Red Hat Certificate System 7.3

Managing Smart Cards with the Enterprise Security Client

7.3

ISBN: N/A

Publication date: Updated August 5, 2008

Red Hat Certificate System 7.3

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

Red Hat Certificate System 7.3: Managing Smart Cards with the Enterprise Security Client

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

About This Guide

The **Enterprise Security Client** is a simple user interface which formats and manages smart cards. This guide is intended for everyday users of Certificate System, who will use the **Enterprise Security Client** to manage their smart cards. Certificate System agents should read the *Certificate System Agent's Guide* for information on how to perform agent tasks, such as handling certificate requests and revoking certificates.

Before reading this guide, be familiar with the following concepts:

- Public-key cryptography and the Secure Sockets Layer (SSL) protocol
- Intranet, extranet, and Internet security and the role of digital certificates in a secure enterprise
- LDAP and Red Hat Directory Server

1. What Is in This Guide

This guide contains the following chapters:

- *Chapter 1, Overview of the Enterprise Security Client* provides an introduction to the Certificate System.
- *Section 1, "Supported Platforms for the Client"* provides information about supported platforms for the **Enterprise Security Client**.
- *Chapter 2, Installing the Enterprise Security Client* provides information on how to install and uninstall the **Enterprise Security Client** on the supported platforms.
- *Chapter 3, Using the Enterprise Security Client* provides instructions on using the **Enterprise Security Client** for token enrollment, formating, and password reset operations.
- *Chapter 4, Using Enterprise Security Client Keys for SSL Client Authentication and S/MIME* describes how to use the **Enterprise Security Client** keys for SSL and S/MIME authentication.
- *Appendix A, Enterprise Security Client Configuration* provides information on configuring the **Enterprise Security Client**.

2. Additional Reading

The documentation for Red Hat Certificate System contains the following guides:

- *Certificate System Agent's Guide* details how to perform agent operations for the CA, DRM, OCSP, and TPS subsystems through the Certificate System agent services interfaces.

- *Certificate System Administration Guide* explains how to install, configure, and use Red Hat Certificate System.

Additional Certificate System information is provided in the Certificate System SDK, an online reference to HTTP interfaces, javadocs, samples, and tutorials related to Certificate System. A downloadable zip file of this material is available for user interaction with the tutorials.

For the latest information about Red Hat Certificate System, including current release notes, complete product documentation, technical notes, and deployment information, see *http://www.redhat.com/docs/manuals/cert-system/*.

3. Examples and Formatting

All of the examples for Red Hat Certificate System commands, file locations, and other usage are given for Red Hat Enterprise Linux 5 systems. Be certain to use the appropriate commands and files for your platform. For example:

To start the Red Hat Directory Server:

```
service dir-server start
```

Example 1. Example Command

Certain words are represented in different fonts, styles, and weights. Different character formatting is used to indicate the function or purpose of the phrase being highlighted.

Formatting Style	Purpose
`Monospace font`	Monospace is used for commands, package names, files and directory paths, and any text displayed in a prompt.

`Monospace`
`with a`
`background`
	This type of formatting is used for anything entered or returned in a command prompt.

Italicized text	Any text which is italicized is a variable, such as *instance_name* or *hostname*. Occasionally, this is also used to emphasize a new term or other phrase.
Bolded text	Most phrases which are in bold are application names, such as **Cygwin**, or are fields or options in a user interface, such as a **User Name Here:** field or **Save** button.

Other formatting styles draw attention to important text.

NOTE

A note provides additional information that can help illustrate the behavior of the system or provide more detail for a specific issue.

TIP

A tip is typically an alternative way of performing a task.

IMPORTANT

Important information is necessary, but possibly unexpected, such as a configuration change that will not persist after a reboot.

CAUTION and WARNING

A caution indicates an act that would violate your support agreement.

A warning indicates potential data loss, as may happen when tuning hardware for maximum performance.

4. Giving Feedback

If there is any error in this *Enterprise Security Client Guide* or there is any way to improve the documentation, please let us know. Bugs can be filed against the documentation for Red Hat Certificate System through Bugzilla, *http://bugzilla.redhat.com/bugzilla*. Make the bug report as specific as possible, so we can be more effective in correcting any issues:

- Select the Red Hat Certificate System product.
- Set the component to `Doc - enterprise-security-guide`.
- Set the version number to 7.3.
- For errors, give the page number (for the PDF) or URL (for the HTML), and give a succinct description of the problem, such as incorrect procedure or typo.

  For enhancements, put in what information needs to be added and why.

- Give a clear title for the bug. For example, `"Incorrect command example for setup script options"` is better than `"Bad example"`.

We appreciate receiving any feedback — requests for new sections, corrections, improvements, enhancements, even new ways of delivering the documentation or new styles of docs. You are welcome to contact Red Hat Content Services directly at *mailto:docs@redhat.com*.

5. Revision History

Revision History

Revision 7.3.2	Tuesday, August 5, 2008	Ella Deon Lackey`<dlackey@redhat.com>`

Additional edits to the token management system section, per review from Christina Fu and related to Bugzilla #455345.
Minor edits per request from Ade Lee, related to Bugzilla #454899, Bugzilla #454900, and Bugzilla #454901.

Revision 7.3.1	Friday, May 30, 2008	Ella DeonLackey`<dlackey@redhat.com>`

Added details about how the token management system works.
Updated supported smart cards and added new esc-prefs.js parameter per errata RHBA-2008:0276 and RHBA-2008:0277.

Revision 7.3.0-6	Tue May 15 2007	DavidO'Brien`<david.obrien@redhat.com>`

Further update related to Bugzilla 239636. Elaborated Security Officer Mode.

Revision 7.3.0-5	Mon May 14 2007	DavidO'Brien`<david.obrien@redhat.com>`

Addresses Bugzilla 239636.

Revision 7.3.0-4	Mon May 8 2007	DavidO'Brien`<david.obrien@redhat.com>`

Addresses Bugzilla 238662.

Revision 7.3.0-3	Mon May 8 2007	DavidO'Brien`<david.obrien@redhat.com>`

Added instructions for downloading CA certificate.

Revision 7.3.0-2	Mon May 7 2007	DavidO'Brien`<david.obrien@redhat.com>`

Addressed layout issues in appendix.
Revised images and procedures for using certificates and SSL.

Revision 7.3.0-1	Thur May 3 2007	DavidO'Brien`<david.obrien@redhat.com>`

Retagged procedures for presentation.
Addressed Bugzilla 238667.

Overview of the Enterprise Security Client

The **Enterprise Security Client** is a tool for Red Hat Certificate System which simplifies managing smart cards. End users can use security tokens (smart cards) to store user certificates used for applications such as single sign-on 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 required to process token-related operations; optionally, the Data Recovery Manager (DRM) can be used with the token management system for server-side key generation and key archival and recovery. 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. Enterprise Security Client provides the following capabilities:

- Supports Global Platform-compliant smart cards like Gemalto Cyberflex Access e-gate 32K and Cyberflex Access 64K V2 Standard tokens.
- 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 so that keys can be archived and recovered on a separate token if a token is lost.

1. About Smart Card Management

Certificate System creates, manages, renews, and revokes certificates, as well as archiving and recovering keys. For organizations which use smart cards, the Certificate System has a token management system — a collection of subsystems with established relationships — to generate keys and requests and receive certificates to be used for smart cards. These relationships are show in *Figure 1.1, “How Certificate System Manages Smart Cards”*.

Four Certificate System subsystems are involved with managing tokens:

- 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 stored on the TPS 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.

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.

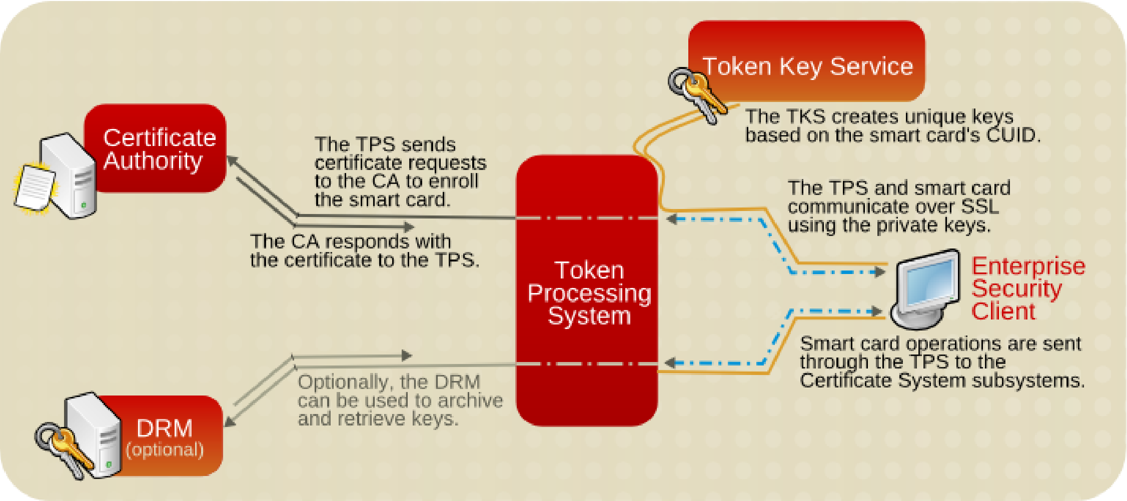

Token Key Service
The TKS creates unique keys based on the smart card's CUID.
Certificate Authority
The TPS sends certificate requests to the CA to enroll the smart card.
The CA responds with the certificate to the TPS.
The TPS and smart card communicate over SSL using the private keys.
Token Processing System
Enterprise Security Client
Smart card operations are sent through the TPS to the Certificate System subsystems.
Optionally, the DRM can be used to archive and retrieve keys.
DRM
(optional)

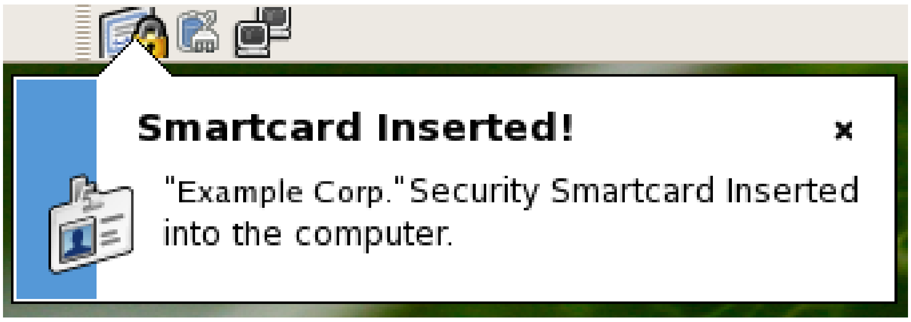
Figure 1.1. How Certificate System Manages Smart Cards

To use the tokens, the Token Processing System must be able to recognize and communicate with them. The tokens must first be *enrolled* to format 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.

2. Features

- The *Phone Home* feature defines the token issuer name, TPS server, and TPS end-entities interface URL without requiring any user configuration.
- Enterprise Security Client has diagnostic logging that records common access and events and records potential errors such as interruptions with the connection between the Enterprise Security Client and TPS server.

- The Enterprise Security Client user interface incorporates Mozilla XULRunner technology. XULRunner is a runtime package which hosts standalone applications based on XUL, an XML markup language with a rich feature set for user interfaces. XUL has the following advantages over HTML for applications:
  - XUL provides a wide UI widget set and greater control over the presentation.
  - XUL markup is local to the client machine, so it has a greater privilege level than HTML.
  - XUL also uses Javascript as the scripting language for convenient program logic scripting.
  - XUL Javascript code can make use of the array of Mozilla functionality by using their XPCOM technology.
- The Mac Enterprise Security Client ships with a smart card-specific TokenD component which bridges the gap between Certificate System-supported tokens and the Mac CDSA security layer, allowing current OS X applications like Apple Mail and Safari to take advantage of the capabilities of Certificate System tokens:
  - The Mac Keychain Access utility can be used to view the certificates and keys on Certificate System tokens.
  - The Apple Mail client can be used to send and view signed and encrypted emails using Certificate System tokens.
  - The Apple Safari browser can use Certificate System tokens to log onto secure SSL web sites.
- This version of Enterprise Security Client provides tray icon functionality on all three platforms, including tool tips for errors and actions such as inserting or removing a smart card.



Smartcard Inserted!

"Example Corp."Security Smartcard Inserted into the computer.

Figure 1.2. Example Token Tray Icon and Tool Tip

On most operating systems, many programs maintain an icon in the tray or notification area. These icons can be used to control the operation of the program, usually through context menus when the icon is right-clicked. In the default Enterprise Security Client configuration, Enterprise Security Client launches and automatically minimizes to the tray. This tray functionality behaves differently on the different operating systems:

- *Windows.* When right-clicked, the tray icon shows a simple menu with options to **Manage Smart Card**, which opens the Enterprise Security Client interface, and to **Exit Smart Card Manager**, which exits the Enterprise Security Client. The exit option in that menu is the only want to exist the Enterprise Security Client on Windows; clicking the **X** in the top right corner minimizes Enterprise Security Client to the tray. Double-clicking the tray icon brings Enterprise Security Client to the front. There are also notification messages, shown as standard balloon tool tips, on events like inserting or removing a card.

- *Linux.* The tray icon appears only if the notification area in Gnome has been enabled. The tray icon options are identical to the Windows options. Clicking the **X** in the top left corner closes the current window and minimizes Enterprise Security Client to the tray.

- *Mac.* On Mac, the tray is called the dock. Since Enterprise Security Client is based on Mozilla, right-clicking on the Enterprise Security Client dock icon reveals all the standard Mozilla Firefox menu options, including options to hide, show, and quit the client. The Enterprise Security Client also has a menu item called **Manage Smart Cards** in the dock menu, which opens the card management UI. The top level application menu has a menu under **Go**, **Manage Smart Card**, which also opens the card management window.

Installing the Enterprise Security Client

The **Enterprise Security Client** is packaged as a set of installation executables or RPMs and other files that are part of the complete Red Hat Certificate System distribution. These are listed in the installation chapter of the *Certificate System Administrator's Guide*.

1. Supported Platforms for the Client

The Enterprise Security Client interface is supported on the following platforms:

- Red Hat Enterprise Linux 4 AS (Intel x86)
- Red Hat Enterprise Linux 4 ES (Intel x86)
- Microsoft Windows XP
- Apple MAC OS X 10.4.x (Tiger)

2. Supported Smart Cards

Enterprise Security Client supports the following smart cards:

- Global Platform-compliant smart cards such as Gemalto Cyberflex Access 32K e-gate tokens
- Gemalto Cyberflex Access 64K V2 Standard smart cards, with the DER SHA1 value configured as described in PKCS#1 v2.1

3. Installing and Uninstalling the Enterprise Security Client on Red Hat Enterprise Linux

3.1. Installing the Client

The first step in installing the **Enterprise Security Client** is to download the required packages. The *Certificate System Administrator's Guide* explains how to retrieve these RPMs and other files through the **Red Hat Certificate System 7.3 (AS v.4 for x86)** or **Red Hat Certificate System 7.3 (ES v.4 for x86)** Red Hat Network channels. There are two ways to obtain the packages:

- Downloading an ISO image or packages through the Red Hat Network channel
- Using the Red Hat `up2date` utility

The preferred method of obtaining RPMs is using the up2date command-line utility, as follows:

```
# up2date esc
```

If the up2date command completes successfully, all of the necessary **Enterprise Security Client** RPMs will be installed and ready for use.

> **NOTE**
>
> If the up2date utility was used 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. Copy the **Enterprise Security Client** installation RPMs packaged with Red Hat Certificate System.

2. Install the RPMs as root in the following order:

```
# rpm -ivh ccid-1.2.1-1.el4.i386.rpm
# rpm -ivh pcsc-lite-1.3.3-1.el4.i386.rpm
# rpm -ivh pcsc-lite-libs-1.3.3-1.el4.i386.rpm
# rpm -ivh ifd-egate-0.05-15.i386.rpm
# rmp -ivh coolkey-1.1.0-1.el4.i386.rpm
# rpm -ivh esc-1.0.1-1.el4.i386.rpm
```

> **NOTE**
>
> Update the version numbers of the RPM files to match the current version.

The **Enterprise Security Client** installation is located in /usr/lib/esc-1.0.1. The esc shell script is installed in /usr/bin/esc. The **Enterprise Security Client** can be launched by typing esc at a command prompt.

The **Enterprise Security Client** for Linux implements a daemon (escd) which 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. The client can also be launched manually by selecting **System Settings**, then **Smart Card Manager**, from the **System** menu.

