



Red Hat Certificate System 10

Managing Smart Cards with the Enterprise Security Client

Updated for Red Hat Certificate System 10.1

Red Hat Certificate System 10 Managing Smart Cards with the Enterprise Security Client

Updated for Red Hat Certificate System 10.1

Florian Delehay
Red Hat Customer Content Services
fdelehay@redhat.com

Marc Muehlfeld
Red Hat Customer Content Services

Petr Bokoč
Red Hat Customer Content Services

Marc Muehlfeld
Red Hat Customer Content Services

Filip Hanzelka
Red Hat Customer Content Services

Ella Deon Ballard
Red Hat Customer Content Services

Tomáš Čapek
Red Hat Customer Content Services

Legal Notice

Copyright © 2020 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-sa/3.0/). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This guide is meant for regular users of Certificate System subsystems. It explains how to manage personal certificates and keys using the Enterprise Security Client, a simple interface to format and manage smart cards.

Table of Contents

CHAPTER 1. INTRODUCTION TO THE ENTERPRISE SECURITY CLIENT	3
1.1. RED HAT ENTERPRISE LINUX, SINGLE SIGN-ON, AND AUTHENTICATION	3
1.2. RED HAT CERTIFICATE SYSTEM AND THE ENTERPRISE SECURITY CLIENT	4
CHAPTER 2. INSTALLING THE ENTERPRISE SECURITY CLIENT	6
2.1. SUPPORTED PLATFORMS FOR THE CLIENT	6
2.2. SUPPORTED SMART CARDS	6
2.3. INSTALLING AND UNINSTALLING THE ENTERPRISE SECURITY CLIENT ON RED HAT ENTERPRISE LINUX	6
CHAPTER 3. USING THE ENTERPRISE SECURITY CLIENT	8
3.1. TRAY ICONS FOR THE ENTERPRISE SECURITY CLIENT	8
3.2. LAUNCHING ENTERPRISE SECURITY CLIENT	8
3.3. CONFIGURING PHONE HOME	9
3.4. SETTING UP USERS TO BE ENROLLED	11
3.5. MANAGING SMART CARDS	12
3.6. DIAGNOSING PROBLEMS	16
CHAPTER 4. USING SMART CARDS FOR WEB AND MAIL CLIENTS	22
4.1. SETTING UP BROWSERS TO SUPPORT SSL FOR TOKENS	22
CHAPTER 5. SETTING UP ENTERPRISE SECURITY CLIENT	24
5.1. DISABLING LDAP AUTHENTICATION FOR TOKEN OPERATIONS	24
APPENDIX A. REVISION HISTORY	25

CHAPTER 1. INTRODUCTION TO THE ENTERPRISE SECURITY CLIENT

The *Enterprise Security Client* is a tool for Red Hat Certificate System which simplifies managing smart cards. End users can employ security tokens (smart cards) to store user certificates for applications such as single sign-on (SSO) access and client authentication. End users are issued the tokens containing certificates and keys required for signing, encryption, and other cryptographic functions.

The Enterprise Security Client is the third part of Certificate System's complete token management system. Two subsystems – the Token Key Service (TKS) and Token Processing System (TPS) – are used to process token-related operations. The Enterprise Security Client is the interface which allows the smart card and user to access the token management system.

After a token is enrolled, applications such as Mozilla Firefox and Thunderbird can be configured to recognize the token and use it for security operations, like client authentication and S/MIME mail. The Enterprise Security Client provides the following capabilities:

- Supports Global Platform-compliant smart cards like Gemalto 64K V2 and Safenet 300J Java smart cards.
- Enrolls security tokens so they are recognized by TPS.
- Maintains the security token, such as re-enrolling a token with TPS.
- Provides information about the current status of the token or tokens being managed.
- Supports server-side key generation through the TPS and DRM subsystems so that keys can be archived and recovered on a separate token if a token is lost.

1.1. RED HAT ENTERPRISE LINUX, SINGLE SIGN-ON, AND AUTHENTICATION

Network users frequently have to submit multiple passwords for the various services they use, such as e-mail, web browsing and intranets, and servers on the network. Maintaining multiple passwords, and constantly being prompted to enter them, is a hassle for users and administrators. *Single sign-on* is a configuration which allows administrators to create a single password store so that users can log in once, using a single password, and be authenticated to all network resources.

Red Hat Enterprise Linux supports single sign-on for several resources, including logging into workstations and unlocking screensavers, accessing encrypted web pages using Mozilla Firefox, and sending encrypted email using Mozilla Thunderbird.

Single sign-on is both a convenience to users and another layer of security for the server and the network. Single sign-on hinges on secure and effective authentication, and the Enterprise Security Client ties into the public-key infrastructure implemented by Red Hat Certificate System.

One of the cornerstones of establishing a secure network environment is making sure that access is restricted to people who have the right to access the network. If access is allowed, users can *authenticate* to the system, meaning they can verify their identities. One such method is presenting a *certificate*: an electronic document that identifies the entity presenting it.

These certificates can be stored on a smart card. When inserted by a user, the smart card presents the certificates to the system and identifies the user so it can be authenticated. One of the two authentication methods for Red Hat Enterprise Linux's single sign-on is smart card authentication. (The other is Kerberos-based authentication.)

Single sign-on using smart cards goes through three steps:

1. A user inserts a smart card into the card reader. This is detected by the pluggable authentication modules (PAM) on Red Hat Enterprise Linux.
2. The system maps the certificate to the user entry and then compares the presented certificates on the smart card to the certificates stored in the user entry.
3. If the certificate is successfully validated against the key distribution center (KDC), then the user is allowed to log in.

The Enterprise Security Client manages the smart cards, which is part of administering single sign-on.

1.2. RED HAT CERTIFICATE SYSTEM AND THE ENTERPRISE SECURITY CLIENT

Red Hat Certificate System creates, manages, renews, and revokes certificates and keys. For managing smart cards, the Certificate System has a token management system to generate keys, create certificate requests, and receive certificates.

Two subsystems – the Token Key Service (TKS) and Token Processing System (TPS) – are used to process token-related operations. The Enterprise Security Client is the interface that allows the smart card and user to access the token management system.

A total of four Certificate System subsystems are involved with managing tokens, two for managing the tokens (TKS and TPS) and two for managing the keys and certificates within the public-key infrastructure (CA and DRM).

- The Token Processing System (TPS) interacts with smart cards to help them generate and store keys and certificates for a specific entity, such as a user or device. Smart card operations go through the TPS and are forwarded to the appropriate subsystem for action, such as the Certificate Authority to generate certificates or the Data Recovery Manager to archive and recover keys.
- The Token Key Service (TKS) generates, or derives, symmetric keys used for communication between the TPS and smart card. Each set of keys generated by the TKS is unique because they are based on the card's unique ID. The keys are formatted on the smart card and are used to encrypt communications, or provide authentication, between the smart card and TPS.
- The Certificate Authority (CA) creates and revokes user certificates stored on the smart card.
- Optionally, the Data Recovery Manager (DRM) archives and recovers keys for the smart card.

