA_instance* configuration by editing the *p1* policy set in the `caTokenUserEncryptionKeyEnrollment.cfg` file, as shown:

```
policyset.set1.p1.constraint.class_id=noConstraintImpl
policyset.set1.p1.constraint.name=No Constraint
policyset.set1.p1.default.class_id=nsTokenUserKeySubjectNameDefaultImpl
policyset.set1.p1.default.name=nsTokenUserKeySubjectNameDefault
policyset.set1.p1.default.params.dnpattern=UID=$request.uid$,
```

```
OU=Engineering,O=Example
policyset.set1.p1.default.params.ldap.enable=true
policyset.set1.p1.default.params.ldap.searchName=uid
policyset.set1.p1.default.params.ldapStringAttributes=uid,mail
policyset.set1.p1.default.params.ldap.basedn=dc=example,dc=com
policyset.set1.p1.default.params.ldap.maxConns=4
policyset.set1.p1.default.params.ldap.minConns=1
policyset.set1.p1.default.params.ldap.ldapconn.Version=2
policyset.set1.p1.default.params.ldap.ldapconn.host=ldaphostA.example.com
policyset.set1.p1.default.params.ldap.ldapconn.port=389
policyset.set1.p1.default.params.ldap.ldapconn.secureConn=false
```

The altered profile serves certificate requests with S/MIME support enabled.

Step 8: Starting All Certificate System 7.3 Instances

1. Restart the Directory Server for the Certificate System 7.3 instance.

```
cd /opt/redhat-ds/slapd-DS-instance

./start-slapd
```

2. Start all of the Certificate System 7.3 instances.

```
/etc/init.d/instance_ID start
```

Step 9: Generate New Certificate System Server Certificates

If the Certificate System 7.3 server is on a different machine than the Certificate Management System 7.x server, then an SSL server certificate associated with each newly-migrated Certificate System server instance *must* be created.

There are three procedures to generate new server certificates, depending on the subsystem: generating self-signed CA server certificates; generating a CA certificate request which is signed by another CA; and generating DRM, OCSP, or TKS server certificates.

1. Self-Signing an SSL Server Certificate for a CA

1. Open the Certificate System 7.3 CA directory. For example:

```
cd /var/lib/rhpki-ca
```

2. Open the CA Console.

```
pkiconsole https://server.example.com:9443/ca
```

3. Select the **Configuration** tab.
4. Select the **System Keys and Certificates** option from the menu on the left.
5. Select the **Local Certificates** tab on the right.
6. Press the **Add/Renew** button to launch the Certificate Setup Wizard.
7. Follow the wizard prompts, and fill in the appropriate information.
   a. In the **Type of Operation** panel, select the **Request a certificate** option (the default).
   b. In the **Certificate Selection** panel, select **SSL Server Certificate** from the pull-down menu.

      Choose the **Sign this SSL Certificate with my CA Signing Certificate** option (the default). The SSL server certificate is automatically generated at the end of the process.
   c. In the **Key-Pair Information for the SSL Server Certificate** panel, select **Create new key pair**.

      Fill in information in the other fields on this panel as necessary.
   d. Select the desired hashing algorithm or use the default of SHA-1 in the **Message Digest**

Algorithm panel.

e. The next panel is **Subject Name for the SSL Certificate**. For the *cn* component, enter the fully qualified domain name, such as `zeta.example.com`, of the Certificate System 7.3 CA instance machine. Fill in information in the other fields on this panel as necessary (Red Hat strongly recommends filling in the *o* and *c* components).

f. For the rest of the panels in the wizard, click next, and either fill in the options as desired or accept all of the default settings.

8. Restart the Certificate System 7.3 CA instance.

```
/etc/init.d/rhpki-ca restart
```

2. Requesting a New SSL Server Certificate from a Third-Party CA

1. Start the CA Console.

```
pkiconsole https://server.example.com:9443/ca
```

2. Select the newly-migrated Certificate System instance, and open the Console for that instance.

3. In the Certificate System Console, select the **Configuration** tab.

4. In the left menu, select the **Keys and Certificates** option.

5. Select the **Local Certificates** tab on the right.

6. Press the **Add/Renew** button to launch the Certificate Setup Wizard.

7. Go through the screens in the wizard to request the certificate.

a. In the **Type of Operation** panel, select the **Request a Certificate** option (the default).