3.2. Uninstalling on Red Hat Enterprise Linux

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

> **NOTE**
>
> Update the version numbers of the RPM files to match your version.

```
# rpm -ev ccid
# rpm -ev pcsc-lite
# rpm -ev pcsc-lite-libs
# rpm -ev ifd-egate
# rpm -ev coolkey
# rpm -ev esc
```

4. Remove any remaining files in the installation directory.

4. Installing and Uninstalling on Windows

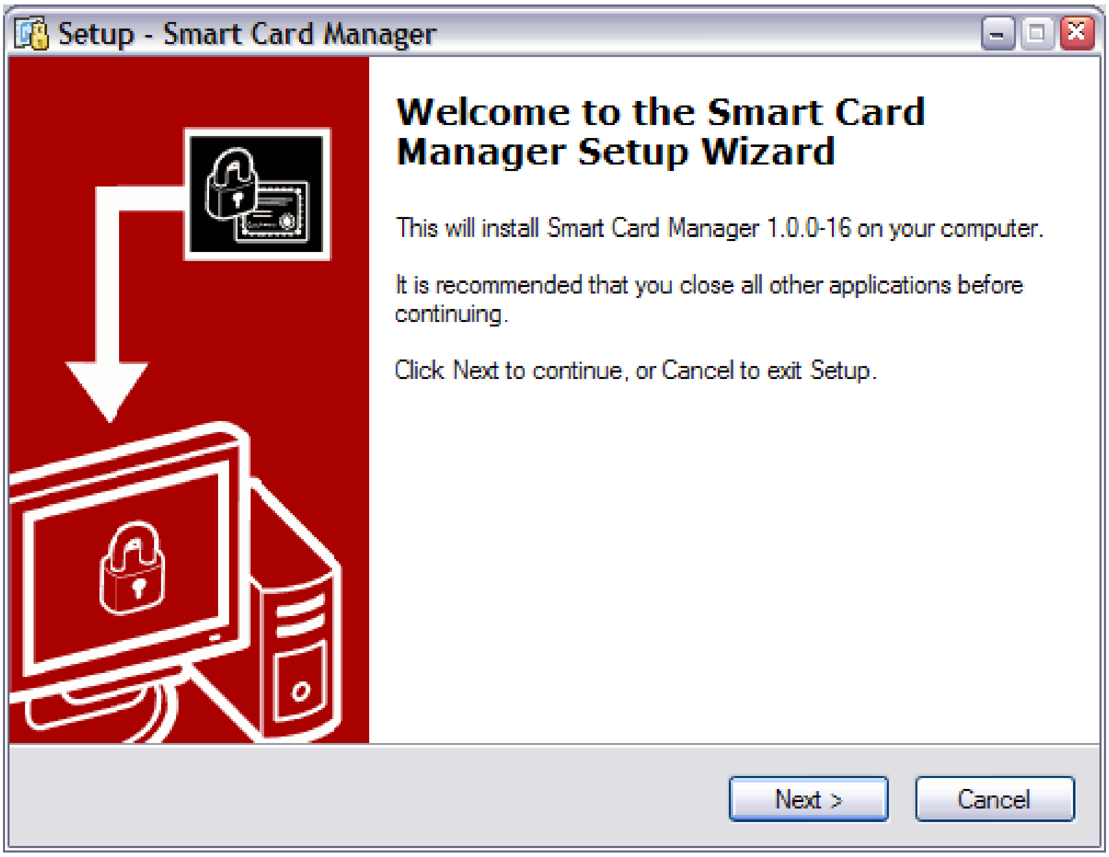
4.1. Installing the Client

The Windows **Enterprise Security Client** packages are available in the **Downloads** area of Red Hat Network. There are two channels for the packages; Windows clients are available in 32-bit. The Windows **Enterprise Security Client** package is called `SmartCardManagerSetup`*version*`.exe`.

To install the **Enterprise Security Client** on Windows:

> **NOTE**
>
> You may need administrator privileges to install the **Enterprise Security Client** on the Windows system.

1. Download the Windows **Enterprise Security Client** installer (`SmartCardManagerSetup-1.0.1-X.win32.i386.exe`) from the Red Hat Network channel.
2. Double-click the `SmartCardManagerSetup-1.0.1-X.win32.i386.exe` file to launch the **Enterprise Security Client** installation program.



Setup - Smart Card Manager

Welcome to the Smart Card Manager Setup Wizard

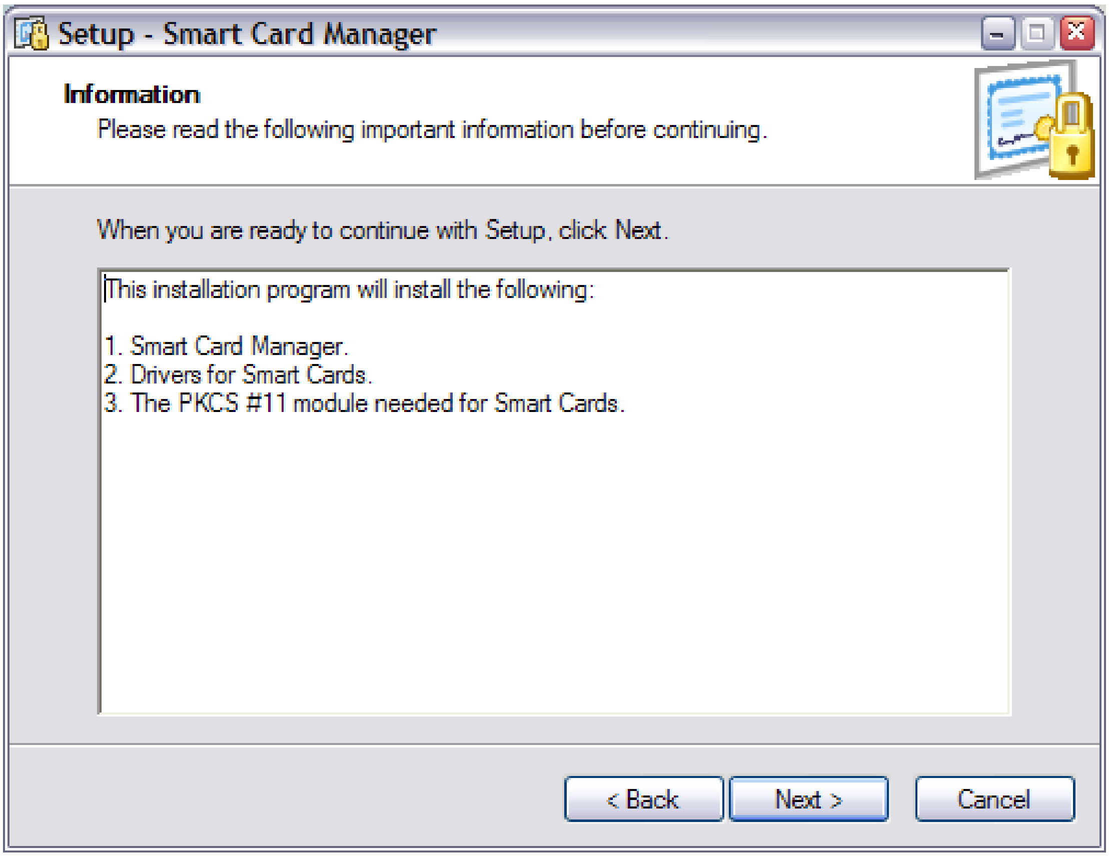
This will install Smart Card Manager 1.0.0-16 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

Figure 2.1. Launching the Installation Wizard

3. The wizard displays the list of packages that will be installed.



Information

Please read the following important information before continuing.

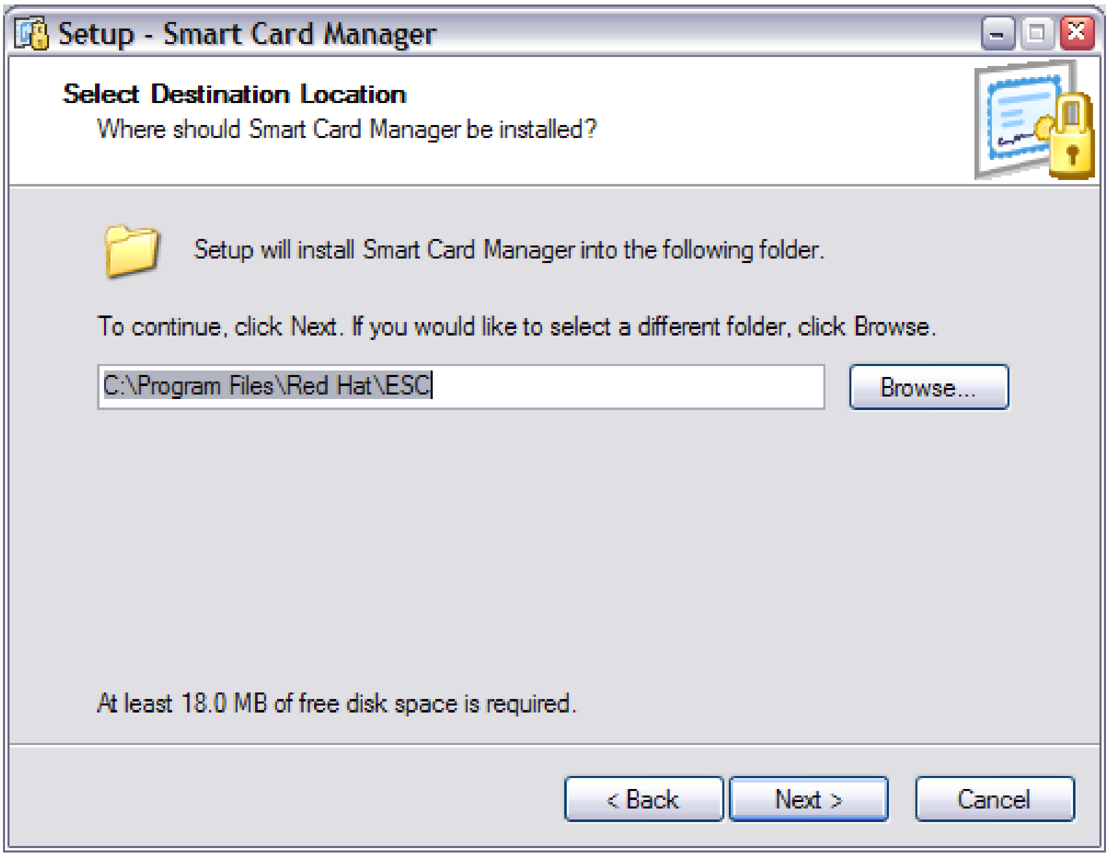
When you are ready to continue with Setup, click Next.

This installation program will install the following:

1. Smart Card Manager.
2. Drivers for Smart Cards.
3. The PKCS #11 module needed for Smart Cards.

Figure 2.2. Launching the Installation Wizard on Windows

4. The wizard prompts for the installation directory for the **Enterprise Security Client**. The default directory is `C:\Program Files\Red Hat\ESC`.



Select Destination Location

Where should Smart Card Manager be installed?

Setup will install Smart Card Manager into the following folder.

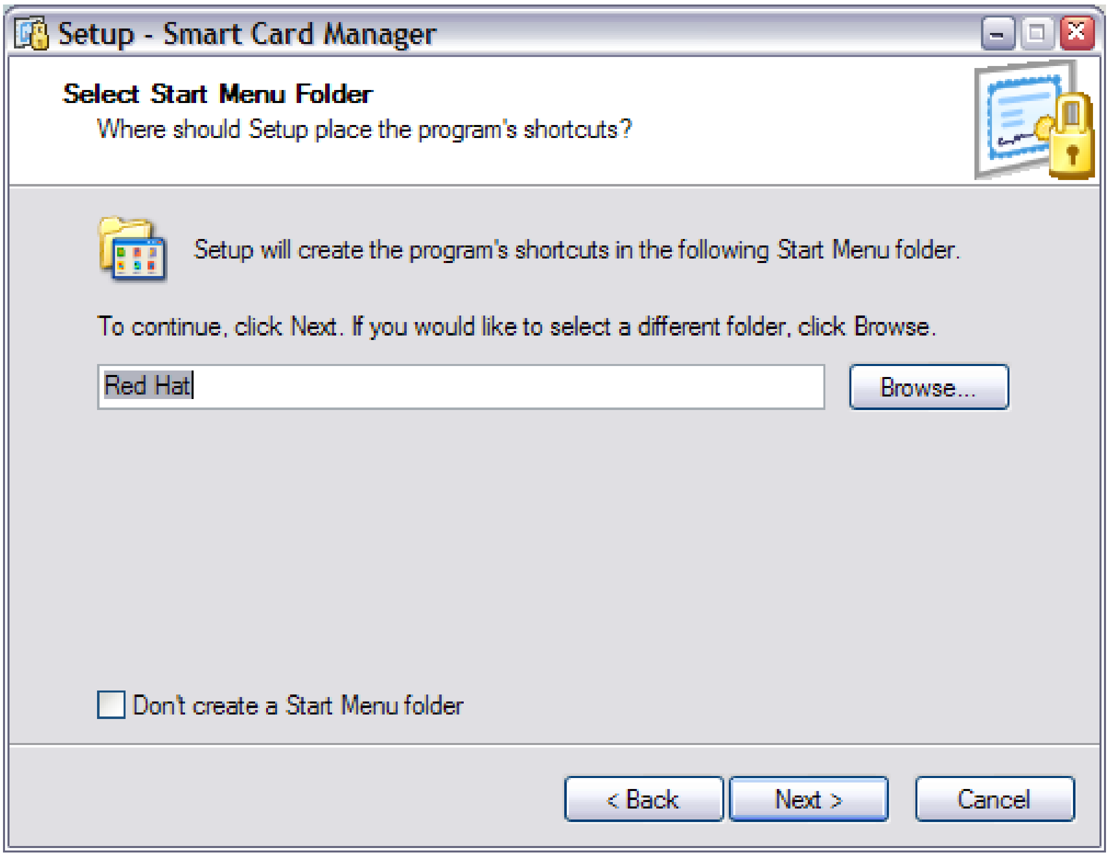
To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files\Red Hat\ESC

At least 18.0 MB of free disk space is required.

Figure 2.3. Specifying the Installation Directory

5. The wizard prompts for the Start Menu directory for the **Enterprise Security Client**. The default directory is `Red Hat`.



Setup - Smart Card Manager

Select Start Menu Folder

Where should Setup place the program's shortcuts?

Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

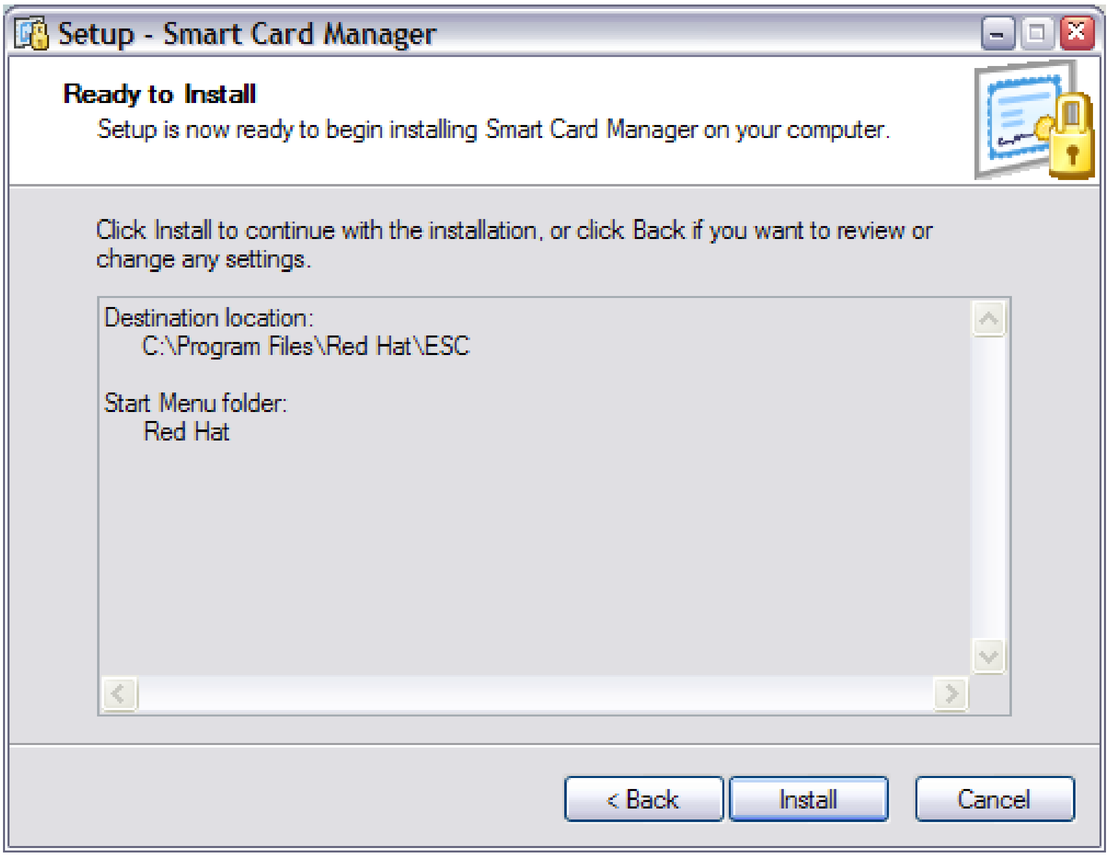
Red Hat

Figure 2.4. Specifying the Start Menu Directory

6. Proceed through the **Enterprise Security Client** installation wizard. Click **Install** to begin installing the **Enterprise Security Client** components.