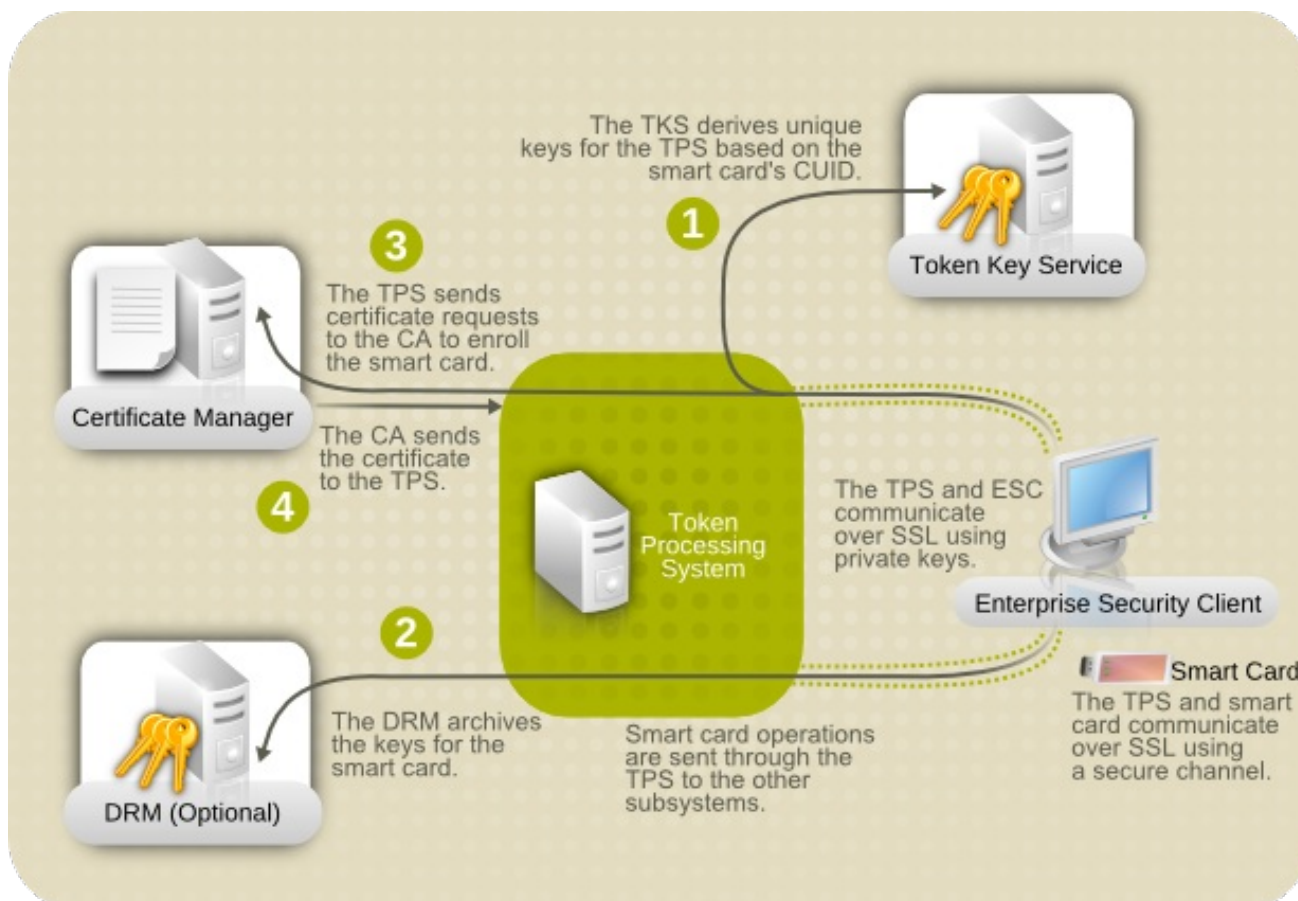


Figure 1.1. How Certificate System Manages Smart Cards

As Figure 1.1, "How Certificate System Manages Smart Cards" shows, the TPS is the central hub in the Red Hat Certificate System token management system. The token communicates with the TPS directly. The TPS then communicates with the TKS to derive a set of unique keys that can be used for TPS-token communication (1). When the smart card is enrolled, new private keys are created for the token; those keys can be archived in a DRM (2), if key archival is configured. The CA then processes the certificate request (3) and issues the certificates to store on the token. The TPS sends those certificates back to the Enterprise Security Client (4), and they are saved to the token.

The Enterprise Security Client is the conduit through which TPS communicates with each token over a secure HTTP channel (HTTPS), and, through the TPS, with the Certificate System.

To use the tokens, the Token Processing System must be able to recognize and communicate with them. The tokens must first be *enrolled* to populate the tokens with required keys and certificates and add the tokens to the Certificate System. The Enterprise Security Client provides the user interface for end entities to enroll tokens.

CHAPTER 2. INSTALLING THE ENTERPRISE SECURITY CLIENT

2.1. SUPPORTED PLATFORMS FOR THE CLIENT

The Enterprise Security Client interface is supported on the Red Hat Enterprise Linux 7.3 and later platforms.

The ESC is also supported on latest versions of Red Hat Enterprise Linux 5 and 6. Although these platforms do not support Red Hat Certificate System 10, those clients can be used against the TMS system in Red Hat Certificate System 10.

2.2. SUPPORTED SMART CARDS

For details, see the corresponding section in the [Red Hat Certificate System 10 Release Notes](#).

2.3. INSTALLING AND UNINSTALLING THE ENTERPRISE SECURITY CLIENT ON RED HAT ENTERPRISE LINUX

2.3.1. Installing the ESC Client

The first step in installing the Enterprise Security Client is to download the required packages. There are two ways to obtain the packages:

- Downloading an ISO image from the Customer Portal.
- Using the Red Hat **yum** utility

The preferred method of obtaining RPMs is using the **yum** command-line utility, as follows:

```
# yum install esc
```

If the **yum** command completes successfully, all of the necessary Enterprise Security Client RPMs and dependencies will be installed and ready for use.



NOTE

If you used the **yum** utility to install the Enterprise Security Client, there is no need for further installation; the client has already been installed. The following procedure is for installing from a CD image.

1. As the **root** user, install the Enterprise Security Client packages:

```
# yum install esc
```

The Enterprise Security Client is located in **/usr/lib/esc-1.1.0** on Red Hat Enterprise Linux 32-bit systems and **/usr/lib64/esc-1.1.0** on Red Hat Enterprise Linux 64-bit system. The **esc** shell script is installed in **/usr/bin/esc**. You can launch the Enterprise Security Client by running the **esc** command.

The Enterprise Security Client for Linux implements a daemon (**escd**) that runs silently, waiting for a smart card to be inserted. When an unenrolled smart card is inserted, the daemon automatically

launches the client UI, and the Enterprise Security Client guides the user through the enrollment process. It is also possible to launch the client manually from the **System** menu, by selecting **System Settings**, then **Smart Card Manager**.

2.3.2. Uninstalling the ESC client

1. Unplug all USB tokens.
2. Stop the Enterprise Security Client.
3. Log in as the **root** user, and use **rpm -ev** to remove the Enterprise Security Client RPM:

```
# yum remove esc
```



NOTE

Update the version numbers of the RPM files to match your version.

4. Remove any remaining files in the installation directory.

CHAPTER 3. USING THE ENTERPRISE SECURITY CLIENT

The following sections contain basic instructions on using the Enterprise Security Client for token enrollment, formatting, and password reset operations.