b. In the **Certificate Selection** panel, select **SSL Server Certificate** from the pull-down menu, and choose the **Create a request for submission to another CA** option. An SSL server certificate request is generated to submit to a CA for approval.

c. In the **Key-Pair Information for the SSL Server Certificate** panel, select **Create new key pair**. Fill in information in the other fields on this panel.

d. The next panel is **Subject Name for the SSL Certificate**. For the *cn* component, enter the fully qualified domain name of the Certificate System 7.3 CA instance machine, such as

`omega.example.com`. Fill in information in the other fields on this panel; it is strongly recommended that the *o* and *c* components also be filled in.

e. Go through the remaining panels in the Certificate Setup Wizard, and fill in the different fields or use the defaults.

8. Obtain the SSL server certificate request, and store it in a base-64 file.

9. Submit the SSL server certificate request to another CA and wait for approval of the request.

10. Once the SSL server certificate has been approved, press the **Add/Renew** button to relaunch the Certificate Setup Wizard.

    a. In the **Type of Operation** panel, select the **Install a certificate** option.

    b. In the **Certificate Selection** panel, select **SSL Server Certificate** from the pull-down menu.

    c. Enter in any necessary information in the **Location of Certificate** panel.

    d. Go through the remaining panels in the Certificate Setup Wizard to install the updated SSL server certificate.

11. Restart the Certificate System CA instance.

```
/etc/init.d/rhpki-ca restart
```

3. Generating a New DRM, OCSP, or TKS SSL Server Certificate

1. Open the subsystem instance's administrative console. For example, for the DRM subsystem:

```
pkiconsole https://server.example.com:10043/kra
```

2. Select the newly-imported Certificate System instance, and log into the Console for the instance.

3. Open the **Configuration** tab.

4. Select the **System Keys and Certificates** option from the menu on the left.

5. Select the **Local Certificates** tab on the right.

6. Click the **Add/Renew** button to launch the Certificate Setup Wizard.

a. In the **Type of Operation** panel, select the **Request a certificate** option (the default).

b. In the **Certificate Selection** panel, select **SSL Server Certificate** from the pull-down menu. An SSL server certificate request is generated, which can be submitted to a CA for approval.

c. In the **Key-Pair Information for the SSL Server Certificate**, select **Create new key pair**. Fill in information in the other fields.

d. The next panel is **Subject Name for the SSL Certificate**. For the *cn* component, enter the fully qualified domain name of the Certificate System subsystem machine, such as `omega.example.com`. Fill in information in the other fields on this panel; Red Hat strongly recommends filling in the *o* and *c* components also.

e. Click through the remaining panels in the Certificate Setup Wizard.

7. Obtain the SSL server certificate request, and store it in a base-64 file.

8. Submit the SSL server certificate request to a CA, and wait for approval of the request.

9. After the SSL server certificate is approved, click the **Add/Renew** button to relaunch the Certificate Setup Wizard.

a. In the **Type of Operation** panel, select the **Install a certificate** option.

b. In the **Certificate Selection** panel, select **SSL Server Certificate** from the pull-down menu.

c. Set the location information in the **Location of Certificate** if required.

d. Click through the remaining panels in the Certificate Setup Wizard to install the new SSL server certificate for the migrated subsystem instance.

10. Restart the Certificate System subsystem instance.

```
/etc/init.d/rhpki-kra restart
```

Step 10: Customizing User Data (Console)

Use the Console to configure any custom behavior of the different subsystems, such as customized plug-ins, logging, and auditing. A subsystem may have to be restarted once all configuration changes have been applied.

Step 11: Verifying Migration

After migrating all Certificate Management System 7.x subsystems to the corresponding Certificate System 7.3 subsystem instances, open the CA end-entities services page and each subsystem agent services pages for the Certificate System 7.3 server to ensure that everything is working properly. For example:

```
http://server.example.com:9080/ca/ee/ca
https://server.example.com:9443/ca/agent/ca
```

Then log into the Certificate System Console and verify that the new server can be managed through the Console.

```
pkiconsole https://server.example.com/ca
```

The port numbers for all the agent services interfaces can be found in the `server.xml` in the `conf/` directory for the Certificate System 7.3 installation.