NOTE

The installation process also installs the CoolKey PKCS #11 driver and e-gate drivers needed for Certificate System-supported keys and automatically installs the Certificate System PKCS #11 module in any Mozilla browsers it can locate. The installer places the Certificate System Cryptographic Service Provider (CSP) on the user's system to allow users to use their smart cards with Microsoft products such as Outlook and Internet Explorer.



Ready to Install

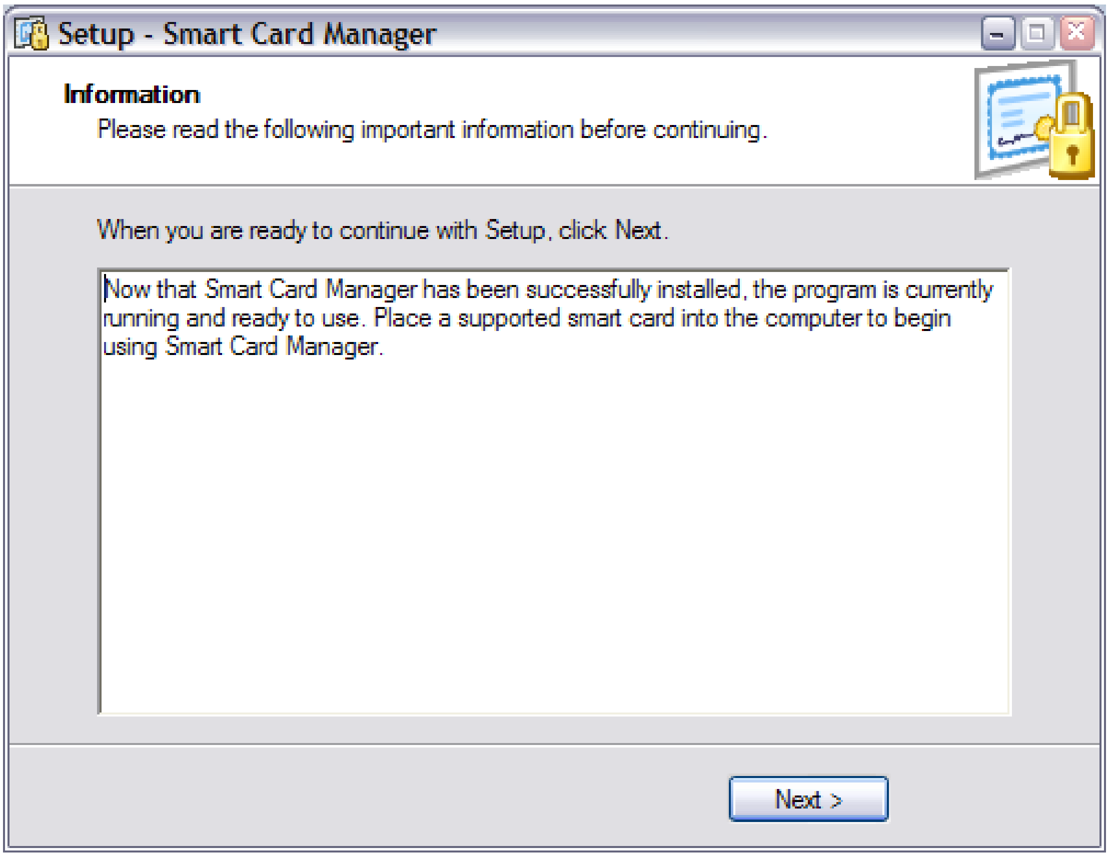
Setup is now ready to begin installing Smart Card Manager on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

```
Destination location:
    C:\Program Files\Red Hat\ESC

Start Menu folder:
    Red Hat
```

Figure 2.5. Ready to Start the Installation

7. When the installation has completed, the **Enterprise Security Client** will prompt the user to insert a token, and can then be launched for immediate use.



Information

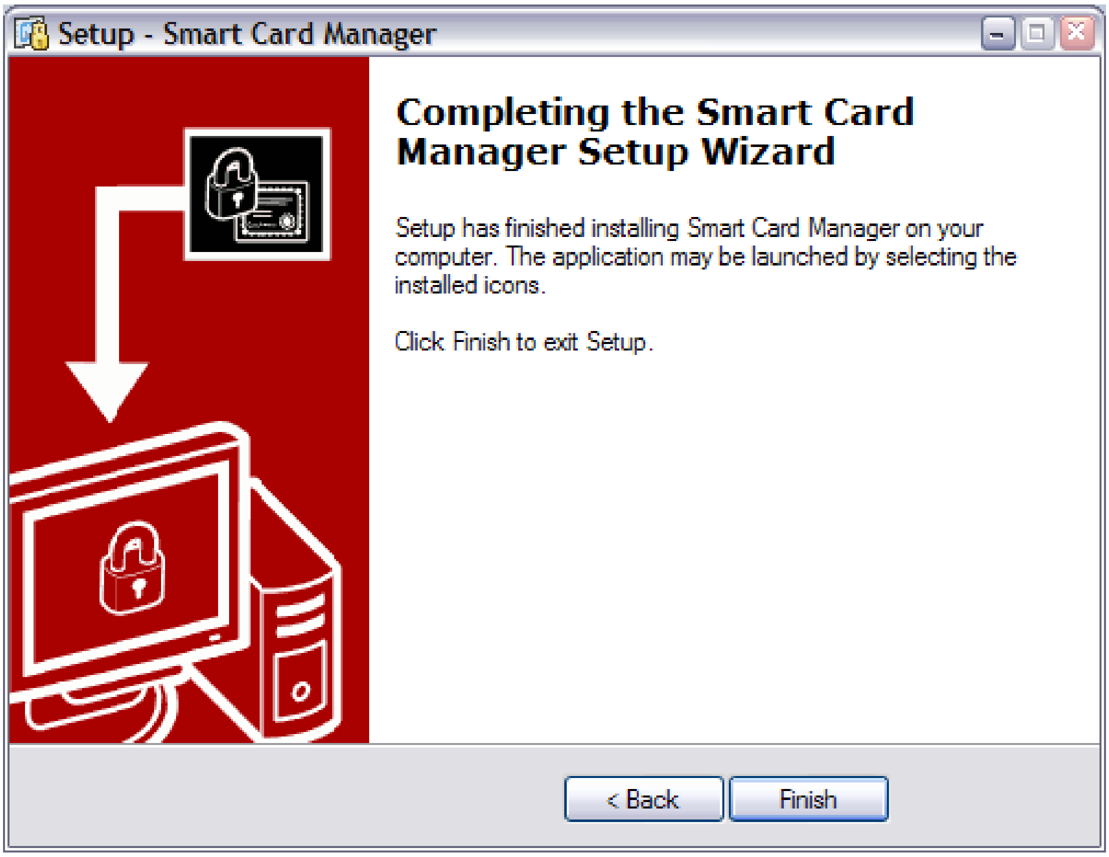
Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

Now that Smart Card Manager has been successfully installed, the program is currently running and ready to use. Place a supported smart card into the computer to begin using Smart Card Manager.

Figure 2.6. Launching the Smart Card Manager

8. Click **Finish** to complete the installation.



Setup - Smart Card Manager

Completing the Smart Card Manager Setup Wizard

Setup has finished installing Smart Card Manager on your computer. The application may be launched by selecting the installed icons.

Click Finish to exit Setup.

Figure 2.7. Completing the Installation

4.2. Uninstalling the Client

1. Unplug all USB tokens.
2. Stop the **Enterprise Security Client**.
3. Open the **Control Panel**, and click the **Add Remove Programs** icon.
4. In the list of available programs, click **Smart Card Manager**, and click **Change/Remove**.
5. When the uninstallation is complete, remove any remaining files in the installation directory.

5. Installing and Uninstalling the Enterprise Security Client on Mac OS X

5.1. Installing the Client

The Mac **Enterprise Security Client** packages are available in the **Downloads** area of Red Hat Network. There are two channels for the packages; Mac clients are available in 32-bit. The Mac **Smart Card Manager** package is `SmartCardManager`*version*`.dmg`.

To install the **Enterprise Security Client** on Mac OS X:

1. Download the `SmartCardManager-1.0.1-X.OSX4.darwin.dmg` file from the Red Hat Network channel.

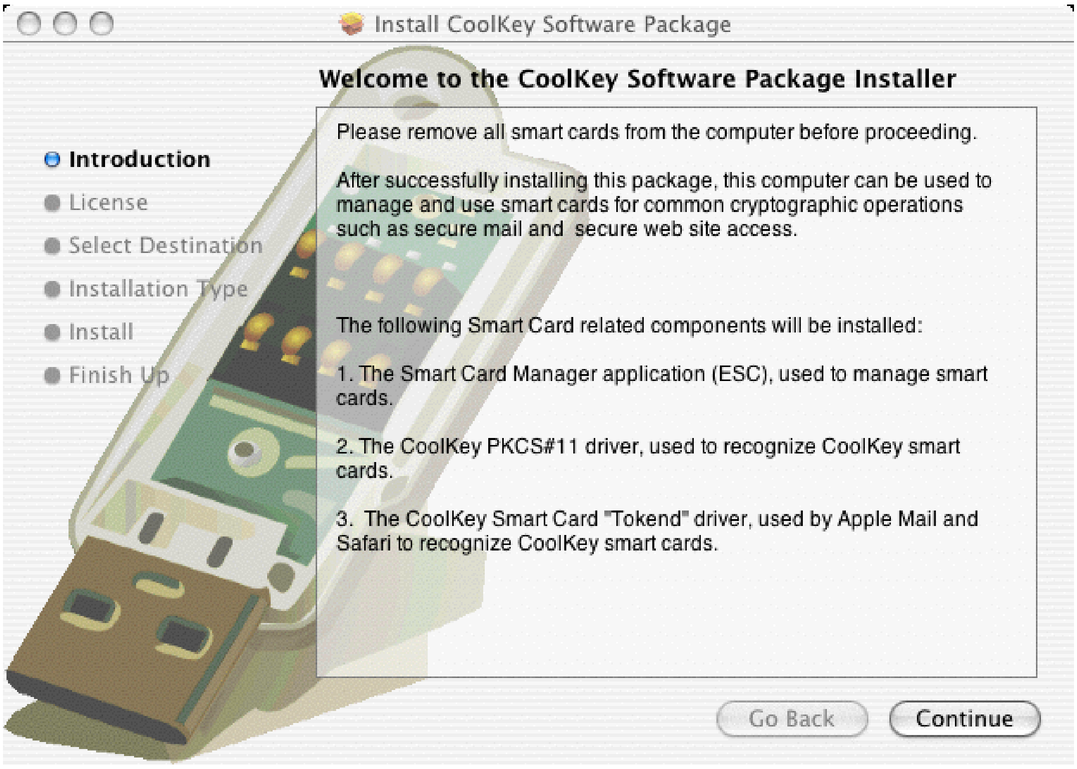
2. Double-click the `SmartCardManager-1.0.1-X.OSX4.darwin.dmg` file to display the **Enterprise Security Client Volume**.

   Inside the Volume are two directories, `ESC.app` and `Coolkey1.14.pkg`. `ESC.app` is the **Enterprise Security Client** application, and `Coolkey1.14.pkg` is the installer for the token support software, including the TokenD system.

3. Drag the `ESC.app` file to an accessible location (for example, the desktop), to install the **Enterprise Security Client**.

4. Install the CoolKey package, as follows:

   a. Double-click the `Coolkey1.14.pkg` file to launch the CoolKey installer, and follow the directions to complete the installation.



Welcome to the CoolKey Software Package Installer

Please remove all smart cards from the computer before proceeding.

After successfully installing this package, this computer can be used to manage and use smart cards for common cryptographic operations such as secure mail and secure web site access.

The following Smart Card related components will be installed:

1. The Smart Card Manager application (ESC), used to manage smart cards.

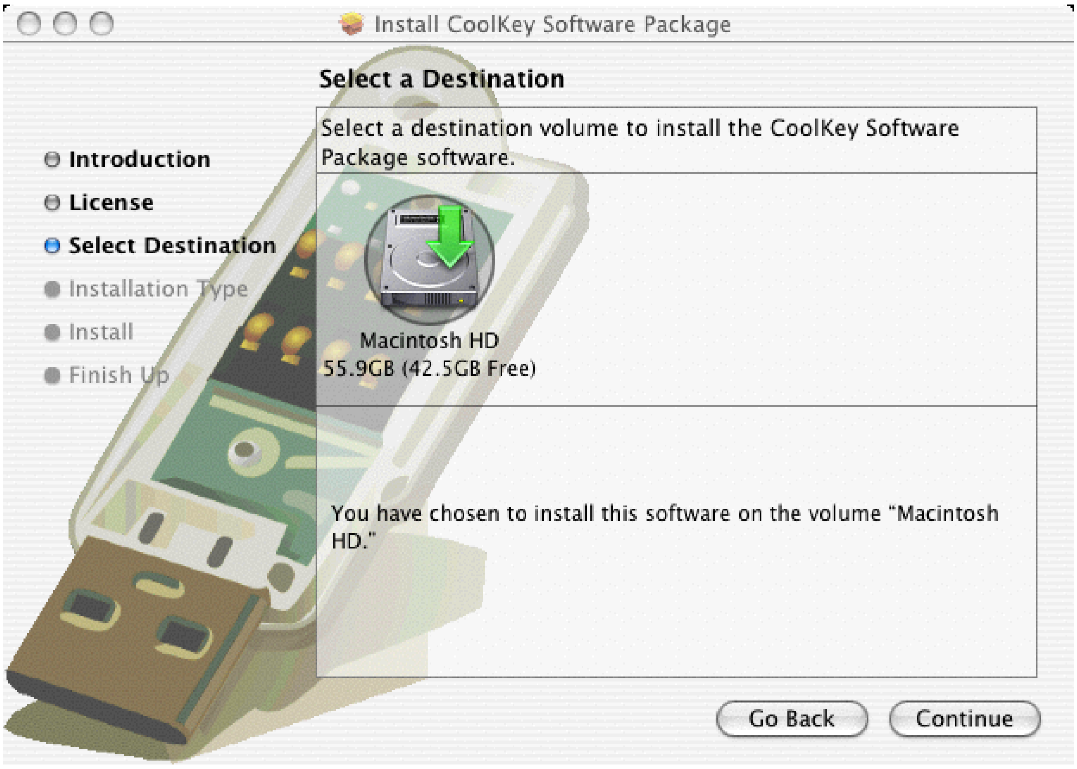
2. The CoolKey PKCS#11 driver, used to recognize CoolKey smart cards.

3. The CoolKey Smart Card "Tokend" driver, used by Apple Mail and Safari to recognize CoolKey smart cards.

Figure 2.8. Mac Installation Program

b. Read the Software License Agreement, and click **Continue** if you accept the terms.

c. Select the installation destination.



Install CoolKey Software Package

Select a Destination

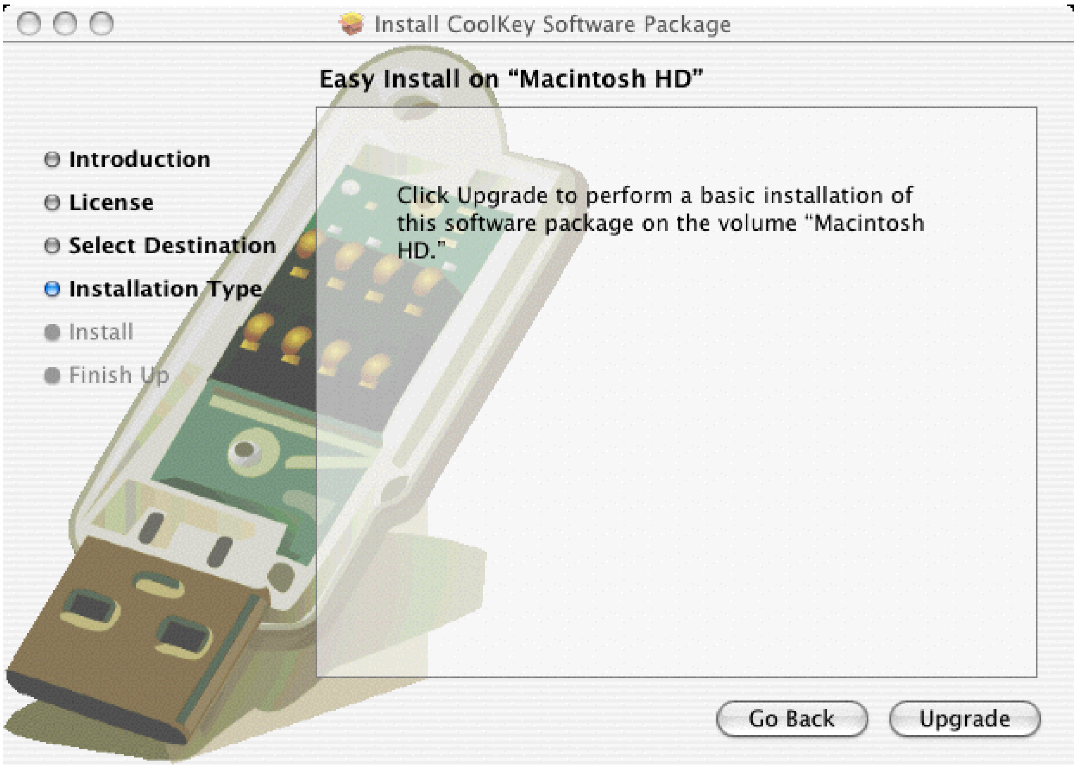
Select a destination volume to install the CoolKey Software Package software.