3.1. TRAY ICONS FOR THE ENTERPRISE SECURITY CLIENT

Many programs maintain an icon in the tray or notification area which can be used to control the operation of the program, usually through context menus when right-clicking the icon. The Enterprise Security Client provides tray icons, including tooltips for errors and actions such as inserting or removing a smart card.



Figure 3.1. Example Token Tray Icon and Tooltip

In its default configuration, the Enterprise Security Client launches and automatically minimizes to the tray. On Red Hat Enterprise Linux, the tray icon appears only if the notification area in Gnome has been enabled.

3.2. LAUNCHING ENTERPRISE SECURITY CLIENT

There are two concepts for launching the Enterprise Security Client. The Enterprise Security Client process must be started and it runs silently, waiting to detect any inserted smart card or token. The user interface for the Enterprise Security Client opens automatically when smart cards are inserted or can be opened manually.

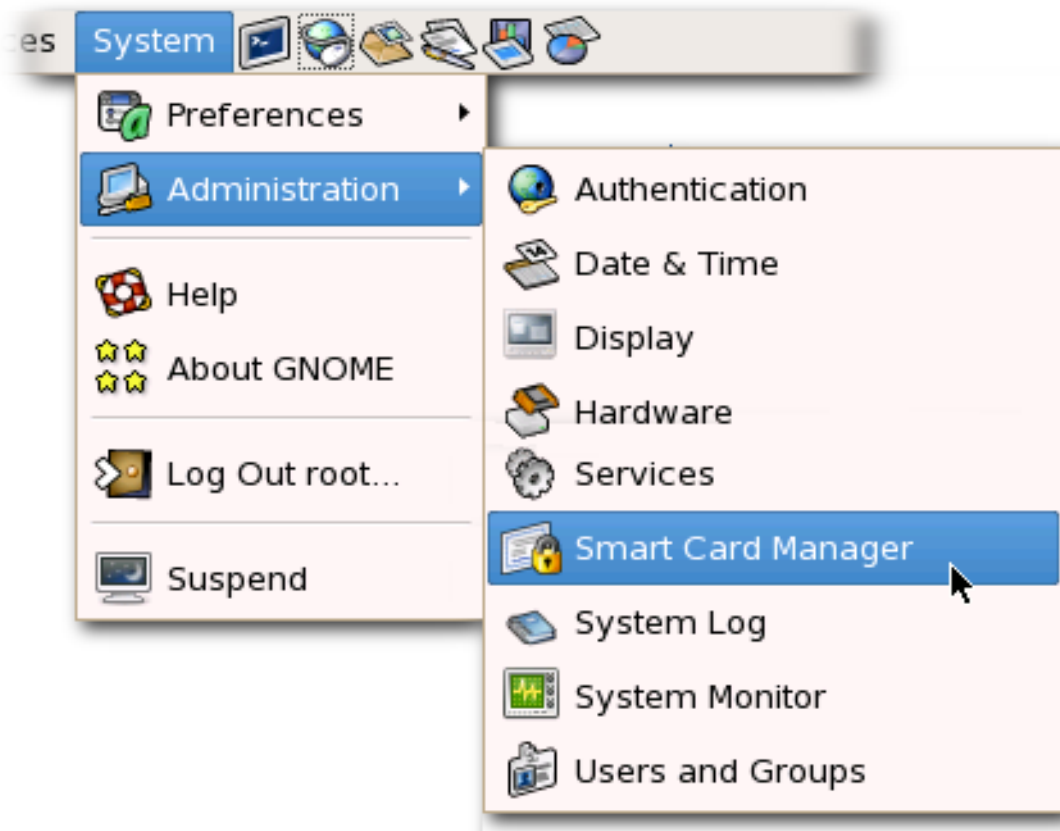
3.2.1. Opening the Enterprise Security Client on Red Hat Enterprise Linux

Initiate the Enterprise Security Client daemon (**escd**) from the command line:

```
esc
```

This daemon listens silently for smart cards and opens the GUI as soon as a smart card is inserted.

To open the Enterprise Security Client GUI manually, click **Applications, System Settings**, and then **Smart Card Manager**.



3.3. CONFIGURING PHONE HOME

The *Phone Home* feature in the Enterprise Security Client associates information within each smart card with information that points to distinct TPS servers and Enterprise Security Client UI pages. Whenever the Enterprise Security Client accesses a new smart card, it can connect to the TPS instance and retrieve the Phone Home information.

Phone Home retrieves and then caches this information; because the information is cached locally, the TPS subsystem does not have to be contacted each time a formatted smart card is inserted.

The information can be different for every key or token, which means that different TPS servers and enrollment URLs can be configured for different corporate or customer groups. Phone Home makes it possible to configure different TPS servers for different issuers or company units, without having to configure the Enterprise Security Client manually to locate the correct server and URL.

NOTE

In order for the TPS subsystem to utilize the Phone Home feature, Phone Home must be enabled in the TPS configuration file, as follows:

```
op.format.userKey.issuerinfo.enable=true
op.format.userKey.issuerinfo.value=http://server.example.com
```

3.3.1. About Phone Home Profiles

The Enterprise Security Client is based on Gnome. When the Enterprise Security Client caches information for each token, the information is stored in the user's configuration file. The next time the Enterprise Security Client is launched, it retrieves the information from the configuration file instead of contacting the server again.

When a smart card is inserted and Phone Home is triggered, the Enterprise Security Client first checks the token for the *Phone Home URL*, which is the default URL the Enterprise Security Client uses to try connecting to the TPS.

If there is no Phone Home information on the token, users can specify the Phone Home URL value manually by clicking the **Phone Home** button in the Enterprise Security Client UI. See [Section 3.3.2, "Setting the Phone Home URL"](#). The other information is supplied and stored when the token is formatted. In this case, the company supplies the specific Phone Home URL for the user. After the user submits the URL, the format process adds the rest of the information to the Phone Home profile. The format process is not any different for the user.

3.3.2. Setting the Phone Home URL

The Enterprise Security Client needs to be configured to communicate with the TPS; this is done via the *Phone Home URL*. Formatted tokens (they can be formatted by the manufacturer or by your IT department) already have this URL set. If a token is unformatted, the Enterprise Security Client cannot find the Phone Home URL: such blank tokens require to define the URL manually.

The **Phone Home** button allows users to specify the Phone Home URL:

1. Once a blank token is inserted, click the **Phone Home** button in the Enterprise Security Client UI to open a configuration dialog.
2. In the **TPS Config URI** field, fill in the new TPS URL.
3. Click **OK** to save. Once the new Phone Home URL is correctly configured, the rest of the information is retrieved and added to the Phone Home profile.

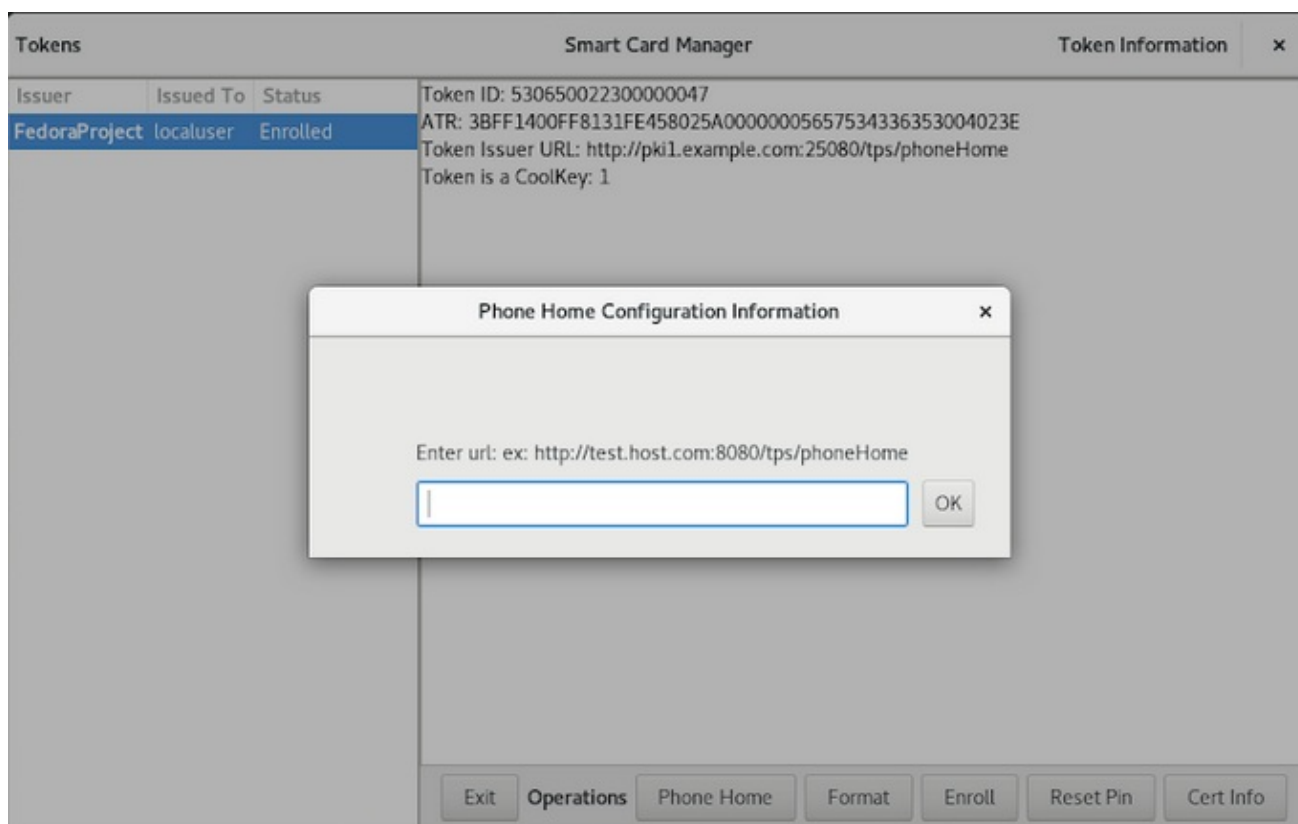


Figure 3.2. Phone Home URL Configuration

3.3.3. Configuring the TPS to Use Phone Home

The Phone Home feature and the different type of information it uses only work when the TPS is properly configured to use Phone Home. If not, TPS ignores this feature. Phone Home is configured in the **phoneHome.xml** in the `/var/lib/pki/pki-tomcat/tps/conf/` directory; this prints the Phone Home information to XML.

[Example 3.1, "TPS Phone Home Configuration File"](#) shows an example XML file used by the TPS subsystem to configure the Phone Home feature.

Example 3.1. TPS Phone Home Configuration File

```
<ServiceInfo><IssuerName>Example Corp</IssuerName>
  <Services>
    <Operation>http://server.example.com:7888/nk_service ## TPS server URL
    </Operation>
    <UI>http://server.example.com:7888/cgi_bin/esc.cgi ## Optional
  Enrollment UI
  </UI>
  <EnrolledTokenBrowserURL>http://www.test.url.com ## Optional
  enrolled token url
  </EnrolledTokenBrowserURL>
  </Services>
</ServiceInfo>
```

The TPS configuration URI is the URL of the TPS server which returns the rest of the Phone Home information to the Enterprise Security Client. An example of this URL is **http://localhost:8443/tps/phoneHome**; the URL can reference the machine name, fully-qualified domain name, or an IPv4 or IPv6 address, as appropriate. When the TPS configuration URI is accessed, the TPS server is prompted to return all of the Phone Home information to the Enterprise Security Client.

To test the URL of the Smart Card server, enter the address in the **TPS Config URI** field, and click **Test URL**.

If the server is successfully contacted, a message box indicates success. If the test connection fails, an error dialog appears.

3.4. SETTING UP USERS TO BE ENROLLED

When the Token Processing System is installed, one of its configuration settings is the LDAP directory which contains the users who are allowed to enroll a token. Only users who are stored within this authentication directory are allowed to enroll, format, or have a token. Before attempting to enroll a token or smart card, make sure that the person requesting the operation has an entry in the LDAP directory.

The TPS is configured to look at a specific base DN in the LDAP directory. This is configured in the TPS's **CS.cfg**:

```
auth.instance.0.baseDN=dc=example,dc=com
auth.instance.0.hostport=server.example.com:389
```

For a user to be allowed to enroll a token, the user must be somewhere below the base DN.

If the user does not already have an entry, then the administrator must add the user to the specified LDAP directory in the specified base DN before any tokens can be enrolled for the user.

```
/usr/bin/ldapmodify -a -D "cn=Directory Manager" -w secret -p 389 -h server.example.com
```

```
dn: uid=jsmith,ou=People,dc=example,dc=com
objectclass: person
objectclass: inetorgperson
objectclass: top
uid: jsmith
cn: John Smith
email: jsmith@example.com
userPassword: secret
```

3.5. MANAGING SMART CARDS

You can use the **Manage Smart Cards** page to perform many of the operations that can be applied to one of the cryptographic keys stored on the token.

You can use this page to format the token, set and reset the card's password, and to display card information. Two other operations, enrolling tokens and viewing the diagnostic logs, are also accessed through the **Manage Smart Cards** page. These operations are addressed in other sections.