Macintosh HD
55.9GB (42.5GB Free)

You have chosen to install this software on the volume "Macintosh HD."

Figure 2.9. Specifying the installation destination

d. Click **Upgrade** (or **Install**, if shown), to begin the installation.



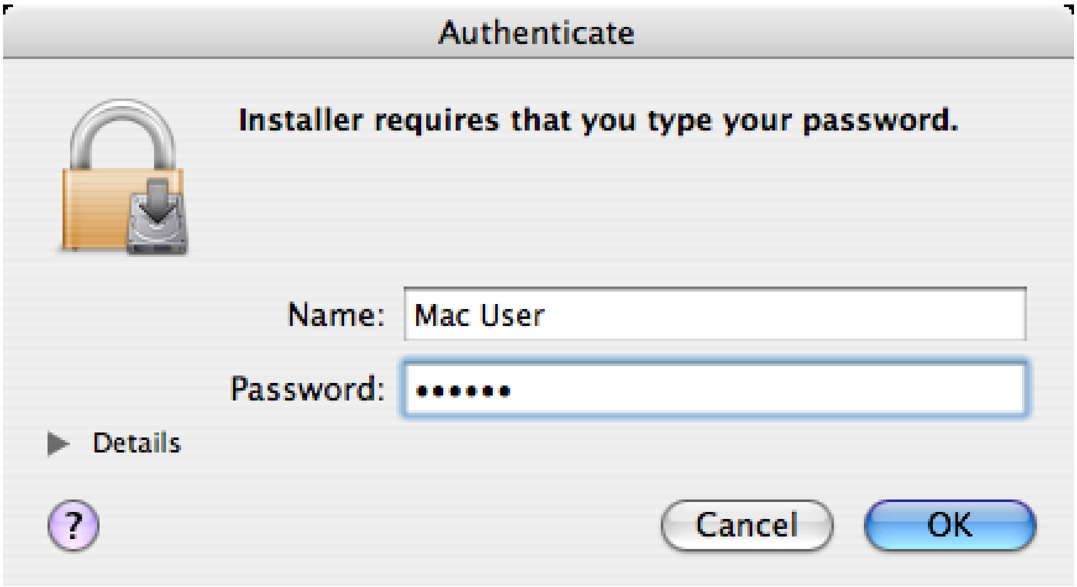
Install CoolKey Software Package

Easy Install on "Macintosh HD"

Click Upgrade to perform a basic installation of this software package on the volume "Macintosh HD."

Figure 2.10. Launch Installation

e. Enter the administrator password, and click **OK** to start the installation.



Authenticate

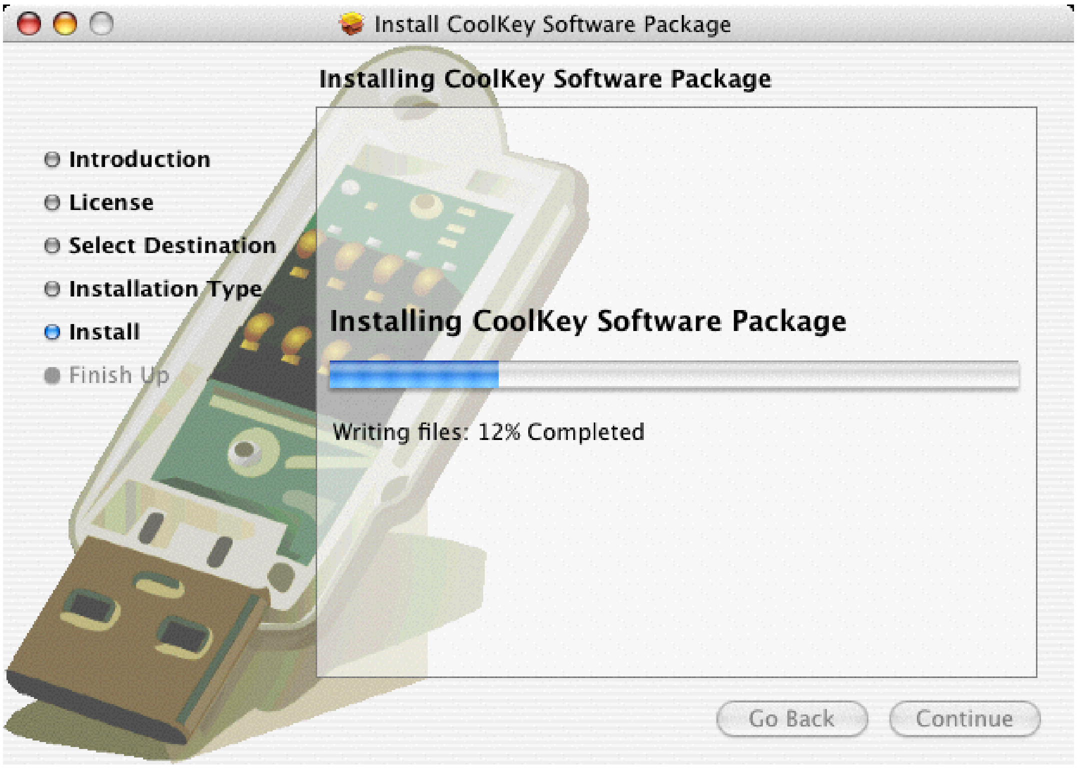
Installer requires that you type your password.

Name: Mac User

Password: ••••••

Figure 2.11. Entering the Administrator Password

f. When the installation is complete, click **Close**.



Installing CoolKey Software Package

Writing files: 12% Completed

Figure 2.12. Coolkey Software Package installation in progress

When the process is complete, the e-gate token drivers, the PKCS11 module, and the TokenD software are all installed on the local system.

5.2. Uninstalling the Client

1. Unplug all USB tokens.

2. Stop the **Enterprise Security Client**.

3. Delete the `ESC.app` icon.

> **NOTE**
>
> There is no uninstallation program for the Mac.

Using the Enterprise Security Client

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

1. Launching Enterprise Security Client

Each of the supported operating systems requires a slightly different method of starting the **Enterprise Security Client,** as follows:

Red Hat Enterprise Linux 4

> Open a command shell and type `esc`. This starts the **Enterprise Security Client** daemon (`escd`), which listens for inserted smart cards.
>
> Alternatively, click **Applications**, **System Settings**, and then **Smart Card Manager**.

Microsoft Windows

> Double-click the **Enterprise Security Client** icon on the desktop, or from the **Start** menu. The **Enterprise Security Client** is also configured to start on reboot.

Mac OS X

> Double-click the **Enterprise Security Client** icon wherever the application was installed.

2. Phone Home

The **Enterprise Security Client** provides a feature called *Phone Home*. This feature 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 connects to the TPS server and retrieves 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.tokenKey.issuerinfo.enable=true
op.format.tokenKey.issuerinfo.value=http://server.example.com
```

2.1. About Phone Home Profiles

The **Enterprise Security Client** is based on Mozilla XULRunner. Consequently, each user has a profile similar to the user profiles used by Mozilla Firefox and Thunderbird. The **Enterprise Security Client** accesses the configuration preferences file. 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.

2.2. Setting Global Phone Home Information

Phone Home is triggered automatically when a security token is inserted into a machine. The system immediately attempts to read the Phone Home URL from the token and to contact the TPS server. For new tokens or for previously formatted tokens, the Phone Home information may not be available to the card.

The **Enterprise Security Client** configuration file, `esc-prefs.js`, has a parameter which allows a global Phone Home URL default to be set. This parameter is *`esc.global.phone.home.url`* and is not in the file by default.

To define the global Phone Home URL:

1. Remove any existing **Enterprise Security Client** user profile directory. Profile directories are created automatically when a smart card is inserted.

   - On Red Hat Enterprise Linux, the profile directory is `~/.redhat/esc`.

   - On Windows, the profile directory is `C:/Documents and Settings/`*user_name*`/Application Data/RedHat/ESC`.

   - On Mac, the profile directory is `~/Library/Application Support/ESC`.

2. Open the `esc-prefs.js` file.

   - On Red Hat Enterprise Linux, the profile directory is `/usr/lib/esc-1.0.1/defaults/preferences`.

   - On Windows, the profile directory is `C:/Program Files/Red Hat/ESC/defaults/preferences`.

   - On Mac, the profile directory is

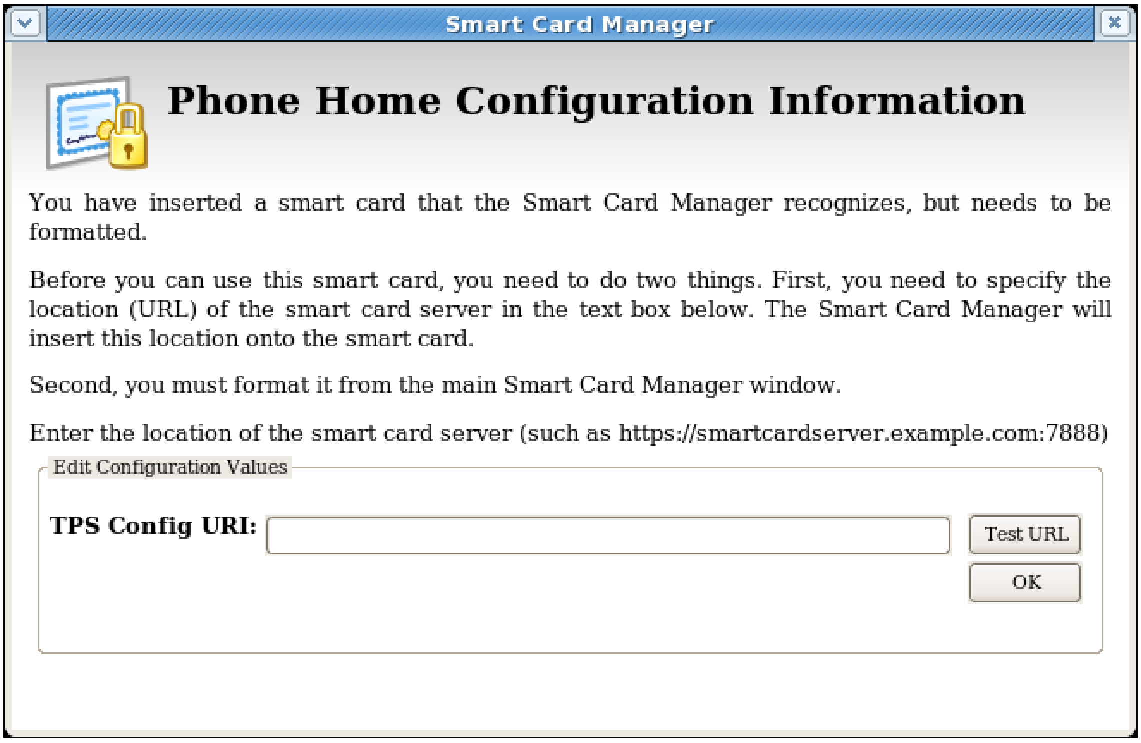
`/Applications/ESC.app/Contents/Resources/defaults/preferences`.

3. Add the global Phone Home parameter line. For example:

```
pref("esc.global.phone.home.url","http://tps.example.com:7888/cgi-bin/home.cgi");
```

When a smart card is inserted and Phone Home is launched, the **Enterprise Security Client** first checks the token for the Phone Home information. If no information is on the token, then the client checks the `esc-prefs.js` file for the *`esc.global.phone.home.url`* parameter.

If no Phone Home information is stored on the token and there is no global Phone Home parameter, the user is prompted for the Phone Home URL when a smart card is inserted, as shown in *Figure 3.1, “Prompt for Phone Home Information”*. 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.



Phone Home Configuration Information

You have inserted a smart card that the Smart Card Manager recognizes, but needs to be formatted.

Before you can use this smart card, you need to do two things. First, you need to specify the location (URL) of the smart card server in the text box below. The Smart Card Manager will insert this location onto the smart card.

Second, you must format it from the main Smart Card Manager window.

Enter the location of the smart card server (such as https://smartcardserver.example.com:7888)

Edit Configuration Values

TPS Config URI:

Figure 3.1. Prompt for Phone Home Information

2.3. Adding Phone Home Information to a Token Manually

The Phone Home information can be manually put on a token in one of two ways:

- The preferred method is that the information is burned onto the token at the factory. When the tokens are ordered from the manufacturer, the company should also supply detailed information on how the tokens should be configured when shipped.
- If tokens are blank, the company IT department can supply the information when formating small groups of tokens.

The following information is used by the Phone Home feature for each smart card:

- The TPS server and port. For example:

```
"esc.key.40900062ff020000ba87.tps.url" =
"http://tps.example.com:12443//nk_service"
```

- The TPS enrollment interface URL. For example:

```
"esc.key.40900062ff020000ba87.tps.url" =
"http://tps.example.com:12443/cgi_bin/esc.cgi?"
```

- The issuing company name or ID. For example:

```
"esc.key.40900062ff020000ba87.issuer.name" = "Example Corp"
```

- The Phone Home URL. For example:

```
"esc.key.40900062ff020000ba87.phone.home.url" =
"http://tps.example.com:12443/phone_home/phone_home.cgi?"
```

- Optionally, a default browser URL to access when an enrolled smart card is inserted.

```
"esc.key.40900062ff020000ba87.EnrolledTokenBrowserURL" =
"http://www.test.example.com"
```

2.4. Configuring the TPS to Use Phone Home

The Phone Home feature and the different type of information used by it only work when the TPS has been properly configured to use Phone Home. If the TPS is not configured for Phone Home, then this feature is ignored. The configuration for Phone Home is configured in the `index.cgi` in the `/var/lib/pki-tps/cgi-bin/home`; 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.