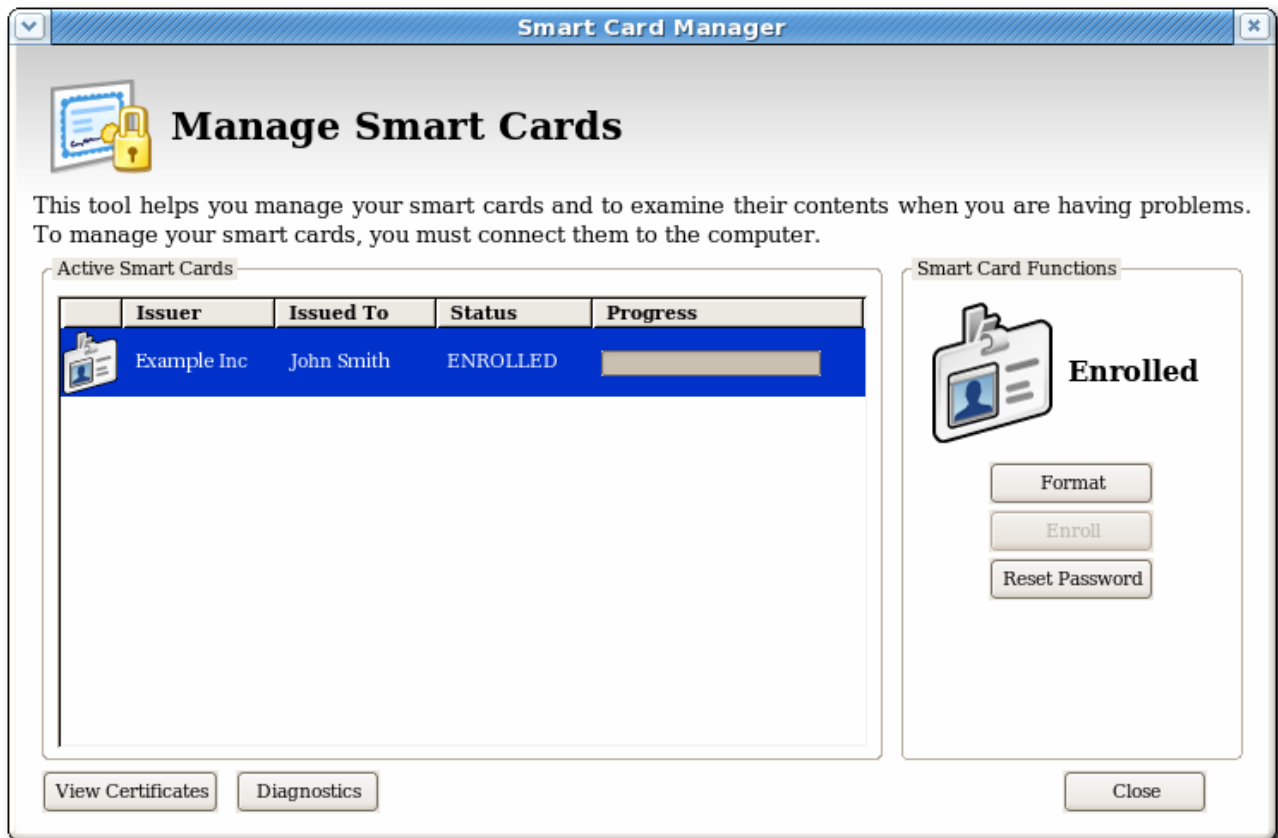


Figure 3.3. Manage Smart Cards Page

3.5.1. Formatting the Smart Card

When you format a smart card, it is reset to the uninitialized state. This removes all previously generated user key pairs and erases the password set on the smart card during enrollment.

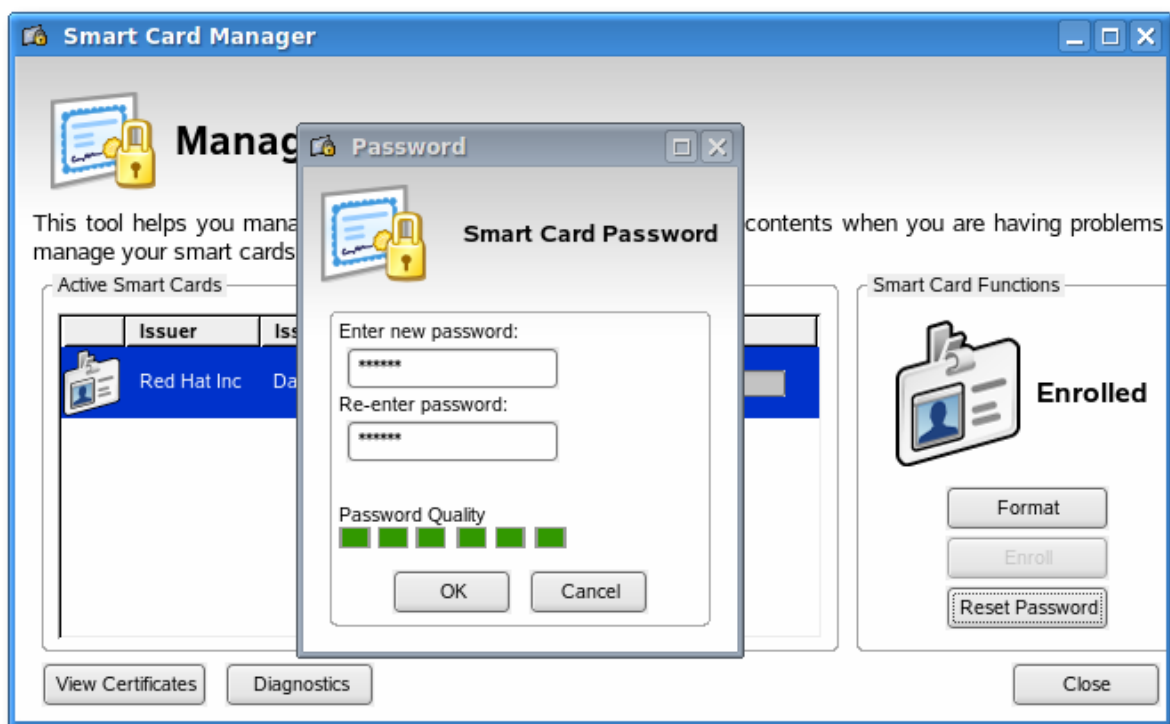
To format a smart card:

1. Insert a supported smart card into the computer. Ensure that the card is listed in the **Active Smart Cards** table.
2. In the **Smart Card Functions** section of the **Manage Smart Cards** screen, click **Format**.
3. If the TPS has been configured for user authentication, enter the user credentials in the authentication dialog, and click **Submit**.
4. During the formatting process, the status of the card changes to BUSY and a progress bar is displayed. A success message is displayed when the formatting process is complete. Click **OK** to close the message box.
5. When the formatting process is complete, the **Active Smart Cards** table shows the card status as UNINITIALIZED.

3.5.2. Resetting a Smart Card Password

If a user forgets the password for a smart card after the card is enrolled, it is possible to reset the password. To reset the password on a smart card:

1. Insert a supported smart card into the computer. Ensure that the card is listed in the **Active Smart Cards** table.
2. In the **Smart Card Functions** section of the Manage Smart Cards screen, click **Reset Password** to display the **Password** dialog.
3. Enter a new smart card password in the **Enter new password** field.
4. Confirm the new smart card password in the **Re-Enter password** field, and then click **OK**.



5. If the TPS has been configured for user authentication, enter the user credentials in the authentication dialog, and click **Submit**.

6. Wait for the password to finish being reset.

3.5.3. Viewing Certificates

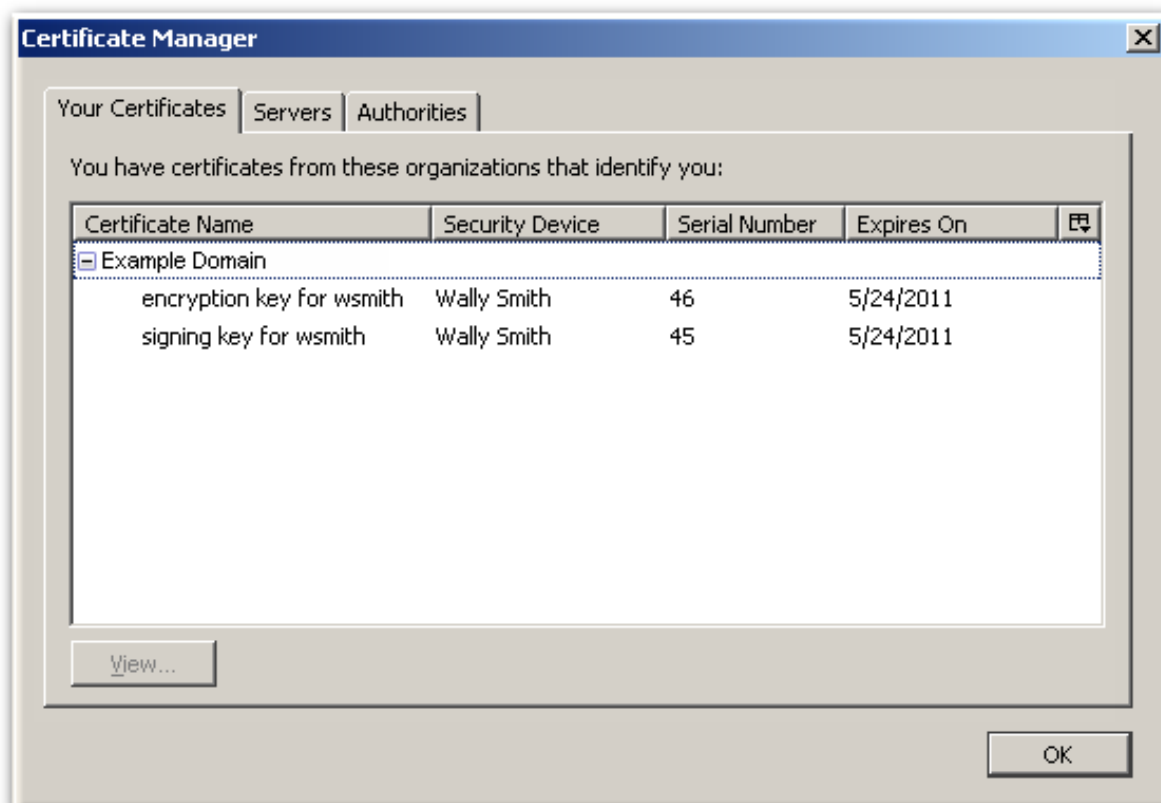
The **Smart Card Manager** can display basic information about a selected smart card, including stored keys and certificates. To view certificate information:

1. Insert a supported smart card into the computer. Ensure that the card is listed in the **Active Smart Cards** table.
2. Select the card from the list, and click **View Certificates**.



This displays basic information about the certificates stored on the card, including the serial number, certificate nickname, and validity dates.

3. To view more detailed information about a certificate, select the certificate from the list and click **View**.



3.5.4. Enrolling Smart Cards

Most smart cards will be automatically enrolled using the automated enrollment procedure. You can also use the **Manage Smart Cards** facility to manually enroll a smart card.

If you enroll a token with the user key pairs, then the token can be used for certificate-based operations such as SSL client authentication and S/MIME.



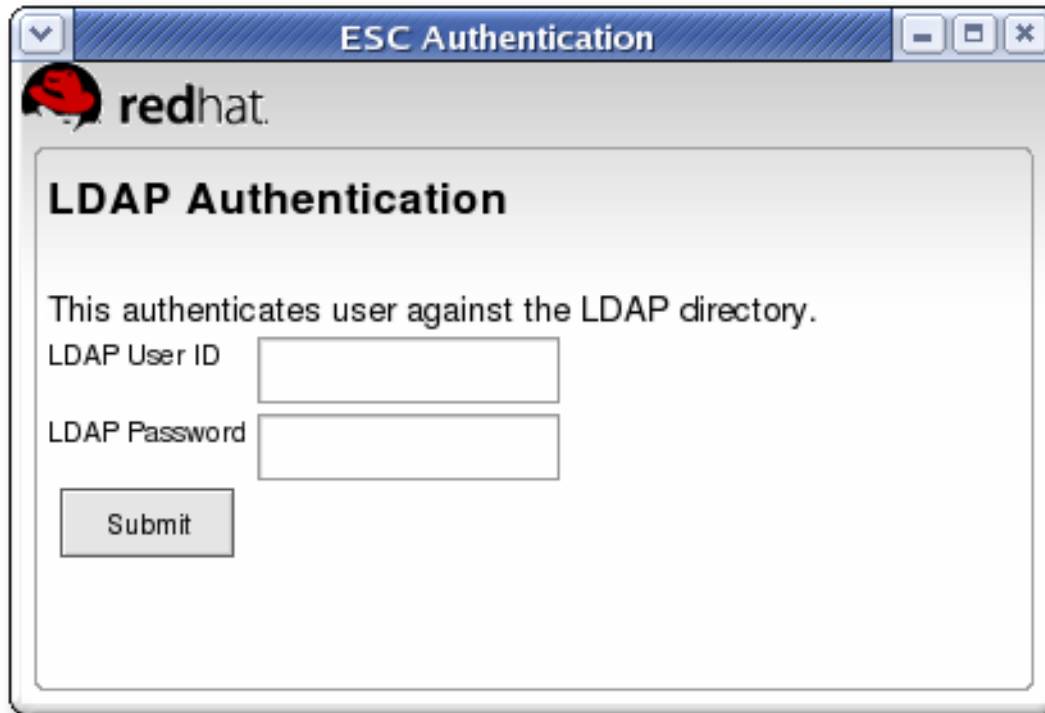
NOTE

The TPS server can be configured to generate the user key pairs on the server and then archived in the DRM subsystem for recovery if the token is lost.

To enroll a smart card manually:

1. Insert a supported, unenrolled smart card into the computer. Ensure that the card is listed in the **Active Smart Cards** table.
2. Click **Enroll** to display the **Password** dialog.
3. Enter a new key password in the **Enter a password** field.
Confirm the new password in the **Re-Enter a password** field.
4. Click **OK** to begin the enrollment.
5. If the TPS has been configured for user authentication, enter the user credentials in the authentication dialog, and click **Submit**.

If the TPS has been configured to archive keys to the DRM, the enrollment process will begin generating and archiving keys.



The image shows a window titled "ESC Authentication" with the Red Hat logo. Inside the window, there is a section titled "LDAP Authentication" with the text "This authenticates user against the LDAP directory." Below this text are two input fields: "LDAP User ID" and "LDAP Password". A "Submit" button is located below the "LDAP Password" field.