```
<ServiceInfo><IssuerName>Example Corp</IssuerName>
    <Services>
        <Operation>http://tps.example.com:12443/nk_service ## TPS server URL
        </Operation>
        <UI>http://tps.example.com:12443/cgi_bin/esc.cgi   ## Optional
Enrollment UI
        </UI>
        <EnrolledTokenBrowserURL>http://www.test.url.com   ## Optional
enrolled token url
        </EnrolledTokenBrowserURL>
    </Services>
</ServiceInfo>
```

Example 3.1. TPS Phone Home Configuration File

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 `https://test.example.com:12443/cgi-bin/home/index.cgi`. 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. Windows Cryptographic Service Provider

The Microsoft Windows version of the **Enterprise Security Client** installs a *Windows Cryptographic Service Provider* (CSP) that is compatible with the Certificate System-supported smart cards.

Microsoft Windows supports a software library designed to implement the Microsoft Cryptographic Application Programming Interface (CAPI). CAPI allows Windows-based applications, such as the Windows version of the **Enterprise Security Client**, to be developed to perform secure, cryptographic functions. This API, also known as CryptoAPI, provides a layer between an application which supports it, such as Certificate System, and the details of the cryptographic services provided by the API.

The CAPI interface can be used to create custom CSP libraries. In Certificate System, custom CSP libraries have been created to use the Certificate System-supported smart cards.

CAPI Store.

The *CAPI store* is a repository controlled by Windows, and which houses a collection of digital certificates associated with a given CSP. CAPI oversees the certificates, while each CSP controls the cryptographic keys belonging to the certificates.

Certificate System CSP.

The Certificate System CSP is designed to provide cryptographic functions on behalf of Windows using our supported smart cards. The Windows CSP performs its requested cryptographic functionality by calling the Certificate System PKCS #11 module.

The Certificate System CSP, which has been signed by Microsoft, provides the following features:

- Allows the user to send and receive encrypted and signed emails with **Microsoft Outlook**.
- Allows the user to visit SSL-protected websites with **Microsoft Internet Explorer**.
- Allows the user to use smart cards with certain VPN clients, which provides secure access to protected networks.

The required CSP libraries are automatically installed with the **Enterprise Security Client**. There are several common situations when a Windows user interacts directly with the CSP.

- When a smart card is enrolled with the **Enterprise Security Client**, the newly created certificates are automatically inserted into the user's CAPI store.
- When a smart card is formatted, the certificates associated with that card are removed from the CAPI store.
- When using applications such as **Microsoft Outlook** or **Microsoft Internet Explorer**, the user may be prompted to enter the smart card's password. This is required when the smart card is asked to perform protected cryptographic operations such as creating digital signatures.

4. Smart Card Auto Enrollment

Because the **Enterprise Security Client** is configured using the Phone Home feature, enrolling a smart card is extremely easy. Because the information needed to contact the backend TPS server is provided with each smart card, the user is guided quickly and easily through the procedure.

The following procedure describes how to enroll an uninitialized smart card. To enroll an uninitialized smart card:

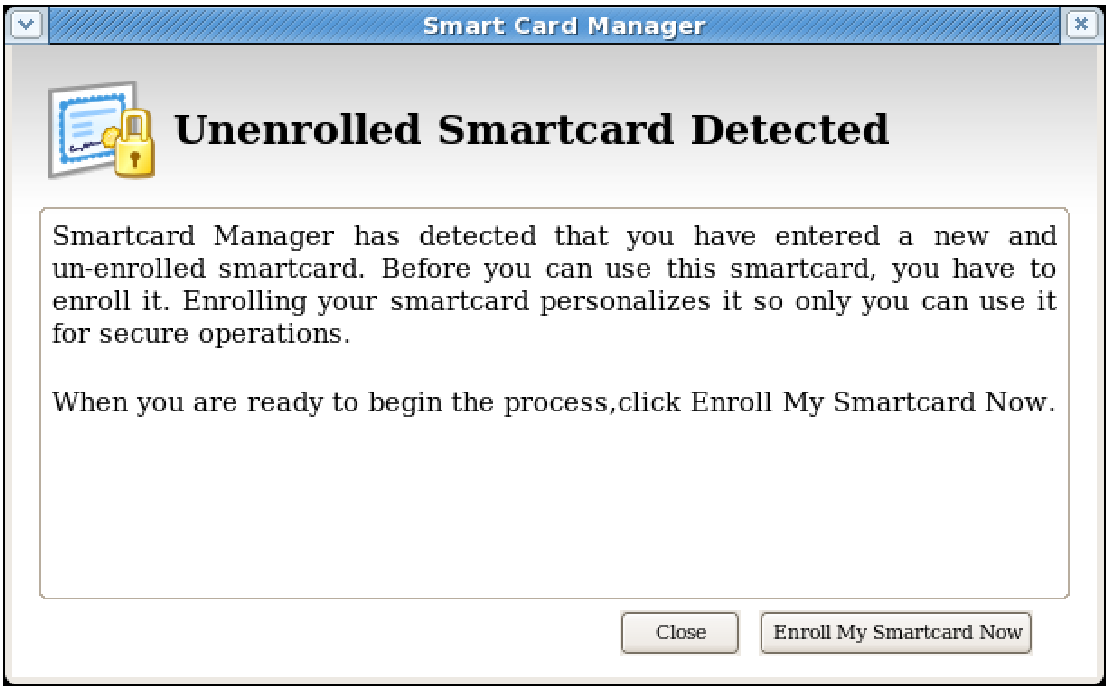
> **NOTE**
>
> This procedure assumes that the smart card is uninitialized and the appropriate Phone Home information has been configured.

1. Ensure that the **Enterprise Security Client** is running.

2. Insert an uninitialized smart card, pre-formatted with the Phone Home information for the TPS and the enrollment interface URL for the user's organization.

   The smart card can be added either by placing a USB form factor smart card into a free USB slot, or by inserting a standard, full-sized smart card into a smart card reader.

   When the system recognizes the smart card, it displays a message indicating it has detected an uninitialized smart card.



Smart Card Manager

Unenrolled Smartcard Detected

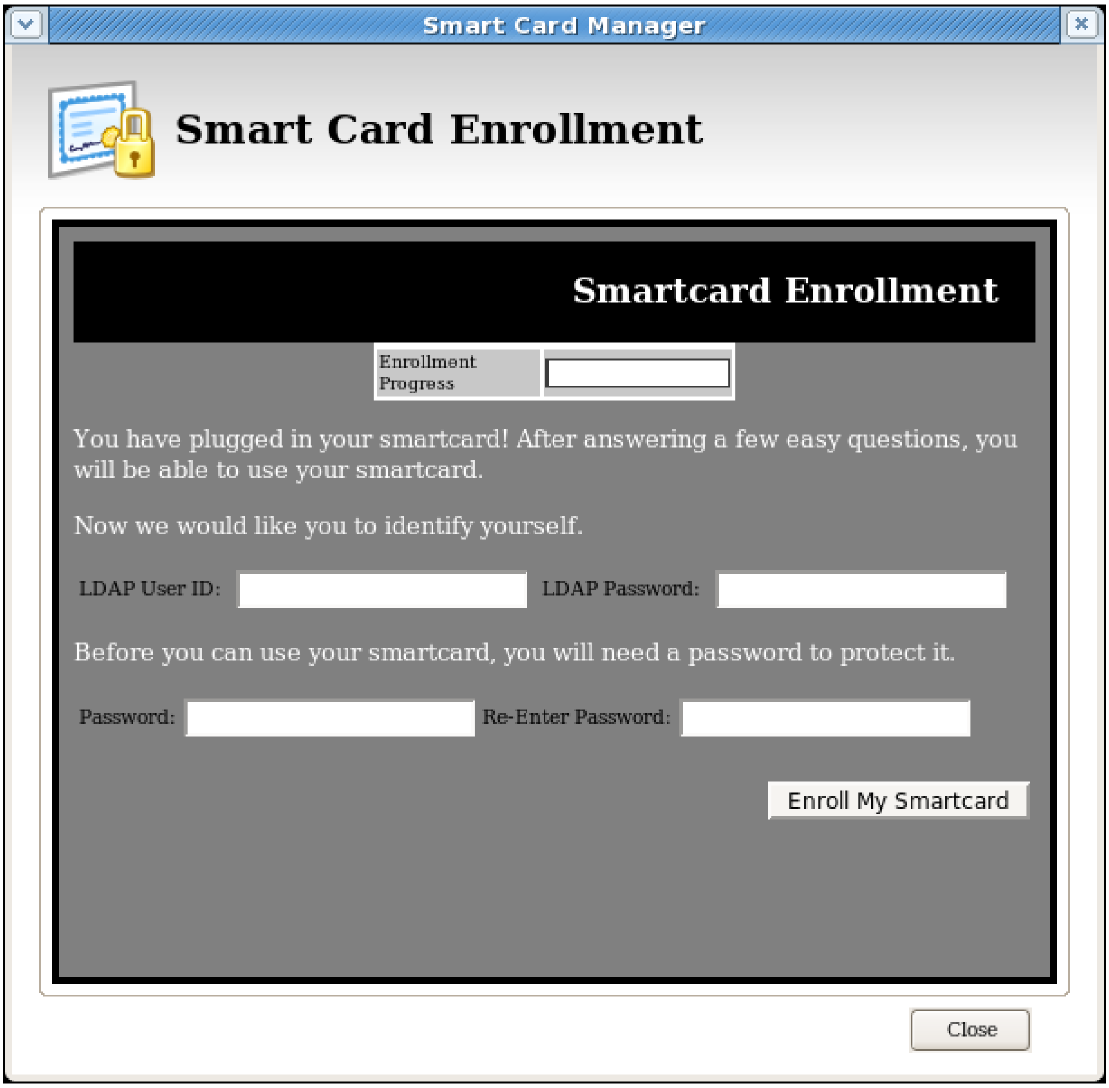
Smartcard Manager has detected that you have entered a new and un-enrolled smartcard. Before you can use this smartcard, you have to enroll it. Enrolling your smartcard personalizes it so only you can use it for secure operations.

When you are ready to begin the process,click Enroll My Smartcard Now.

3. Click **Enroll My Smart Card Now** to display the smart card enrollment form.

> **NOTE**
>
> If you remove the card at this point, a message displays stating that the smart card can no longer be detected. Reinsert the card to continue with the enrollment process.

4. Because the Smart Card Manager now knows where the enrollment UI is located (it is included in the Phone Home information), the enrollment form is displayed for the user to enter the required information.



You have plugged in your smartcard! After answering a few easy questions, you will be able to use your smartcard.

Now we would like you to identify yourself.

LDAP User ID: LDAP Password:

Before you can use your smartcard, you will need a password to protect it.

Password: Re-Enter Password:

Figure 3.2. Smart Card Enrollment Page

The above illustration shows the default enrollment UI included with the TPS server. This UI is a standard HTML form, which you can customize to suit your own deployment requirements. This could include adding a company logo or adding and changing field text, etc.

Refer to *Section 5, “Customizing the Smart Card Enrollment User Interface”* for information on how to customize the UI.

5. The sample enrollment UI requires the following information for the TPS server to process the smart card enrollment operation:

   LDAP User ID
   This is the LDAP user ID of the user enrolling the smart card; this can also be a screen name or employee or customer ID number.

LDAP Password

This is the password corresponding to the user ID entered; this can be a simple password or a customer number.

NOTE

The LDAP user ID and password refer to the fact that the TPS server is usually associated with a Directory Server, which stores user information and to which the TPS refers to authenticate users.

Passwords must conform to the password policy configured in the directory server.

Password

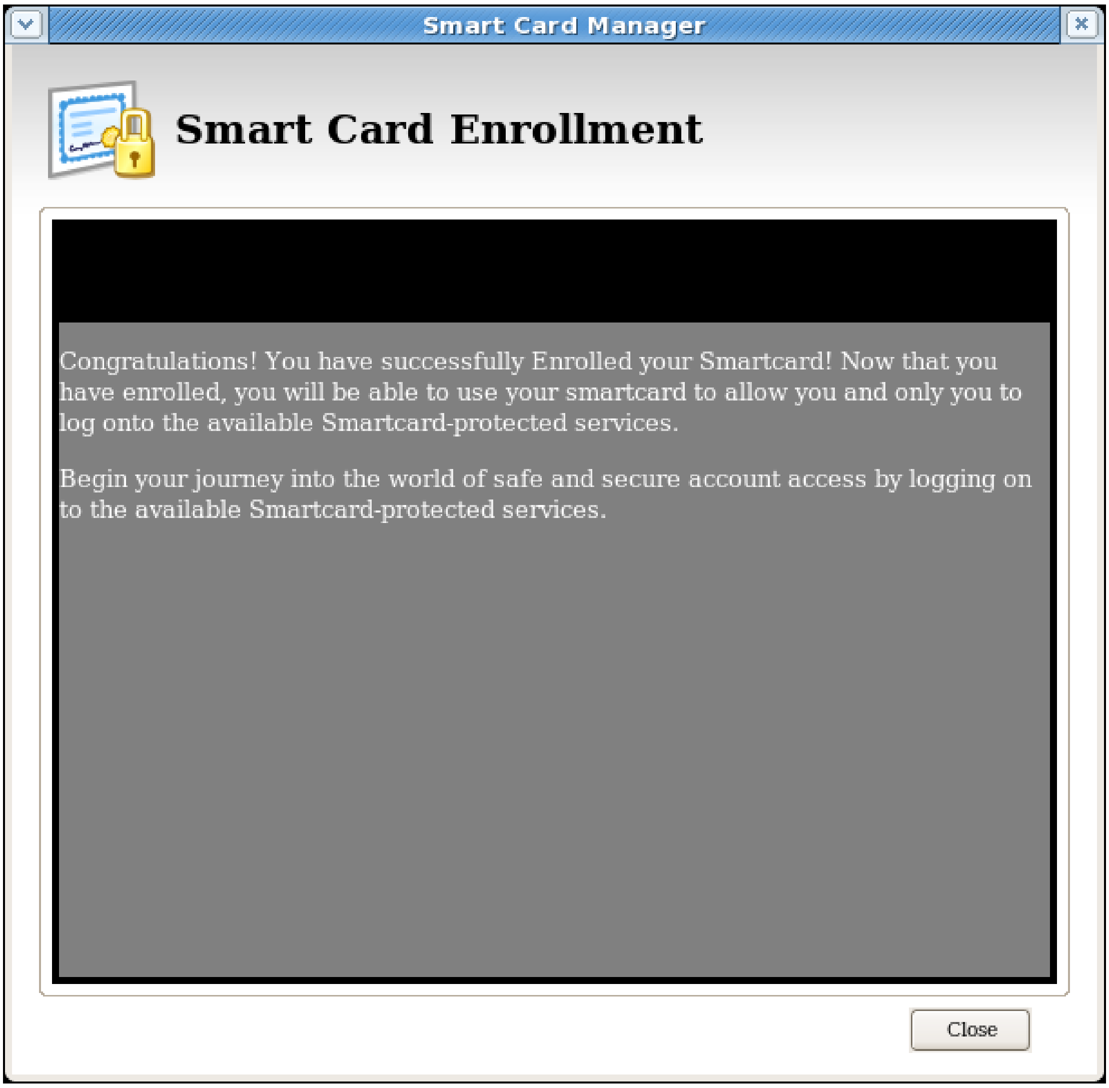
This sets the smart card's password, used to protect the card information.

Re-Enter Password

This confirms the smart card's password.

6. After you have entered all required information, click **Enroll My Smart Card** to submit the information and enroll the card.

7. When the enrollment process is complete, a message page opens which shows that the card was successfully enrolled and can offer custom instructions on using the newly-enrolled smart card.



Congratulations! You have successfully Enrolled your Smartcard! Now that you have enrolled, you will be able to use your smartcard to allow you and only you to log onto the available Smartcard-protected services.

Begin your journey into the world of safe and secure account access by logging on to the available Smartcard-protected services.

Figure 3.3. Smart Card Enrollment Success Message

5. Customizing the Smart Card Enrollment User Interface

Red Hat Certificate System (specifically the TPS subsystem) ships with a generic, external smart card enrollment user interface (UI). This UI consists of HTML and Javascript, and consequently can be customized to suit individual deployments.

The default HTML file for the Enrollment UI is located at `/var/lib/rhpki-tps/cgi-bin/home/Enroll.html`

The UI references resources such as images and Javascript files within its code. These resources are located in `/var/lib/rhpki-tps/docroot/home/`

The following is an extract from the default UI HTML file, and it includes comments on how you might customize it to suit your requirements.

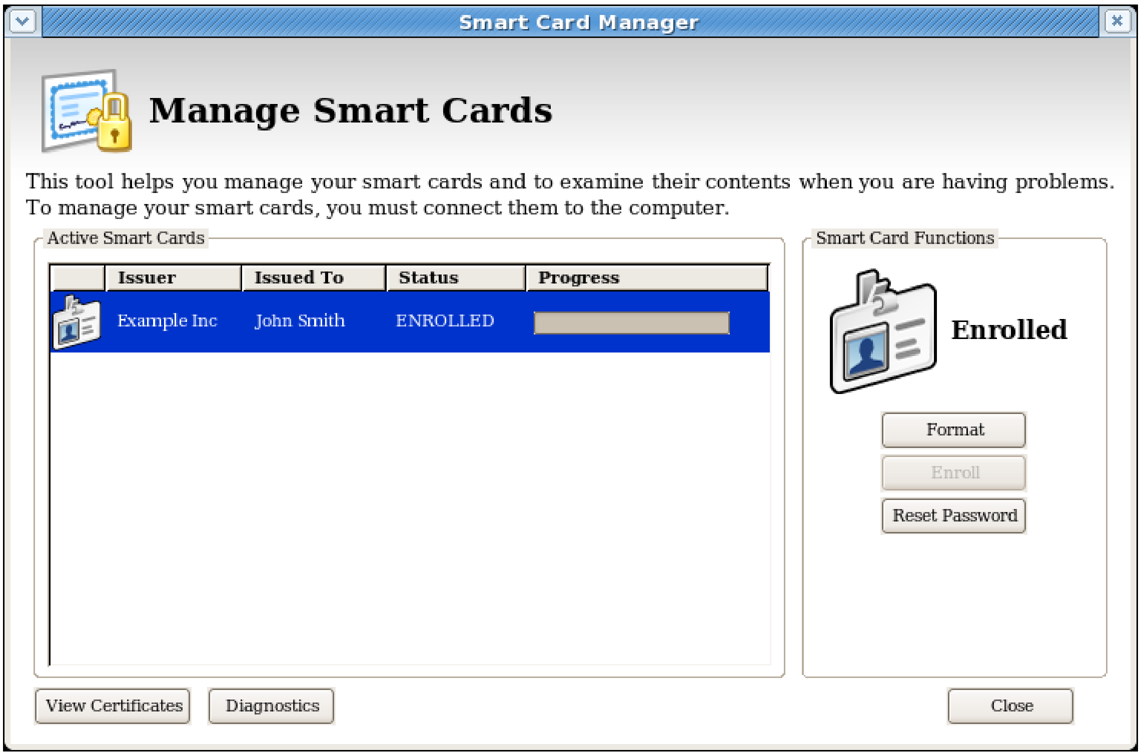
```
<html>
        <head>
                <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
                <link rel=stylesheet href="/home/style.css" type="text/css">