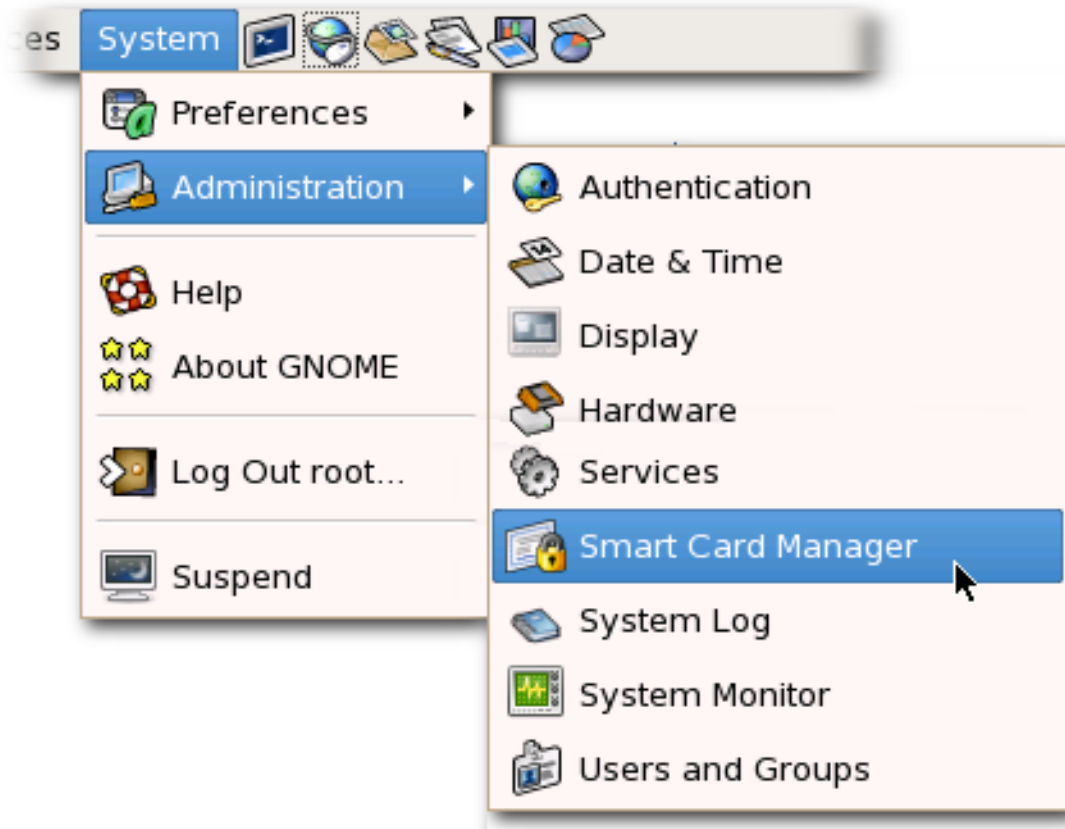
When the enrollment is complete, the status of the smart card is displayed as ENROLLED.

3.6. DIAGNOSING PROBLEMS

The Enterprise Security Client includes basic diagnostic tools and a simple interface to log errors and common events, such as inserting and removing a smart card or changing the card's password. The diagnostic tools can identify and notify users about problems with the Enterprise Security Client, smart cards, and TPS connections.

To open the **Diagnostics Information** window:

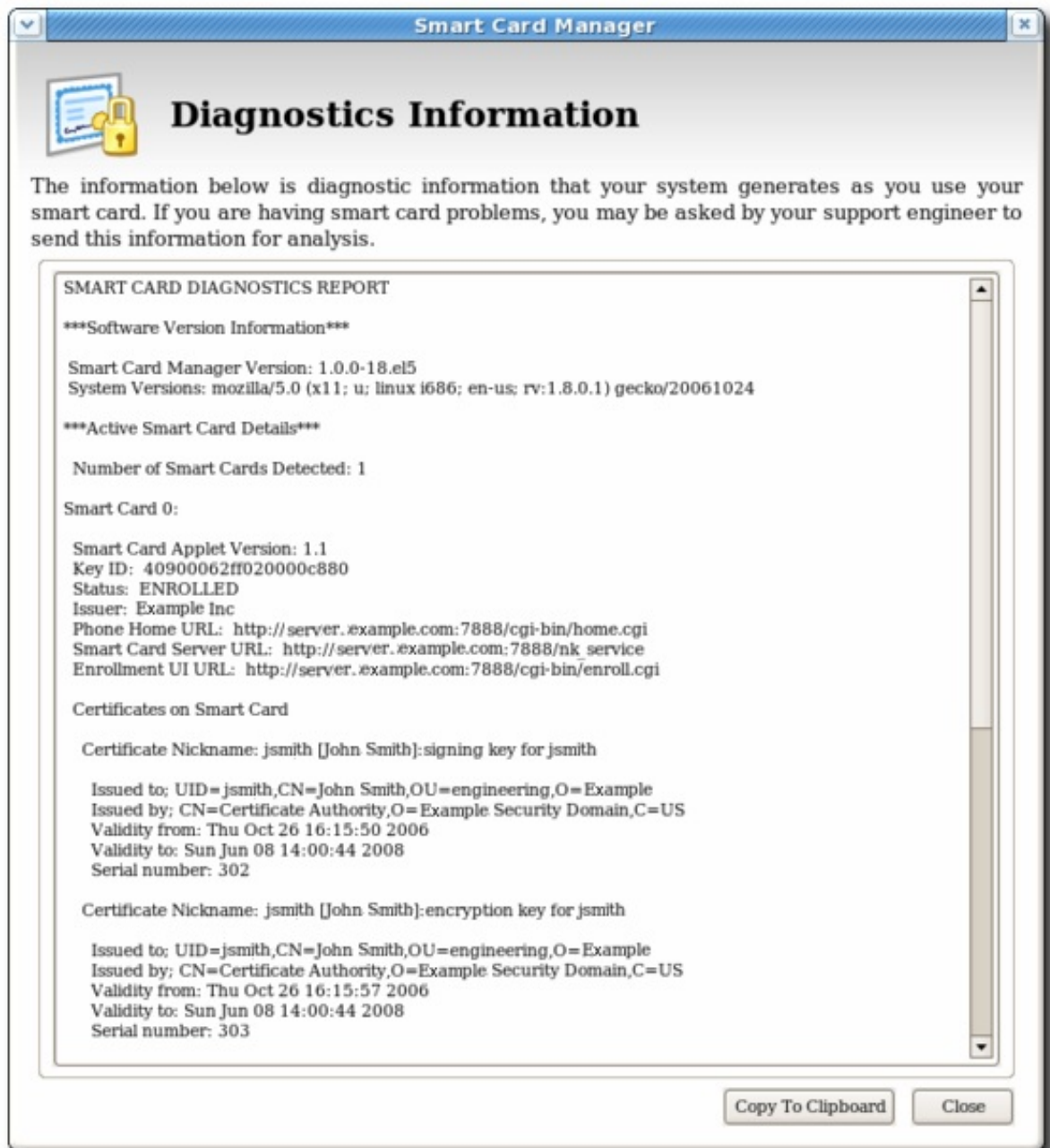
1. Open the Enterprise Security Client.



2. Select the smart card to check from the list.
3. Click the **Diagnostics** button.



4. This opens the **Diagnostic Information** window for the selected smart card.



The **Diagnostics Information** screen displays the following information:

- The Enterprise Security Client version number.
- The number of cards detected by the Enterprise Security Client.

For each card detected, the following information is displayed:

- The version of the applet running on the smart card.
- The alpha-numeric ID of the smart card.
- The card's status, which can be any of the three things:
 - *NO_APPLET* No key was detected.
 - *UNINITIALIZED*. The key was detected, but no certificates have been enrolled.
 - *ENROLLED*. The detected card has been enrolled with certificate and card information.

- The card's Phone Home URL. This is the URL from which all Phone Home information is obtained.
- The card issuer name, such as **Example Corp.**
- The card's answer-to-reset (ATR) string. This is a unique value that can be used to identify different classes of smart cards. For example:

```
3BEC00FF8131FE45A0000000563333304A330600A1
```

- The TPS Phone Home URL.
- The TPS server URL. This is retrieved through Phone Home.
- The TPS enrollment form URL. This is retrieved through Phone Home.
- Detailed information about each certificate contained on the card.
- A running log of the most recent Enterprise Security Client errors and common events.

The Enterprise Security Client records two types of diagnostic information. It records *errors* that are returned by the smart card, and it records *events* that have occurred through the Enterprise Security Client. It also returns basic information about the smart card configuration.

3.6.1. Errors

- The Enterprise Security Client does not recognize a card.
- Problems occur during a smart card operation, such as a certificate enrollment, password reset, or format operation.
- The Enterprise Security Client loses the connection to the smart card. This can happen when problems occur communicating with the **PCSC** daemon.
- The connection between the Enterprise Security Client and TPS is lost.

Smart cards can report certain error codes to the TPS; these are recorded in the TPS's **tps-debug.log** or **tps-error.log** files, depending on the cause for the message.

Table 3.1. Smart Card Error Codes

Return Code	Description
General Error Codes	
6400	No specific diagnosis
6700	Wrong length in Lc
6982	Security status not satisfied
6985	Conditions of use not satisfied

Return Code	Description
6a86	Incorrect P1 P2
6d00	Invalid instruction
6e00	Invalid class
Install Load Errors	
6581	Memory Failure
6a80	Incorrect parameters in data field
6a84	Not enough memory space
6a88	Referenced data not found
Delete Errors	
6200	Application has been logically deleted
6581	Memory failure
6985	Referenced data cannot be deleted
6a88	Referenced data not found
6a82	Application not found
6a80	Incorrect values in command data
Get Data Errors	
6a88	Referenced data not found
Get Status Errors	
6310	More data available
6a88	Referenced data not found
6a80	Incorrect values in command data
Load Errors	
6581	Memory failure

Return Code	Description
6a84	Not enough memory space
6a86	Incorrect P1/P2
6985	Conditions of use not satisfied

3.6.2. Events

- Simple events such as card insertions and removals, successfully completed operations, card operations that result in an error, and similar events.
- Errors are reported from the TPS to the Enterprise Security Client.
- The NSS crypto library is initialized.
- Other low-level smart card events are detected.

CHAPTER 4. USING SMART CARDS FOR WEB AND MAIL CLIENTS

After a smart card is enrolled, the smart card can be used for SSL client authentication and S/MIME email applications. The PKCS #11 module has different names and is located in different directories depending on the operating system.

Table 4.1. PKCS #11 Module Locations

Platform	Module Name	Location
Red Hat Enterprise Linux	onepin-opensc-pkcs11.so	/usr/lib64/

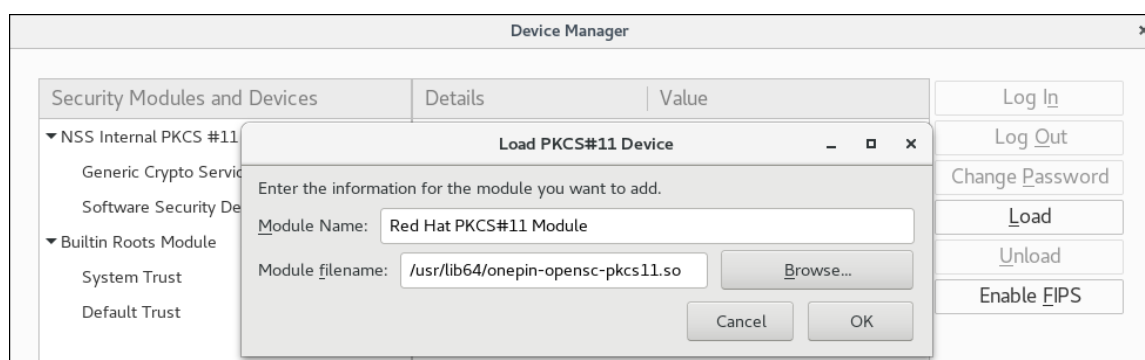
4.1. SETTING UP BROWSERS TO SUPPORT SSL FOR TOKENS

To set up the Firefox browser to support SSL for tokens:

1. Open the **Edit** menu and select **Preferences**.

If the menu bar is not visible in Firefox, press the **Alt** key to temporarily display it.

2. In the **Advanced** entry, select the **Certificates** tab, and click the **Security Devices** button.
3. Add the PKCS #11 driver:
 - a. Click the **Load** button.
 - b. Enter a module name.
 - c. Click **Browse**, select the Enterprise Security Client PKCS #11 driver library, and click **OK**.



4. If the CA is not yet trusted, download and import the CA certificate.

1. Open the **SSL End Entity** page on the CA. For example:

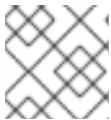
<https://server.example.com:9444/ca/ee/ca/>

2. Click the **Retrieval** tab, and then click **Import CA Certificate Chain**.
3. Click **Download the CA certificate chain in binary form** and then click **Submit**.
4. Choose a suitable directory to save the certificate chain, and then click **OK**.

5. Click **Edit > Preferences** and select the **Advanced** tab.
 6. Click the **View Certificates** button.
 7. Click **Authorities**, and import the CA certificate.
5. Set the certificate trust relationships.
1. Click **Edit > Preferences** and select the **Advanced** tab.
 2. Click the **View Certificates** button.
 3. Click **Edit**, and set the trust for websites.

The certificates can be used for SSL.

CHAPTER 5. SETTING UP ENTERPRISE SECURITY CLIENT



NOTE

The Enterprise Security Client can be launched without requiring extra configuration.

5.1. DISABLING LDAP AUTHENTICATION FOR TOKEN OPERATIONS

By default, each user who requests a token operation is authenticated against an LDAP directory. If the user has an entry, then the operation is allowed; if the user does not have an entry, then the operation is rejected.

For testing purposes or for certain types of users, then it can be simpler or preferable to disable LDAP authentication. This is not configured in the Enterprise Security Client configuration, but in the Token Processing System configuration, and must be done by a TPS administrator.

1. Stop the TPS subsystem.

```
# systemctl stop pki-tps
```

2. Open the TPS configuration file.

```
# vim /var/lib/pki-tps/conf/CS.cfg
```

3. Set the authentication parameters to **false**.

```
op.operation_type.token_type.loginRequest.enable=false
op.operation_type.token_type.auth.enable=false
```

The *operation_type* is the token operation for which LDAP authentication is being disabled, such as **enroll**, **format**, or **pinreset**. Disabling authentication for one operation type does not disable it for any other operation types.

The *token_type* is the token profile. There are default profiles for regular users, security officers, and the users enrolled by security officers. There can also be custom token types for other kinds of users or certificates.

For example:

```
op.enroll.userKey.loginRequest.enable=false
op.enroll.userKey.pinReset.enable=false
```

4. Restart the TPS subsystem.

```
# systemctl restart pki-tomcatd@pki-tomcat.service
```

Editing the TPS configuration is covered in the [Red Hat Certificate System 10 Administration Guide](#).

APPENDIX A. REVISION HISTORY

Note that revision numbers relate to the edition of this manual, not to version numbers of Red Hat Certificate System.

Revision 10.1-0

Fri Nov 20 2020

Florian Delehay

Published the guide for Red Hat Certificate System 10.1.