                <!-- Change the title if desired -->
                <title>Enrollment</title>

        </head>

        <script type="text/JavaScript" src="/home/util.js">
        </script>
        <body onload="InitializeBindingTable();" onunload=cleanup();>
                <progressmeter id="progress-id" hidden="true" align =
"center"/>

                <table width="100%" class="logobar">
                        <tr>
                                <td>

                                        <!-- Use customized logo here... -->
                                        <img alt="" src="/home/logo.jpg">

                                </td>
                                <td>
                                        <p class="headerText">Smartcard
Enrollment</p>
                                </td>
                        </tr>
                </table>

                <table id="BindingTable" width="200px"align="center">
                        <tr id="HeaderRow">
                        </tr>
                </table>

                <!-- Insert customized descriptive text here. -->
                <p class="bodyText">You have plugged in your smart card!
                        After answering a few easy questions, you will be
able to use your smart card.
                </p>

                <p class="bodyText">
                        Now we would like you to identify yourself.
                </p>

                <table>
                        <tr>
                                <td><p >LDAP User ID: </p></td>
                                <td> </td>
                                <td><input type="text" id="snametf"
value=""></td>
                                <td> </td>
                                <td><p>LDAP Password: </p></td>
                                <td> </td>
                                <td><input type="password" id="snamepwd"
value=""></td>
                        </tr>
                </table>

                <p class="bodyText"> Before you can use your smart card, you
will need a password to protect it.</p>

                <table>
                        <tr>
                                <td><p >Password:</p></td>
                                <td><input type="password" id="pintf"
name="pintf" value=""></td>
                                <td><p >Re-Enter Password:</p></td>
                                <td><input type="password" id="reenterpintf"
name="reenterpintf" value=""></td>
                        </tr>
                </table>
                <br>
                <table width="100%">
                        <tr>
                                <td align="right">
                                <input type="button" id="enrollbtn"
name="enrollbtn" value="Enroll My Smartcard"
                                       onClick="DoEnrollCOOLKey();">
                                </td>
                        </tr>
                </table>
        </body>
</html>
```

Example 3.2. Customizing the Smart Card Enrollment User Interface

6. 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 keys.

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.



Manage Smart Cards

This tool helps you manage your smart cards and to examine their contents when you are having problems. To manage your smart cards, you must connect them to the computer.

Issuer	Issued To	Status	Progress
Example Inc	John Smith	ENROLLED	

Figure 3.4. Manage Smart Cards Page

6.1. Formatting the Smart Card

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

The TPS server can be configured to load newer versions of the applet and symmetric keys onto the card.

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.

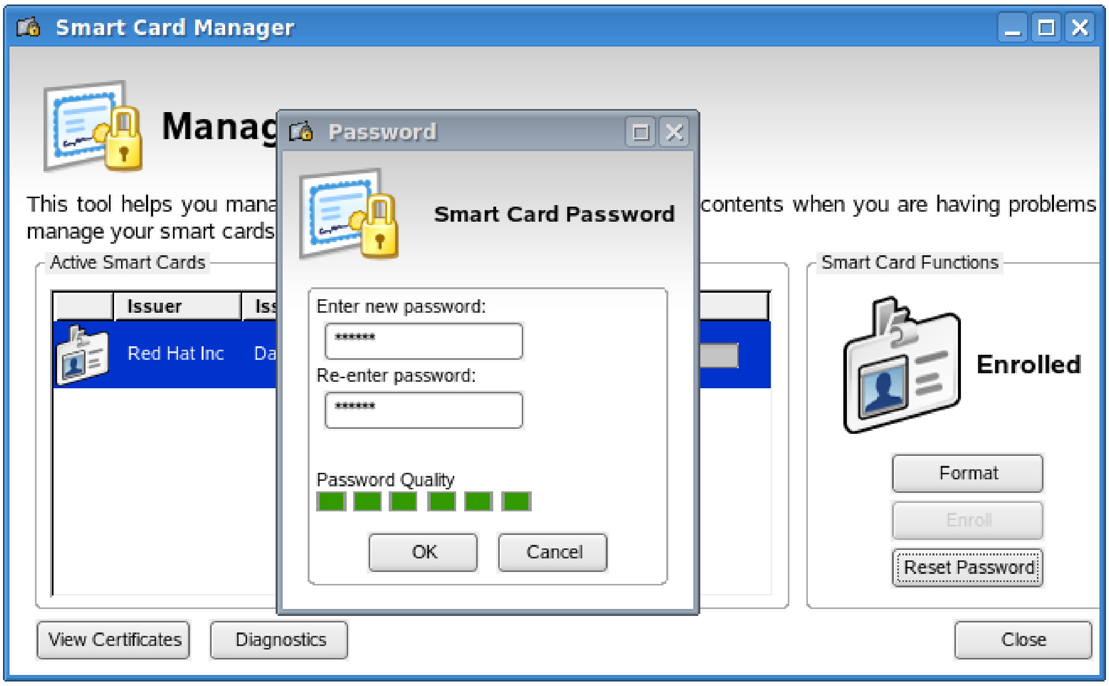
5. When the formatting process is complete, the **Active Smart Cards** table shows the card

status as UNINITIALIZED.

6.2. 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**.



This tool helps you mana
manage your smart cards

contents when you are having problems

Enter new password: ******

Re-enter password: ******

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.

6.3. Viewing Certificates

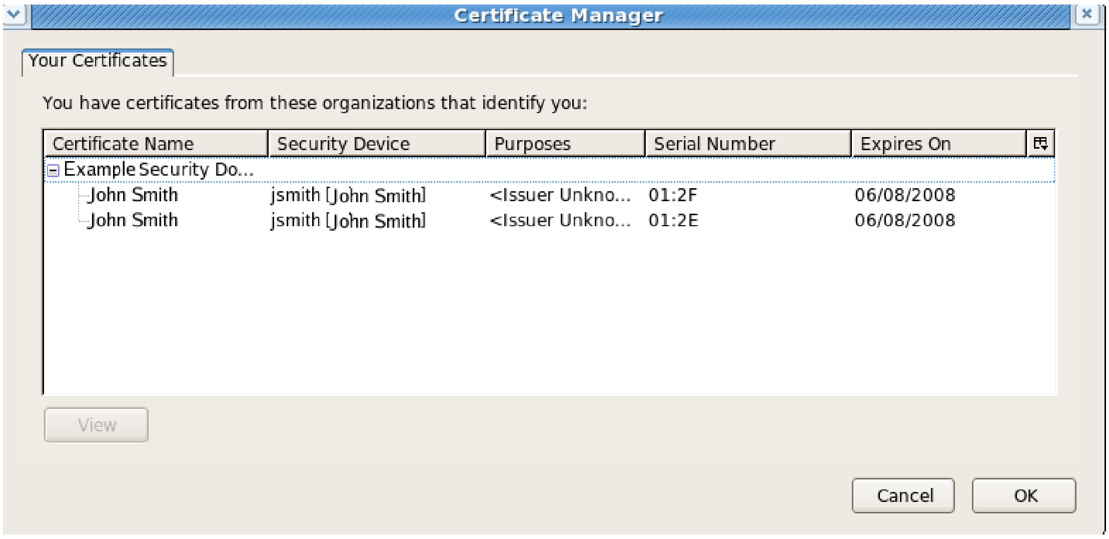
You can use the **Smart Card Manager** to display basic information about a selected smart card, including stored keys and certificates. To view certificate information:

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**.



You have certificates from these organizations that identify you:

Certificate Name	Security Device	Purposes	Serial Number	Expires On
⊟ Example Security Do...				
John Smith	jsmith [John Smith]	<Issuer Unkno...	01:2F	06/08/2008
John Smith	jsmith [John Smith]	<Issuer Unkno...	01:2E	06/08/2008

6.4. Enrolling Smart Cards

Most smart cards will be automatically enrolled using the automated enrollment procedure, described in *Section 4, “Smart Card Auto Enrollment”*. 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.

> **NOTE**
>
> This button is active only if the inserted card is unenrolled.

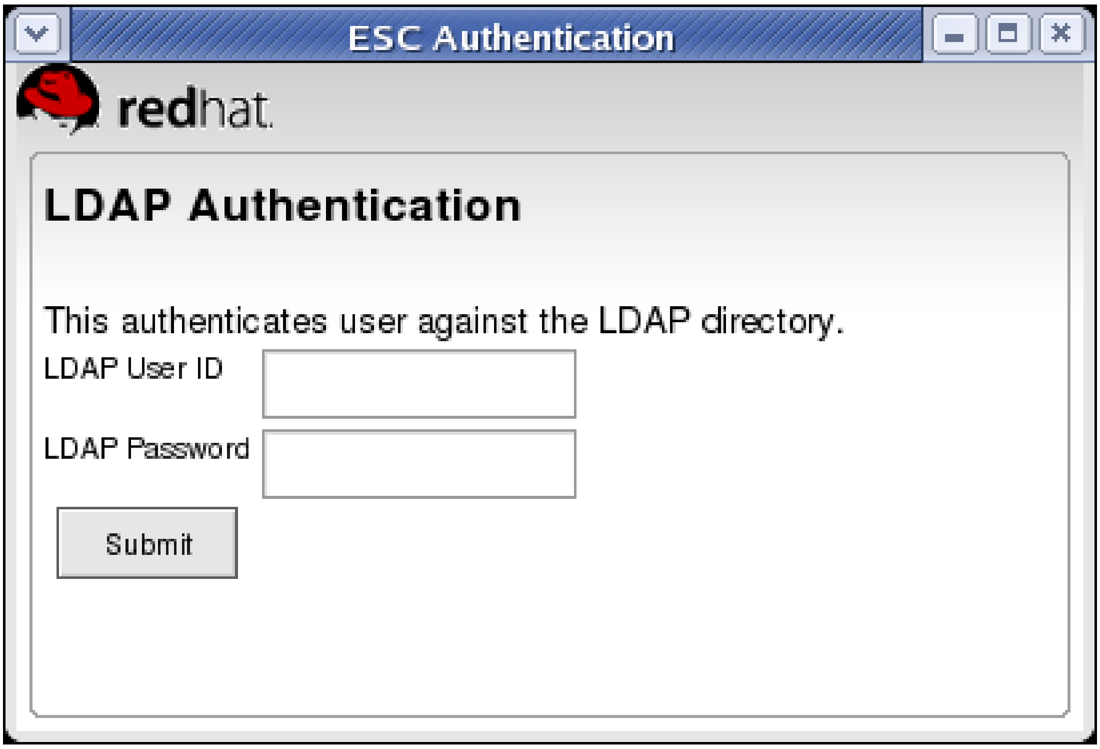
3. Enter a new key password in the **Enter a password** field.

   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.



ESC Authentication

redhat

LDAP Authentication

This authenticates user against the LDAP directory.

LDAP User ID

LDAP Password

Figure 3.5. LDAP Authentication Prompt

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

7. Security Officer Mode

The Smart Card Manager, in conjunction with the latest TPS Server software, now supports a special "security officer" mode of operation. This mode, presented as an alternative to the standard user-centric mode, allows a supervisory individual, (a security officer), the ability to oversee the face to face enrollment of regular users in a given organization.

After an individual has been enrolled, they are issued a smart card containing the necessary certificates required to perform company-supported cryptographic operations, such as encrypted email and secure web site access.

Security officer mode provides the following functionality:

- Enrollment of security officers.
- Enrollment of individuals under the supervision of a security officer, including:
  - The ability to search for an individual within an organization.
  - An interface that displays a photo and other pertinent information about an individual.
  - The ability to enroll approved individuals.
- Other functions provided by this mode include:
  - An interface to format or reset a user's card.
  - An interface to format or reset a security officer's card.
  - An interface to enroll a temporary card for a user that has misplaced their primary card.
  - An interface to store TPS server information on a card. This server information, or "Phone Home" information, is used by the Smart Card Manager to contact a given TPS server installation.

Working in the security officer mode falls into two distinct areas:

- Creating and managing security officers.
- Managing regular users by security officers.

When security officer mode is enabled, the Smart Card Manager uses an external user interface provided by the server. This interface takes control of smart card operations in place of the local XUL code that the Smart Card Manager normally uses.

The external interface maintains control until security officer mode is disabled.

7.1. Enabling Security Officer Mode

To set up security officer mode:

1. Open the Smart Card Manager installation directory.

   On **Microsoft Windows**, this is `C:/Program Files/Red Hat/ESC/esc.exe`.

   On **Red Hat Enterprise Linux**, this is `/usr/lib/esc-1.0.1/`

2. Open the `defaults/preferences/esc-prefs.js` file.

3. Edit the password prompt parameter in the `esc-prefs.js` file and set the value to `no`, meaning the prompt is enabled:

```
pref("esc.disable.password.prompt","no");
```

> **NOTE**
>
> Ensure that the Smart Card Manager is running and that its icon is displayed in the system tray (Windows) or in the notification area (Red Hat Enterprise Linux).

7.2. Managing Security Officers

- *Section 7.2.1, “Enrolling a New Security Officer”*
- *Section 7.2.2, “Formatting an Existing Security Officer Smart Card”*
- *Section 7.2.3, “Closing Security Officer Mode”*

7.2.1. Enrolling a New Security Officer

1. Run the `esc` command with the `-secmod` to open the security officer enrollment form. This has the format:

```
./esc -secmode SECURITY_URL/cgi-bin/so/enroll.cgi
```

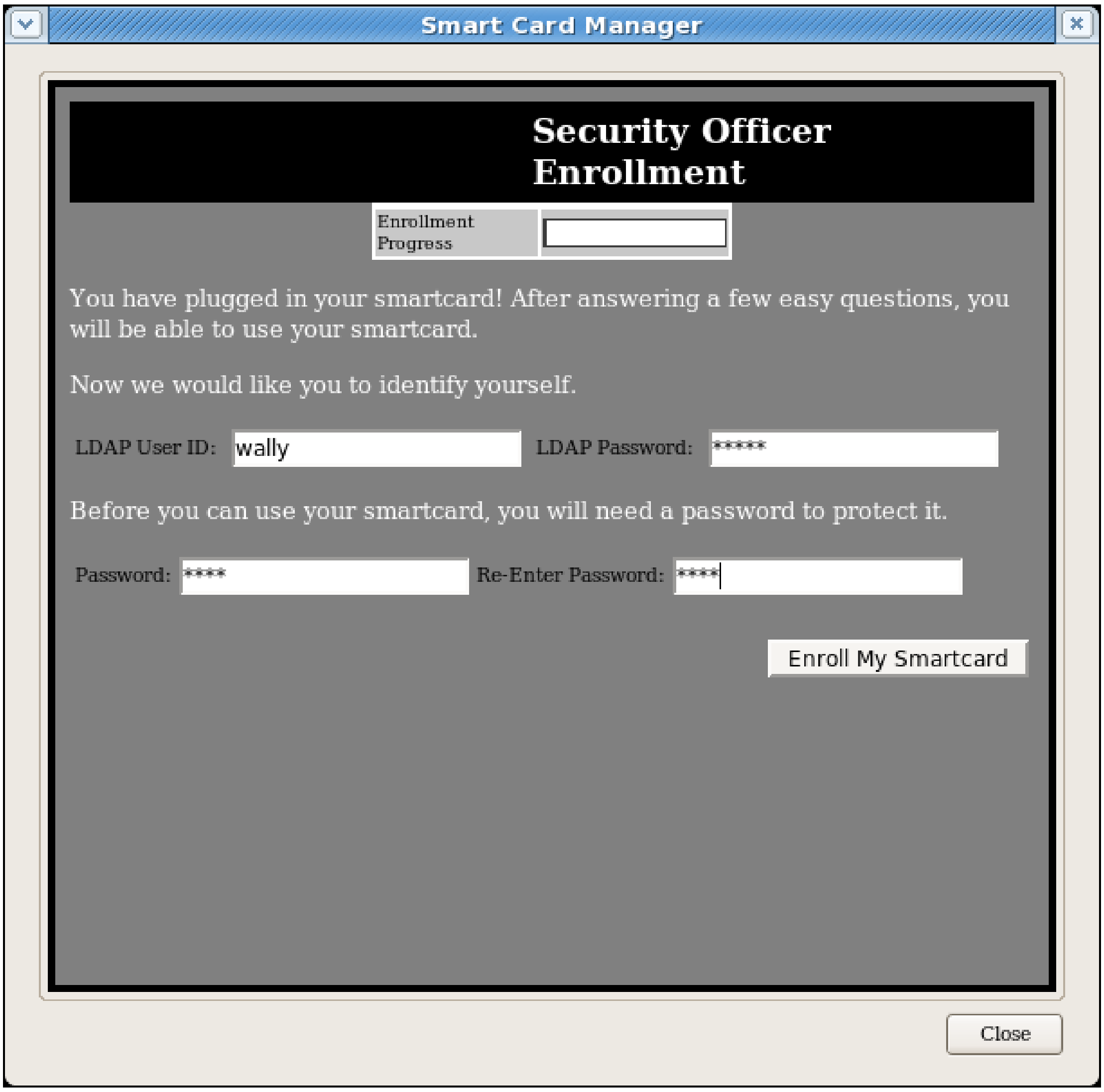
> **NOTE**
>
> The `esc` command can be run from any location.

SECURITY_URL is the pre-determined URL of the external TPS security interface. Referencing the `enroll.cgi` file opens the security officer enrollment page. For example:

```
./esc -secmode http://test.host.com:7888/cgi-bin/so/enroll.cgi
```

This opens the security officer enrollment page.

2. In the **Security Officer Enrollment** window, enter the LDAP user name and password of the new security officer and a password that will be used with the security officer's smart card.



You have plugged in your smartcard! After answering a few easy questions, you will be able to use your smartcard.

Now we would like you to identify yourself.

LDAP User ID: wally LDAP Password: *****

Before you can use your smartcard, you will need a password to protect it.

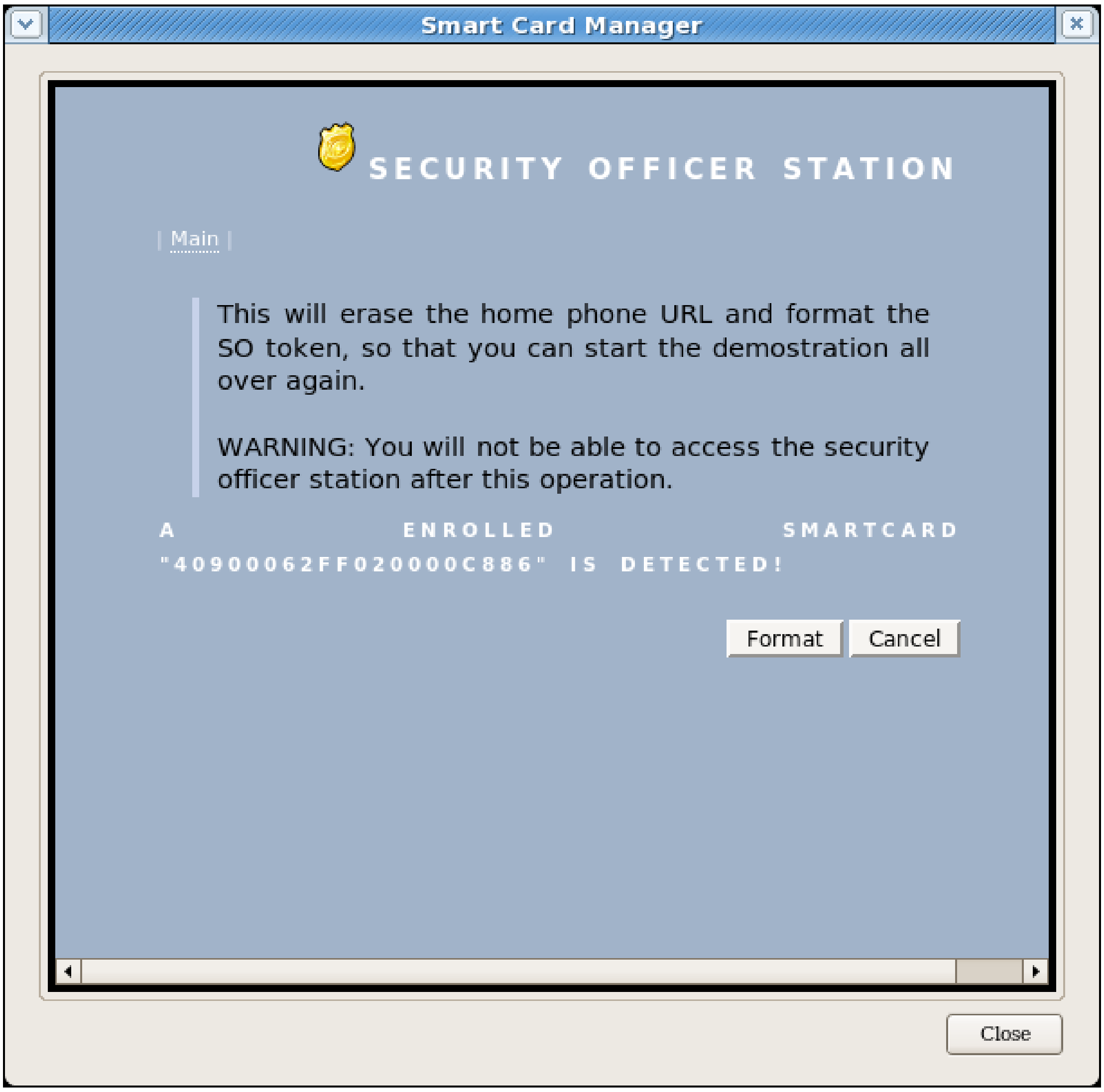
Password: **** Re-Enter Password: ****

3. Click **Enroll My Smartcard**.

This produces a smart card which contains the certificates needed by a security officer to gain access to the application, so that regular users can be enrolled and managed within the system.

7.2.2. Formatting an Existing Security Officer Smart Card

1. Click **Format SO Card**. Because the security officer card is already inserted, the following screen displays:



This will erase the home phone URL and format the SO token, so that you can start the demostration all over again.

WARNING: You will not be able to access the security officer station after this operation.

A ENROLLED SMARTCARD "40900062FF020000C886" IS DETECTED!

2. Click **Format** to begin the operation.

When the card is successfully formatted, the security officer's card values are reset. Another security officer's card must be used to enter security officer mode and perform any further operations.

7.2.3. Closing Security Officer Mode

Click **Close** to leave security officer mode. The Smart Card Manager now operates normally.

7.3. Managing Regular Users

The security officer Station page manages regular users through operations such as enrolling new or temporary cards, formatting cards, and setting the Phone Home URL.

- *Section 7.3.1, “Opening the User's Smart Card Interface”*
- *Section 7.3.2, “Enrolling a New User”*
- *Section 7.3.3, “Enrolling a Temporary Card for an Existing User”*
- *Section 7.3.4, “Formatting a Smart Card”*
- *Section 7.3.5, “Setting a Home URL on a Smart Card”*

7.3.1. Opening the User's Smart Card Interface

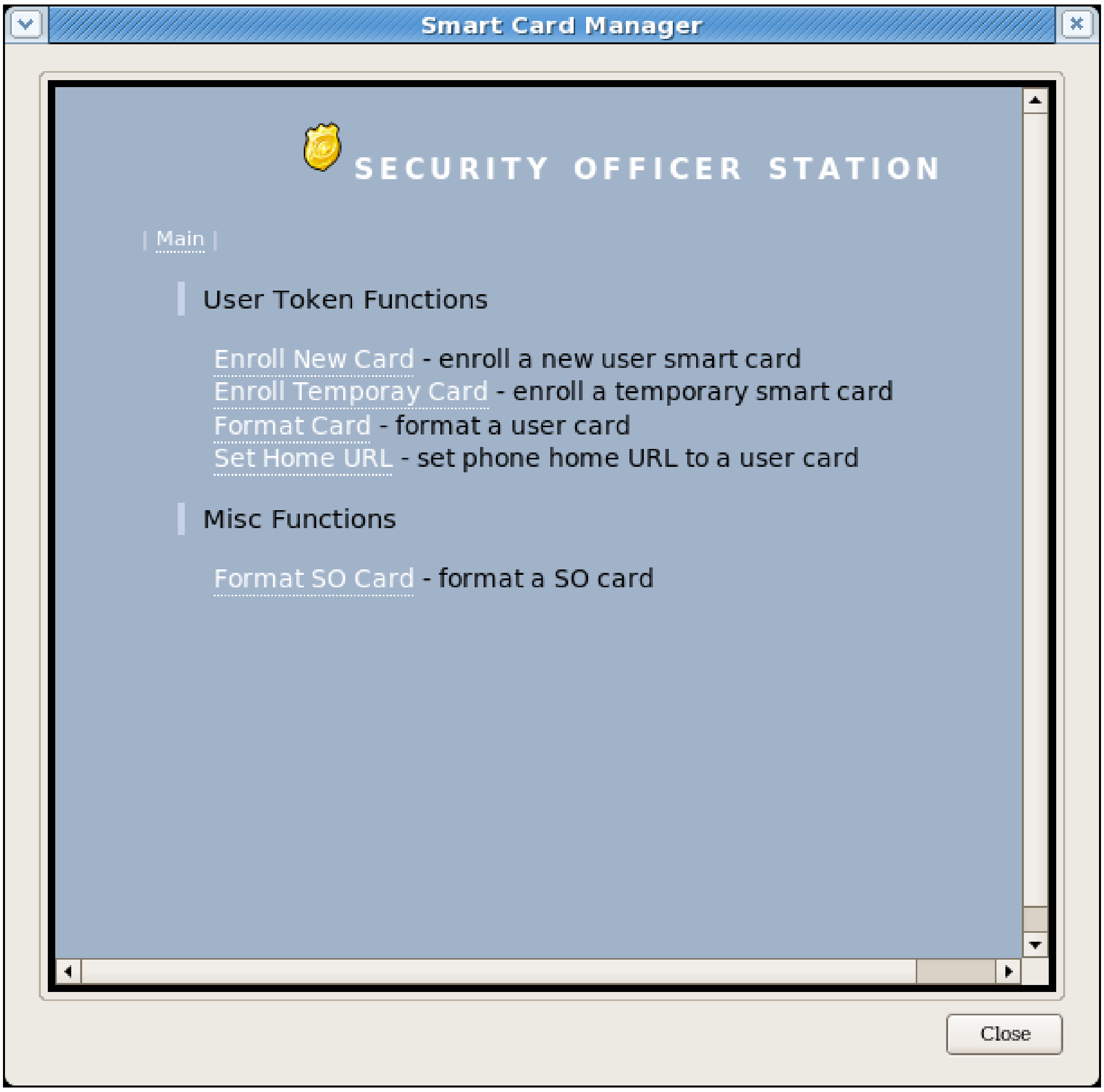
1. Run the `esc` command with the `-secmod` to open the security officer mode. This has the format:

```
./esc -secmode SECURITY_URL/cgi-bin/sow/welcome.cgi
```

> **NOTE**
>
> The `esc` command can be run from any location.

SECURITY_URL is the pre-determined URL of the external TPS security interface. Referencing the `welcome.cgi` file opens the security officer station page. For example:

```
./esc -secmode http://test.host.com:7888/cgi-bin/sow/welcome.cgi
```



SECURITY OFFICER STATION

| Main |

User Token Functions

Enroll New Card - enroll a new user smart card
Enroll Temporay Card - enroll a temporary smart card
Format Card - format a user card
Set Home URL - set phone home URL to a user card

Misc Functions

Format SO Card - format a SO card

This opens the security officer welcome page.

NOTE

Ensure that there is a valid and enrolled security officer card plugged into the computer. A security officer's credentials are required to access the following pages.

2. Click **Continue** to display the security officer Station page. You may be prompted to enter the password for the security officer's card. This is required for SSL client authentication.

7.3.2. Enrolling a New User

1. Click the **Enroll New Card** link to display the **Security Officer Select User** page.
2. Enter the LDAP name of the user who is to receive a new smart card.
3. Click **Continue**. If the user exists, the **Security Officer Confirm User** page opens.
4. Compare the photo of the user with the person actually present. Verify any other information presented on the screen.
5. If all the details are correct, click **Continue** to display the **Security Officer Enroll User** page. This page prompts the officer to insert a new smart card into the computer.
6. If the smart card is properly recognized, enter the new password for this card and click **Start Enrollment**.

A successful enrollment produces a smart card that a user can use to access the secured network and services for which the smart card was made.

7.3.3. Enrolling a Temporary Card for an Existing User

1. Click **Enroll Temporary Card**.
2. Follow the instructions given in *Section 7.3.2, “Enrolling a New User”*.

A successful enrollment produces a smart card which is valid for a limited time. The user can use this card until it expires.

7.3.4. Formatting a Smart Card

1. Click **Format Card** to display the page used to format or reset an existing smart card.
2. Ensure that a card is inserted into the computer.
3. Click **Format** to reset the card.

After successful completion, this card can be used to enroll another user.

7.3.5. Setting a Home URL on a Smart Card

1. Click **Set Home URL** to display the screen used to write a new home URL onto the card. This "home URL" tells the Smart Card Manager where to locate the TPS servers.

   This feature may prove useful if any of the TPS server information has changed since a given card was used.

2. Ensure that a valid card is inserted into the computer.

3. Click **Format**. The result will be a card with the new phone home information stored on the card.

8. Diagnosing Problems

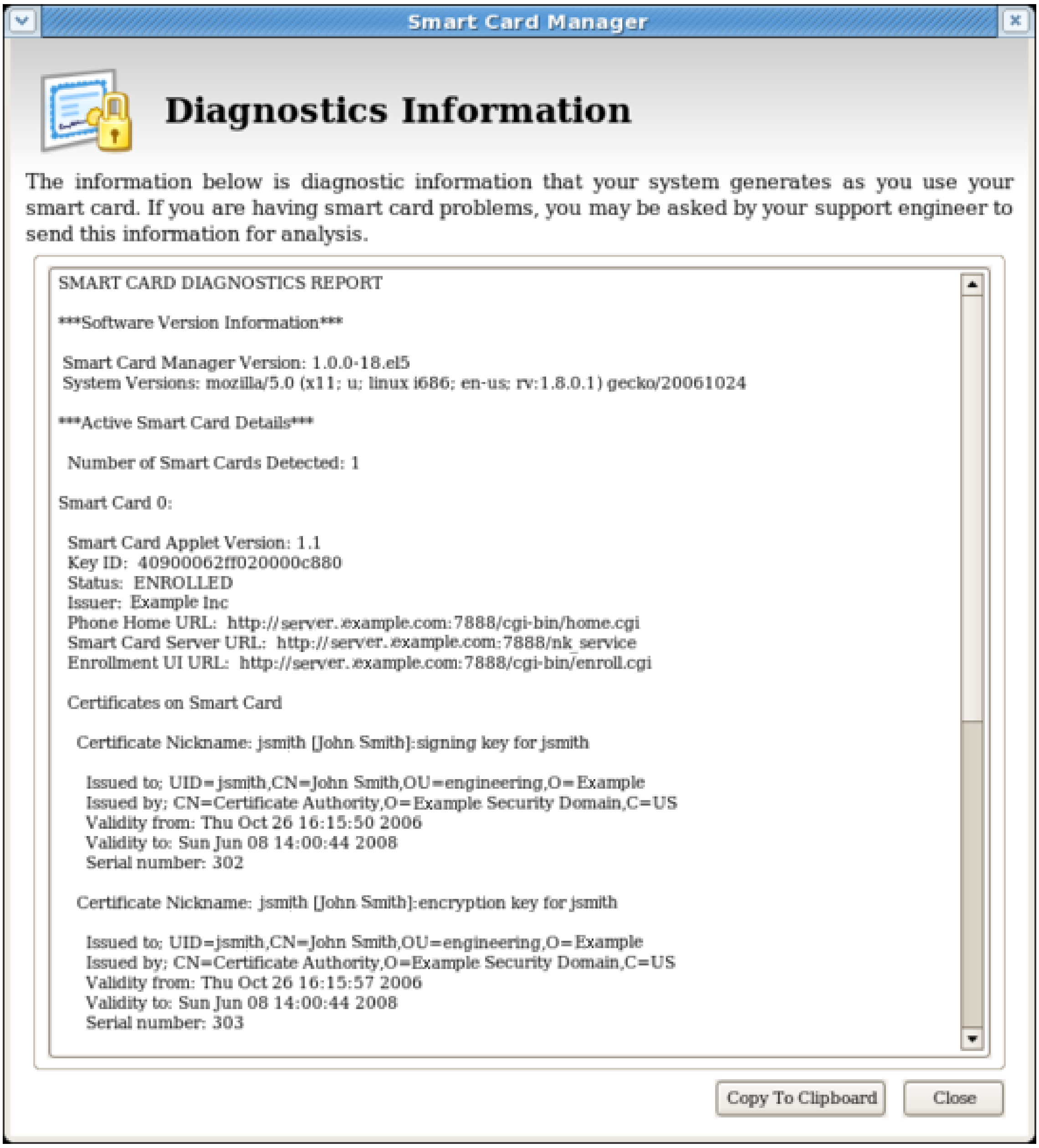
The **Enterprise Security Client** includes basic diagnostic tools and a simple interface to log errors and common events, such as inserting and removing a smart card or changing the card's password. The diagnostic tools can identify and notify users about problems with the **Enterprise Security Client**,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

To open the **Diagnostics Information** screen:

- In the **Manage Smart Cards** screen, click **Diagnostics**.

The following problems and events are logged by the **Enterprise Security Client**:

- 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.
- 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.



Diagnostics Information

The information below is diagnostic information that your system generates as you use your smart card. If you are having smart card problems, you may be asked by your support engineer to send this information for analysis.

```
SMART CARD DIAGNOSTICS REPORT

***Software Version Information***

 Smart Card Manager Version: 1.0.0-18.el5
 System Versions: mozilla/5.0 (x11; u; linux i686; en-us; rv:1.8.0.1) gecko/20061024

***Active Smart Card Details***

  Number of Smart Cards Detected: 1

Smart Card 0:

  Smart Card Applet Version: 1.1
  Key ID:  40900062ff020000c880
  Status:  ENROLLED
  Issuer:  Example Inc
  Phone Home URL:  http://server.example.com:7888/cgi-bin/home.cgi
  Smart Card Server URL:  http://server.example.com:7888/nk_service
  Enrollment UI URL:  http://server.example.com:7888/cgi-bin/enroll.cgi

  Certificates on Smart Card

    Certificate Nickname: jsmith [John Smith]:signing key for jsmith

      Issued to: UID=jsmith,CN=John Smith,OU=engineering,O=Example
      Issued by: CN=Certificate Authority,O=Example Security Domain,C=US
      Validity from: Thu Oct 26 16:15:50 2006
      Validity to: Sun Jun 08 14:00:44 2008
      Serial number: 302

    Certificate Nickname: jsmith [John Smith]:encryption key for jsmith

      Issued to: UID=jsmith,CN=John Smith,OU=engineering,O=Example
      Issued by: CN=Certificate Authority,O=Example Security Domain,C=US
      Validity from: Thu Oct 26 16:15:57 2006
      Validity to: Sun Jun 08 14:00:44 2008
      Serial number: 303
```

Figure 3.6. The Smart Card Manager Diagnostics Information Screen

Information Displayed on the Diagnostics Information Screen.

The Diagnostics Information screen displays the following information:

- The **Enterprise Security Client** version number.
- The version information for the system upon which the client is running.
- 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 following:

  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 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.

Using Enterprise Security Client Keys for SSL Client Authentication and S/MIME

After a token is enrolled, the token 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. These are described in the following table:

Platform	Module Name	Location
Windows	`coolkeypk11.dll`	`C:\Windows\System32\`
Red Hat Enterprise Linux	`libcoolkeypk11.so`	`/usr/lib/`
Macintosh	`libcoolkeypk11.dylib`	`/Library/Application Support/CoolKey/PKCS11`

Table 4.1. PKCS #11 Module Locations

1. Using the Certificates on the Token for SSL

To use the certificate on the token for SSL in an application such as Mozilla Firefox:

1. In Mozilla Firefox, open the **Tools** menu, choose **Options**, and then click **Advanced**.

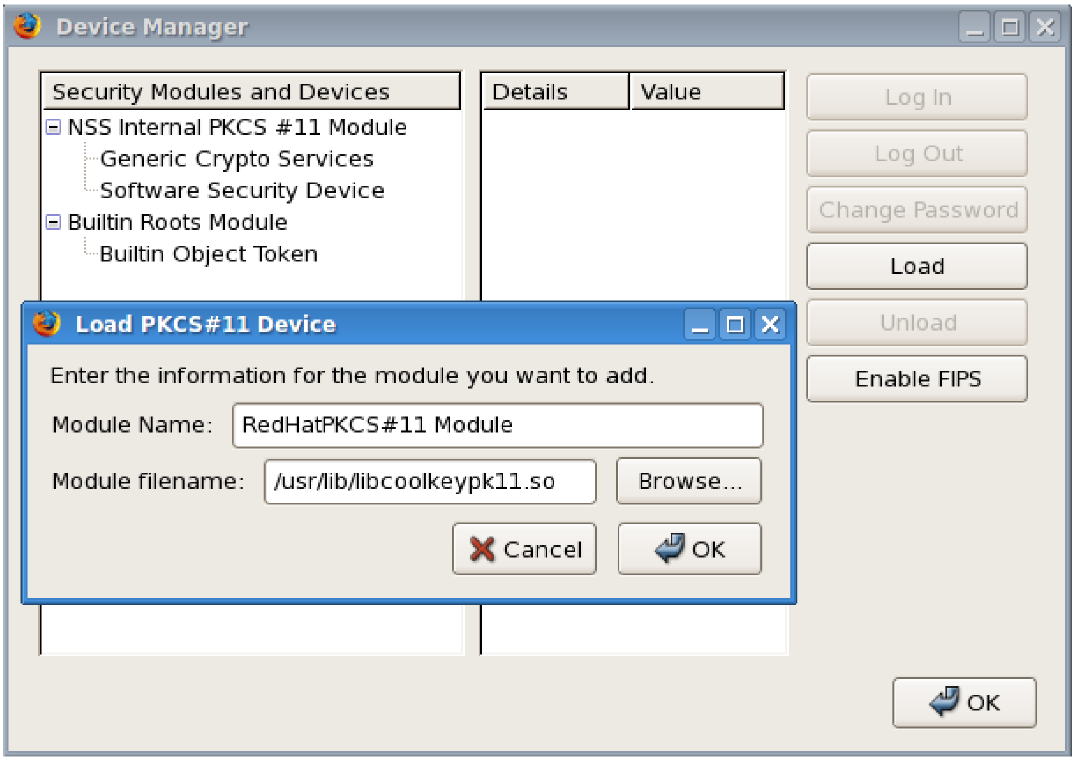
2. Add a PKCS #11 driver.

> **NOTE**
>
> Windows and Macs automatically attempt to load the PKCS #11 module to any Mozilla browsers they find.

   a. Click **Manage Security Devices** to open the **Device Manager** window, and then click the **Load** button.

   b. Enter a module name, such as `token key pk11 driver`.

   c. Click **Browse**, find the Enterprise Security Client PKCS #11 driver, and click **OK**.



Enter the information for the module you want to add.

Module Name: RedHatPKCS#11 Module

Module filename: /usr/lib/libcoolkeypk11.so

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

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

```
https://example.com:9443/ca/ee/ca
```

   b. Click the **Retrieval** tab, and then click **Import CA Certificate Chain**.

   c. Click **Download the CA certificate chain in binary form** and then click **Submit**.

   d. Choose a suitable directory to save the certificate chain, and then click **OK**.

   e. Click **Edit > Preferences**, and select the **Advanced** tab.

   f. Click the **View Certificates** button.

   g. Click **Authorities**, and import the CA certificate.

4. Set the certificate trust relationships.

   a. Click **Edit > Preferences**, and select the **Advanced** tab.

   b. Click the **View Certificates** button.

   c. Click **Edit**, and set the trust for websites.

The certificates can be used for SSL.

2. S/MIME Applications

To enable S/MIME on mail applications such as Mozilla Thunderbird:

1. In Mozilla Thunderbird, open the **Edit** menu, and select **Account Settings**.
2. Select **Security** on the left.
3. Add a PKCS #11 driver.
   a. Click **Manage Security Devices** to open the **Device Manager** window.
   b. Click the **Load** button.
   c. Enter the module name, such as `token keypk11 driver`.
   d. Click **Browse**, find the Enterprise Security Client PKCS #11 driver, and click **OK**.
4. If the CA is not yet trusted, download and import the CA certificate.
   a. Open the **SSL End Entity** page on the CA. For example:

```
https://example.com:9443/ca/ee/ca
```

   b. Click the **Retrieval** tab, and then click **Import CA Certificate Chain**.
   c. Click **Download the CA certificate chain in binary form** and then click **Submit**.
   d. Choose a suitable directory to save the certificate chain, and then click **OK**.
   e. In Thunderbird, open the **Edit** menu, and select **Account Settings**.
   f. Select **Security** on the left, and click the **Manage Certificates** button.
   g. Click the **Authorities** tab, and import the CA certificate.
5. Set up the certificate trust relationships.
   a. In Thunderbird, open the **Edit** menu, and select **Account Settings**.
   b. Select **Security** on the left, and click the **Manage Certificates** button.
   c. In the **Authorities** tab, select the CA, and click the **Edit** button.
   d. Set the trust settings for identifying websites and mail users.
   e. In the **Digital Signing** section of the **Security** panel, click **Select** to choose a certificate to use for signing messages.

6. In the **Encryption** of the **Security** panel, click **Select** to choose the certificate to encrypt and decrypt messages.

Appendix A. Enterprise Security Client Configuration

Previously, **Enterprise Security Client** relied on an application-specific configuration file. **Enterprise Security Client** is now based on Mozilla XULRunner technology, which allows the preferences facility built into Mozilla to be used for simple configuration of the **Enterprise Security Client**. A simple UI, discussed in *Chapter 3, Using the Enterprise Security Client*, manages most important configuration settings.

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

1. Configuration

The **Enterprise Security Client** uses the Mozilla configuration preferences for each of the supported platforms. A default configuration file is located in the following directories on each platform:

Platform	Location
Windows	C:\Program Files\Red Hat\ESC\defaults\preferences\esc-prefs.js
Red Hat Enterprise Linux	/usr/lib/esc-1.0.1/esc/defaults/preferences/esc-prefs.js /usr/lib64/esc-1.0.1/esc/defaults/preferences/esc-prefs.js
Macintosh	~/Desktop/ESC.app/defaults/preferences/esc-prefs.js

Table A.1. Location of Default Configuration Files

These files specify the default configuration to use when the **Enterprise Security Client** is first launched.

When the **Enterprise Security Client** is launched, it creates a separate, unique profile directory for each user on the system. These profiles are stored in different locations on each platform, as described below:

Platform	Location
Windows	`C:\Documents and Settings\$USER\Application Data\RedHat\ESC\Profiles`

Platform	Location
Red Hat Enterprise Linux	`~/.redhat/esc`
Macintosh	`~/Library/Application Support/ESC/Profiles`

Table A.2. Location of Enterprise Security Client User Profiles

> **NOTE**
>
> When the **Enterprise Security Client** requires any changes to a user's configuration values, the updated values are written to the user's profile area, not to the default Javascript file.

The `esc-prefs.js` file extract shown below lists the **Enterprise Security Client**-supported configuration values.

```
#################################################################
#The entry below is the XUL chrome page where Enterprise Security
#Client proceeds on startup.
#
 pref("toolkit.defaultChromeURI",
  "chrome://esc/content/settings.xul");

#The entry below is the URL Enterprise Security Client consults
#for back end TPS functionality.
 pref("esc.tps.url","https://test.host.com:7888/nk_service");

#The following three entries are for internal use
 pref("signed.applets.codebase_principal_support",true);
 pref("capability.principal.codebase.p0.granted",
    "UniversalXPConnect");
 pref("capability.principal.codebase.p0.id", "file://");

#The entry below sets how many seconds Enterprise Security Client
#should wait while TPS is processing a message
 pref("esc.tps.message.timeout","90");

#The entry can be set allow Enterprise Security Client to write
#newly created certificates
#to the local CAPI store after an enrollment operation.
#Also, when a format is done, those same certs will be removed
#from the local CAPI store.

pref("esc.windows.do.capi","yes");
##################################################################
```

Example A.1. Example Configuration File

2. Enterprise Security Client Mac TokenD

The TokenD software installed on a Macintosh computer provides a link between the Certificate System CoolKeys and the Mac CDSA security API, which provides a wide variety of security functionality.

For example, the Apple Mail application can use a *KeyChain* to perform security-related tasks. A KeyChain can store entities such as certificates, passwords, and private and public keys. Although most KeyChains are stored in software, the CDSA API allows KeyChains to be stored on smart cards or keys. CoolKey TokenD allows a Certificate System key to be displayed as a KeyChain.

To verify that the TokenD software is working correctly:

1. Ensure that the **Enterprise Security Client** has been installed on the Mac computer.
2. Use the **Enterprise Security Client** to enroll a token, enabling it with the correct certificates and key information.
3. Insert the enrolled token into a USB slot.
4. If TokenD is working, the token blinks for a few seconds while the information is obtained from the token. This is because the Mac CDSA layer is making a request for data.
5. Open the Mac `Keychain Access` utility in `Applications/Utilities/`
6. Find the new `$Keychain` entry in the list of valid chains. The chain has the key's UID in its name.
7. Click the CoolKey KeyChain to view the certificates and keys on the token.

3. Enterprise Security Client XUL and Javascript Functionality

Smart Card Manager stores the XUL markup and Javascript functionality in `<`*`ESC_INSTALL_PATH`*`>/chrome/content/esc/`, where `<`*`ESC_INSTALL_PATH`*`>` is the **Smart Card Manager** installation directory.

The primary **Smart Card Manager** XUL files are listed in the following table.

Filename	Purpose
`settings.xul`	Contains the code for the **Settings** page.
`esc.xul`	Contains the code for the **Enrollment** page.

Filename	Purpose
`config.xul`	Contains the code for the configuration UI.
`esc_browser.xul`	Contains the code for hosting the external HTML **Smart Card Manager** enrollment UI.

Table A.3. Main XUL Files

The primary **Smart Card Manager** Javascript files are listed in the following table.

Filename	Purpose
`ESC.js`	Contains most of the **Smart Card Manager** Javascript functionality.
`TRAY.js`	Contains the tray icon functionality.
`AdvancedInfo.js`	Contains the code for the **Diagnostics** feature.
`GenericAuth.js`	Contains the code for the authentication prompt. This prompt is configurable from the TPS server, which requires dynamic processing by the **Smart Card Manager**.

Table A.4. Main Javascript Files

4. Quick Javascript UI Guide

Certificate System 7.1 deployments may be using a customized external UI for key enrollment. Changes have been made to the names of internal **Enterprise Security Client** XPCOM objects in later versions of Certificate System, so changes need to be made to the `ESC.js` file to adapt an older UI. The places for these changes are shown in the file section below.

```
//ESC.js : Core Enterprise Security Client functionality
....
//
// Attach to the Enterprise Security Client XPCOM object on load
//
  try {
netscape.security.PrivilegeManager.enablePrivilege("UniversalXPConnect");
    netkey = Components.classes["@redhat.com/rhCoolKey"].getService();
    netkey = netkey.QueryInterface(Components.interfaces.rhICoolKey);
    gNotify = new jsNotify;
    netkey.rhCoolKeySetNotifyCallback(gNotify);
  } catch(e) {
     alert("Can't get UniversalXPConnect: " + e);
  }

//Sample function to complete Enrollment of a key.
```

```
function EnrollCoolKey(keyType, keyID, enrollmentType, screenname,
                       pin,screennamepwd,tokencode)
 {
   try {

     netkey.EnrollCoolKey(keyType, keyID, enrollmentType, screenname,
                          pin,screennamepwd,tokencode);
   } catch(e) {
     ReportException("netkey.EnrollCoolKey() failed!", e);
     return false;
   }

   return true;
 }
```

5. Enterprise Security Client File Locations

This reference shows the different directories and file locations for the different client machines.

The location of the **Enterprise Security Client** main directory on the different client platforms is as follows:

Platform	Location
Windows	C:\Program Files\Red Hat\ESC
Red Hat Enterprise Linux	/usr/lib/esc-1.0.1/esc /usr/lib64/esc-1.0.1/esc
Macintosh	User preference for the `ESC.app` directory

Table A.5. Main Directories for the Enterprise Security Client

5.1. Windows

On Windows, **Enterprise Security Client** uses the following directories and files:

File or Directory	Purpose
`C:\Program Files\Red Hat\ESC`	Main directory.
`application.ini`	XULRunner application configuration file
`components\`	XPCOM components directory.
`chrome\`	Directory for Chrome components and additional application files for **Enterprise Security Client** XUL and Javascript.
`defaults\`	**Enterprise Security Client** default preferences.

File or Directory	Purpose
`esc.exe`	The executable which launches **Enterprise Security Client** in XULRunner.
`xulrunner\`	Privately-deployed XULRunner bundle.

Table A.6. Enterprise Security Client File and Directory Locations on Windows

5.2. Red Hat Enterprise Linux

On Linux, **Enterprise Security Client** is installed by its binary RPM to the default location `/usr/lib/esc-1.0.1/esc/`.

File or Directory	Purpose
`application.ini`	XULRunner application configuration file.
`components/`	XPCOM components.
`chrome/`	Directory for Chrome components and additional application files for **Enterprise Security Client** XUL and Javascript.
`defaults/`	**Enterprise Security Client** default preferences.
`esc`	The script which launches the **Enterprise Security Client**.
`xulrunner/`	Privately-deployed XULRunner directory.

Table A.7. Enterprise Security Client File and Directory Locations on Red Hat Enterprise Linux

5.3. Mac OS X

On Mac OS X, the XULRunner framework located in `ESC.app` as follows:

File or Directory	Purpose
`Contents/`	Privately deployed XUL framework • `Info.plist` • `Frameworks/` • `XUL.framework/` • `Resources`

File or Directory	Purpose
`application.ini`	**Enterprise Security Client** XULRunner application configuration file.
`components/`	**Enterprise Security Client** XPCOM components.
`chrome/`	Directory for Chrome components and additional application files for **Enterprise Security Client** XUL and Javascript.
`defaults/`	**Enterprise Security Client** default preferences.
`xulrunner`	The script which launches **Enterprise Security Client**.

Table A.8. Enterprise Security Client File and Directory Locations on Mac

Index